Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Cursive Bengali Script Recognition for Indian Postal Automation

THÈSE

présentée et soutenue publiquement le 12/11/2008

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Szilárd VAJDA

Composition du jury

Président :	Thierry Paquet	Professor, University of Rouen
Rapporteurs :	Jean-Marc Ogier	Professor, University of La Rochelle
	Laurence Likforman-Sulem	Associate Professor, Telecom ParisTech
Examinateurs :	Thierry Paquet	Professor, University of Rouen
	René Schott	Professor, University of Nancy 1
	Abdel Belaïd	Professor, University of Nancy 2
Invité :	Christophe Choisy	Research Engineer, Itesoft Company

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

First of all I would like to express my gratitude to Prof. Abdel Belaïd for supervising me during this thesis and giving me precious hints concerning research. He was the person who initiated me in handwriting recognition and he also helped me during the hard moments.

I would like to thank everyone else in the Read Group, especially to Hubert Cecotti, Yves Rangoni, Hatem Hamza and André Alusse. The laboratory has been an ideal environment, both socially and technically, in which to conduct research.

Special thank must go to Dr. Christophe Choisy for all the help concerning the basic NSHP-HMM recognition system.

Special thanks must go to Prof. B. B. Chaudhury and Dr. Umapada Pal for all the help supplied during my stay in CVPR Unit, Indian Statistical Institute, Kolkata, India.

I would also like to thank everyone in the Loria Research Center, specially Nadine Beurne for her indulgence and help concerning the administration related problems.

The Indian Post should be thanked for providing the Indian postal document to carry out our research work. The Service de Recherche Technique de la Poste (SRTP), is to be thanked for providing the bank check amount dataset.

The Indo-French Center for the Promotion of Advanced Research (IFCPAR), is to be thanked for providing the financial support necessary for me to carry out this work.

I'm dedicating this thesis to my mother, for the endless love and support.

Table des matières

Table des figures

1

Introduction

Nowadays, various works were proposed to realize the core of the recognition systems to satisfy the needs raised by different real-life applications like automatic reading of postal documents, bank check reading, form processing, printed document recognition, etc. Despite the impressive progress achieved during the last few decades in this field, the performances of the handwriting recognition systems are still far from human performances. Most of these systems, while presenting a large spectrum of perspectives to the problem, share the same difficulties.

The automatic reading of handwritten addresses is quite a dynamic research field and several research teams all over the world are interested in. Such reading systems are typically composed by several processing stages : image acquisition and image pre-processing followed by address bloc location, segmentation into lines and words, location of ZIP code and city names, recognition of the ZIP code and city name and finally the fusion of the results to produce a final decision. Each of these processes are hiding quite serious challenges which guided us to investigate such a mail processing system.

During this thesis, we have tried to consider a restricted part of these handwriting recognition issues throughout the design and implementation of a system for cursive Bengali handwritten address recognition. Considering this research work positioned in the field of postal document recognition, different type of questions have been raised like :

- What kind of word recognizer should be considered ? Should we segment or not in a script environment(Bengali) which has never been segmented before ?
- How to exploit the graphical richness of this mainly unknown script which Bengali is ? How can this extra information integrated in the recognizer ?
- Is it possible to extend the word recognizer to handle larger vocabularies ? If so, how we can do this ?
- What kind of digit recognizer should be considered for this purpose ? What kind of learning strategy should be considered ?

– What kind of improvements can be proposed to improve the digit recognition scores ?

Considering all these issues, we have concentrated our efforts to apply and extend an existing word recognizer, so called NSHP-HMM (Non Symmetric Half-Plane Hidden Markov Model) which is a totally $2D$ model able to recognize words without any kind of physical segmentation. This choice was motivated by the fact that there was no existing solution to segment handwritten Bengali words into letters or graphemes. To exploit the specific graphical shape property of this ancient script, we propose a combination of low-level information with the high-level one considering them as one entity instead of using them separately as other models do. This natural combination follows the human reading habits where the whole word is considered and there is no physical separation between the different type of information.

In order to extend the existing model for larger vocabularies, we propose an appropriate stopping mechanism in the decomposition process where there is no physical segmentation, so there are no well defined boundaries between the letter components.

Equally, we were interested to recognize pin codes coming from these postal documents. In this topic we have focused our attention around some neural network solutions and we have proposed a new learning mechanism. Meanwhile, we have conducted some research around classifiers combination to reach higher accuracy and robustness as well.

However, the main challenge of this work was to apply these solutions to Indian postal documents written in Bengali. While the Latin scripts which are familiar to us, Europeans, contains just a restricted number of letters and digits, the Bengali script (the second most popular script in India and the 5^{th} most popular script over the world) is much more rich in number as well more complex in graphical shapes. Our target was to apply and adapt with success the existing word and digit recognizers by exploiting the specificities of this Indian script.

The proposed HWR (Handwriting Recognition System), which is the outcome of a strength collaboration work [1] between Indian and French scientists, has been used with success on different recognition tasks. The main application area is the recognition of Bengali city names and pin codes coming from Indian postal documents. Several experiments that have been carried on Latin and Bangla scripts also allowing us to consider the accuracy and the robustness of the model. A success can also be observed for separated handwritten digits, where the system gives also promising results.

The recognition of handwritten Bangla city names is a pioneering work as in our best knowledge there is no existing research in this field.

The thesis structure can be described as follows :

1. The international project 2702-1 "HANDWRITING RECOGNITION FOR POSTAL AUTOMATION" has been hosted by IFCPAR (Indo French Centre for the Promotion of Advanced Research) and has been deployed by CVPR (Computer Vision and Pattern Recognition), Calcutta, India and READ (REcognition of writing and Document Analysis), Nancy, France.

- In chapter 2. we try to describe the different attempts achieved to design and create such postal address recognition systems, highlighting the different challenges and hardships encountered by the researchers during the last thirty years. We will review some word recognition strategies and digit recognition strategies as well . In order to get a clear idea about the current strategies applied to reduce the vocabulary in the handwriting recognition paradigm, a section is addressing to this issue.
- In chapter 3. a detailed formal description is given concerning the NSHP-HMM system and its different applications in handwritten word and digit recognition, highlighting the system's advantages and the drawbacks derived from the model's nature.
- Chapter 4. contains a personal contribution concerning the implant of high-level information in the NSHP-HMM HWR system to create a more reliable and robust system. A detailed description is given concerning the extraction of the features, the combination of the low-level and high-level features in the framework of the NSHP-HMM and the different normalization techniques proposed by us. The evaluation of this new technique is performed on the handwritten Bangla city name dataset and the SRTP dataset which is a handwritten French bank check amount collection.
- Chapter 5. contains the contribution concerning a new pruning methodology in the Viterbi decoding process. We describe the theoretical aspects of the threshold mechanism used by this strategy followed by an evaluation of the technique for handwritten Bangla city names.
- Chapter 6. presents a comparison study between the stochastic and neural models used for separated handwritten digits. This chapter has the role to show our achievements on digit recognition using HMM based techniques and respectively neural network based approaches. In order to highlight the strengths and weakness of each type of method we propose some combination schemes to exploit the complementarity of these classifiers. The evaluation of the methods is performed on different handwritten digit datasets.
- The final chapter 7. is consecrated to the conclusions concerning our contribution to the field of handwriting recognition by appointing new ways to explore based on the work proposed by us.

2

Postal documents recognition

Throughout this chapter we would like to review the difficulties raised by the automatic postal document recognition with specific reflection to the Indian postal documents, which is the main concerns in this thesis. The main objective is not to give you a full description of the field but to highlight the specificities of this task. Similarly, we will review the handwritten word recognition and handwritten digit recognition issues but mainly just some specific domains will be considered in order to allow a direct comparison with the solutions proposed in this research work.

2.1 Postal documents recognition

Automatic sorting of handwritten mail pieces is a very challenging task. The main problem in handwritten address recognition are parsing and recognizing a set of correlated entities such as the ZIP codes, street names and building numbers, in the presence of incomplete information. It is a computer vision problem which has stringent performance requirements in commercial applications

The task of accurately recognizing and interpreting a handwritten address is complicated by the variability and the complexity of the address, word shape distortion due to non-linear shifting, unpredictable writing style and failure to locate the actual address in the database due to severe postcode recognition errors and intrinsic deficiencies in the address database.

2.1.1 History

As a result of extensive socioeconomic activity in recent years, Japan's information traffic has been increasing rapidly. As stated by Wada in [Wad33] the total volume of mail in Japan is equally rising and increasing by 7% ever since 1975. Considering the fact that the amount of mail per capital in Japan is 1/2 of that in the US, they are estimating with a kind of confidence

that the volume of mail will become twice that of the early 90s. For that purpose the postal mechanization was considered a high priority issue by the Postal Bureau of the Ministry of Posts and Telecomunication and has a history of more than 4 decades.

In order to allow such an automatic sorting strategy the standardization of envelopes as well as the introduction of postal codes was necessary. In 1962, in order to ensure postal item harmonization , the JIS standard [Tok93] was initially instituted, with eight such standards being formulated as recommended envelope standards by the Ministry. Meanwhile, the introduction of the postal code system has been a longer process and was accepted by the public just in 1975.

The first automatic postal mail sorting system has been installed at Tokyo Central Post Office and it was the world's first machine that could read 3-digit numerals within red postal code frames through OCR equipment. Similarly in 1968 the first culling, facing and canceling machine (CFC) has been started working on Shinjuku Post Office. In 1971 they made it possible to interconnect the OCR sorters with the CFC establishing one of the first entirely automatic postal document sorting system.

The same development has been started in France due to the growing demand for such automatic mail sorting. The Technical Research Department of La Post (SRTP) established in 1984 was responsible for many research projects in the field, but the mainstream was to adapt the existing systems rather than innovate [Bur93].

In order to follow-up the modern solutions a total rethink was necessary as stated by Burbaud. An outcome of this strategy is the Rennes-Cesson parcel center, a sorting unit experimentally using self-guided vehicles which serve the container unloading platform. The same direction has been followed by projects focused on address recognition on small envelopes where a former segmentation method has been considered and for digit recognition a neural network has been found as being the ideal solution. An extended version of the former project stated by Gilloux in [Gil93] was the address recognition of flat mail, where the address is often surrounded by other informations like advertisement, sender identification, magazines, etc. For that reason, a preliminary address location process should precede the recognition. As La Poste offers banking solutions for the clients a project has been oriented toward such a bank check reading system [LLGL97] carrying many advantages : the size of the vocabulary is reduced (the shape profile of the courtesy amounts in French are quite discriminant) and the location of the different contents are restricted.

The Unites States Postal Service (USPS) has also invested significantly toward automated processing of mail-pieces to speed up the sorting as well as to reduce the labor cost. The letter mail automation program of the United Stated hosed by USPS utilize recognition software developed by numerous vendors [SLGS02]. Using such a strategy the cost of the processing per 1000 mail pieces drops from 47.78 USD for manual processing to 27.46 USD for mechanized processing and 5.30 USD for automated processing. The savings are multiplying rapidly as about 400 million

pieces of letter mail per day are considered for sorting by the USPS. Setlur and his colleagues [SLGS02] are describing the different databases and standards imposed by the USPS to resolve the address issue on the mail pieces. The standards are fairly well adhered to especially in the machine printed mail-streams. Handwritten mail finds greater deviations from the standards.

2.1.2 Postal document preprocessing

When mixed mail enters in a postal facility, it must be first faced and oriented, so that the address to be readable by the used mail processors. Existing USPS systems face and orient the domestic mail pieces based on the fluorescing indicia on each mail pieces. However, as stated by the authors in [NCA+03] stamps and foreign-originated mail pieces do not fluoresce so the processing systems can not sort foreign mails. For this purpose, they are proposing a system which analyze both face of the mail piece and faces it and orients it in right order. After a preliminary binarization process, the address is located without considering the position of the address candidate. From these candidates, based on a priori knowledge (stamp position, presence of postal delimiters, bar codes, etc.), the best one is selected. Even the results are promising, 7.1% of error has been done during the facing and orientation, the system is still dependent on the a priori knowledge given by the human operator. However, as stated by the authors the beta test of their system will save around 500.000 hours anually. The system has been deployed by USPS in November 2002.

Once the facing and orientation is performed the location of the address bloc should be considered. Different attempts have been done based mainly on the structural composition of the address. El Yacoubi et al. are proposing quite an interesting solution [YBG95]. Instead of using the classical way, they are using word spotting for that purpose. Using this technique they are not just locating but also recognizing the word they are looking for between the address structure. Their model is based on HMMs. For each digit they are designing a letter HMMS and the word HMMs are made of a simple concatenation of the corresponding letter HMMs. As features they are considering the presence of : upper strokes, lower strokes and closed loops. The system has been tested on a reduced size database (122) containing 350 street names. The achieved 92.1% is quite impressive and the authors concluded that the remaining error are coming from pieces where some segmentation errors occured or the image was quite noisy.

Lii and Shrihari [LS95] have considered a similar challenge for fax cover pages. They were trying to locate the name and the address of these special type of documents. The system attempts to locate and recognize words which are data field indicators so as to figure out the position of such keywords as "TO" and "COMPANY". These keywords are considered as references in block segmentation to segment out their associated data regions. Firstly they are building a spatial map grid where all the spatial relationships between the text objects are stored. Locational

information concerning the individual connected components can directly be retrieved from this map. The connected components are serving for labeling purpose. Instead of word recognition they are concentrating more on character recognition using a two-layer back propagation neural network using chain code feature. The word recognition for the reduced vocabulary (To, Attention, From, Company, Message, Pages) is based on matching at letter level. The results are both for machine printed and handwriting fax cover pages. While for printed documents 100% good location performance has been done, for printed the result was just 80%. The results are quite impressive but the size of the dataset (12 machine printed documents versus 115 handwritten) can not really show the quality of the technique.

For the address block location a contour clustering algorithm has been proposed by Govindaraju and Tulyakov [GT03] meeting the criteria to be invariant to the document style (printed or handwritten). Their strategy is to extract connected components contours, extract contour features and cluster these features in the feature space. Once these points are clustered using heuristics the cluster corresponding to the address block is detected. Finally, the other clusters close to the designated one are discarded. The algorithm was developed for parcel image set which contains images with well separated address blocks as well as non separated/incorrectly separated address blocks. For that purpose they have considered the HWAI system described in detail in [Sri00]. While without this address location algorithm the system has considered 240 images as being finalized, when the algorithm has been deployed this finalized document number has increased to 272 which is quite a success. The advantage of the system is its invariant aspect which can be exploited in such a document environment as postal documents.

Another kind of strategy is considered by the authors [WP94] to locate the address block. They are considering the same strategy as Le Cun with handwritten digits [LBBH01]. They also consider this issue, as a challenging one and they propose a convolutional neural network with four outputs to find the different corner of the address block.

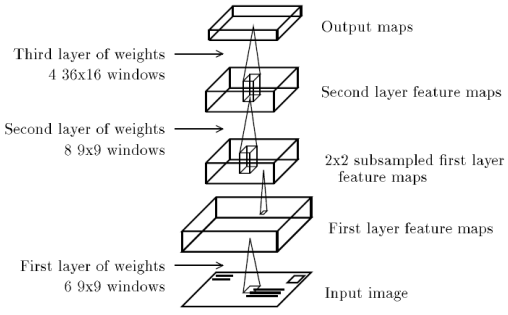

FIGURE 2.1 – The structure of the convolutional neural network used by Wolf and Platt

The system has been tested on 500 test images. One challenging test image and the cor-

responding output are presented in Fig. 2.2. The 98.2% score is a very good one and we also consider such a strategy as being useful for this address bloc location issue.

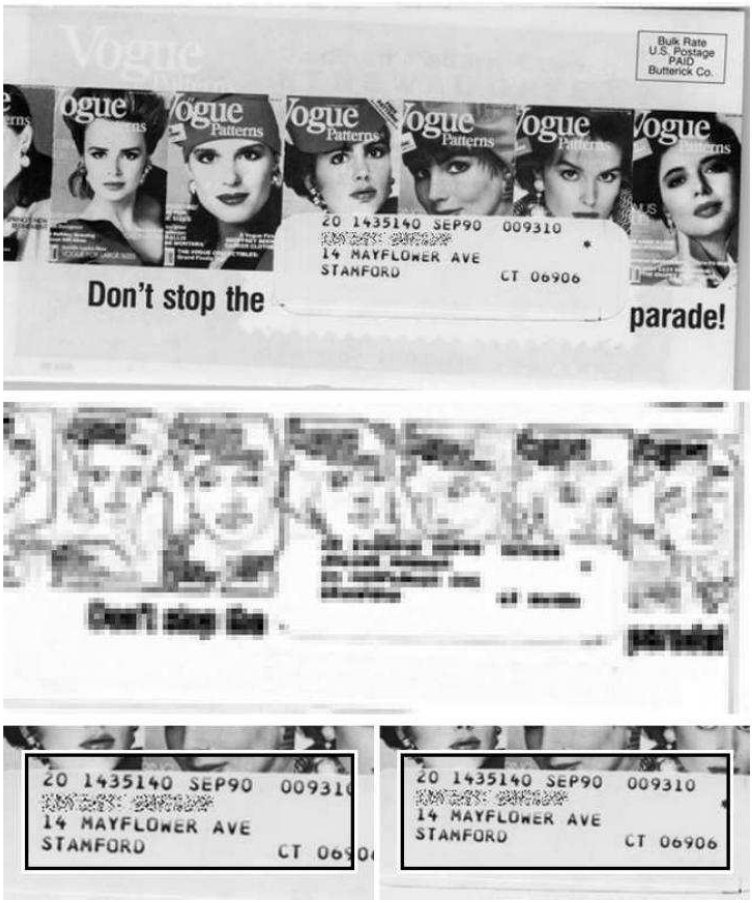

FIGURE 2.2 – A challenging sample file. As it can be observed, even after preprocessing there is still a large amount of background noise. While the left address candidate is almost correct the ZIP code is truncated. The right candidate (shown in the lower right) gives the complete address.

2.1.3 Automatic address recognition systems

The goal of this section is to give a brief idea about the current systems and their architecture. Unfortunately there are just a few research papers describing the whole system from image acquisition to final recognition. Mainly the research groups are focusing on some particular problems coming from this managing flow like : image pre-processing, address location, line and word segmentation, digit recognition and word recognition. We have also decided to follow this structure, presenting above just a few systems and after we will develop in more details the word recognition part as well as the digit recognition part as these are also the main concerns for our postal automation system.

Blumenstein and Verma [BV97] proposed the implementation of a recognition system for printed and handwritten postal addresses based on Artificial Neural Networks(ANN). They were interested in comparing different type of networks to analyze the recognition performance as well as the accuracy.

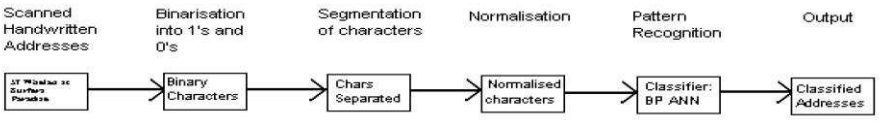

FIGURE 2.3 – The system flowchart considered by Blumenstein et al. for postal address recognition

They performed all the steps necessary before sending the image for recognition. The acquisition was made by a scanner and a simple binarization technique has been implemented. The segmentation is also based on connected components and paying attention to handwritten words where the cutting path has been defined by a sparse pixel density. In order to allow the recognition of characters by neural ANN a size regulated normalization has been done. The recognition has been performed by a multi-layer ANN trained with backpropagation algorithm. The system flowchart of the system is depicted in Fig. 2.3

While the results for printed characters are good (84.72% accuracy), the same situation can not be noted for the handwritten characters where just 58.59% has been reached. Similarly, for the RBF type ANN the results are even worse. However, the result for the whole address recognition scheme is quite promising. For printed addresses the system is varying between 83.33%-97.62% good accuracy, while as it was expected for handwritten addresses it can not reach more than 68.75% accuracy. The low result can be explained by the quality of the images as well as the reduced number of datasets. Unfortunately no comparison can be made due to the variations in the database.

Another complete system is proposed in [MSM98]. The authors present a system based on four modules : over-segmentor, dynamic zip locator, zip candidates generator and city-state verifier.

Instead of using a linear system architecture as we have seen in [BV97], the authors are using a more complicated structure with a possible return to the segmenter if no zip code is found correctly. First the address lines are separated using projection allowing a skew angle of -10 to +10 degree. The over segmentor is responsible for finding a set of split points for a word or text line image. They have applied heuristic for this purpose like location of a set of split point on the upper contour or lower contour of each connected component, looking for sharp or smooth valleys, horizontal and vertical overlaps of the graphemes, etc. The zip code locator is powered by a ANN in order to generate posterior probabilities in the matching. As they can

not know about the size of the ZIP code, they are looking for both 9-digit and 5-digit ones as well. Once this segmentation is successful, an HMM is considered to generate the candidates list. The output of the HMM is an ordered list of valid zip codes. These zip codes are coupled with the corresponding city names available in the database. The flexible matcher is used for matching the list of graphemes with every entry in the lexicon. For the selection, a criterion is defined : when the match is done of a sequence of graphemes to a string entry, it is not necessary that each character in the string is on the top among all the possible character classes for the corresponding segment. Based on a match ranking, the final decision is taken. The overall system achieves an accuracy rate of 83.5% with 3.6% error for 5-digit encoding on 805 cursive addresses. Similarly, as in the previous case, based on the specificity of the dataset, no direct comparison can be made.

The real-time system proposed by Kim et al. [KG95] is really usable for a small size lexicon. They are considering for preprocessing the chain code extraction coded in an array. Each data node in the structure represents one of the eight grid nodes that surround the previous data node. They also consider slant correction, noise removal, smoothing and normalization. After this process they are segmenting the characters in graphemes based on the following assumptions : the number of segments per character must be at most 4 and all touching characters should be separated. For features they are considering 74 chain code based features. The recognition is based on dynamic matching by comparison between several possible combinations of segments and reference feature vectors of codewords.

The results are performed on 3,000 images including firm names, street names, personal names and state names. Using all the 74 features they achieved 96.23°% (10 words lexicon), 87.40% (100 words lexicon) and for large vocabulary the results are decreasing to 72.30% (1000 words lexicon). Using a subset of these features, lower results have been achieved. First of all this is a complete recognition system giving high results for reduced vocabularies but in the mean time is quite fast (100-200 msec) because of the chain code representation, so it can meet the requirements of a real-time application.

In [Sch78] the authors describe the design principles of a multi-font word recognition system developed for German postal documents reading. He also describes the whole work flow but he is concentrating more on the separated character recognition and contextual post processing instead of the image preprocessing. The main idea is to feed each separated character into a SCR (Single Character Recognizer). These SCRs are standing for different purposes, they will decide if it is capital letter, small letter or the analyzed character belongs to the numerical dataset. Each recognizer will output a rank ordered list and these outputs will serve for the final decision to match them against a hash coded table look-up. Even if the author gives a very detailed description of the method it can not be considered a complete reading system and there is a lack of precision as there is no kind of result given about the accuracy of the SCR and neither about

the table look-up strategy.

2.1.4 The particularities of the Indian postal documents

As the main aspect concerning this thesis is the recognition of Indian postal documents so we would like to show the specificity of such documents.

India has a multi lingual and multi-script behavior. In India there are about 19 official languages and an Indian postal document can be written in any of these official languages. Moreover, some people write the destination address part of a postal document in two or more language scripts. For example in Fig. 2.4, the destination address is written partly in Bengali script and partly in English. Bengali is the second most popular language in India after the Hindi and the fifth most popular language in the world.

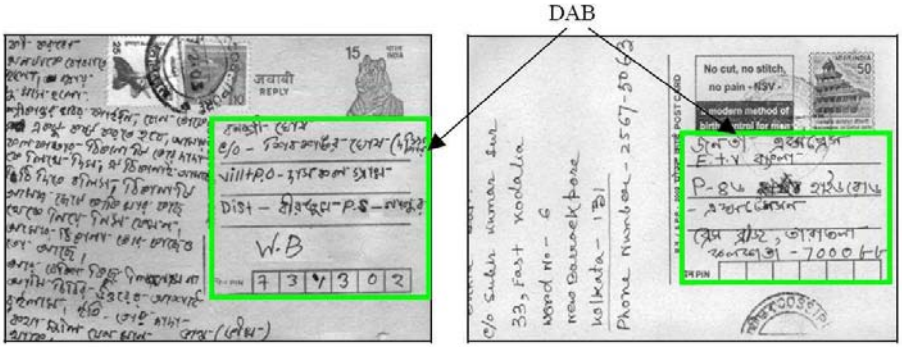

FIGURE 2.4 – Indian multi script postal documents with the corresponding DAB (destination address block) identified

Indian postal code is a six-digit number. Based on this six-digit number pin-code we cannot locate a particular post office in a village. We can locate a post office of a town/sub-town by this six-digit pin-code. Representation of the pin-code digits is shown in Figure 2.5. The Fig. 2.6 is the spatial representation of pin-codes all over India.

In India there is a wide variation in types of postal documents. Some of these are post-cards, inland letters, special envelopes, etc. Post-cards, inland letters, special envelopes are sold in Indian post offices and there is a pin-code box of six digits to write pin number in the postal document. Also, because of the educational backgrounds, there is a wide variation in writing style and medium. For example Kol-32 is written instead of Kolkata-700032. Also, sometime people do not mention pin-code on the Indian postal document. Thus, the development of Indian postal address recognition system is a challenging issue [RVP+05a].

The first digit and covering region		First two digit and their representation	
First Digit	Region	First 2 Digit	States/Circle Covered
1 2	Northern	11	Delhi
		12 to 13	Harvana
		14 to 16	Punjab
		17	Himachal Pradesh
		18 to 19	Jammu & Kashmir
		20 to 26	Uttar Pradesh
		27 to 28	Uttaranchal
3 4	Western	30 to 34	Raiasthan
		36 to 39	Guiarat
		40 to 44	Maharastra
		45 to 48	Madhva Pradesh
		49	Chattisgarh
5 6	Southern	50 to 53	Andhra Pradesh
		56 to 59	Karnataka
		60 to 64	Tamil Nadu
		67 to 69	Kerala
7 8	Eastern	70 to 74	West Bengal
		75 to 77	Orissa
		78	Assam
		79	North Eastern
		80 to 85	Bihar
		86 to 88	Jharkand

FIGURE 2.5 – Representation of first and second digit in an Indian pin-code

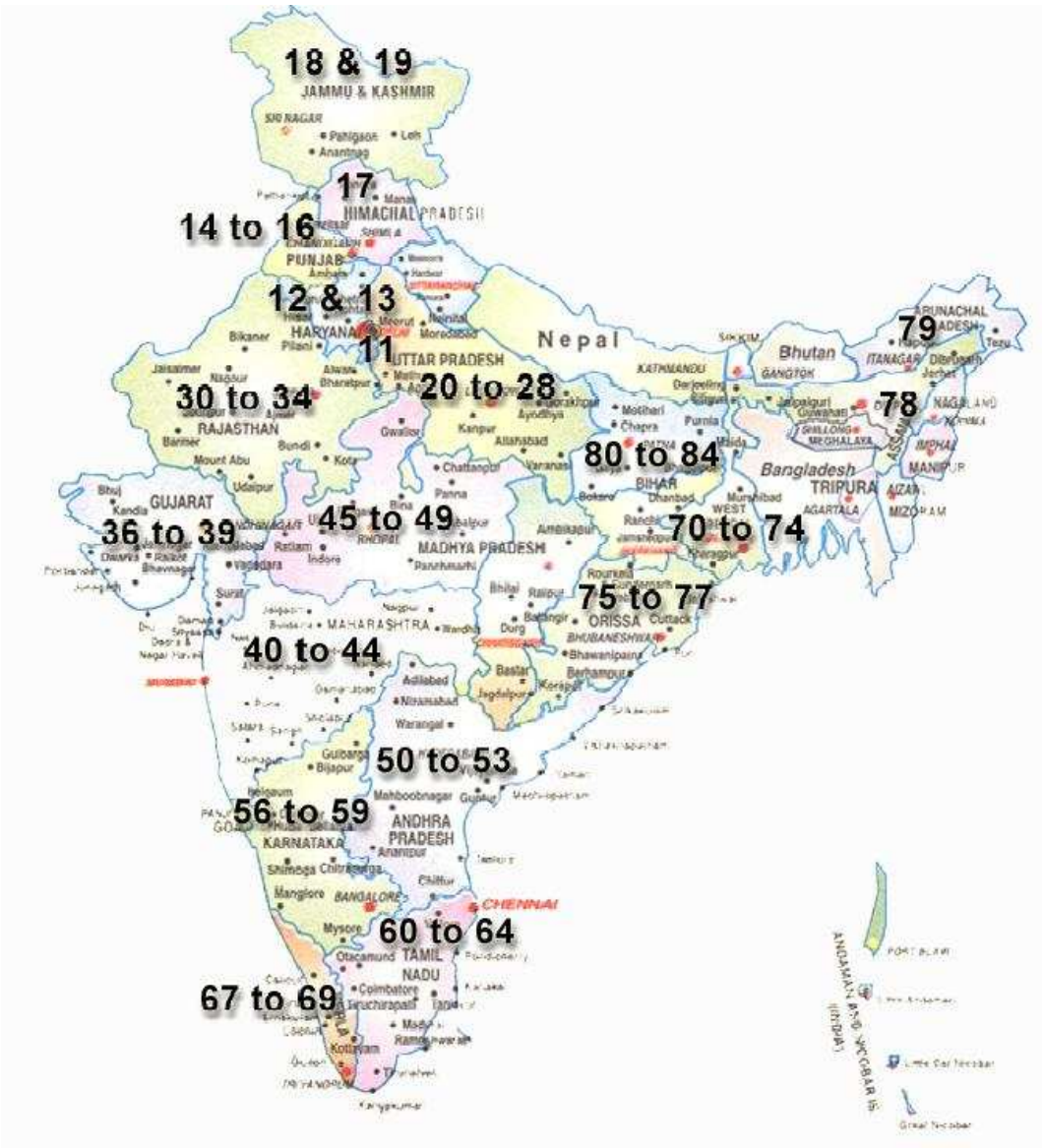

FIGURE 2.6 – Indian postal codes distribution on the map of India

2.1.5 Conclusions

Considering the postal address recognition subject, we can conclude several things. Due the growing traffic of postal documents all over the world the designed and development of such systems is top priority for the different Postal Services. During the development of such systems we can encounter different challenging issues like : find the right orientation of the document, perform noise cleaning, locate the destination address block, segment the DAB in lines and words, spot the city/town name and the pin-code and finally the recognition.

All these issues have been addressed in different scientific papers but in the whole literature you can find just a few pieces where complete systems are described with the corresponding details. The different research groups are focusing on specific problems like segmentation, DAB location, word recognition or digit recognition. Due to the waste amount of different postal datasets used for test purpose it is quite impossible to compare the results of the different systems.

Finally, as this thesis is focusing on Indian postal document recognition we should note the difficulty of this task. As it was described the Indian documents are much more complex than other documents due to the multi-script environment which India has. The addresses are often written in different scripts and the quality of the medium is changing, so, often the image acquisition is also low. In the mean time another challenge is that nobody has worked in handwritten Bengali word recognition.

In the next few sections we will discuss in detail the word recognition achievements as well as the different attempts for handwritten digit recognition focusing on issue like feature extraction, feature combination in order to allow a global view about the existing systems and models. For digit recognition we are focusing more on neural network strategies and classifiers. We are discussing such a point of view of these issues, because we would like to show what kind of extensions we are proposing in this research work.

2.2 Handwritten word recognition

2.2.1 Introduction

The recognition of handwritten words by computers is a challenging task. Despite the impressive progress achieved during the last few decades and the increasing power of computers, the performances of the handwriting recognition systems are still far from human performances. Words are fairly complex patterns owing to the great variability in handwriting style ; handwritten word recognition is a difficult matter.

The first difficulties are due to the high variability and uncertainty of human writing. Not only because of the great variety in the shape of characters but also because of the overlapping

and the interconnection of the neighboring characters. In handwriting we may observe either isolated letters such as hand printed characters, groups of connected letters, i.e. sub-words or entirely connected words.

Furthermore, when observed in isolation, characters are often ambiguous and require context to minimize the classification error. The most natural unit of handwriting is the word and it has been used by many HWR systems. One of the main advantages using whole-word models is that they are capable to capture within-word co-articulations. When such whole word models are adequately trained they will usually yield the best recognition performance. Global or holistic approaches treat words as single indivisible entities and attempt to recognize them as whole, bypassing the segmentation issue. Therefore for small vocabulary recognition such as bank check reading applications, where the lexicon does not have more than 30-40 entries, whole-word models are the preferred choice.

While words are suitable baseline units for recognition, they are not a practical choice for large vocabulary handwriting recognition. Since each word has to be processed individually and data cannot be shared between word models, this implies prohibitively large amounts of training data. Instead of using whole-word models, analytical approaches use sub-word units such as characters or pseudo-characters called also *graphemes*, requiring the segmentation of words into these units.

The second type of difficulties lie in the segmentation of handwritten words into characters. While in case of hand printed characters, the segmentation is not so difficult, as the characters are more or less written separately. For cursive words, this task becomes very difficult.

Even with this difficulty and errors introduced by the segmentation, the most successful approaches are segmentation based recognition in which words are firstly segmented into characters or part of them and after that dynamic programming techniques are used driven by a lexicon to find the best word hypothesis.

2.2.2 Handwriting recognition systems

Considering the different handwriting recognition systems they can be classified concerning different criteria like :

- the nature of features used by the different systems
- the size of lexicon considered by the system
- the analyzed shape is considered as an entity or not (analytical vs. holistic)
- the analyzed script is printed or handwritten
- the nature of the recognizer

Our classification criteria adopted in this thesis will be based on the nature of the input as we can consider systems where low-level features are used, others where the features contain a

semantical aspect transmitted by the human vision and more recently the systems combine the discriminative power of these low-level and perceptual features. A special section will be dedicated to the $2D$ bi-dimensional models as our extended model is also based on $2D$ architecture. Considering such a classification will allow us also to describe the different systems considering the lexicon and the writing dimensionality too. The writing dimension is defined as the dimension of the writing which can be considered as a pattern realized on a 2D plan. This classification is considered as being important as our improvements proposed in this thesis are also based on such criteria.

Low-level features based handwriting recognition systems

Over the last several years, machine learning techniques particularly when applied to neural networks, have played an increasingly important role in the design of the different pattern recognition systems.

As stated in [LBBH01], better recognition systems can be built by relying more on automatic learning and less on hand-designed heuristics. In the case study for separated handwritten digits, the authors show that hand-crafter feature extraction can be replaced by carefully designed learning machines (classifiers) that operate on pixel level.

In a classical pattern recognition system a feature extractor gathers the relevant information from the input pattern and then a trainable classifier categorizes the relevant feature vectors into classes. The new idea was to rely on as much as possible on the feature extraction itself. Precisely in such a case the classifier could be fed with almost raw images and the feature extraction process is embedded in the system which can extract the different features and in the same time is able to learn them.

In [KFK02] the authors are using a basic classifier based on the Euclidean distance for unconstrained handwriting but the feature extraction is tremendous. After preprocessing containing skew correction, slant removal, a script identification is performed. The line segmentation into words is based on horizontal projection. The word segmentation into characters is based on a technique which automatically extracts the required knowledge in the form of *IF-THEN* rules. Once the segmentation is finished for each character, a 280 dimensional feature vector is extracted. After a size normalization to 32×32 from each shape the horizontal and vertical profile is extracted containing the number of black pixels counted during the sweap.

They also define some new features by radial histogram the number of black pixels existing on a rad that starts from the center of the character matrix and ends at its edge. The radial histogram is calculated by rotating the rad by step of 5 degrees. Additionally an out-in radial profile is defined as the position of the first black pixel on the rad, looking from the center of the character to the periphery.

Elms et al. proposed a comparison study [EPI98] between a commercial OCR and an HMM approach for faxed word recognition. In such documents, transmissions distortions can be observed. The results achieved by the HMM approach are greater than the OCRs results. Here the researchers have been concentrating on the problem of isolated character recognition, assuming that words are easy to segment into characters prior to character recognition, whereas the differences between images from books and faxes is that the facsimile images commonly have the characters blurred together, making them very difficult to segment. For the OCR the OmniPage has been used, while for the HMM the characters have been viewed as a sequence of columns. For each pixel column the shape aspect (the arrangement of pixel values within the line) and the location (the position of the line with respect to preceding lines in the sequence) have been considered.

The considered HMM is a classical Bakis-chain, where the number of states have been set-up based on average length of observations to be modeled. For training purpose the classical Baum-Welch formula has been used. While the reported results for the OCR are much more better for clean documents the superiority of the HMM model powered by a lexicon is shown for noisy faxed inputs.

In order to solve the problems raised by the different affine transformations in [SLD94] the authors define a new distance measure which can be made locally invariant to any set of transformations of the input and can be computed efficiently. The metric so-called *tangent distance* is based on the iteration of a Newton type algorithm which finds the points of minimum distance on the true transformation manifolds. The test results shown that the algorithm can handle a rotation in the range of $(-15^{\circ}, 15^{\circ})$. It is mentioned that other spaces than pixel space should give better results.

The method presented in [CK00] has been used for handwritten Arabic word recognition for a reduced size vocabulary. The approach does not require segmentation into characters and it is applied to a script, where ligatures, overlaps and style variations pose challenges to the segmentation-based methods. While the other methods extract high-level perceptual features, in this method there is no need for such an extraction process.

The authors propose to transform each word in polar coordinates, then apply a two-dimensional Fourier transform to the polar map. The resultant spectrum tolerates variations like size, rotation and displacement which can often occur in handwriting. For this purpose just half of the Fourier spectrum was used and just the lower frequencies have been selected. As classifier the simple Euclidean metric was used. The word templates were built using an average of the coefficient values.

The obtained results (93% accuracy) for both printed and handwritten words are encouraging. The extension of the model to a large vocabulary becomes difficult due to the resemblance of the shapes.

To absorb the rotation for handwritten characters, the authors in [CCB04] propose a dynamic network topology. To preserve as much as possible the available information the raw image is considered as input.

The interest is to handle dynamically the network architecture by taking into account the rotation variation of the analyzed shape. In that sense the rotation problem in $2D$ is transformed into a $1D$ problem which is easier. The given results are performed for reduced vocabulary size like 30 characters, where some classes are grouped based on similar shape considerations.

The comparison study has shown the superiority of this method among the others like *Fourier transform* or *Fourier-Mellin transform* but a net superiority can be achieved if the character is deslanted.

For omni-font English and Arabic open vocabulary in [BSM99] a complete OCR system is described. The system is script-independent, the feature extraction techniques based mainly on low-level information are also script -independent and for modeling and recognition purpose a segmentation-free technique have been used. The analyzed shape is divided into overlapped frames which is a system parameter. And each frame is decomposed in 20 cells as presented in Fig. 2.7.

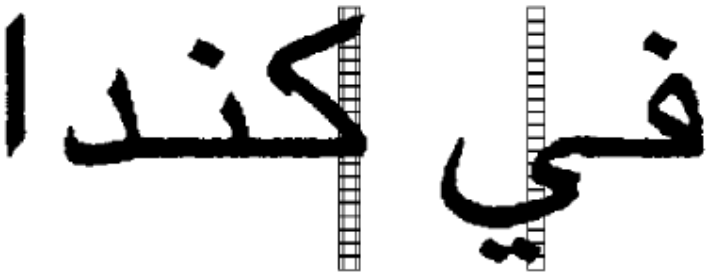

FIGURE 2.7 – For Arabic handwriting each line of text is divided into frames and each frame is divided into cells [BSM99].

As features some low-level features have been used like : intensity (percentage of black pixels within each cell) as a function of vertical positions, vertical derivative of intensity, horizontal derivative of intensity and local slope and correlation across a window of two cells to avoid to extract script dependent features. The results is a set of 80 simple features per frame. For letter models a 14 state left-to-right HMM is used. The achieved recognition scores are excellent but the data is clean data with no much variations.

For writer dependent vocabularies containing 150 words, Bunke et al. [BRST95] propose a Hidden Markov Model based technique. The input vector is composed by shape descriptors extracted form the skeleton graph. These features are somehow at an intermediate level of abstraction, providing a good compromise between discriminatory power and extraction reliability

and reproducibility. The disadvantage of the system is the assumption of a cooperative writer who is willing to adapt his or her personal writing style such that the recognition performance of the system is improved. The ISADORA system used for this purpose allows to use a highly flexible HMM-based pattern recognition architecture to build structural models from simple constituents. The number of states for each letter HMM is fixed in function of some heuristics based on the number of minimal edges for the given letter in the skeleton graph. The word HMM is a concatenation of the letter HMMs. So all the words in the vocabulary share the same letters which allows to have a much more larger training data.

We can conclude than a correct recognition rate of over 98% can be considered as a quite satisfactory result but the data has quite good quality without much variability. Hence an eshaustive comparison with the other techniques using noisy data is not possible.

For postal OCR system, Kornai is proposing an experimental HMM approach [Kor97] based also on some low-level features extracted from a height normalized word shape to 64 pixels using sliding window technique and feature extraction by pre-segmentation. The features coming form the window frames are based on the pixel density, upper/lower contour, etc. While for the sliding window method a 12-16 dimensional feature vector is proposed for the segmentation is based mainly on valleys (local minima) in the contour. To increase the perplexity of the system some language models based on the vocabulary have been developed and implanted in the system. The results obtained for handwritten zip code (84,5%) coming from the CEDAR dataset is quite good but the second experiment concerning the city/state name recognition (63,6%) has shown that such a method is not tuned for such a task.

Using discrete HMM for word recognizer the authors in [GB03a] propose an interesting feature selection technique based on two empirical observations : 1) two HMMs classifiers using different feature sets but the same HMM topology often have similar (or identical) paths for the correct class. 2) the HMM with the highest score given one feature set is also very often among the HMMs with very high scores using another feature set.

After a slant and skew correction and a normalization procedure a sliding window is moved from left to right over the word. The extracted features are : the proportion of black pixels in the window, the center of gravity, and the second order moments. These features are characterizing the word from a global point of view. The other features like the position of the upper and the lowermost pixel, the number of black and white transitions in the window and the fraction of black pixels between the upper and the lowermost black pixels is considered. As lower and upper case are considered for each letter a HMM is built. The character HMMs are concatenated into word models, so this approach allows to share training data across different words. The result of 77,2% can be considered a good score if we consider the fact that 2,296 word classes have been used for the different experiments.

Even if is still a $1D$ HMM model, Park and Lee in [PL96] propose a combination of 4 discrete

linear HMMs to recognize handwritten Hangul characters belonging to a large vocabulary. Each HMM has as input a given regional *projection contour profile* (RPCP) like horizontal, vertical, horizontal-vertical and diagonal-diagonal to consider the all possible senses of writing. Such an RPCP allows to transform a compound pattern or a multi-contour pattern into a unique outer contour. The combination is based on the idea that classifiers with different methodologies or features are usually complementary to each other. Fur this purpose weight combining and majority voting [BVM+04] were used.
This approach allows us to think that a simple linear HMM is not sufficiently enough to consider the dimensionality of handwriting and more sophisticated methods should be proposed for such a task where not just the temporal aspect should be preserved but the spatial aspect too.

So far the different system presented can be classified as mono-dimensional $1D$ systems that means the models developed consider the handwriting as a one-dimensional signal. Namely the observation symbols are coded accordingly as presented in the pioneering work of Rabiner [Rab89].

As stated before, a truly $2D$ extension of the architecture raises high computational complexity problems, some scientist have proposed different techniques to bypass this drawback. Our intention is not to give an exhaustive survey of these systems but to review some of the more interesting ones.

Bi-dimensional system architectures using low-level features

An innovative idea is proposed by Levin et al. [LP92] to model handwritten digits. As stated by the authors the one-dimensional models proposed by Rabiner for speech cannot work properly for signal which are $2D$ in their nature, more precisely $1D^{1/2}$ as handwriting is.

To bypass the handwriting constraint, they propose a planar modeling for the handwritten digits where the information is pixel based considering the color of each pixel composing the word shape. The results achieved using this new technique has shown the force of the model to deal with handwriting and the choice of the pixels as input seems to be a good solution.

The *dynamic time warping* (DTW) known as a suitable solution to match a reference vector against an extracted measure vector giving an exact distance, should be extended to *dynamic planar warping* (DPW). One solution is to divide the image into sub-images where the classical warping function can be found but such solution is sub-optimal as stated by [DE87]. Therefore the algorithm is impractical for real size images. The solution proposed by the author is to impose some constraints in the model to be able to reduce the algorithm complexity as being polynomial.

The idea is to limit the number of admissible warping sequences in such a way that an optimal solution to the constrained problem can be found in polynomial time. The additional constraints used are not arbitrary, but instead reflect the geometric property of the specific set

of images being considered. Considering a statistical independence among the image columns, the authors have introduced the *PHMM* (*Planar Hidden Markov Model*) or *Pseudo* 2*D Hidden Markov Model*. Each local state in the PHMM was represented by its own binary probability distribution, i.e., the probability of a pixel being 1 (black) or 0 (white).

The achieved results for separated handwritten digits show the superiority of the model. The constraints imposed at the beginning help to reduce the computational complexity drastically and find the optimal solution in linear time which designates the PHMM as a powerful tool in $2D$ object recognition problems.

Based on the work proposed by Levin [LP92], Gilloux proposed a new system for handwritten digits recognition [Gil94]. The PHMM (see Fig. 2.8) observes the pixel colors. Such a low-level approach adopted also by Gilloux, shows its importance among the others, where perceptual features have been considered. The PHMM used here can be considered as a continuation of the basic PHMM proposed by Levin but it was extended in different points. In that case the model structure is different. Instead of considering the distribution in the super-state of the PHMM, the authors consider super-state classes, where the secondary HMM states can be integrated. The approach is really innovative as the distribution is calculated not column-wise but state-wise allowing to model more precisely the $2D$ deformations of writing. This method also preserve the hypothesis concerning the independence between the different image columns.

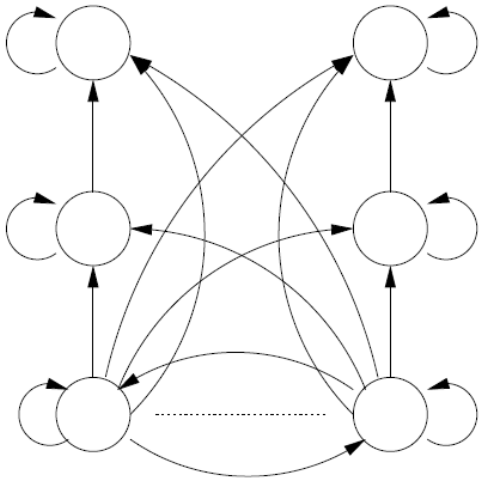

FIGURE 2.8 – The PHMM proposed by Gilloux [Gil94].

A major contribution of the author is the usage of the Markov network which can be trained with exponential complexity. The training process of such a model is exponential as stated also by Levin [LP92]. However, considering that the information repartition is given by the previous PHMM, the bi-dimensional dependency between the states can be calculated directly as their distribution is given a priori. The drawback of this model is than the repartition of the

information in the different states is sub-optimal. Even if the repartition is correct (which is not necessary assured) the algorithm is based on Viterbi search which is of course a sub-optimal search mechanism.

Park and Lee propose a totally bi-dimensional Markov model [PL95], namely the Hidden Markov Mesh Random Field (HMMRF) for handwritten character recognition. The images are decomposed in $n \times n$ windows where the black pixel density is considered as observation for the model. The authors propose in this model a new decoding algorithm which allows to preserve the completely bi-dimensional relation between the different observation. For this reason the decoding is based on the hypothesis called "look-ahead" which means the marginal distribution is considered as being optimal.

The experimental results reported by the authors concerning the digit database of Concordia University outperform the results reported by $1D$ linear HMMs or PHMMs for the same dataset.

For printed Arabic word recognition, Amara [ABE98] propose also a PHHM without any a priori segmentation. The approach is global trying to model pseudo words (see Fig. 2.9) occurring often is Arabic which is a semi-cursive script. A word can be constructed from up to 10 pseudo-words called also *PAW*(*Piece of Arabic Word*). Such a modeling approach is considered because these elements can quickly be isolated in the script using connected component finding schemes. Even if we can find them in different positions they not change very much their shape while the letters can have different shapes in function of their position in the word.

The topology used here is derived directly from the input as the horizontal pixel sequences having the same color are considered as being the observations. The observation is dependent on the duration of the identical pixel sequence and in the mean time for the black color sequence the immediate upper neighborhood is considered. This allows to highlight the correspondence between the image lines. The secondary HMM observes the image lines more precisely the succession of black and white pixel sequences, while the main HMM states observe the succession of the lines which provides to the model a bi-dimensional aspect. The results achieved for printed Arabic city names is excellent (96.87%-100%) but the size of the vocabulary is reduced. 100 PAWs have been considered containing up to 3 characters.

As stated also by Choisy [Cho02] this work shows the generality of the model used for Arabic script also and in the same time the discriminative power of the pixel information characterizing the different pseudo words in a pseudo $2D$ representation.

Derived from this theory based on the extension of the DTW to DPW, Saon [Sao97] proposed a system so called NSHP-HMM (*Non Symmetric Half-Plane Hidden Markov Model*) for the recognition of handwritten words on literal bank check amounts. The designed scheme (see Fig. 3.4) combines advantageously a HMM (*Hidden Markov Model*) and a MRF (*Markov Random Field*). It operates on pixel level, in a holistic manner, on height normalized images which are considered as random field realizations. The HMM analyzes the image along the horizontal

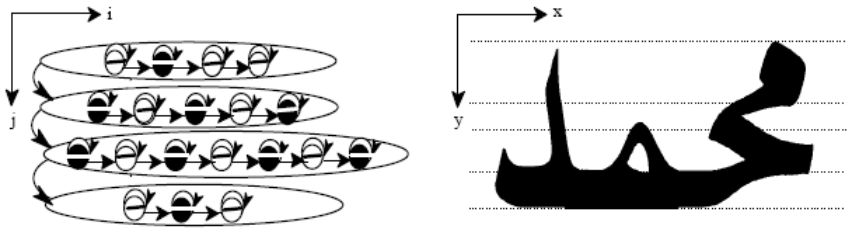

FIGURE 2.9 – The corresponding PHMM for the Arabic paw [ABE98]

direction of writing, considering in the different states of the HMM the observation probabilities given by the different image columns estimated by causal MRF-like pixel conditional probabilities. Since the considered vocabulary has a reduced size containing just 26 words such a holistic method is applied. No grapheme segmentation step is required, so the commonly encountered under or over-segmentation problems are avoided.

To extend the previous system, Choisy proposed to introduce in the NSHP-HMM an implicit segmentation [Cho02]. This system (see Fig. 3.1) is also based on pixel column observations produced by the NSHP but instead of using general word models as in case of Saon, the author proposes to build a general word NSHP-HMM. The word model based on letter NSHP-HMMs and word meta-models is able to re-estimate the letter models and the ligatures between letters throughout the general word model. Such kind of re-estimation is much more precise as the basic letter HMM concatenation used so often in the literature.

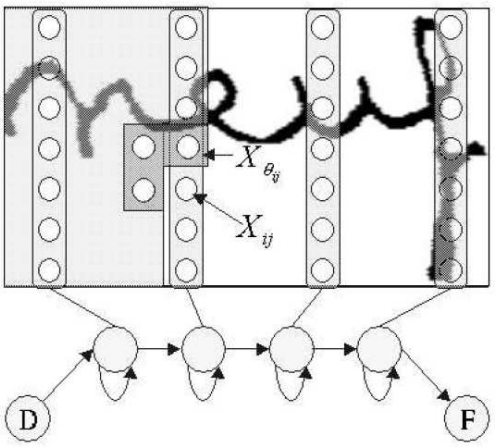

FIGURE 2.10 – The NSHP-HMM considered by Choisy for bank check amounts recognition

For this scheme a *cross-learning mechanism* [CB02] was developed and implemented with success for different handwritings belonging always to reduced size vocabularies. The cross-learning resides in the classical re-estimation of the global word model and based on this re-estimation, the letter models and the word meta model are also re-estimated allowing to consider the dif-

ferent context for the letter models and the different ligatures for the word meta models. The novelty of the approach resides in the training mechanism based on the convergence of the well known Baum-Welch algorithm [Rab89] and the information dispatch in the meta-models and letter models considering the general word models. While for the training such re-estimation flow is considered, to test the models the general word models are built based on the letter models and the meta models respectively.

For Arabic handwriting recently Touj et al. [TNEBA04] consider a planar architecture for modeling and recognition. The scheme proposed is based on the work of Levin [LP92] where five different horizontal HMMs have been considered. Each of them is associated to a one horizontal zone of the Arabic handwriting. The different zones are : upper diacritics zone, upper zone part, middle-zone or busy zone, lower zone part and lower diacritics respectively. These HMM are considered as being the observations for the up-down HMM which models the variations between the different writing zones considering also the different morphological variations of Arabic script. In that sense the segmentation procedure for such a scheme is vital. The segmentation is subdivided into four parts : firstly a horizontal segmentation followed by a vertical one in the middle zone is performed while the third and fourth segmentation concerns the position of the graphemes associated to extensions and diacritics. After the segmentation process a feature extraction is performed based mainly on perceptual features like diacritics which can be distinguished based on their dimension and pixel density, ascenders and descenders and in the middle zone containing a large variety of information a 8 dimension vector is extracted.

The results obtained on the IFN/ENIT dataset containing handwritten Tunisian city names are encouraging (72%) but considering the size of the vocabulary (25 entries) the results steps behind. As mentioned above the main drawback of the system is its sensitivity to the different geometrical transformations as the different horizontal parts of the writing should be clearly distinquishable.

Considering the same baseline scheme as Touj, Wang et al. in [WBKR00] propose a HMM based modeling together with an extended sliding window feature extraction method to decrease the influence of the baseline detection error. The results shown that the model can achieve better recognition performances and reduce the error rate significantly compared with classical models.

The coding of the frames into observation has a weak point. The generated feature vectors are depending upon the accuracy of the baseline detection. As stated by the authors such a reliable detection method does not exist so they are proposing a new feature extraction scheme which is much more tolerant to the errors committed by the baseline extractor. The new feature vector is composed by local means of the different writing zones divided in frames were the percentage of black pixel is calculated.

To achieve higher accuracy in [MG96] the authors combine a segmentation-free technique based on matching with a segmentation based one, where dynamic programming has been used.

The feature extraction is based on low-level features extracted from each image column like location and number of transitions from background to foreground pixels along the processed image vertical lines (columns). The combination based on thresholds and Borda count is successful but the results of the classifiers are still not satisfactory due to the sensitivity of the models to the different slant and skew modification.

In summary, as we can observe the different handwriting recognition systems working on low-level features achieve good recognition scores for small size and middle-size vocabularies but they are very sensitive to the different variations, distortions introduced by the writer, the writing device and the digitization process.

For a reduced vocabulary like separated digits, the superiority of the neural based approach instead of the stochastic one is considerably. The results can be explained by the fact that in case of digits the number of classes is reduced, the variability is not so huge while for words recognition such technique does not work satisfactory as just a stochastic model considering the temporal aspect of the input signal is able to model correctly the cursive handwriting.

High-level perceptual features based handwriting recognition systems

For off-line unconstrained handwritten word modeling and recognition [YGSS99] El-Yacoubi et al. proposed a hidden Markov model-based approach designed to recognize handwritten words for a large vocabulary. To reduce the irrelevant information such as noise and intra-class variations, a four step preprocessing mechanism is proposed. Firstly a baseline slant normalization is performed, followed by a lower letter area (upper-baseline) normalization and when dealing with handwritten cursive words character skew correction. Finally a smoothing is applied in order to be able to extract features like ascenders, descenders, loops, etc.

As a context is available the feature extraction is performed at segment level but considering also the positions of the loops. The explicit segmentation is based on image upper contour minima allowing to the segmentation to propose a high number of segmentation points. After the extraction of global features (27), a feature set based on the analysis of the contour transition histogram is performed (14 symbols). Also some segmentation features (5) have been used.

The complex letter method presented in Fig. 2.11 allows to model the different letters as a succession of two or three graphemes. To model the different words written on lowercase or uppercase, two parallel models have been integrated in the general word model to be able to consider the word in uppercase the word in lowercase and a mix-up of lower-case and upper-case letters. The results obtained for real French city names extracted manually from the envelopes are excellent considering the huge variability of writing in the dataset size. (10w, 100w, 1000w) (99,02%, 96,3%, 87,9%) without any kind of rejection criteria. We should also mention than these high-class results were achieved by considering the information coming from the pin code

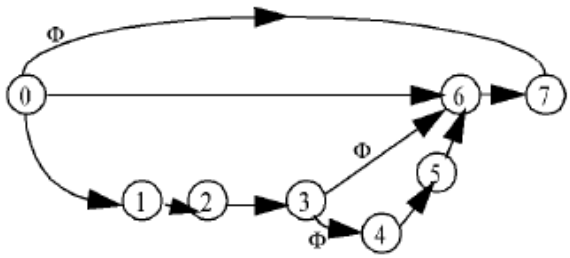

FIGURE 2.11 – The complex letter model considering the different graphemes proposed by El-Yacoubi et al. [YGSS99]

recognizer having a confidence value.

Similar approach can be found in [Koe02] where the author discuss in details this letter modeling aspect with special consideration to the complexity. If we consider a large vocabulary and we are taking into account all the occurring possibilities to mix uppercase and lowercase characters in the same word, the number of decoded states blows up.

Up to now this problem has not been addressed in handwriting recognition. The complexity of the search in lexical trees using multiple character models is a real challenge. To overcome the complexity of the problem, Koerich uses the maximum approximation to select only the more likely combination, considering the local context.

Guillevic and Suen [GS95] propose a method for recognizing unconstrained, writer independent handwritten cursive words belonging to a small static lexicon, i.e. legal bank check amounts. After preprocessing, slant correction mainly, amount segmentation into words and extraction of global features for the recognition module are performed. Seven types of global features are extracted from the word image : ascenders, descenders, loops, estimate length of the word, vertical strokes, horizontal strokes, diagonal strokes. Threshold for ascenders and descenders are determined empirically and are expressed as a percentage of the main body height. Word length is estimated as the number of central threshold crossings. Strokes are extracted using mathematical morphology operations. For classification purpose nearest neighbor classifier is used.

Madhvanath et al in [MG01b] discuss the use of holistic features for an address reading classifier implemented at CEDAR. Features used by the system are word length, number and position of ascenders, descenders, loops, and points of return. Macro features or composite features such as "ff" and "ty" are also extracted and used to enhance the classifier scores. Feature equivalence rules provide means of normalization among different styles.

An innovative fuzzy approach is proposed by Rodrigues and Ling in [RL01] to extract features from handwriting based on a corpus of Brazilian bank checks and to classify them with a fuzzy

Hidden Markov Model. After a pre-processing phase containing smoothing, rotation and slant correction, the word image is segmented on different line segments. For each line segment a fuzzy method is considered to establish its membership to different type of curve lines or straight lines. Such representation allows to reduce the variability of handwriting.

The feature extraction is based on the idea that is possible to recognize a letter by observing the position and the type of its segment lines. Scanning the word segment from top to bottom three different line segments can be found according to the top, middle and bottom part. Similarly a horizontal left to right scanning allows to distinguish between left, center and right word segments. As the pre-processing cannot avoid totally the handwriting variability, the authors have proposed fuzzy sets to deal with this variation and to obtain membership functions in each of the 6 cases. To apply such a procedure it is necessary to decompose the word segments in line segment so called branches. The membership functions are generated from a relation between the amount of points in the branch and its word segment. These branches can be : vertical lines, lines with positive inclination, lines with negative inclination, horizontal lines, C type curve, D type curve, A type curve, U type curve and Z type curves. Holes are also considered but this type of feature is not represented by any fuzzy set. After a classification based on membership, a feature vector can be created, where an element is a membership value to a fuzzy set representing a line segments regarding its position in word segment. Considering such a codebook, the authors propose a parallel Fuzzy Hidden Markov Model (FHMM) which is a concatenation of letter models able to handle the uppercase, the lowercase and the mixture of characters inside the word. Some post-processing are also performed in order to improve the system accuracy based on the position of ascenders and descenders in the analyzed word. The general performance of the system is very low (50%) considering the performances of other systems working on similar datasets [FYBS00].

Hybrid features based handwriting recognition systems

To avoid the restrictions imposed by the $1D$ model, in [FGB98] the authors propose a unidimensional HMM model but the observation symbol observed by the discrete HMM is a combination of low-level features based on analysis of horizontal and vertical projected transition histograms and another set of features devoted to the representation of cursive script based on detection of holes, ascenders and descenders. As the system handles large vocabulary, the HMM model is designed using elementary letter or graphemes models which are concatenated to create the word models.

The used letter model is able to consider the cases when a letter is segmented in 1, 2 or 3 sub-images or segments. Even for such a discrete HMM model the results obtained for different vocabularies is considerable. (10w -98,7% ; 100w -94,3%, 1000w - 86,5%). To improve the results

a contextual HMM is designed using the context of the sub-images in the form of its two neighbors. So 4 multi-layer perceptrons have been designed to recognize the elementary image, the elementary image with the left context, the elementary image with right context and the segment with left and right context. Even if the MLP recognizers' performance it is not satisfactory (51,5%) for the fusion the new features increase considerably the recognition scores for the same datasets (10w - 99,3% ; 100w - 97,4%, 1000w - 93,3%).

This system is a hybrid system concerning the used features and the outputs of the recognizers are used as observations in the HMM rather than being directly combined into a word recognition score as in segmentation by recognition approaches.

A new strategy is proposed for improving feature sets in a discrete HMM-based handwriting recognition system [GS00]. The strategy proposed by Grandidier et al. are integrating several information sources from specialized feature sets. The basic idea is to retain the most discriminative features and to replace the others with the new ones obtained from new feature spaces. This idea comes from some observations obtained from the evaluation of the SRTP [2] handwriting system described in [GSEY+99]. The authors have concluded the followings : 1) the word length has a strong influence on the recognition performances. 2) in case of long words the system has more features and contextual information. 3) the presence of the most discriminative features is more probable in long sequences of observation.

Firstly an evaluation should be performed in order to calculate the discriminative power of each single feature. For this purpose the conditional perplexity was chosen. This indicator is based on the statistical notion of entropy and perplexity. The conditional entropy is defined as follows :

$$H(f_j) = -\sum_{i=1}^{N_c} p(c_i \mid f_j) \cdot logp(c_i \mid f_j) \tag{2.1}$$

where c_i are the classes considered in the modeling and N_c the number of those classes. $H(f_j)$ quantifies the capability of feature f_j to discriminate between the classes c_i.
The conditional perplexity $PP(f_j)$ of a feature f_j is obtained from the relation :

$$PP(f_j) = 2^{H(f_j)} \tag{2.2}$$

The conditional perplexity quantifies the capability of each single feature to discriminate between all classes, without the help of recognition results so called in the feature selection branch as *filter method*. To quantify the discriminative power of a feature set, the global entropy H is calculated :

$$H = \sum_{j=1}^{N_f} p(f_j) \cdot H(f_j) \tag{2.3}$$

2. Service de Recherche Technique de la Poste

where N_f is the number of features and $p(f_j)$ the a priori probability of the feature f_j. Using this formalism it is possible to rank a feature set E_i according to its conditional perplexity values.

Secondly, in the proposed strategy the features judged non-discriminative should be replaced. For this reason a new feature space will be used to obtain a new characterization of the information present in the handwriting. This procedure is called descent of the perceptual level. As the features can be considered as the perception of the shape by the recognition system ; then the shift of feature space can be considered as a change of the perceptual level.

The novelty of the technique is to discard the non-discriminative features by replacing them with others judged more significantly and creating a new observation space which describes better the shape. Some similar techniques can be found in the literature but their action mechanism is different.

Based on classifiers combination [RF03] for each feature space a classifier can be built and after based on some heuristics (softmax, majority voting, weighted majority voting [BVM+04], etc.) a combination is performed to find the optimal solution. Such kind of methods give good results when the complementarity is considerable.

Two classifiers are complementary : the errors committed by one classifier can be corrected by the other and vice-versa. If the classifiers are committing the same type of errors such types of combination are useless.

In [FYBS00] the authors developed a system for handwritten legal amount recognition of Brazilian bank checks using a global approach not requiring an explicit segmentation and the word modeling is supplied by HMMs. To extract robust features a slant correction and a smoothing process have been applied to regulate the continuous contour of the word, eliminating low noises in the image.

Considering the lexicon containing 39 words, we can find sub-groups of words where no perceptual features are present so this well known feature set cannot be applied. This occurs mainly in words like "um", "cinco", "seize", "nove" etc., where there is no context as the words are short or does not contain ascenders/descenders which can distinguish the word shapes. For that reason a second feature set is proposed based on representation of concavities and convexities that exist in the middle zone of writing. As just a reduced vocabulary is considered a model discriminant left to right Bakis model is used for model and recognition purpose. Considering the results as expected a better representation especially for the words with an absence of perceptual features can be observed. Unfortunately the recognition accuracy reported is not sufficient (67.7%) even if we consider the fact that the lexicon size is slightly different from cases when English or French legal amount are considered.

Another strategy for improving the performance of the recognition system is to combine the feature sets by constructing their Cartesian product [GSEY+99, YGSS99] and creating a new feature space describing the form. The drawback of this approach lies in the exponential increase

in the number of parameters.

Judging not suitable a $2D$ Markov Random Field for handwriting recognition, the authors propose a complete scheme for totally unconstrained handwritten word recognition based on a single contextual HMM type stochastic network [CKZ94]. After some pre-processing like slant normalization and morphological filtering to discard noise, a segmentation is performed. The observation sequence extracted from the segments is composed by 35 elements. The feature vector computation is based on several type of features. Different moment features have been used to capture the global shape information, perceptual features like hole, X-joint, zero-crossing, and to precise, zonal features have been used for the previous topological and geometrical features, and finally pixel distribution features and reference line features have been extracted for this purpose. The same features have been used with success by Kundu in [KHB89] for first order and second order HMM based handwriting recognition.
The path discriminant HMM used does not give high accuracy in Top1 (51,9%) but as in Top5 the results achieve around 91,2%, they propose the usage of a dictionary. A comparison is given using 36 Legendre moments which seems to be worst than the feature set proposed by the authors.

Scagliola et al. [CS00] propose a segmentation by recognition approach, frequently adopted to recognize off-line cursive handwritten words. Considered as necessary, the authors have been pre-processing the image (binarization, skew angle estimation and correction, slant estimation and correction). After an oversegmentation process based on lower and median profiles of the word the different segments are sent than to the hypothesis evaluator which prepares the data for the optimal interpretation algorithm. The hypothesis evaluator is a distance evaluator based on Euclidean distance for each letter shape characterized by a 34 dimension vector composed by 32 local features indicating the proportion of black pixels and their prevalent direction in a grid enriched by 2 global features indicating the height to width ratio and the proportion of the character image height below the baseline. To enhance the recognition performance it was deemed necessary to integrate several other sources of information to contribute additional terms to the overall matching score. The hypothesis generator cannot essentially capture the relationship of the hypothesis under evaluation with respect to such global characteristics of the image as its size and position of baseline and upperline. So some penalties have been introduced in the system based on the location of the hypothesis and its size. The results shown prove the importance of such extra information as up to 20% good recognition improvement can be observed.

In summary, we have considered different kind of features based on their nature used to describe hard patterns as handwritten words. The different features can be classified as low-level features, intermediate-level features and high-level also called perceptual features based on their abstraction level. While the low-level features can be extracted easily in case of the perceptual feature a complex, not always precise extraction mechanism should be called. The low-level features can give a global or local estimation of the form, while the perceptual features are

much more descriptive ones modeling the human vision but their extraction and interpretation is heuristic dependent.

In the scientific community different approaches have been considered for the word modeling and recognition, mainly based on HMM type models where the handwriting was considered as $1D$, $2D$ and $1D^{1/2}$ signal. The common approach considered by the system is the usage of low-level features mainly for $2D$ and $1D^{1/2}$ models while for the $1D$ models perceptual features are considered.

To improve the accuracy of the systems there are some attempts to use hybrid features, but sometimes these are considered separately or even if there are integrated in the same system there is no conditional relationship between. We consider this aspect as being crucial as in human vision these features are not separated. The shape is considered as an entity with the features extracted from it.

2.2.3 Lexicon reduction strategies in handwriting recognition

As we can observe nowadays for the different handwriting recognition systems three main approaches have been proposed : the *holistic methods*, where the whole words shape is considered to bypass the difficult problem of segmenting the word into its individual parts (letters/graphemes), the *segmentation based approach* trying to segment the given word in smaller entities and the *HMM based approaches* which seems to be a suitable tool for cursive script recognition.

While the holistic methods give interesting results for reduced size vocabularies, they cannot handle larger vocabularies. Similarly, the segmentation based techniques have the advantage to reduce the complex problem of word recognition to isolated character recognition but the segmentation and grapheme recombination are both based on heuristics, rules that are derived from the human intuition. The HMM based techniques are the more appropriate ones to solve such a recognition task. They are stochastic models able to deal with noise and shape variations that occur in cursive handwriting. The number of feature vectors representing the unknown word may be of variable length. Such a requirement is fundamental in cursive handwriting because the length of the individual input words exhibits a great degree of variations. This requirement cannot hold for the neural network based approaches, where the size of the input should be fixed.

As we can see, one of the most common constraint of the current recognition systems is that they are only capable to recognize words that are present in a restricted vocabulary [GS95, Sao97, Cho02, TLK+01] typically comprised of 10-1,000 words. The restricted vocabulary, usually called a *lexicon*, is a list of valid words that are expected to be recognized by the HWR system. As there is no established definitions concerning the size of the vocabulary we will use the definitions given by Koerich [KSS03] : a *small vocabulary* contains tens of words, a *medium size vocabulary* contains hundreds of words, a *large vocabulary* counts thousands of words while a *very large*

vocabulary has tens of thousands of word entries.

According to Plamondon and Shrihari [PS00], the ultimate handwriting computer will have to process electronic handwriting in an unconstrained environment, deal with many handwriting styles and languages, work with arbitrary user-defined alphabets and understand any handwritten messages provided by any writer. So, it is unquestionable the importance of large vocabulary handwriting recognition techniques to reach some of these goals. The capability of dealing with large vocabularies, however, opens up many more applications.

As the lexicon is a key point to the success of the different handwriting recognition systems the scientists propose *lexicon-driven approaches* [CGS99] in order to reduce the number of templates/models to be match in models discriminant approaches or the number of paths to be followed in the path discriminant approaches.
Such a lexicon reduction is really necessary because :

1. by reducing the vocabulary size we can reduce the possibility of miss-recognition probability [ZM99, KSSEY00]
2. by reducing the vocabulary there is a considerable time gain parameter which plays an important role in the real-time systems as in case of the mail sorting [DG00, KSS03] where the time factor is really important.

Therefore, it is very important to perform lexicon reduction in the different handwriting recognition systems.

There are several techniques proposed in the literature for lexicon reduction. The most common used techniques are based on holistic perceptual features as : the length of the word, the presence or absence of ascenders, descenders, t-crossings, diacritics, etc. In this approach, holistic word features of the input word shape are matched against holistic features of every model of the lexicon. Lexicon entries which do not match with the holistic features of the input image are discarded. Typically, more than one exemplar must be stored or synthesized for each lexicon entry because of various writing styles encountered. The efficiency of this approach is limited by the computational overhead for extracting holistic features and feature matching with more than one exemplar for each lexicon entry. For the segmentation based system the scheme is more complicated.

Given a sequence of N graphemes and a string (lexicon entry) of length W, the dynamic programming technique can be used to obtain the best grouping of the N graphemes into W segments. A dynamic table of size $(N \times W)$ must be constructed to obtain the best path. Given a lexicon of L entries, the complexity of the lexicon driven matching is $\mathcal{O} = L \times N \times W$. As stated by Zimmerman the speed of the lexicon-driven system decreases linearly with the lexicon size while the accuracy also decreases when the lexicon size becomes larger.

Kimura et al. propose a lexicon reduction [KG97] in which the input image is first segmen-

ted into segments based on a set of heuristics and an ASCII string is generated based on the recognition results of these segments. A dynamic program is than used to match this ASCII string with each lexicon entry. Lexicon entries with high matching cost are eliminated for further consideration. The disadvantage of this system is that it heavily relies on the initial segmentation of words into characters, task which is problematic for cursive words.

To reduce the lexicon for a postal reading system Zimmermann et al. propose a more sophisticated model based on *character spotting* [ZM99]. The notion of *key character* is introduced here. Key characters identify unambiguous characters of cursive words which can be segmented and recognized without performing word recognition or contextual analysis.

The extraction and recognition of key characters work directly on the sequence of graphemes which can be obtained by any oversegmentation methods. The key characters should satisfy some basic properties based on the confidence value attributed by the classifier which recognizes them and in the mean time some geometric constraints (average number of horizontal transitions, normalized vertical position and normalized height of the supposed key character) should also be satisfied.

To power the vocabulary reduction, a length estimation is performed based on a neural network. The goal of length estimation is to provide an estimate of how many characters are present in a given image without performing any expensive recognition. The estimation is based on the connected components, the number of graphemes, the number of horizontal transitions per scan line and the average height of the graphemes. The applied two-stage lexicon reduction is based on a length estimation followed by a key character spotting. The order is based on the fact that examining a lexicon entry in the first stage is much more faster than in a second stage. Using just the length estimation the reduction of the lexicon is 54.4% while the key character method has a 37.5% accuracy which resulted in a total average reduction of 72.9%. The time saving was 54.6%.

Koerich et al. in [KLSS02] proposed a hybrid technique to recognize handwritten words belonging to a large vocabulary. The baseline method is based on a lexicon driven word recognizer based on discrete HMM models which generate a list of candidate N-best scoring word hypothesis ordered according to the a posteriori probability assigned to each word hypothesis as well as the segmentation of such word hypothesis into characters. Once the word hypothesis are omitted by the HMM based on profiles, the different segments are recognized by a classical multi-layer perceptron. Finally, in a probabilistic framework the different results are integrated to allow to assign the final class label for each word. The 10% of recognition rate improvment over the HMM system alone shows the importance of the character classifier in this hybrid scheme.

Another technique for dynamical lexicon reduction for Finnish city names is proposed by Guillevic et al. in [DG00]. This approach is also based on character spotting. In a basic postal application scheme the reduction of the vocabulary is based on the zip code recognition along

with a database of zip code to city name correspondence, to generate a dynamically a reduced lexicon. Here the authors make the assumption that either the zip code could not be located, it was missing or it was simply not recognizable. This lead them to a method called *character spotting*. After some preprocessing techniques to reduce the variability of handwriting a word length estimation is performed based on similar measures as in case of Zimmermann. This kind of reduction brings no more than 50% of gain as most of the city names have a length between 7 and 10 characters. To achieve a better reduction performance the authors are trying to identify and extract isolated characters that do not overlap vertically. This kind of scheme is limited as often neighbouring characters even slightly overlap. Therefore a more sophisticated analysis is performed based on heuristics, contours, etc. If the character spotting fails the word is rejected. Once the spotted bittmaps have been extracted there are sent to the character recognizer. Based on the position and the character label best N-hypothesis of the spotted character, a gramar HMM is constructed where there is no training process, the parameters are fixed manually being the a prori knowledge of the lexicon. Considering the 500 images of the test set, in 7% of the cases the spotting has failed but approximately 95% of the all upper words were correctly processed.

The given system has the same drawback as the system proposed by Kimura, where the spotting depends on the precision and accuracy performed by the segmentation module or on the chance to find or not separated characters in the analyzed word shape.

Shridhar et al. [SKTH02] have considered the issue of *lexicon completeness* to measure the impact on the accuracy in the framework of a Dutch city name recognition system. The *completeness* can be defined as the probability that an incoming word belongs to the lexicon. It has been documented in many handwriting recognition systems that once the size of the vocabulary increase the recognition accuracy falls [KG97, ZM99] but the availability of a partially complete lexicon has not been analyzed. After slant estimation and correction, segmentation points are detected providing over-segmented parts. The lexicon directed algorithm based on dynamic programming is applied, using the total likelihood of characters as objective function.

For the United States Postal Service (USPS) a Handwritten Address Interpretation (HWAI) system is proposed by Srihari [Sri00]. The work presented here can be considered as one of the base systems ever developed for postal automation. After performing separated digit recognition using several classifiers (polynomial, k-NN, etc), the word classification is done either by analytical approach recognizing characters or by holistic approach where the whole word like an entity is considered.

To reduce the vocabulary size the ZIP code results are used. The author presents a direct relation between the size of the lexicon and the number of ZIP codes. The different HWAI clones implanted in different countries like Australia, United Kingdom, Canada have shown the enormous success of the system in the domain of automatic postal sorting but due to the different writing habits and slightly different post-code and address structure, slightly different recognizers

are called.

A new pruning technique has been proposed in [MK97] based on a general word shape descriptor. As the used vocabulary size is huge containing 21,000 words, an efficient pruning procedure is necessary. The authors propose a generalized descriptor based on strokes which provide a coarse representation of the word shape. Downstrokes are identified by matching local extrema on the contour of the cursive word and are stored in a code book as : M for "medium", A for "ascender", D for "descender", F for "f-stroke" and U for "unknown". The pruning is based on an elastic matching of the image descriptor with the ideal descriptor of lexicon entries organized in the form of a trie. The ideal shape is extracted from an "ideally written" word, where : pure cursive style has been used, no baseline skew, no character slant. The ideal descriptor for a given word is built by the concatenation of the ideal descriptors of the constituent words. The trie representation allows a fast matching procedure but the mechanism is still sensitive and depends on the stroke extraction.

A comparison study is performed in [SHK97] between two recognition mechanisms. In first case a lexicon-free approach has been applied while in second case a lexicon-directed recognition mechanism is used. The contextual information is incorporated using the total likelihood of each character. The likelihood of each character is calculated using the modified quadratic discriminant function. With some modification the lexicon-free algorithm can be derived from the other. As codebook for the character recognition local chain code histogram of the characters contour is used. The results performed on USPS postal images shown the superiority of the lexicon bases system (90,38%) while using the lexicon-free system the results cannot achive more than 85,90%.

While the other techniques use some heuristics to prune the lexicon based on length estimation, presence or absence of different perceptual features like ascenders, descenders, diacritics, in [Gil00] Gilloux is proposing a meta-heuristic for a very large vocabulary (up to 100,000 word entries) reduction for handwriting for French proper names recognition.

The meta-heuristic used here is based on *Tabu search* described in detail in [HF98a]. In that recognition scheme the word recognition is considered as a research of an optimal configuration in a configuration space organized on neighboring. The Tabu search is based on searching among the different configuration and going for the optimal solution throughout these configuration. For the searching mechanism a distance measure is necessary but as HMMs have been used for the different word models, a modified *Kullback distance* (described in detail by Rabiner in [Rab89]) has been used.

A lexical post-processing is proposed by Carbonnel and Anquetil [CA04] based on word filtering. The filtering is based on global word shapes in order to build static sub-lexicon during the recognition process. The pertinent information are based on downstrokes composed by : ascender, descender, long and median. Once the word is coded accordingly, the shape is compressed as different successive median downstrokes are replaced by a single one. This coding increases the

importance given to the prominent downstrokes because they are more robust than the median downstrokes. The authors call such a compressed coding a generic shape. (see Fig. 2.12). Using this codebook the shapes can be classified in function of their generic shape representation. For recognition purpose a modified edit-distance is used. The results performed on a 2,5k dictionary are very encouraging.

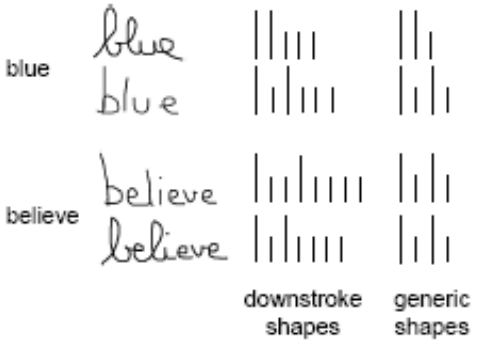

FIGURE 2.12 – Generic word shape coded by segments in [CA04]

For large vocabulary off-line handwriting recognition an exhaustive and systematic survey is given by Koerich et al. in [KSS03, Koe02] where the authors give all the possible details concerning the different solutions on the different recognition strategies developed by researchers during the last decades.

A general recognition scheme for handwriting recognition in the vision of Koerich is presented in Fig. 2.13.

The model begins with an unknown handwritten word which is presented at the input of the recognition system as a raw image. To convert this image into information understandable by computer requires the solution to a number of challenging problems. Firstly a front-end parametrization is needed which extracts from the image all the necessary meaningful information in a compact form. This involves pre-processing like slant and slope correction, smoothing, normalization of the image to reduce the undesirable variability that only contributes to complicate the recognition process.

The second step is relative to the front-end parametrization of the segmentation of the word into sequences of basic recognition units such as characters or character segments (graphemes). However, the segmentation is not necessary to be present. The final step is to extract discriminant features from the input image to either build up a feature vector or to generate a graph, string of codes or sequence of symbols whose class label in unknown. A crucial step in handwriting is the pattern training which consists of the usage of one or more pattern corresponding to handwritten words of the same class to create a representative pattern to this class.

The resulting pattern generally called *reference pattern* or *class prototype* or *template* can

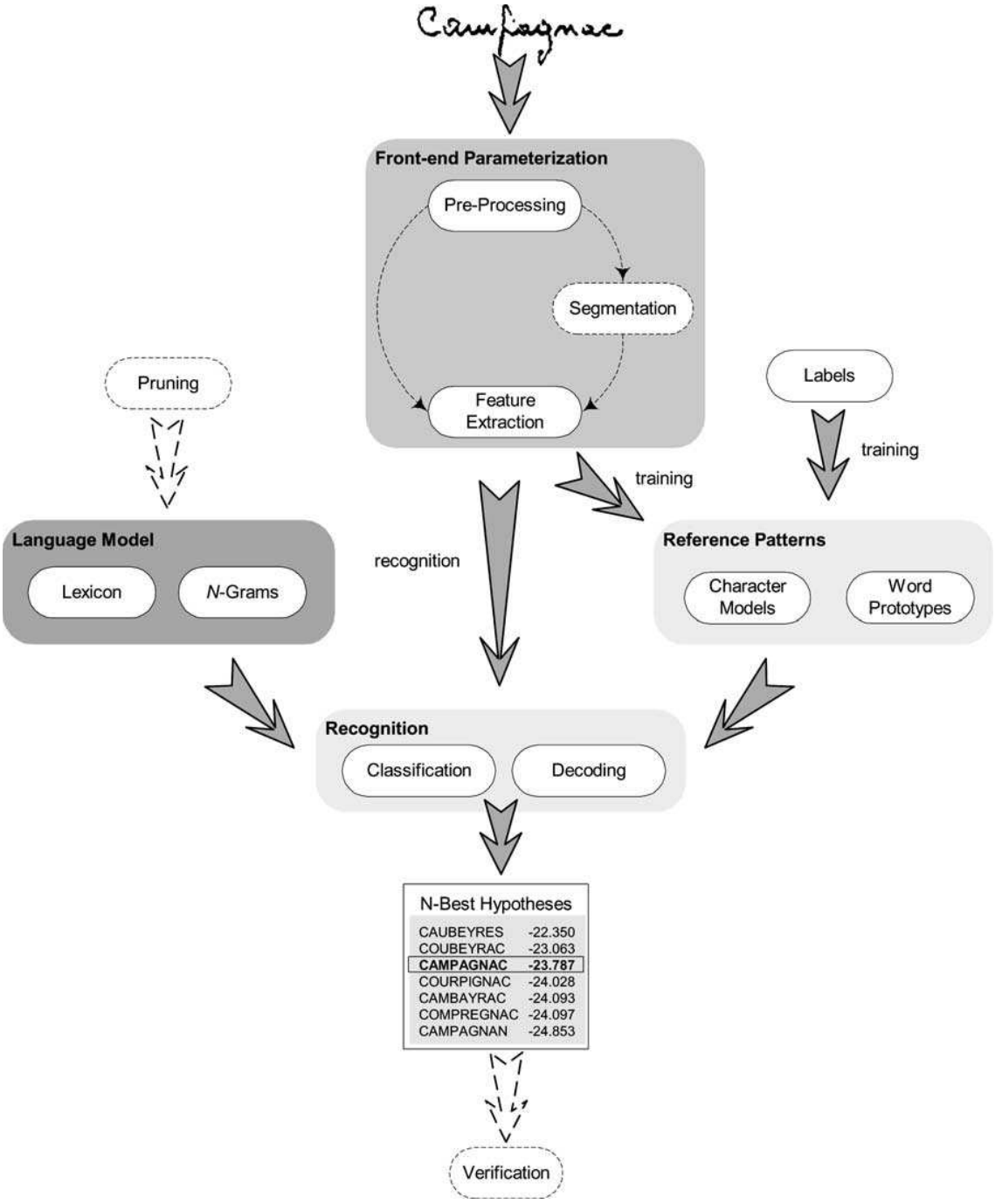

FIGURE 2.13 – An overview of a basic handwriting recognition system as described by Koerich [KSS03]

be derived from a set of averaging technique. Inasmuch as recognizing is a difficult task, mainly sub-word parts such as letters or graphemes are recognized and the using standard concatenation methods the final word model is recognized. The recognition includes a comparison of the unknown pattern with each class reference pattern by measuring a similarity score which can be a distance or either a probability.

For meaningful improvement however is necessary to incorporate different a priori knowledge into the recognition process such as language models. While for limited size vocabularies the different recognition techniques perform satisfactory results, for large size vocabularies the language model contributes to improve the accuracy as well as the computational complexity of the recognizer.

However, the key idea discussed in this survey is how to prune the lexicon which means to reduce the number of words to be compared during the recognition, and how to measure the quality of this reduction.

The notion of *coverage* introduced by the authors can measure the capacity of the reduction mechanism to include the right answer as the procedure may throw away the true word hypothesis. Hence such a measure is necessary. Unfortunately, most of authors does not report the results in term of coverage but just in terms of recognition accuracy.

The different reduction techniques depend upon the application environment. While for banking applications like bank checks reading [GS95, GAA+01] can be used to reduce the possible hypothesis, for postal address systems [Bel96] the lexicon reduction can be based on the ZIP cod recognizers [MSM98, Sin97] which are much more reliable than the word recognizers.

Conventionally, depending on the reliability of the ZIP code digits recognizer, the lexicon containing thousands of entries can be reduced to a few hundred [YGSS99]. When there is no available additional information source other alternatives are necessary like linguistic knowledge which plays also an important role in limiting the lexicon size. As discussed above, the length of the analyzed word shape is an easily detectable feature which can already distinguish between short and long words. Based on the length of the feature vector extracted from the word we can already have a hint of the length of the word.

The shape of the word can be also used for reduction purpose as presented above in case of Guillevic and Zimmermann in [DG00, ZM99, MKG01]. In Tab. 2.1 we can find the effectiveness of some methods reducing the lexicon based on word shape analysis.

Method	**Lexicon size**	**Test set**	**Reduction(%)**	**Coverage**	**Speedup**
[DG00]	3,000	500	3.5	95.0	-
[ZM99]	1,000	811	72.9	98.6	2.2
[MKG01]	23,600	760	95	75.5	-

TABLE 2.1 – Different lexicon reduction strategies using word shape analysis

A more adequate method described in detail in [Koe02] is based on the lexicon representation and the different searching mechanism in such data structures. Considering mainly HMM based systems where the temporal aspect of the signal creation is tracked, the search space reorganization plays an important role as it is possible to exploit the different word similarities at letter level. As the words are letter concatenations we can consider the common prefix parts, suffix parts which occurs in the different lexicon entries. There are two ways to organize a lexicon : *flat* representation and *tree* representation.

The flat lexicon or linear representation denotes the fact that the word is kept strictly separate in the recognition process. The matching between a given sequence of observations of unknown word and each word model is calculated independently. The different word models are constructed a priori by concatenation of graphemes in letters and letters in words and further the unknown word is matched to all the word models belonging to the dictionary. So the complexity increases linearly with the number of words considered in the vocabulary and the average length of the word.

Organizing the lexicon to be searched as a character tree instead of a linear structure of independent words has some advantages. The structure is called *lexical tree*. Taking advantage of the word spelling, two or more words can contain the same initial characters so called prefixes. Hence in a lexical representation they will share the same sequence of characters as presented in Fig. 2.14.

Having such a data representation, the *dynamic programming* (DP) based computation of the a posteriori probability can be factorized. Hence, a speed-up and reduction in storage can be obtained by using a trie. Each node in the the trie corresponds to a letter. As an exhaustive DP calculation is performed for all word entries in the lexicon, the results are more accurate, and thanks to the sharing of intermediate results, the running time is also improved compared to the traditional flat approaches. In handwriting recognition, this kind of approach has been used firstly with success by Man et al. [MFW96] for their $Npen^{++}$ system.

A more complicated organization procedure based on *Directed Acyclic Word Graph* (DAWG) can be designed, which can be considered as an extension of the prefix tree. Here not just the common prefix parts are shared but the other common parts also, such as : terminations or suffixes [GFK02]. Some work has been done in that sense by Lifchitz et al. [LM00] using an ad-hoc method based on a non deterministic state automaton. However, it is not clear how such a method could work in handwriting as there are no available experimental results.

Once the lexicon is organized properly based on common parts like in [KSSEY00] or using some other criteria like self organizing maps [GM97] which maps the words in a two dimensional plane by preserving the neighborhood relation between words, a proper searching mechanism should be applied.

The search problem in handwriting recognition can be formulated as follows : select a word

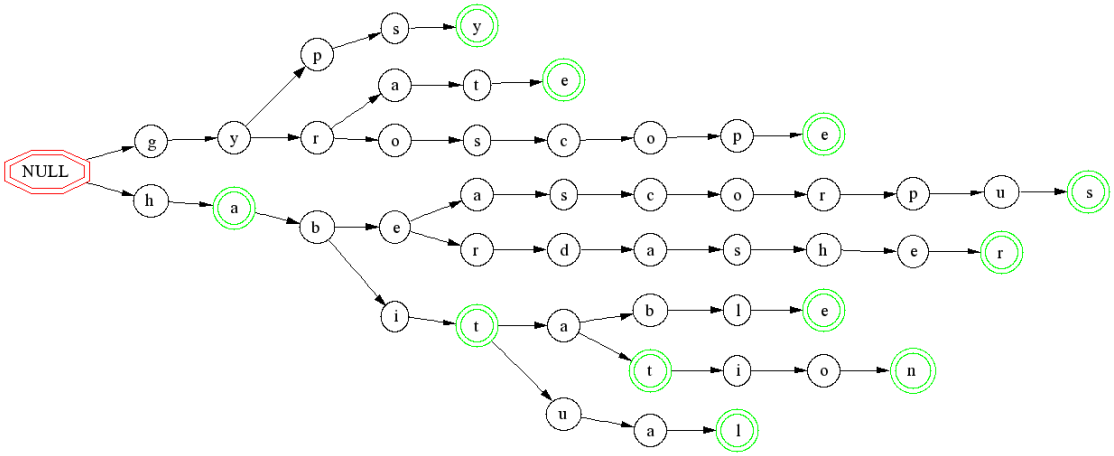

FIGURE 2.14 – Tree representation of English words coming from a dictionary

reference with the highest score, given a test pattern corresponding to an unknown handwritten word represented as a sequence of observations O a set of reference patterns denoted as R_n, $1 \leq n \leq V$ for a V-size word vocabulary in which each pattern is of the form $R_n = (c_1, c_2, \ldots, c_L)$.

Each character is usually modeled by different parameters (features) like : $c_i = (f_1, f_2, \ldots, f_M)$ where M denotes the feature vector size, and f_m represents the mth feature extracted. Unfortunately the most part of the research is carrying out for small size and medium size vocabularies and mainly DP based searching schemes are proposed which become time-consuming once the lexicon size grows as the number of hypothesis blows up. The goal is to align the sequence of observation and compute the likelihoods once for each node by traversing the tree in breadth and retain this value for each frame without knowing a priori to which word such letter node belongs.

To reduce considerably the decoding process, Koerich [KSSEY00] propose instead of using Viterbi search, Beam search, A^* and multi-pass search a *Syntax Directed Level Building Algorithm* (SDLBA) has been considered. The key point here is also the search space representation by using a lexicon tree. In the Tab. 2.2 we can see how such a lexicon reorganization can reduce the complexity of the searching space.

Lexicon size	**No. of char in Flat**	**No of char in Tree**	**Reduction factor**
10	119	113.5	1.05
100	1,198	987	1.21
1,000	11,988	8,361	1.43
10,000	120,035	66,558	1.80
30,000	360,012	173,631	2.07

TABLE 2.2 – Complexity study for different vocabularies represented in flat and tree structure

While the Viterbi algorithm matches a model to a sequence of observations, determining the maximum likelihood state sequence of the model given the sequence of observations, the SDLBA match an observation sequence to a number of models.

The level building algorithm jointly optimize the segmentation of the sequences into sub-sequences produced by different models, and the matching of sub-sequences to particular models. Since the lexical tree guides the recognition, the algorithm needs to incorporate some constraints to handle the language syntax provided by the lexical tree as well as the contextual information related to the class transition probabilities.

The comparison study between the baseline system composed by a Viterbi search combined with a flat representation and the SDLBA has shown the superiority of the lexicon reduction strategy. While for reduced size vocabularies the recognition accuracy, speed is more or less similar, for extended vocabularies the accuracy and the speed gain can be highlighted.

In order to present the different achievements for large vocabulary datasets, we consider being important to review the most important results obtained by Koerich in this domain than other results reported in the literature.
The Table 2.3 gathers the recognition scores obtained by Koerich for different vocabularies.

Lexicon Size	**Recognition rate**	Speedup factor
10	98.50%	7.8
100	94.95%	8.2
1k	88.42%	9.5
10k	77.60%	11.1
30k	71.03%	11.8

TABLE 2.3 – Word recognition scores obtained by Koerich in [Koe02] for the system based on the constrained level building algorithm and a lexical tree representation

A careful analysis of the handwriting recognition field reveals that most of the research is oriented towards simple problems, where just a reduced size vocabulary is considered. Once the size of the vocabulary increases the data size required to develop a good recognition system increases. The Table 2.3 presents some recent results from the literature for the problem of middle size vocabularies and large vocabularies. It should be stressed that these studies have used different datasets and experimental conditions, which make a direct comparison very difficult. However, the results are very helpful to illustrate the current state on this field.

2.2.4 Conclusions

Considering the handwriting recognition by a computer we can declare that it is a hard and challenging issue. Despite the impressive progress achieved during the last few decades and

Ref.	Method	Lexicon size	Accuracy
[FGB98]	HMM	100	98.70%
[CS00]	LVQ	100	93.40%
[KC02]	HMM	100	88.2%
[MG96]	DP	100	89.03%
[WBKR00]	HMM	300	96.51%
[BM99]	HMM	400	89.00%
[GWGH95]	DP	746	80.41%
[BBD+93]	DP/NN	1k	47.00%
[FGB98]	HMM	1k	94.30%
[CS00]	LVQ	1k	83.80%
[AYV02]	HMM	1k	90.80%
[KSS04]	HMM	1k	91.00%
[KPH04]	HMM	1k	67.80%
[MB00]	HMM	7,719	60.05%
[Koe02]	HMM	10k	81.60%
[CK94]	HMM	10k	67.09%
[BRKR00]	HMM	30k	89.20%
[Koe02]	HMM	40k	73.23%
[Koe02]	HMM	80k	68.65%

TABLE 2.4 – Recent results concerning off-line handwriting for middle size and large size vocabularies

the increasing power of computers the performances of the handwriting recognition systems are still far from human performance. Words are fairly complex patterns and owing to the great variability in handwriting style, handwritten word recognition is a difficult one.

Even if different features are extracted from the word patterns at different abstraction levels, guided by the human perception [Whe70, McC76], the features on their own are not sufficiently discriminant for such a recognition task. For that reason it is necessary to combine the different features extracted with different precision in order to interact in the recognition system. This combination should be complete merge as a simple combination seems to be not exploiting the interaction of the different features of the same pattern.

Considering the feature extraction in general, we can also conclude that it is much simpler to extract low-level features instead of high-level ones but while the high-level features describe the word shape the low-level features give just a quantitative measure.

One of the most common constraint of the current recognition systems is that they are only capable to handle words that are present in a restricted vocabulary typically comprised of 10-1,000 words which does not satisfy the requirements of the industrial applications like postal automation, etc., where large vocabularies are considered. To tackle the reduction of the vocabulary some special data representation are necessary and new search techniques should be considered.

2.3 Handwritten digit recognition

2.3.1 Introduction

The digit recognition being a subfield of character recognition is a subject of interest since the first years of research in the field of handwriting recognition. The motivation of such kind of research is multiple. The first is based on the high demand of industrial applications (ZIP code recognition, date recognition, courtesy amount recognition, census form recognition, etc.) which needs to recognize the printed or handwritten digits from the different paper supports. The second consideration is based much more on the theoretical aspect of the problem as such a pattern recognition task is a simple and a robust one, containing just a reduced number of classes to be separated with considering not so complex shapes to be distinguished.

The different subjects addressed in digit recognition domain available in the literature can be separated as follows :

- feature extraction methods
- classification methods
- system architectures

The investigation of the different feature extraction techniques has gained a considerable

attention since discriminative feature set is considered the most important factor in achieving high recognition performances. A comprehensive survey for feature extraction for off-line character recognition is given by Trier et al. in [TJT96]. The authors have been considered many different important aspects in the feature extraction which should be taken into account before selecting a specific feature and classifier. Similarly, in [Lon98], Loncaric presents an interesting review of shape analysis techniques used for the same purpose.

For digit recognition, two main types of feature have been used : *statistical* and *structural features*. The statistical features are derived from the statistical distributions of points, such as zoning, moments, projection histograms or direction histograms. Structural features are based on topological and geometrical properties of the analyzed character, like strokes and their directions, end-points, intersections of segments or loops. As stated also by Bortolozzi et al. in [BdSBJOM05] the researchers have explored the integration of these two kind of features as they are considered as being complementary.

Reducing the *data dimensionality* is also an important factor for the further classifications. That is the reason why such *feature selection* mechanisms have appeared recently in the machine learning community. Reducing the dimensionality means to reduce the input vector which is considered by the recognizer. The objective of variable selection is three-fold : 1) improving the prediction performances of the predictors, 2) providing more effective and faster predictors and 3) providing a better understanding of the the underlaying process that generates the data. Eliminating some input components from the original input stream can be done by Principal Component Analysis (PCA) [RB05] or other methods like : genetic algorithms (GA) [SBM04], neural networks (NN), etc [GE03].

Besides, the investigation of the different feature extraction and feature selection methods, an important role has played by the classification methods and strategies. The task of classification is to partition the future space into regions corresponding to source classes or assign class confidence to each location in the future space. Statistical techniques and neural networks have been widely used due to the efficiency of implementation. Statistical classifiers are divided into parametric and into non-parametric ones. They include the linear discriminant function (LDF), the quadratic discriminant function (QDF), the nearest-neighbor 1-NN and k-NN classifiers, the Parzen window classifier, etc. Neural networks for classification include the multi-layer perceptron (MLP) [Bis95], recurrent neural networks (RNN) [VdC99], self-organizing maps (SOM) [AS00], radial basis functions (RBF) [LBBH01], etc. Recently, a new type of statistical classifier, Support Vector Machine (SVM) [CV95] has been in the attention of the machine learning community. The SVM is based on the statistical learning theory of Vapnik [Vap95] and quadratic programming optimization. Initially designed for two-class problem separation, the SVM became a powerful tool used in text categorization [AM04], digit recognition [SC03], etc.

The main difference between statistical and neural classifiers is that the parameters of the

neural networks are optimized in discriminative supervised learning with aim to separate the patterns of different classes. When the network topology (number of units, number of connections, transfer function, etc.) are well defined and the number of training samples is large, neural networks are able to give high classification accuracy to unseen test data.

Considering the system architectures in case of NNs, two main aspects should be considered. First, the different classifiers have been boosted by using specific initial parameter initialization on the weights, the number of units have been estimated throughout some heuristics, different activation functions have been considered, different topologies have been proposed and finally different training algorithms have been developed to train the different kind of networks. Unfortunately all these attempts are based mainly on heuristics and there is no mathematical rigor behind. The success of these methods is mainly based on the nature of considered database.

The second aspect concerns the classifiers combination which deals with the challenge : *How to exploit the classification capabilities of the different classifiers in the same classification framework ?*

It has been found that multiple expert decision combination strategies can produce more robust, reliable and efficient recognition performances than the application of single expert (classifier) [RF03]. It is also noted that a single classifier with a single feature set [BVM+04] and a single generalization strategy often does not comprehensively capture the large degree of variability and complexity encountered in many practical tasks. Multiple expert decision combination can help to tackle many of these problems by acquiring multiple source information through multiple features extracted from multiple processes. Introducing different classification criteria and a sense of modularity in the system design, it will lead to more flexible recognition systems. While some decision combination strategies are task specific, the most are generic and usually it is possible to apply the same techniques to a variety of tasks.

In order to get a comparative idea about the best achievements on handwritten digit recognition field, a comparative result table can be found in Table 2.5 and Table 2.6 respectively. The reported results here are based on one classifier system.

The MNIST [LBBH01] and the CEDAR datasets are well-known benchmark datasets used by the community to test the different proposed off-line recognition methods [LF05]. The Table 2.5 contains also the best recognition score ever achieved on MNIST authored by Simard et al. [SSP03]. We should notice that this result was obtained by using affine transformations so the training database is not similar with the original MNIST training corpus.

Analyzing the results of the Table 2.5 and Table 2.6 a net superiority of the NN based methods can be highlighted. This can be explained with the fact than the NN based approaches can better adapt to such a pattern recognition tasks where the number of classes to be separated is low (10 class problem) and there is a considerable training dataset available to adjust the model parameters. In case of the HMM there is not necessary to have such a considerable training

Ref.	Method	Rec. rate
[SSP03]	Convolutional NN + Data distorsion	99.60%
[BMP02]	Flexible image matching	99.37%
[DS02]	SVM	99.58%
[LBBH01]	LeNet1	98.30%
[LBBH01]	LeNet4	98.90%
[LBBH01]	LeNet5	99.05%
[BS97]	SVC-poly	98.60%
[BS97]	Virtual SV	99.00%
[TL02]	SVM	99.41%
[ZBS05]	NN	98.55%

TABLE 2.5 – Important contributions in the field of isolated digit recognition for the MNIST benchmark dataset

Ref.	Method	Rec. rate
[CL99]	HMM	96.16%
[DS98]	Morphological and Topological properties	99.37%
[SLS99]	NN	99.77%
[LS93]	NN	98.87%
[FNVZ98]	NN	99.54%
[dSBJSBS04]	HMM	98.00%

TABLE 2.6 – Important contributions in the field of isolated digit recognition for the CEDAR benchmark dataset

corpus but the discriminative power of such a stochastic model is much more reduced than in case of a neural network.

In the next few sections, we will focus on different digit recognition systems based on NNs and HMMs. The aim is to show a few existing systems and to review their properties. The goal is not to give an exhaustive state of the art on handwritten digit recognition but more to give some hints of the domain in order to be able to compare the existing methods with the methods proposed by us.

2.3.2 Neural network based classifiers for handwritten digit recognition

This section is dedicated to the connectionist paradigm oriented toward digit recognition. We will discuss in detail the Multi-layer Perceptron and its extension : the convolutional neural network proposed by LeCun in [LBBH01]. Our choice is oriented toward such type of networks as they are the most efficient ones in this field. Throughout these networks we will present different issues raised in the connectionism, like :

- network architecture
- feature selection
- pattern selection
- training algorithm

The feedforward neural network

The multi-layer perceptron (MLP) is a layered feedforward network, that can be represented by a directed acyclic graph (see Fig. 2.15). Each node in the graph stands for an artificial neuron of the MLP, and the labels in each directed arc denote the strength of synaptic connection between two neurons and the direction of the signal flow in the MLP. For the different pattern classification task it is the most used network type.

For pattern classification the number of neurons in the input layer of an MLP is determined by the number of features selected for representing the relevant patterns in the feature space. For the output layer this is set up by the number of classes to which the input data belongs. The neurons in the hidden and output layers compute in general the sigmoid function (see Fig. 2.16) on the sum of the products of input values and weight values of the corresponding connections to each neuron in order to break the linear aspect of the activation performed by the MLP. The activation function is :

$$y = f(\sigma) = \frac{1}{1 + e^{\sigma}} \tag{2.4}$$

The forward step in the MLP paradigm means to compute the activation of a given pattern through the different units of the different layers. In mathematical terms speaking it is a cu-

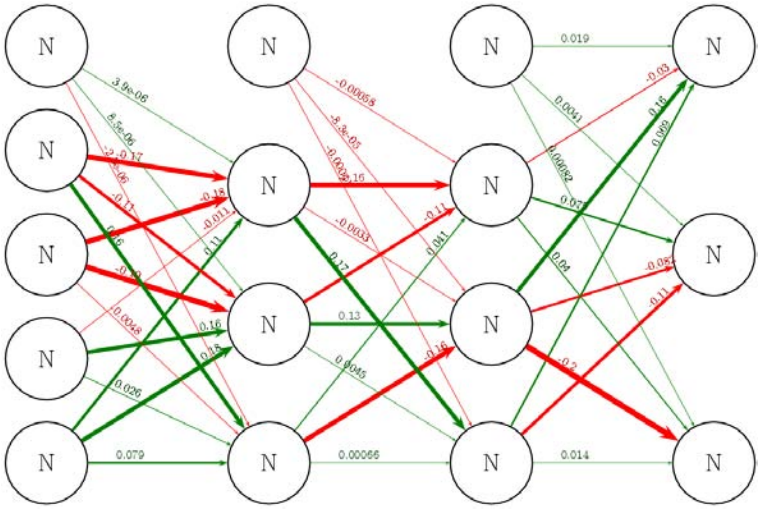

FIGURE 2.15 – A multi-layer perceptron scheme with the corresponding weights

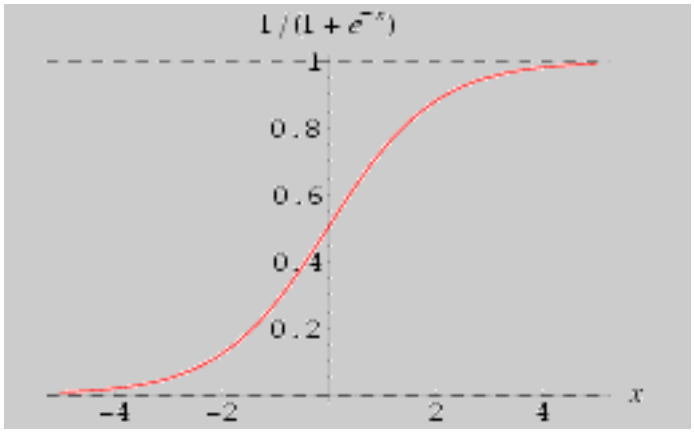

FIGURE 2.16 – The sigmoid function

mulated sum of the products between the inputs of the layers and the corresponding synaptic weights. This information is propagated from the input layer toward the output layer.

In order to spread out the information into the network, a propagation rule has been defined. The activation of a given neuron n_j^l is given by :

$$\sigma_j^l = \sum_{i \in Input(j)} w_{ij}^l \times y_i^{l-1} \tag{2.5}$$

where $Input(j)$ is the set of input neurons for the neuron j. In a fully connected multi-layer perceptron scheme this notion corresponds to all the units in the previous layer. The σ_j^l is calculated as a weighted sum. In order to bias this sum a special value w_0^l is added for each layer. This can be considered in practice as a special neuron which has as input the value 1. Considering this bias, the previous formula is :

$$\sigma_j^l = w_0^l + \sum_{i \in Input(j)} w_{ij}^l \times y_i^{l-1} \tag{2.6}$$

The output value of a neuron n_j^l is given by :

$$y_j^l = f\left(w_{0j}^l + \sum_{i=1}^{L_t} w_{ij}^l \times x_i\right) \tag{2.7}$$

where l is the number of layer, while j corresponds to the number of the neuron in the given layer. The y_j^l value is calculated as a weighted sum of the inputs where an activation function f is applied.

The training process of a MLP involves tuning the strengths of its synaptic connections so that it can respond appropriately to every input taken from the training set. The number of hidden layers and the number of neurons in a hidden layer required to design an MLP are also determined during the training phase. The training process incorporates learning ability in an MLP. The generalization ability of a NN is tested by checking its responses to input patterns which does not belong to the training set.

The *back propagation algorithm*, which uses patterns of known classes to constitute the training set represents a *supervised learning* method. After supplying each training pattern to the MLP, it computes the sum of the squared errors at the output layer and adjusts the weight values of the synaptic connections to minimize the error sum. Weight values are adjusted by propagating the error sum from the output layer to the input layer through the intermediate layers. Such a back propagation allows to estimate the importance of each neural unit, hence the error of each unit is estimated.

A formal description of the training procedure is described hereinafter. The training of the network is based on gradient descent concept. The forward propagation of the error in the

network is based on the error calculus on the output layer between the expected value and the calculated value and the propagation of this information between the different layers. Considering as activation function the sigmoid, the error is :

1. For the output layer considering C class and u the expected output, $\forall j \in \{1, \cdots C\}$

$$\delta_j^l = (u_j - f(\sigma_j^l)) \times (f(\sigma_j^l) \times (1 - f(\sigma_j^l))) \tag{2.8}$$

2. The error for the hidden layers can be written as follow :

$$\delta_j^l = (f(\sigma_j^l) \times (1 - f(\sigma_j^l))) \times \sum_{k \in Output(j)} \delta_k^{l+1} \times w_{jk}^{l+1} \tag{2.9}$$

where $Output(j)$ denotes all the neurons connected to neuron j in the previous layer.

Once the error is calculated trough the formula given above, we can adjust the weight by updating them. The generic weight updating can be described as follow :

$$w_{ij}^l = w_{ij}^l + \Delta w_{ij}^l \tag{2.10}$$

where Δw_{ij}^l is :

$$\Delta w_{ij}^l = \epsilon \times \delta_j^l \times y_i^{l-1} \tag{2.11}$$

where ϵ stands for the learning rate, controlling the speed of the weights' change.

The choice of the activation function as the sigmoid function is motivated by the fact that it is not linear and in the same time $f'(x) = f(x) \times (1 - f(x))$ where $f(x) = \frac{1}{1+e^{-x}}$. However, there are many other activation functions used to normalize the output of the network units. The $\tanh(x) = \frac{e^{-x}-e^{-x}}{e^{-x}+e^{-x}}$ is one of the commonly used activation function besides the sigmoid.

To train the network an iterative training process is considered presenting each training pattern sample to the network several times. Based on the error propagation method we can distinguish : *batch learning* (all the pattern are processed and at the end the update of the weights is performed), *incremental learning* (once a training pattern is passed in the network, the error correction is right away performed) and there is a trade-off so called *mini-batch learning* (after some samples the correction is performed).

The inconvenience of such a network is the huge number of free parameters. To train such a network many training samples are necessary and several epochs should be invoked to realize the convergence of the training procedure.

The LeNetX digit recognition system

To solve the dilemma between small networks that cannot learn the training set and large networks which seems to be over-parametrized we can design specialized network architectures

which are specifically designed to recognize two-dimensional shapes such as digits, while eliminating irrelevant distortions and variability. These considerations lead the authors in [LBBH01] to use *convolutional neural networks*. In a convolutional net each unit takes input from a local receptive field on the layer below, forcing it to extract a local feature. Furthermore, units located in different places of the image are grouped in planes, called *feature maps*, within units are constrained to share a single set of weights. This makes the operation performed by a feature map shift invariant and equivalent to a convolution followed by a squashing functions. This weight sharing technique considerably reduces the number of free parameters. A single layer is formed by multiple feature maps extracting different feature types.

Complete networks are built of multiple convolutional layers, extracting features of increasing complexity and abstraction. Sensitivity to shift and distorsions can be reduced by using lower-resolution feature maps in the higher layers. This is achieved by inserting *subsampling layers* between the convolutional layers. It is important to stress that all the weights in such a network are trained by gradient descent. Computing the gradient can be done with a slightly modified version of the classical backpropagation procedure. The first application of such a network is the LeNet1, shown in Fig. 2.17.

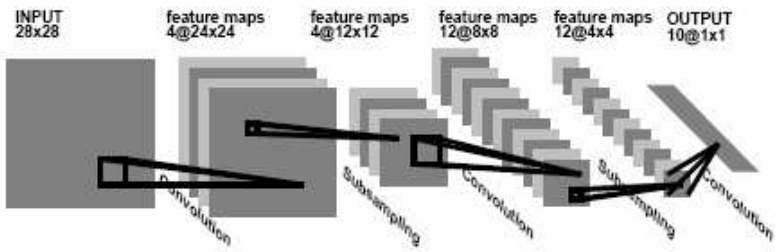

FIGURE 2.17 – Architecture of LeNet1. Each plane represents a feature map i.e. a set of units whose weights are constrained to be identical. Input images are sized to fit in a $16x16$ pixel field, but enough blank pixels are added around the border to this field to avoid edge effects in the convolution calculations

Because of LeNet1 has a small input field, the images were down-sampled to 16×16 pixels are centered in the 28×28 input layer. Although about $100,000$ multiply/add operations are required to evaluate the LeNet1. Its convolutional nature keeps the number of free parameters to about $3,000$. The LeNet1 architecture was developed using a special USPS[3] dataset [LBBH01].

The different experiments with LeNet1 made it clear that a larger convolutional network was needed to make optimal use of the larger size of the training set. The LeNet4 and after that the LeNet5 was designed to address this problem. The LeNet4 is an extended version of the original

3. United States Postal Service

LeNet1, where more feature maps have been considered. An extra layer of hidden units is added which is a fully connected to both the last layer of feature maps and to the output units. It contains about $260,000$ connections and about $17,000$ free parameters.

The LeNet5 has a similar architecture to LeNet4, but has more maps, a larger fully connected layer, and it uses a distributed representation to encode the categories at the output layer, rather than the more traditional "1 to N" code. LeNet5 has a total of about $340,000$ connections and $60,000$ free parameters. Most of them lays in the last two layers. Once again the non-deslanted 20×20 images centered by center of mass were used, but the training procedure includes a module that distorts the input images during the training using small randomly picked affine transformations (shift, scaling, rotation and skewing).

Generally to train a neural system a huge data amount is needed to cover the different intra-class and inter-class variations. Hence in the neural network scheme based approaches we can find equally advantages and disadvantages. Among the advantages we can enumerate : good generalization property based on a solid mathematical background, good convergence rate, fast recognition process, etc. [Bis95]. Among the disadvantages can be enumerated some restrictions like : the size of the input should be fixed, the nature of the input is not obvious for the different pattern recognition tasks, it is hard to implant in the network topology some a priori knowledge based on the data corpus. In the same time the system convergence speed can be long [LBBH01], as the adjustment of the decision surface (hyperplane) in the function of the network's free parameters is a time costly process. The excessive data amount, the network architecture and the course of dimensionality are always an endless trade-off in the neural networks theory [Bis95, RHW86].

Different optimization attempts in the connectionist framework

Nowadays, more or less each baseline neural recognition system has reached its limits as there is no existing system allowing to realistically model the human vision. Hence, the research was oriented toward different improvements of the existing systems by refining the mathematical formalism or by implanting in the systems some empirical knowledge.

In order to tackle these problems, different techniques have been proposed in the literature.Some of them use some a priori knowledge derived from the dataset, some other methods modify the network topology in order to reduce the number of free-parameters, some other techniques try to reduce the dimension of the input using feature selection techniques and some others try to develop the so called active learning techniques whose goal is to use an optimal training dataset through selection of most informative patterns from the original dataset. Considering the nature of the improvements, different research axis can be distinguished.

In order to select the most descriptive features, in the last few years many feature extraction

algorithms were proposed. In character recognition these features can be grouped as : *statistical features*, *geometrical features* [LBBH01, Guy91, HF98b, RBTT95], *size and rotation invariant features* [AOC+99, Gos84, LPT91, SR92], etc.

For the combination of these features many feature subset selection mechanisms were designed. The feature subset selection is based mainly on neural networks (NN) and genetic algorithms (GA) [YH98] operating with randomized heuristic search techniques [CLR95].

Concerning the network topology there is a consensus in the NN community. One or two hidden layer is sufficient for the different pattern classification problems [BW00]. Even so, LeCun and his colleagues proved that it is possible to use multiple hidden layers based on multiple maps using convolution, sub-sampling and weight sharing techniques in order to achieve excellent results on separated digit recognition [LBBH01, SSP03]. To be noticed, this is the only type of network using several layers to our best knowledge which was used with success for handwritten digit recognition purpose.

Spirkovska and Reid using a higher order neural network introduced inside the topology some position, size and rotation invariant a priori information [SR92].

Another solution is the *optimal brain damage* (OBD), proposed by LeCun in [CDS90] which removes the unimportant weights in order to achieve a better generalization and a speed-up of the training/testing procedure. The *optimal cell damage* (OCD) and its derivatives are also based on the idea to prune the network structure. All approaches solve more or less the encountered problems but each of them has a considerable time complexity.

The best approach seems to be the so called *active learning* and its different derivatives. In such an approach the learner, the classifier is guided during the training process. Some information control mechanism is implanted in the system. Rather than passively accepting all the available training samples, the classifier on his own guides its learning process by finding the most informative patterns. With such a guided training, where the training patterns are selected dynamically, we can reduce considerably the training time duration and a better generalization can be obtained. All non interesting data can be discarded. As stated in [SOS92] the generalization error decrease more rapidly for active learning than for passive learning.
Engelbrecht in [Eng01] is grouping these techniques in two classes in function of their action mechanism :

1. *Selective learning*, where the classifier selects at each selection interval a new training subset from the original candidate set.

2. *Incremental learning*, where the classifier starts with an initial subset selected somehow from the candidate set. During the learning process, at specified selection intervals some new samples are selected from the candidate set and these patterns are added to the training set.

While in [Eng01, RPH01, SOS92] the authors have been developed *active learning techniques* for feedforward neural networks, for SVM approaches similar systems have been proposed in [KH04, SC03, WNC05]. In this second case all the pattern selection algorithms are based on the idea than the hyperplane constructed by the SVM is depending only on a reduced subset of the training patterns called also support vectors that lies close to the decision surface. Mostly the selection methods in that case are based on kNN, clustering, confidence measure, Hausdorf distance, etc. The drawback of such systems is the difficulty to fix the different parameters of the systems as stated by Shin and Cho [SC03]. Another limitation of the approach is a second training procedure which is necessary while for NN the training process is applied just once. This drawback can also be found in case of the different network pruning strategies proposed by LeCun.

Conclusions concerning the neural networks in digit recognition

Considering the comparison between the fully-connected neural network (MLP) and convolutional network we can conclude different things. Instead of using features extracted from the analyzed shape is better to use the raw information considering the $2D$ aspect of the shape. While the MLP is not so complex, the program implementation is easy to do, the LeNetX based systems have much more complex structures extracting local information using the maps built in the different layers. The number of parameters in case of the convolutional network explode while the number of parameters in an MLP is not numerous. To reduce the number of free parameter in a convolution network weight sharing technique is used. That means using different windows we can extract the same features using the same weights. Such an approach allows a local analysis of the image. The results achieved by the two methods are similar, with a slight superiority of the convolutional network due to the local analysis of the shape.

The backpropagation algorithm applied to train the different networks is the most appropriate but has a convenience. It is time consuming. To train, to adjust the free parameters of the networks the number of epochs and the number of the training samples should be considerable. This aspect is a very important one being based on the initial weight values, the momentum and the learning rate. The tuning of these parameters could take several days. To do a quick test process (a few hours), a more efficient and fast training could be helpful. Such a fast algorithm can help to test the different parameter settings of a network without spending a tremendous time to find the optimal topology and parameter setting.

Taking into account the different advantages/disadvantages, our neural network architecture proposition is based on the idea to preserve as much as possible the different raw information coming from the input image without presuming any dependencies between the different input components.

The fast learning technique developed by us in the framework of a separated digit recognition task is based on active learning technique more exactly belonging to the branch of incremental learning where the dataset is constructed dynamically during the training. Using the *FDDLCB* algorithm (Fast Data Driven Learning Corpus Building Algorithm) based mainly on least mean square (LMS) error minimization, we have reduced considerably the training time factor without any loss of accuracy.

2.3.3 Stochastic approaches for separated handwritten digit recognition

In this section we review some works concerning the usage of the HMM based models for separated digit recognition. The aim is not to give an exhaustive review in the field but to outline the different architecture used recently to perform such type of recognition. Instead of using high accuracy neural approaches the goal here is to exploit the $2D$ nature of the analyzed shape as well the elasticity given by the stochastic process driven by the temporal quality of the models.

Recent stochastic models in handwritten digit recognition

The drawback of probabilistic methods such as HMMs, it is their reduced capacity to recognize 2D signals as handwriting. The model developed originally for speech recognition [Rab89] has interesting capacities as they can adapt (model) to the different distortions of the input observations, especially to elastic distortions [Cho02].

While there are some works, where the handwriting is considered as one dimensional signal and the data sampling is based on this fact [PL96, FGB98], our interest is oriented towards the 2D models. Here a real two-dimensional signal is considered as input for the classifier.

One recent extension to HMMs are Bayesian networks. Bayesian networks have incredible power to offer assistance in a wide range of endeavors. They support the use of probabilistic inference to update and revise belief values. Bayesian networks readily permit qualitative inferences without the computational inefficiencies of traditional joint probability determinations. In doing so, they support complex inference modeling including rational decision making systems, value of information and sensitivity analysis. As such, they are useful for causality analysis and through statistical induction they support a form of automated learning. This learning can involve parametric discovery, network discovery, and causal relationship discovery.

In [CK03] the authors have considered a Bayesian network to model on-line Hangul characters. A Bayesian network is a graph with probabilities for representing random variables and their dependencies. Such a model efficiently encodes the joint probability distribution of a large set of variables. Its nodes represent random variables and its arcs represent dependencies between random variables with conditional probabilities at nodes. It is a Directed Acyclic Graph (DAG) so all the edges are directed and there is no cycle when edge directions are followed. They

have proposed to model the different graphemes, strokes, etc. and their relation into a single Bayesian network. The Hangul digit model is composed of stroke models and point models. Their relationships are modeled as conditional Gaussian distributions. The system outperformed an HMM system with chain code features and a neural network based system in terms of recognition rate. We should note that is the first model where a large dataset is considered (2,350 digit classes) and it is also scaling and rotation invariant which has a great impact on on-line characters where such deformations are often encountered.

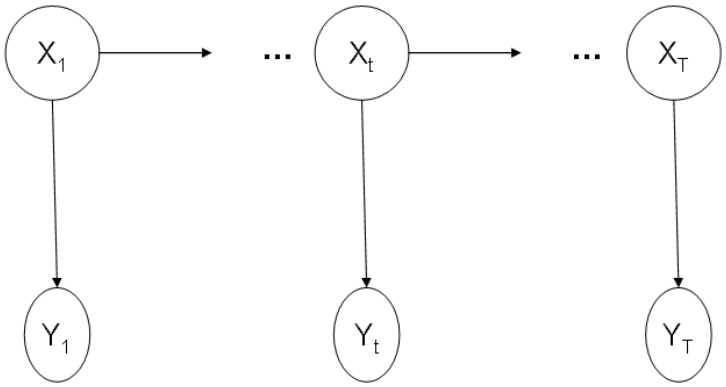

FIGURE 2.18 – An HMM modeled by a Dynamic Bayesian network, where $(X_t)_{1\leq t\leq T}$ are the hidden states and $(Y_t)_{1\leq t\leq T}$ are the observations.

A similar system has been proposed by Hallouli et al in [HLS04], where off-line digit recognition has been considered. Here the Bayesian network models the dependencies between two HMM based models. They model the dependencies between the nodes (state and/or observation variables) creating interactions between the horizontal HMM and the vertical-HMM. The horizontal-HMM (respectively the vertical-HMM) considers as input pixel lines (respectively pixel columns) of the analyzed shape. The 2D modeling resides in coupling state variables or observation variables of these two HMM models. The model can be considered as a two dimensional model as it considers the horizontal and the vertical HMM model simultaneously. While the original HMMs results are not satisfactory, the coupled model shows a 2% net amelioration but even these result is far from the top results achieved by the neural network based models or the support vector machines.

Another kind of approach also based on Hidden Markov Models is proposed by Chevalier [Che04, CGPL05]. The method is totally $2D$ as it is using different regions where Gibbs distributions are considered for a nxn size image with a set of labels having the cardinality L :

$$P(\omega) = \frac{1}{Z} exp(-\sum_{c\in C} V_c(\omega)) \tag{2.12}$$

where $\omega = \{\omega_{(i,j)}, 1 \leq i,j \leq n\}$ and $\omega_{(i,j)} \in \{1,...,L\}$is the label of the site (i,j), C is the set of

cliques, V_c is a potential function associated to cliques c and Z is a normalization constant so that $\sum_\omega P(\omega) = 1$, which is equivalent to Markov Random Field described in detail in Chapter 3.

For the different regions some local features are extracted based on a window analysis. For that reason $2D$ Fourier transform is computed. The low frequency coefficients keep information on strokes and directions. The states are associated to homogeneous portions of strokes in the image. In this model 5×7 state models have been considered. This parameter was tunned based on trial runs. The training mechanism has two phases : in a first step a regular segmentation of the digit in 35 states is performed. This first segmentation allows the computation of the initial model (observations and transition probabilities). In the second step this model is then used to process $2DDP$ decoding and getting new segmentations which give the new parameters of the model. This process is iterated till convergence. More details can be found in [Che04].

The method has been used for separated digit recognition and some attempts have been made to use it for handwritten word recognition but not with so much success. The results obtained for the MNIST data are not satisfying and based on the different regions assigned to the states, the processing of a digit samples is tremendous. The originality of the method is the usage of regions and the extraction of strokes in these regions. The $2D$ dynamic programming is similar with our case, but based on some assumptions, some regions can be merged in order to reduce the complexity of the model. The drawback of this process : there is no well defined merge strategy. Just some basic heuristic based on the measure of uncertainty is considered in the merging policy.

Conclusions concerning the stochastic models in digit recognition

The general conclusion for the stochastic models is that HMM based approaches are more time consuming and the accuracy is lower then in case of the neural approaches. The advantage of these models is their capacity to model the $2D$ signal. The constraint raised by the bi-dimensional nature of the problem for the dynamic programming raises a real time problem for such models. While for the neural approaches we can easily recognize 100-200 digits/sec, considering the same test condition for HMM based approaches it is impossible to achieve better performances than 2-25 digits/sec.

2.3.4 Conclusions

In this section we have addressed the issue of digit recognition considering two main strategies : the first one represented by the ANNs and the second one by HMMs. We have selected this kind of reflection about the digit recognition in general as we consider as being those major attempts which has been done in this field. We have listed all the details of these two strategies highlighting their strengths and weaknesses. Based on the results achieved and our analysis, we can conclude that it is more indicated to perform digit recognition by ANNS but some compen-

sation can be brought also by the HMMS, where the temporal aspect is also considered. While in case of ANNs a preliminary normalization process is a primary condition, the HMMs do not require such a process as the size of the input is not limited by the structure of the recognizer.

Another important aspect is the usage of such system in a real-time application as postal address recognition, bank check reading, tax form reading, etc. Even with the growing power of the modern computers the speed performed by the HMMs can not compete with the recognition and speed performances produced by the ANNs. In our research we also concentrated mainly on the ANNs solutions to meet the requirements imposed by the postal automation system for Indian documents.

2.4 Conclusion

Considering the handwriting recognition and in special the handwritten postal document recognition we can state this is still a hard and challenging issue. Despite the impressive computer power deployed in the last few years and the evolution of vision strategies designed to model and recognize handwriting, the current vision systems are far away from replacing the human vision. They are so many partial problems related to this issue. Each of them requires high precision, machine calculus and last but not least more complete vision models to succeed. However in some specific cases, using some restrictions and a priori knowledge can help to build-up good solutions which can be used in real life applications.

One of the most challenging task in the postal document automation is the recognition with high precision of the address part with the corresponding information like city name, street name and the ZIP code. While the recognition of the digits composing the ZIP code nowadays became approved,the word recognition is much more challenging. Words are fairly complex patterns owing to the great variability in handwriting style, handwritten word recognition is a difficult one. This challenge is extended by the fact that in other script environments than Latin -for example Bengali- the complexity of the recognition is higher because of the huge number of characters and shape modifiers.

Even if the different features are extracted from the word patterns at different abstraction levels, guided by the human perception [Whe70, McC76], the features are not sufficient for such a task. For that reason it is necessary to combine the different features extracted with different precision in order to interact in the recognition system. This combination should be complete merge as a simple combination seems to be not exploiting the interaction of the different features of the same pattern.

One of the most common constraint of the current word recognition systems is that they are only capable to handle words that are present in a restricted vocabulary typically including of 10-1,000 words which does not satisfy the requirements of the industrial applications like

postal automation, etc., where large vocabularies are considered. To tackle the reduction of the vocabulary some special data representation are necessary and new search techniques should be considered.

Considering the comparison done for the different digit recognition attempts we can conclude that the solutions proposed by the neural network strategies are much more faster and precise than the solutions in the time space domain represented by the HMMs. However, it is quite interesting that in these particular cases feature extraction methods are not so efficient as in case of word recognition. This paradox can be explained by the fact that for digit recognition we need as much information as we can gather, while for word recognition this is not possible due to the complexity.

Similarly, the selection of the models is also driven by that fact. For word recognition, a word being a large signal, we should consider the time domain aspect so the most successful solutions are given by the HMMs, while for digit recognition the time domain aspect can be neglected because the space domain contains enough information for the neural networks to be successful.

3

Limits of the baseline NSHP-HMM handwriting recognition model

As our work is oriented towards an extension of the NSHP-HMM model, in this chapter we remind its functioning with the corresponding theoretical and practical aspects. First, a formal description will be detailed, concerning the initial stochastic model proposed by Saon [Sao97], while the second section presents the improvement proposed by Choisy. This transforms the holistic system to an analytical one [Cho02] considering an implicit segmentation strategy. The approach is based on letter NSHP-HMMs and word meta-models for the intra-letter transitions to model reduced size unconstrained handwritten vocabularies. Finally, a conclusion is devoted to highlight the advantages and drawbacks of the model.

3.1 The NSHP-HMM on digit and word recognition

3.1.1 General framework

Convinced by the limitation of $1D$ models for handwriting recognition, Saon [Sao97] proposed to overcome this limitation by a two-dimensional model for modeling and recognition. For this reason, the applicability of Markov Random Fields (MRF) has been studied. Unlike PHMMs (Planar Hidden Markov Model), these fields possess a real $2D$ structure as long as the probability of a random variable of the field is conditioned by neighboring ones. The MRFs perform essentially low-level tasks in image processing or artificial vision [HB93]. So the authors concluded in [SB97] that this is the reason why such modeling was not attempted to model handwriting, which can be considered a complex pattern recognition issue. In their opinion this is due to the fact that MRFs, as initially conceived, are only able to detect simple features in images like lines, edges of given orientations. This turns out to be insufficient for higher level recognition purposes. The idea of Saon is to provide MRFs with a switching mechanism between conditional probability

distribution in order to improve the capacity of the model to detect new features within the image (strokes of different orientations inherent to handwriting).

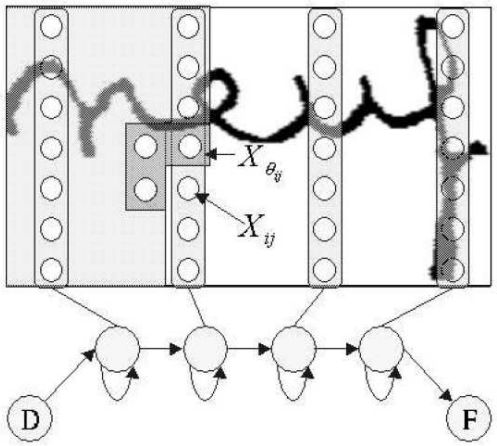

FIGURE 3.1 – The column probabilities observed by the different HMM states in the system of Saon

This is done by tying column probability distribution to the states of a classical HMM (see Fig. 3.1). A transition to another state of the HMM is implying an optimal change of these distributions in order to maximize word-image likelihood. The study has been restricted to causal MRFs for two major reasons. First, as explained by Saon, one cannot specify arbitrary conditioning neighborhoods for consistency reasons (existence of the joint field probability), whereas there are several theoretical achievements on causal MRFs. On the other hand, recursive training and recognition procedure are more easily applicable on causal fields allowing a natural progression of the joint field mass probability calculus. The concept of causality may have different interpretations since the plane is not provided with a natural order.

Two types of causal MRFs are widely used in image processing : the Markov Random Mesh (MRM) and the unilateral Markov Random Field also called non-symmetric half plane (NSHP). As noted by Jeng in [JW87], the NSHPs are more appropriate than MRMs when an accurate model for representing two dimensional data is required (MRMs are conditionally independent on 45° diagonals which diminish their capability to detect strokes having these orientations [Sao97]).

3.1.2 Non-symmetric Half-plane Random Fields

The discussion of the model is restricted to random fields over an $m \times n$ integer lattice L. In that sense, for handwriting the m and n parameters can be considered as the width and height of word image bounding box, respectively. Each site $(ij) \in L$ corresponds to a pixel in the image. Let $X = \{X_{ij}\}_(ij) \in L$ be a random field defined over the lattice L. X^j stands for the column j of the random field X. Moreover, $P(X_{ij} \mid X_{k,l})$ denotes the conditional probability

of the realization $x_{i,j}$ of $X_{i,j}$ knowing realizations x_{kl}, that is $P(X_{ij} = x_{ij} \mid X_{kl} = x_{kl})$. Finally, the notation $P(X_{ij} \mid X_A)$, where $A \subset L$, stands for $P(X_{ij} \mid X_{kl})$ where $(k,l) \in A$. Since the analyzed image is a binary one, the considered random field is also binary, meaning that random variables take values of $\{0,1\}$ where 0 is considered as a white-pixel while 1 is considered as a black one. According to the previous assumptions, a word image can be considered as one possible realization of the random field.

The NSHP Markov chain can be defined as follows. Considering the following sets :

$$\Sigma_{ij} = \{(k,l) \in L \mid l < j \text{ or } (l = j, k < i)\} \tag{3.1}$$

and $\Theta_{ij} \subset \Sigma_{ij}$ where Σ_{ij} is called non-symmetric half plane where Θ_{ij} is considered as being the support of the pixel $(i,j) \in L$.

Definition 1 : X is considered as a non-symmetric half-plane Markov chain if and only if :

$$P(X_{ij} \mid X_{\Sigma_{ij}}) = P(X_{ij} \mid X_{\Theta_{ij}}) \; \forall (i,j) \in L \tag{3.2}$$

The joint field mass probability $P(X)$ can be computed following the chain decomposition rule of conditional probabilities :

$$P(X) = \prod_{j=1}^{n} P(X^j \mid X^{j-1} \dots X^1) = \prod_{j=1}^{n} \prod_{i=1}^{m} P(X_{ij} \mid X_{\Sigma_{ij}}) = \prod_{j=1}^{n} \prod_{i=1}^{m} P(X_{ij} \mid X_{\Theta_{ij}}) \tag{3.3}$$

, where X^j is the conditional probability of the jth column.

Commonly, authors using NSHP Markov chains, choose for all Θ_{ij} the same form, that is $\Theta = \{\Theta_{ij}\}_{1 \le i \le m, 1 \le j \le n}$ (see Fig. 3.2) $\Theta_{ij} = \{(i - i_k, j - j_k) \mid 1 \le k \le P, j_k > 0 \text{ or } (j_k = 0, i_k > 0)\} \cap L \neq 0$ where P represents the number of neighboring pixels. Note that definition of Σ_{ij} satisfies equation 3.1. (see Figure 3.2)

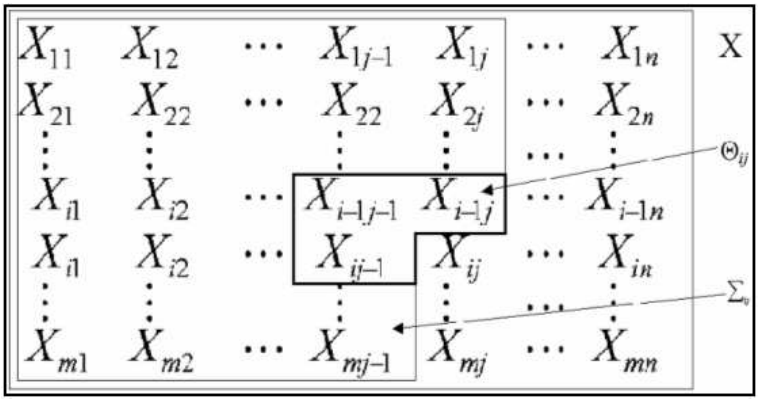

FIGURE 3.2 – Sets of pixels $\Theta_{(i,j)}, \Sigma_{ij}$ related to site (i,j)

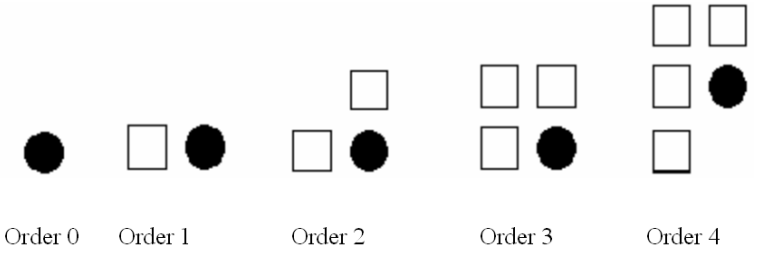

FIGURE 3.3 – The neighborhood orders which can be used

In [Sao97, SB97] the authors are giving an exhaustive study concerning the influence of the different Θ_{ij} collection (see Fig. 3.3) based on the average recognition scores.

The order of a given Θ_{ij} neighborhood is considered by the number of neighbors counting to re-estimate the $P(X_{ij})$.

3.1.3 Formal definition of the NSHP-HMM

To exploit the realization of the NSHP on an image, Saon [Sao97] associates an HMM to observe the different columns. In a given state of the HMM, the observation probability is given by the column product of conditional pixel probabilities. A transition from a state to another results in changing the log probability distributions. After the training phase, the model associates states to particular features (some kind of pixel distributions characterizing different writing strokes) within the word image area. The Fig. 3.4 illustrates the implementation of the NSHP by Saon.

Let λ be the HMM. The equation 3.3 can be rewritten in term of pattern likelihood considering λ :

$$P(X \mid \lambda) = \prod_{j=1}^{n} P(X^j \mid X^{j-1} \dots X^1, \lambda) = \prod_{j=1}^{n} \prod_{i=1}^{m} P(X_{ij} \mid X_{\Sigma_{ij}}, \lambda) \tag{3.4}$$

Supposing that a stochastic process is associated to column X^j. We denote this process by $Q = q_1 \dots q_n$, where the random variable q_j can take values in a finite set of states $S = \{s_1, \dots, s_N\}$

The equation 3.4 can be transformed as follows :

$$\begin{aligned} P(X \mid \lambda) &= \sum_{Q} P(X, Q \mid \lambda) = \sum_{Q} P(X \mid Q, \lambda) P(Q \mid \lambda) \\ &= \sum_{Q} \prod_{j=1}^{n} P(q_j \mid q_{j-1}) P(X^j \mid X^{j-1} \dots X^1, q_j, \lambda) \\ &= \sum_{Q} \prod_{j=1}^{n} P(q_j \mid q_{j-1}) \prod_{i=1}^{m} P(X_{ij} \mid X_{\Sigma_{ij}}, q_j, \lambda) \end{aligned} \tag{3.5}$$

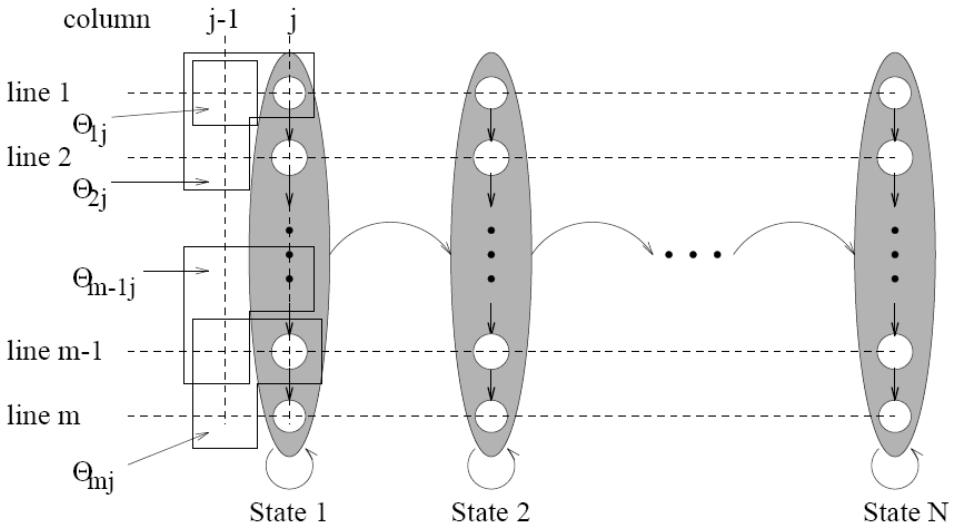

FIGURE 3.4 – The NSHP-HMM scheme by Saon [Sao97] where the states represent the states of the HMM mapping the different image columns

under assumption that Q is a first order Markov process and that pixel distribution for the column j depends only on the state q_j.

Considering the different assumptions and conditions, the NSHP-HMM can be defined as follows.

Definition 2 : A non-symmetric half-plane Hidden Markov Model of order P is defined as $\lambda = (\Theta, A, B, \pi)$ with the following details :

$\Theta = \{\Theta_{ij}\}_{1\leq i\leq m, 1\leq j\leq n}, \Theta_{ij} = \{(i-i_k, j-j_k) \mid 1 \leq k \leq P, j_k \geq 0 \; or \; (j_k = 0, i_k > 0)\} \cap L$ where P represents the number of neighboring pixels per site. Θ is called the NSHP support set collection (or just simply neighborhood set).

$V = \{0, 1\}$, the vocabulary considered for example are pixel colors. The pixel realization of X_{ij} is denoted by $x_{ij} \in V$.

$S = \{s_1, \ldots, s_N\}$ the set of the N possible states of the model. $q_j \in S$ denotes the state associated to the column X^j.

$A = \{a_{kl}\}_{1\leq k,l\leq N}$ $a_{kl} = P(q_{j+1} = s_l \mid q_j = s_k)$, the state transition probability matrix.

$B = \{b_{il}(x, x_1, \ldots, x_P)\}_{1\leq i\leq m, 1\leq l\leq N}$ $x, x_1, \ldots, x_P \in V$ where $b_{il}(x, x_1, \ldots, x_P) = P(X_{ij} = x \mid X_{u_k v_k} = x_k, q_j = s_l)$, $(u_k v_k) \in \Theta ij, 1 \leq k \leq P$ the conditional pixel observation probabilities.

$\pi = \{\pi_i\}_{1\leq i\leq N}$ where $\pi_i = p(q_1 = s_i)$ is the initial state probability.

In the classical HMM definition given by Rabiner in [Rab89], a string resemblance can be observed with the $1D$ HMM but the observation sequence is different as a spatial context defined by Θ is considered. In this case the observations are related to pixel columns calculated as a product of the conditional pixel probabilities belonging to the observed column.

3.1.4 Likelihood calculus for the NSHP-HMM

Similarly as in [Rab89] the optimal evaluation of the likelihood $P(X \mid \lambda)$ is given by the forward-backward function. The forward function α is defined as being the cumulated field probability until column X^j of X when ending in state s_i, $\alpha_j(i) = P(X^1 X^2 \dots X^j, q_j = s_i \mid \lambda)$

1. $\alpha_1(i) = \pi_i \prod_{k=1}^{m} b_{ki}(X_{k1}, X_{u_1 v_1}), \dots, X_{u_p v_p},\ (u_p, v_p) \in \Theta_{k1}\ 1 \leq i \leq N$
2. $\alpha_j(i) = \left[\sum_{l=1}^{N} \alpha_{j-1}(l) a_{li}\right] \prod_{k=1}^{m} b_{ki}(X_{kj}, X_{u_1 v_1}, \dots, X_{u_p v_p},\ (u_p, v_p) \in \Theta_{kj}\ 1 \leq i \leq N\ 2 \leq j \leq n$
3. $P(X \mid \lambda) = \sum_{i=1}^{N} \alpha_n(i)$

In the same manner, the dual function β can be also defined which is similar to α but the resemblance in that case is calculated from the end.

The complexity of the α calculus is $\mathcal{O}[N^2 \times m \times n]$. With a particular left-to-right architecture, Saon has shown that the complexity may decrease to $\mathcal{O}[N \times m \times n]$.

3.1.5 Training of the model

During the training mechanism the objective is to determine the (A, B, π) parameters of the model λ which maximizes the product $\prod_P (X^{(r)} \mid \lambda)$, where $X^{(r)}$ are image samples used to train the model λ.

The well known Baum-Welch mechanism has been used for the maximum likelihood parameter estimation. To define the re-estimation process, first the β backward function should be defined :

1. $\beta_n(i) = 1\ 1 \leq i \leq N$
2. $\beta_j(i) = \sum_{l=1}^{N} a_{il} \prod_{k=1}^{m} b_{kl}(X_{kj}, X_{u_1 v_1}, \dots, X_{u_p v_p},\ (u_p, v_p) \in \Theta_{kj}\ 1 \leq i \leq N\ j = n-1, \dots, 1$

For the state transition probability matrix A, the re-estimation :

$$\overline{a}_{il} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{j=1}^{n_r - 1} \alpha_j^r(i) a_{il} \prod_{k=1}^{m} b_{kl}(X_{kj+1}^{(r)}, X_{u_1 v_1}^{(r)}, \dots, X_{u_p v_p}^{(r)}) \beta_{j+1}^r(l)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{j=1}^{n_r - 1} \alpha_j^r(i) \beta_j^r(i)},\ 1 \leq i, l \leq N \qquad (3.6)$$

Similarly, the conditional pixel observation probabilities :

$$\overline{b}_{il}(x, x_1, \dots, x_P) = \begin{cases} \dfrac{\sum_{r=1}^{R} \frac{1}{P_r} \sum^{n_r}_{\substack{j=1\ s.t. \\ X_{ij}^{(r)} = x\ X_{u_k v_k}^{(r)} = x_k}} \alpha_j^r(l) \beta_j^r(l)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum^{n_r}_{\substack{j=1\ s.t. \\ X_{u_k v_k}^{(r)} = x_k}} \alpha_j^r(l) \beta_j^r(l)} & \text{if denominator} \neq 0 \\ b_{il}(x, x_1, \dots, x_P) & \text{otherwise } x, x_1, \dots, x_P \in \{0, 1\}, \end{cases} \qquad (3.7)$$

where by $P_r = P(X^{(r)} \mid \lambda)$ stands for the emission probability of sample $X^{(r)}$ and n_r is its length. The pixel probability re-estimation is based on counting the number of times when a given pixel configuration is encountered.

3.1.6 Decoding in the NSHP-HMM

Once the parameter re-estimation using the training patterns is finished, the model is able to recognize pattern samples not belonging to the training corpus. As model discriminant approach has been chosen for this purpose, for each entry in the vocabulary, a separate word NSHP-HMM is created. Each model λ_i has been trained with samples belonging to this word model. The recognition process is performed on height normalized images. The conditional probability of models given the image is computed via Bayes rule. This probability gives the likelihood that the image comes from a particular class. The class for which this likelihood is maximum is chosen as the required class. In other words, the image sample X comes from the model λ^* where :

$$\lambda^* = argmax_{\lambda \in \Lambda} P(\lambda \mid X) = argmax_{\lambda \in \Lambda} \frac{P(X \mid \lambda)P(\lambda)}{P(X)} = argmax_{\lambda \in \Lambda} P(X \mid \lambda)P(\lambda) \qquad (3.8)$$

since $P(X)$ is constant for a given image. Here Λ denotes the set of models for all classes.

3.1.7 Experiments and results

The model was used with success for different recognition purposes considering handwritten digit and word datasets. In this section just a few results will be given considering just the most important task as separated character and digit recognition and word recognition for reduced size vocabularies. For an exhaustive result survey with all the parameter details, please refer to [Sao97]. The results concerning the separated characters and digits can be found in Tab. 3.1

Database	**Database type**	**Vocabulary**	**Accuracy**
-	printed characters	26	98,60%
C-SRTP	handwritten characters	26	53,84%
C-LIX	handwritten characters	26	63,32%
UNIPEN	handwritten digits	10	93,36%

TABLE 3.1 – Saon NSHP-HMM results concerning different separated digit character datasets

For reduced size vocabularies the author has tested the NSHP-HMM for different benchmark datasets like SRTP, A2iA, LIX. A brief result report can be found in Tab. 3.2.

Database	**Database type**	**NSHP-HMM order(P)**	**Accuracy**
STRP	handwritten check amounts	3	78,00%
STRP	handwritten check amounts	4	90,08%
A2iA	handwritten check amounts	4	82,50%
LIX	handwritten check amounts	4	91,10%

TABLE 3.2 – Saon NSHP-HMM results concerning different handwritten word datasets

3.1.8 Conclusions

Taking into account the theoretical framework developed by Saon and the application of the NSHP-HMM for reduced size vocabularies, we can state the followings. The novelty of the method is in coupling the NSHP performing conditional joint pixel probability estimation with an HMM to track the temporal aspect of handwriting. The $2D$ context assigned by the Θ_{ij} allows to consider a local pixel neighborhood, which extends the classical $1D$ model to a model appropriate for handwriting modeling and recognition.

Another important aspect which is the main advantage of the system, it is a holistic one. That means, there is no segmentation, the analyzed shape is considered as an entity avoiding the errors coming from the segmentation of the word into letter or graphemes. The results performed by Saon on different handwritten character and word datasets have shown the superiority of the model in accuracy terms speaking, by over performing the results given by other researchers and models.

The drawbacks of the model can be derived by the nature of the of the NSHP-HMM. The structure of the model and the constraints raised by the non-symmetric aspect of the NSHP-HMM leads to a complex handwriting recognition system. As a 3rd order neighborhood has been mainly used, the re-estimation of the conditional pixel probability is not symmetric so instead of one NSHP-HMM in this case 4 similar NSHP-HMM are considered. Each NSHP-HMM is carrying out one of the possible 4 symmetries, while the final decision is given be the cumulus of the 4 separate NSHP-HMM models.

Even if there is no feature extraction, which is quite a time consuming process, the re-estimation of the NSHP which provides the observations for the HMM is based on a local estimation, which does not perform sufficient information on the form. A more considerable neighborhood can be also considered but such an estimation conducts to a exponential complexity as stated by Saon [Sao97] and Choisy [Cho02].

The sensitivity of the model for slant and skew can be explained with the fact than the model observes columns which in such cases can be observed by different states which can deeply modify the likelihood calculus. Such a variation leads to an unstable system.

While the results are very encouraging for reduced size vocabularies, as the method is a holistic one, a possible extension to a larger dictionary will be not possible as stated by Choisy [Cho02] and by others in the literature [KSS03, Bun03]. As there is no segmentation, the model discriminant approach cannot work for a large vocabulary where is necessary to model the different letters and letter-ligatures in order to distinguish between the different word models.

To summarize, the holistic model proposed and tested by Saon for reduced size vocabulary is opening a new trend in the handwriting recognition by considering the handwriting as a bi-dimensional signal instead of following the classical HMM theory. The result achieved outperforms

all the results claimed by other scientists. However, an efficient trade-off should also be made in order to achieve good results taking into account the computational complexity of the model.

3.2 Analytical extension of the NSHP-HMM

3.2.1 General framework

Considering the drawbacks enumerated before especially concerning the vocabulary opening, Choisy proposed an extension of the system [CB02, Cho02]. The author is considering a handwriting modeling based on an analytical approach. Instead of considering the word shape as a whole entity, the he proposes an analytical approach, where the baseline element is the letter. The concatenation of letters into words is assured by the *word-meta models* allowing to handle more precisely the inter-letter connections called *ligatures*. The general word NSHP-HMM models are generated by integrating the different letter NSHP-HMMs into the word-meta model allowing to avoid an explicit word segmentation into letters or graphemes which raise problems concerning optimality and reliability.

The challenge of such kind of system is to solve the training issue of the letter models and the word meta-models. For this purpose Choisy is proposing a solution based on *cross-learning* of letters through the words. The cross-learning mechanism is based on the well-known Baum-Welch algorithm, allowing the emergence of the relevant letter information considering the different contexts wherein the letter can be found and the redundant information extracted from the same letters in different positions in the word shape.

The training procedure optimizes at word level. Only the orthography is required and using the word meta-models different ortographical variations can be also modeled with such an approach. There is no initialization process, the system can manage automatically all the parameter estimation through the Baum-Welch algorithm for *Maximum Likelihood Estimation* (MLE).

Another research aspect proposed by Choisy [CB03] concerns the NSHP-HMM in handwritten word image normalization, where based on the different model states, the model is used to normalize the image and once the normalization is performed, statistical classifiers as Neural Networks (NN) and Support Vector Machines (SVM) [ACRS01, SBS99] have been used for recognition purpose. These type of classifiers need initial data size normalization due their constraints [Bis95, CST00].

The normalization is based on the number of states of each word model. The normalization criteria is to calculate the average of the pixel columns read by the same state supposing the fact than these columns are more or less dependent of each other. Once the image is normalized (vertically by the NSHP-HMM, horizontally by the classical normalization method) the classifiers can perform their job.

3.2.2 Analytical approach

In the proposed analytical system, for each letter of the considered alphabet there is a corresponding letter NSHP-HMM. The meta-models are considered to model the link between letters as in other works just a simple concatenation is performed without any consideration for the different ligatures which can occur to connect two letters. In Fig. 3.5 we can observe different forms of the French word "francs" and the corresponding meta-model which can handle the errors committed by the writers. It is also able to model the different abbreviations of the same word.

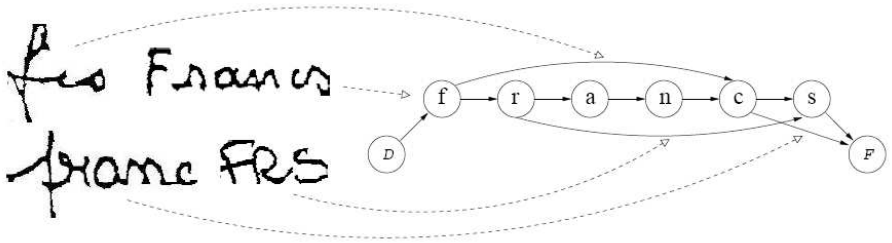

FIGURE 3.5 – Word meta-models for the French words "francs" and the different abbreviations occurring in the bank checks

3.2.3 Formal definition of the models

For this analytical handwriting approach the author is using the same models described already by Saon in [SB97]. The only modification is performed by adding two specific states to the classical NSHP-HMM model in order to model the initial state probability to start in a state or to end in a given state of the model. The notation and definition can be derived directly from the HMM definition given by Rabiner in [Rab89].

Definition 3 : A HMM with specific starting and ending states is defined :

$S = \{s_1, \ldots, s_N, D, F\}$ denotes the N states of the model and D, F are the two specific states added to the model. A state at time t is denoted by q_t where $q_t \in S$.

$V = \{v_1, \ldots, v_M\}$ denotes the number of the possible M symbols observed by the HMM. $O_t \in V$ denotes a symbol at instance t.

$A = \{a_{ij} \cup \{a_{Di}, a_{iF}\}\}_{1 \leq i,j \leq N}$

where $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$, $a_{Di} = P(q_1 = s_i \mid D)$, $a_{iF} = P(F \mid q_T = s_i)$ where A is the state transition probability matrix. The a_{Di} denotes the probability to start the analysis in state i, while a_{iF} denotes the probability to stop in a state i.

$B = \{b_j(k)\}_{1 \leq j \leq N, 1 \leq k \leq M}$, where $b_j(k) = P(O_t = v_k \mid q_t = s_j)$. B is the observation probability matrix for the different states j. The specific states denoted by D and F cannot observe anything.

To simplify the notation such a model is denoted by $\lambda = \{A, B\}$. The π is discarded as the specific states substitute this parameter.

The goal of a HMM with specific state is to better distribute the data in the different states. The final state allows to improve the distribution of data favoring the final states to analyze the information [Cho02]. Considering the different Viterbi paths, we can observe the fact that the first states absorb the noises which can come from the image context. In Fig. 3.6 a classical left to right hidden Markov model and a left to right model with specific states is presented.

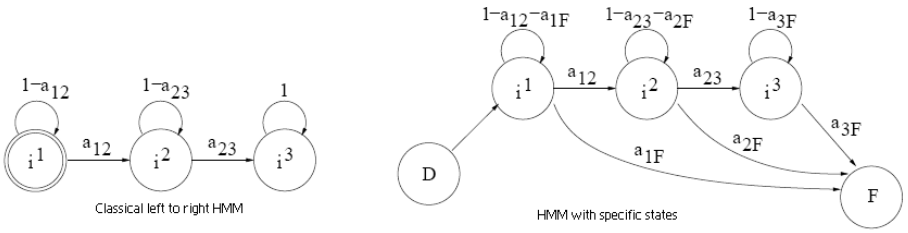

FIGURE 3.6 – A left to right model and a model with specific states where the state duration in the final state is modified

Considering the model with specific end state, the profit can be expressed in term of state duration. While for the classical left to right model the self transition is 1, in the second case integrating the specific state in the model allows to diminish the importance of the final state. So, it will be a penalty for word shapes with extra length, not considered by the model during the training (see Fig. 3.6).

Some works in the literature use equivalent notion called as *null-states* or *empty-states* which serve to concatenate consecutive letters. While in that case the letter-transitions are not considered, in Choisy's system we are able to model the inter-letter connections and the letter endings thanks to the meta-models.

Considering the definition of the NSHP-HMM given by Saon in Section 3.1.3, the extension of the NSHP-HMM with specific states is as follows :

Definition 4 : A non-symmetric half-plane Hidden Markov Model with specific states :

$V = \{0, 1\}$ is the set of observation symbols

$S = \{s_1, \ldots, s_N, D, F\}$ the set of normal states plus the specific states where N is the number of normal states used by the HMM.

$A = \{a_{ij} \cup \{aDi, aiF\}\}_{1 \leq i,j \leq N}$ where $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i), 1 \leq i, j \leq N, a_{Di} = P(q_1 = s_i \mid D), a_{iF} = P(F \mid g_T = s_i)$

$B = \{b_i(y, \Theta, c)\}$ where $s_i \in S \setminus \{D, F\}$ is the observation probability for a given state i to analyze a pixel of color c at ordinate y, considering a given neighborhood Θ.

To simplify the notation, we denote by $b_i(O_t^k)$ the observation probability of column t of

image sample k in state s_i, which is calculated as a product of conditional pixel probabilities belonging to the column.

$$b_i(O_t^k) = \sum_{y=1}^{M} b_i(y, \Theta_{iy}, c) = \prod_{y=1}^{M} P(X_{ij} \mid X_{\Theta_{ij}}, q_i) \tag{3.9}$$

$P_k = P(O^k \mid \eta)$ is the probability to observe the image k considering the NSHP-HMM η .

Considering the previous notation we should also define the Θ_{ij} denoting the neighborhood of X_{ij}. Similarly as in case of Saon (Fig. 3.2), the author is fixing the neighborhood to be exactly the same for each analyzed pixel. This Θ_{ij} is fixed in terms of pixels which will contribute in the calculus and their relative position to the analyzed pixel. The number of contributing pixels will determine the order of the NSHP as presented in Fig. 3.3.

The parameters which determine the NSHP-HMM are similar as in case of Saon described in Section 3.1.3. In conclusion, the model is composed by the height of the analyzed columns, the neighborhood, the number of states and finally the structure of the HMM defined by the allowed state transitions.

3.2.4 Model fusion

The goal of the model fusion process is to build a general word model, considering as baseline elements the letter models and the word meta-models which establish the order of the letters in the word and the link between them. The meta-states in the meta-word models will be replaced with the corresponding letter models and in that way the general word model is automatically generated. This word model is called by Choisy [Cho02] as being the *general word model* or *word NSHP-HMM*.

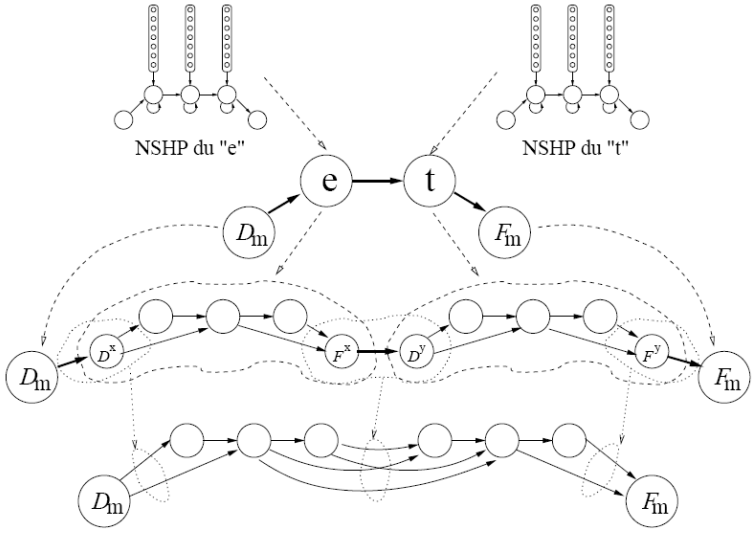

FIGURE 3.7 – The general word model creation process of the word "et" in [Cho02]

In Fig. 3.7 the author gives a detailed scheme of the fusion process to generate the word NSHP-HMM models. During the letter fusion, the specific states will be eliminated in order to link the different letter states in the word model. In that sense the model includes some constraints, as for the word-meta models no self-transition is allowed. The final word NSHP-HMM model allows to estimate the word entity, while the word meta-model allows to estimate the letter transitions and indirectly the re-estimation of the letter models. For more detailed description of the fusion process, please refer to the thesis of Choisy in [Cho02].

3.2.5 Cross-learning concept

Considering the model fusion based on the letter models and word meta-models, during the Baum-Welch parameter re-estimation is not just for the general word NSHP-HMM model but also for the re-estimation of the letter models and the word-meta models too. The former two estimations are based entirely on the re-estimation of the general word model. For that reason, the author is proposing a cross-learning of letters based on the re-estimation performed on the general word models. The Fig. 3.8 illustrates this re-estimation cascade.

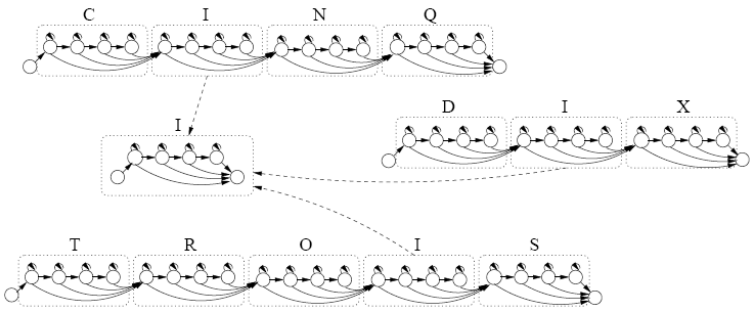

FIGURE 3.8 – The cross training mechanism for the letter "i" considering different word models in [Cho02]

The cross-learning concept contains three distinguish steps :

1. Firstly, considering the different letter models and the word meta-models, the general word model is created based on the concatenation by fusion as described in Section 3.2.4.
2. In the second step the general word model re-estimation is performed based on MLE criterion adapted to the Baum-Welch training, serving to re-estimate the HMM parameters.
3. In the final third step, the information obtained during the word model re-estimation is dispatched into the meta-model and the letter models respectively, allowing also their re-estimation. This information derived at word level will maximize their resemblance throughout the letter and the word meta-models.

As stated by Rabiner [Rab89] during the re-estimation process, the λ^* can be considered more optimal than λ. The convergence of the algorithm is assured but in order to achieve the λ^*, some iterations are necessary. After each iteration, the letter models and word meta-models are re-estimated and the general word models are rebuilt after each iteration step taking into account the new letter and meta-models.

3.2.6 Word normalization by the NSHP-HMM

Considering the success of different type of HMM systems used in speech recognition and handwriting modeling and recognition is because of their capacity to model the different time based signals and the absorption of noises and distortions which occurs often in handwriting. This advantage is coming from the fact of coupling local observations with dynamic programming type matching, allowing to distribute the signal information into different HMM states, considering the different noises which will be also integrated in the model. The advantage of the model can be considered as a penalty, as the local vision observed should be based on the assumption that the different observation are not correlated and there is no conditional dependency between the different observations.

Choisy introduces in his formalism the notion of *local vision* and *global vision* and the corresponding models to imitate the human reading.

A *local vision model* (LVM) is considered if the observations observed by the model are independent and the modeling of the shape is based on the maximization of the local observation probabilities. The dual system, the *global vision model* (GVM) is considered when there is no independence constraint between the observations. A correlation between the observations can exist and the shape probability is estimated through its global view.

The SVM and the classical NN based models can be considered as GVM while the HMM approaches and some specific NN models like the convolutional network [LBBH01] and the *Time Delayed Neural Network* (TDNN) [SGH94], which can be considered as special feedforward neural network where the layers performs successively high-level features extraction. The produced outputs can be interpreted as probabilities.

The central idea considered by Choisy is to combine the strength of the GVM and the LVM. While the LVM is considered for normalization, the GVM is considered for recognition issues. It is well-known that HMM can model the signals but for class separation task the NNs have shown their supremacy in different pattern recognition problems.

The NSHP-HMM is considered for normalization purpose, while a classical MLP is called for the global recognition. In Fig. 3.9 the complete system scheme is presented.

In order to get an idea about the normalization, a brief description will be given in the following section.

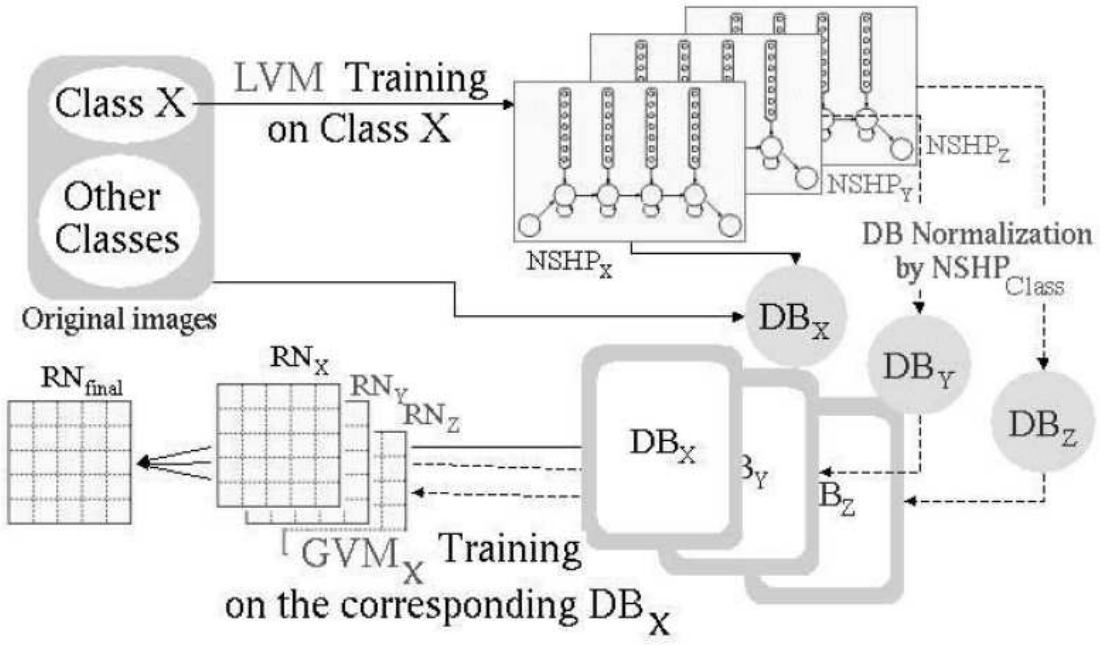

FIGURE 3.9 – The complete scheme for using the HMM in normalization and NN in recognition

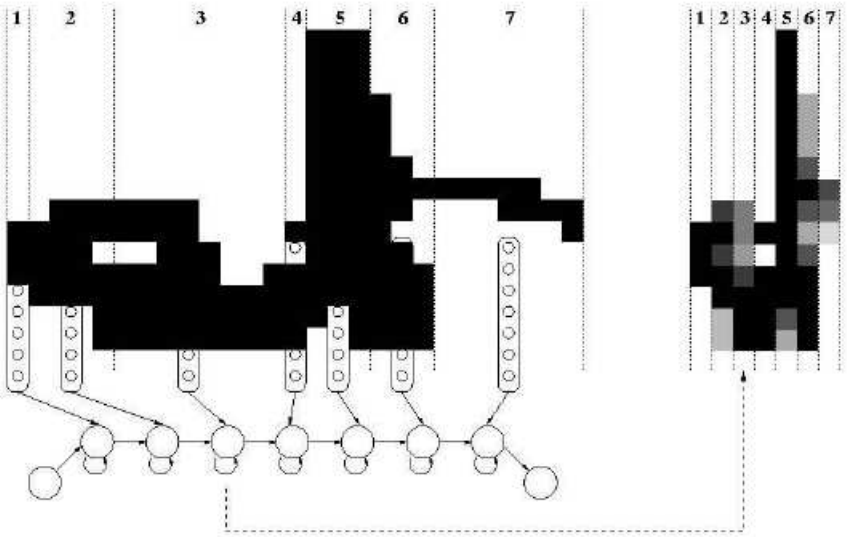

FIGURE 3.10 – Normalization of the French word "et" by the corresponding NSHP-HMM. The normalization is based on the mean value of the columns observed by the same state of the model

The proposed approach is based on the Viterbi algorithm, more exactly on the Viterbi path considered as direct outcome of the recurrent algorithm. Taking into account the state sequence considered as being optimal, the column(s) analyzed by the same state of the NSHP-HMM are grouped. The columns grouping is achieved based on the mean value of the pixel columns (see Fig. 3.10). Using this technique each image normalized by the same NSHP-HMM will have the same width, equal to the number of states in the model. The input image before passing into the normalization model, is height normalized, so that the fixed size constraint is performed. Hence, a GVM like MLP or SVM can be applied in cascade to perform the final recognition based on statistical learning.

3.2.7 Experiments and results

The results obtained for different handwritten datasets (SRTP(26), LIBRE(26), VM(28) and LIX(26)) described in details by Choisy in [Cho02] are very interesting.

After a differential height normalization [Sao97, Cho02] the image is transformed to a new dimension, where the image height is fixed to 20 pixels, while the image width is proportional to the original image width. In Tab. 3.3 are shown the results obtained by the Viterbi algorithm, the Baum Welch mechanism, the combination considering the product of the likelihoods produced from each of these two systems and finally some result are given for SVM and MLP in recognition, while the NSHP-HMM is used as normalizer.

Method	**SRTP(26)**	**LIBRE(26)**	**LIBRE(28)**	**VM(26)**	**VM(28)**	**LIX(26)**
Baum-Welch	86.19%	81.71%	81.84%	85.29%	85.49%	92.15%
Viterbi	86.52%	81.09%	80.87%	85.04%	84.72%	92.30%
Combination	86.44%	82.05%	81.84%	85.18%	85.28%	92.43%
SVM	-	-	81.84%	-	-	-
MLP	-	-	-	-	85.49%	-

TABLE 3.3 – Recognition results concerning the NSHP-HMM using different algorithms

The results obtained for the database LIBRE and VM show the superiority of the NSHP-HMM normalization against the classical linear normalization.

3.2.8 Conclusions

Considering the NSHP-HMM in word recognition, we can conclude the followings. The system does not contain any a priori information, just the dictionary should be given and the system is building automatically the different word meta-models constructing them using the letter-models and word meta-models. The analyzed pixel column information is spread out in the

different states of the model allowing to absorb the different noises and allowing a horizontal elasticity necessary to treat a signal like handwriting.

The choice of an analytical approach, justified as a holistic approach adopted by Saon, cannot work for an extended size vocabulary. The cross-learning mechanism proposed takes care not just on word models but on letter models too, by building a more relevant letter model and respectively a ligature model obtained by the others, using just a simple letter concatenation implanting a certain rigidity in the system.

The normalization by NSHP-HMM is also considerable, as the author has shown the supremacy of the technique against the common linear normalization. While the linear transformation is just a simple global modification of the word shape the non-linear operation performed by the NSHP-HMM allows to focus the normalization at letter level.

The elastic NSHP-HMM in role of LVM is justified as it acts like a local letter estimator, while the SVM and NN considered as GVM performs a global vision of the form.

3.3 General conclusions concerning the NSHP-HMM

In that section we attempt to describe the advantages and disadvantages respectively of the two systems proposed by Saon and Choisy for handwriting recognition using a segmentation-free, context based HMM models. This study permits to consider the qualities of these two recognition systems used for modeling a $2D$ signal like handwriting. In the same time gives us the possibility to propose different improvements considered by us as being vital to keep the robustness and the reliability of the system. Such improvements can stand even in a case of a more considerable size vocabulary, containing much more complex letter shapes as the letters we have used from the Latin alphabet.

The concept of the NSHP coupled with an HMM is new and allows an optimal repartition of the data in the different states of the model. The left context considered by the NSHP-HMM gives to the classical $1D$ HMM a truly $2D$ aspect allowing a more precise and realistic modeling.

At formal level, the extension of the Rabiner's Baum-Welch and Viterbi algorithm open new perspectives in the $2D$ stochastic modeling. The analytical extension of the Saon's system is necessary as a more precise modeling, based on letters and meta-models performs a realistic vision close to human reading capacity. It has been proved that the word reading is considered in its global way, but considering respectively the local specificities emitted by the letters. However, as invoked by many authors, the HMMs have a very good and reliable modeling capacity but their recognition ability is much more reduced.

A main advantage of the system is its capacity to converge to an optimal solution. While the neural network based approaches count many free-parameters, the parameters number of the NSHP-HMM are negligible. That is the reason why the HMM based models converge much more

faster and they do not need a considerable dataset to refine the system parameters [LBBH01].

As drawbacks we can invoke different facts. The usage of the NSHP is new in the handwriting domain but its data sampling process is incomplete or poor, as just low-level information has been considered till now to feed the system. As mentioned by Choisy, the order of the neighborhood has a considerable impact on the recognition score [Cho02, Sao97]. Analyzing the memory complexity of a model having N normal states, with a neighborhood of order V, analyzing an image of height Y is $\mathcal{O}[N(N+2^V Y)]$. Choisy has used also a 3^{rd} order NSHP-HMM. This seems to be an equitable trade-off between algorithm complexity and recognition performance.

The order of the neighborhood controls the mass of the information implanted in the model column wise but meanwhile the required memory size explodes so the used context is not sufficient. A possible growing neighborhood order drives the system to an impossibility to be realized.

In order to compare the results obtained by the two models, a comparative table is given in Tab. 3.4.

Ref.	**Dataset**	**Method**	**Top1**	**Top2**	**Top3**
[SBG95]	SRTP	analytic	82.80%	86.90%	89.40%
[Sao97]	SRTP	global	90.01%	-	92.60%
[Sao97]	A2IA	global	82.50%	89.60%	92.70%
[Sao97]	LIX	global	91.10%	95.80%	97.40%
[Cho02]	SRTP	analytic	86.20%	92.50%	95.20%
[Cho02]	LIBRE	analytic	81.80%	89.20%	92.40%
[Cho02]	LIX	analytic	92.20%	96.50%	97.60%

TABLE 3.4 – Comparative results obtained by Saon and Choisy for different datasets

Considering the global approach proposed by Saon and the analytic one proposed by Choisy, we can state the followings. The results obtained by Saon for the SRTP dataset in Top1 outperform the results reported by Choisy for similar conditions. The Top2 reported by Choisy outperforms the Top3 result of Saon which can be explained by the fact that the analytic method it giving better performances. This fact is also confirmed by the results reported for the LIX dataset. The differential height normalization introduced by Choisy helps to ameliorate the performances of the analytic approach. Meanwhile, Saon has introduced corrections in the training process and he used the knowledge of the natural length allowing him a gain of 4.5% compared to the baseline system. Choisy in his system does not use any kind of extra information.

We can conclude than that information mass based on conditional pixel probabilities has no sufficient descriptive force to feed the model. Even if such data sampling presume a context, the quality of this context is poor as it is based just only on pixel information without considering

any kind of high-level perceptual context often used in the literature and by humans. One of our improvement is based on this conclusion. We would like to extend this low-level information analyzed by the NSHP-HMM with some perceptual content.

Considering the entire NSHP-HMM system, we can remark that it is a complex system. The memory complexity grows exponentially in function of the neighborhood order while the algorithm complexity is $\mathcal{O}[N(2NT-T-N+2)]$, where T is the length the considered observation sequence.

It is clear that the NSHP-HMM is acting on pixel level, so no complex, time costly feature extraction mechanism is used. But the order of the neighborhood and the corresponding sub-models are time costly. Instead of using the Baum-Welch training and the Viterbi decoding algorithm once for each model in that case due to the non-symmetric property of the NSHP-HMM, for each model 4 corresponding sub-models should be considered. This aspect has a similitude in case of the convolutional networks [LBBH01, CVB05a], where the layers are composed by different maps extracting different features.

This is the second main drawback of the system as in case of real time applications like postal address reading [KCGM93, Sri00, DFV97, MS99], bank check amount reading [GAA$^+$01, GS98], form processing, etc., the time factor is very important issue. A possible run-time complexity reduction could be a benefit for the further investigations.

3.4 Proposed approach

Looking around in the vast amount of work proposed by different authors during the last few decades, we consider that an appropriate solution is to keep the $1D^{1/2}$ analytical NSHP-HMM as a baseline modeling for handwriting. The $1D$ models are too rigid while the truly $2D$ models proposed are not efficient in complexity terms speaking.

The first work developed in this research thesis is oriented towards the introduction in the former system (based totally on low-level pixel information) of some high-level perceptual information derived from the geometrical specificity of the analyzed shape. The idea is not totally new as many works can be found in that sense but in those works the pixel information and the perceptual information is treated separately instead of trying to merge together the low-level information with the high-level one. Such mechanism is much more realistic approach taking into account the human reading process where these information are processed together.
The main concern was to merge properly these information having different semantical meanings, without changing the analytical NSHP-HMM framework described in details above and the corresponding stochastic constraints [Rab89]

The second issue concerns a strategy to reduce the time complexity of the Viterbi decoding process. Even if the NSHP-HMM is built-up on letters there is no explicit segmentation. A level-

building type algorithm proposed by Koerich [KSS03] cannot work properly in this formalism. Hence, we propose a new technique based on some threshold values calculated at letter level. More exactly, a likelihood value which can be explained intuitively by the probability to match an unknown analyzed word shape to a given model considering just the prefix part of the word. Such kind of partial analysis can reduce substantially the search complexity of the decoding algorithm based on dynamic programming strategy. The strategy is based on cumulative letter thresholding.

Finally, a comparison study is developed in order to show the efficiency of different neural network based techniques and stochastic models for handwritten digit recognition. We consider it as an important issue to explain why a significant superiority can be observed in the literature in the usage of neural based techniques for separated digits recognition and similarly for HMM based techniques when connected digits have to be recognized. Besides the most recent works in the field, some personal contributions will be also highlighted considering the digit recognition by a neural network scheme, by the NSHP-HMM and finally some combination schemes will be discussed in a multi-classifier framework.

4

High-level information implant in the baseline NSHP-HMM

We propose in this chapter a generic method to implant high-level perceptual information in the stochastic 2D model presented above (see Chapter 3) without any consideration about the type of the perceptual content. Such a technique allows to preserve the stochastic aspect of the model with its constraints. In the same time the new perceptual information will enrich the quantity and the descriptive quality of the analyzed information considered by the model.

This chapter describes the theoretical framework and the application of the model for real data. For an easy comprehension, the chapter is organized in 2 sections as follows :

- In the first section the objective of our research is invoked.
- In the second section a theoretic overview of the system is given with the corresponding experiments performed to validate the system.

4.1 Objectives

The baseline NSHP-HMM described by Saon [Sao97] is working on pixel observations. At the NSHP level, the observation probability is calculated along a column as a product of the conditional pixel probabilities composing the given column allowing to measure the quantity of the pixels and their position in the column. Such a description does not contain enough capacity to describe complex shapes such as handwritten words.

We have observed that all the pixel zones are considered similarly, while some zones containing more discriminative information should be more considered by the system.

In order to increase the discriminating power of the model, we want to introduce inside or outside the model information carrying a semantical meaning of the analyzed shape components. Our approach is based on human vision where not just the form is considered in his own but the

specificities. In case of handwritten words these are : presence or absence of loops, ascenders, descenders. Highlighting these information carried by the different pixel zones, we help the system to better adapt the letter-models, the word meta-models and the general word model to recognize the different words. The different perceptual features like cutting points, ascenders, descenders, etc. can improve in different manner the system. For example, the cutting points can help the system to better approximate the different letter limits in the model as stated by Choisy in [Cho02]. The letter limits decided by the model cannot be considered as being real cutting points.
The ascenders, descenders can help the general word model to better distinguish between shapes as this information can accentuate a Viterbi path instead of another.

The aim is to study the different possibilities, implant strategies to combine the low-level pixel information with high-level ones in the framework of the NSHP-HMM based on column observations.

4.2 General description of the implant problem

Considering the approaches proposed by the literature two possibilities can be discussed for the combination of different type of information. The first one separates the problem in sub-problems considering the low-level information as input for a recognizer, while the high-level information for another and a combination of the different classifiers gives the final result.

The second solution is not to separate the different type of information and to use them in the same classifier. Instead of using a technique familiar to neural approaches, where the input vector values can be totally independent each from another, our aim is to combine these information in realistic manner, giving a physical sense to the combination. This physical meaning comes from the fact that the human reader also considers specific parts in the word shape and their analysis and recognition highly contributes to recognize in mass the whole word entity. We are trying to model such a human behavior inside the NSHP-HMM.

As the NSHP-HMM is working on pixel information, our aim is to implant the extra information on this level.

Taking into account the objectives mentioned before, the problem is to know what kind of information we can assign to the different image pixels considered by the NSHP and how we can integrate it in the formal description of the model. The challenge is to introduce into the model some extra information without disturbing it (i.e. the extra information can also be considered as possible noise) and to transform the model to accept such extra information. A graphical representation of the general system overview is shown in Fig. 4.1. We can observe here than instead of using just the conditional pixel probabilities estimated by the NSHP, we are also considering the structural nature of the shape and we implant these two sources into the

observation analyzed by the HMM.

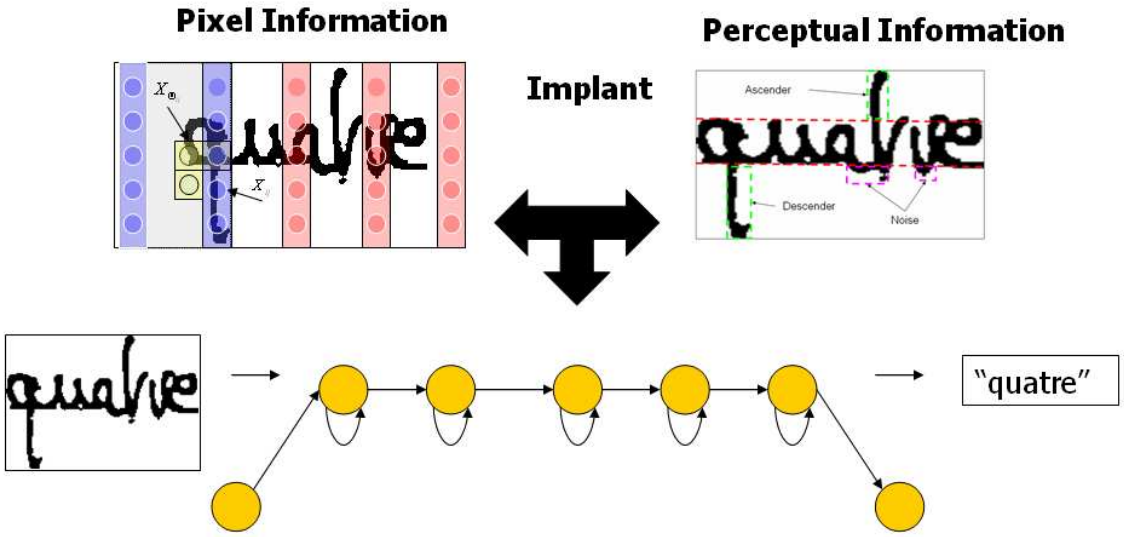

FIGURE 4.1 – The general system overview of the structural information implant in the NSHP-HMM

It is well-known that in handwriting recognition, high-level perceptual information is generally helpful for recognition purposes, so we have oriented our work towards the implant of such kind of high-level information. Unfortunately, this extra information can be perceived as useful information but in the same time can be considered as noise.

Analyzing the noise effect, we discard this possible problem source, by transforming the perceptual information at pixel level, which can be perceived as a weight for the pixel. This weight carries the information of the pixel if this pixel has perceptual qualities or not. This weight can be considered as a measure of importance of the pixel in the observed column. This weight assigned to a pixel in the column will give him an extra power allowing to control its importance among the others. The implant mechanism is based on the classical mathematical weighting being useful in many applications.

Another issue is to know if this weight will be mixed with the pixel observation probability or if it will be used as external global weight. These two possibilities will be studied later considering the training algorithm acceptance. The trade-off should be done between the information performed by the NSHP-HMM and the structural information extracted from the word shape. For example, the NSHP-HMM uses a meta-model composed of letter models, hence it is obvious that if we add information concerning the letter limits, the system should improve its modeling capacity. Using the knowledge of explicit letter limits (i.e. using a segmentation process) we can force the Baum-Welch training to estimate the state transition probability matrix (A) according to these limits. This kind of adding mechanism can be extended to features even within the letter

models. However, in the Baum-Welch training, we re-estimate the letter models just considering the meta-models, where there is no kind of explicit segmentation information (for details see Section 3.1.5). Therefore, the letter model re-estimation is not precise. In other terms speaking, we do not know where are the precise limits of the different letters composing the word. By adding such kind of extra information like cutting points, the letter model re-estimation should become more precise, as instead of considering just the conditional pixel probabilities we accentuate the importance of certain pixels which carry perceptual information. Hence, the meta-models and the general word models should better model the word patterns. As the baseline system cannot precisely locate the letter limits (the segmentation is implicit instead of an explicit one, so consequently not sure enough in comparison with the real segmentation) the NSHP-HMM is based on the information coming from the letters' inside. So the mass of the letter has greater impact than its limits. We can argue that other kind of structural information like ascenders, descenders, etc. representing the letter morphology can really have a substantial impact on the letters' inside in order to better distinguish on letter level. The actual baseline system based just on the conditional pixel probability of pixels in the columns cannot accentuate the presence of a structural information as each pixel component has the same importance.

Considering what we have proposed above for the structural information introduction and aiming to conserve the same theoretical modeling with Markov random fields an the corresponding NSHP-HMM constraints, a common idea for the enhancement is to multiply the column probability by extra information converted in some weight. This weight can accentuate the contribution of a pixel or another in function of its qualities. This extra information can be characterized by the quantity or the quality. The quality can be expressed by assigning a weight to the pixel according to the feature nature. The quantity is expressed by counting the number of features extracted which will be converted into a weight according to the whole feature occurrence.

The difference between the quality and quantity is double. If we characterize the perceptual features by quantity, we are not considering the type of the perceptual feature. In that case an ascender or a descender is considered as being similar contributing in the same manner for the pixel/column observation. Using a quantitative measure allows to distinguish between the different features. It is possible to assign different weight factors to the different features based on the discriminative power of the given features. For example, in Bangla script a higher importance should be assigned to ascenders as the number of descenders is not so important for this script. Meanwhile, the occurrence of such infrequent features can also be a reliable source to distinguish a word from another one.

In that first approach the new column observation is calculated as a product of the conditional pixel probabilities and the weighting is performed by the features (feature points) belonging to the column, but the re-estimation is based just on the pixel re-estimation. The features contribution in that approach is static. No kind of re-estimation is performed at this level. In the second

approach, the idea is to modify the re-estimation process, namely the observation probability in order to implant inside the training mechanism the structural information. In that case, the pixel observation will be slightly different. While in the baseline system the observation is based only on the conditional pixel probability throughout a column, in this case this column observation probability will be based on the pixel probability and their structural power.

In order to be generic in the further investigations, just a high-level information will be considered without any reference to some kind of well defined perceptual/structural features.

4.3 Formal description of the implant

4.3.1 The NSHP-HMM formalism

In order to implant the mechanism, in the NSHP-HMM formalism, some basic notions have to be recalled from Chapter 3. Considering the simplified schema of Fig. 3.1 given in Fig. 4.2 we can pursue the formal description proposed by us.

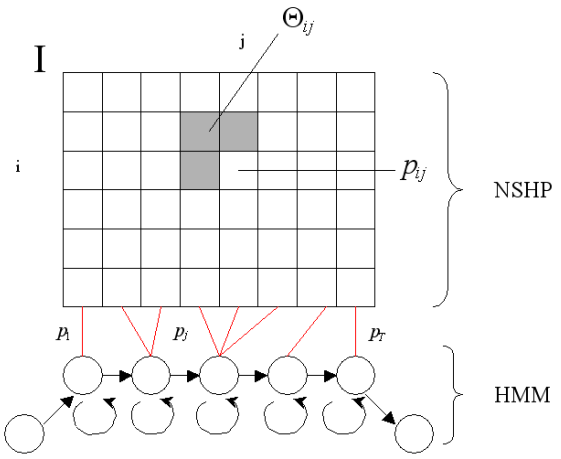

FIGURE 4.2 – The NSHP-HMM model

Let X be the analyzed image having m rows and respectively n columns observed by the NSHP. The joint field mass probability denoted by $P(X)$ of the image X can be computed following the chain decomposition rule of conditional probabilities :

$$P(X) = \prod_{j=1}^{n} \prod_{i=1}^{m} P(X_{ij} \mid X_{\Theta_{ij}}) \tag{4.1}$$

where m denotes the number of rows while n denotes the number of columns in the image X. Let the conditional pixel probability of a pixel (i, j) be denoted by p_{ij} :

$$p_{ij} = P(X_{ij} \mid X_{\Theta_{ij}}) \tag{4.2}$$

Let denote j the current column and i the current row. The column probability processed by the NSHP can be calculated as follows :

$$P_j = \prod_{i=1}^{m} p_{ij} \tag{4.3}$$

where m is the image height and p_{ij} is the conditional probability of pixel i in column j knowing its neighborhood Θ_{ij} in image X.

Considering equations (4.2) and (4.3) equation (4.1) can be computed as follows :

$$P(X) = \prod_{j=1}^{n} P_j \tag{4.4}$$

The notation used in equations (4.1-4.4) is similar as depicted in Fig. 4.2.
In this case P_j denotes the column observation given by equation (4.1). In order to simplify the notation in the further discussions just the notation given by the equation (4.4) will be used.

Our interest is to extend the meaning of P_j at pixel or column level by inserting high-level information at these perception levels considered by the NSHP-HMM analyzer.

4.3.2 The weighting mechanism

We consider that if the pixel carries high-level information, the weight derived from this type of information could be either shared on all the column pixels (i.e. each pixel probability can be weighted individually) or factorized along the column (i.e. the whole column is weighted by the same structural weight).

By weighting we mean :

- *Pixel level* : if the weighting is at pixel level, we can accentuate (weight) a pixel giving it an extra power which means in physical terms that we have seen the same pixel several times, where the number of times means its power among the other pixels.
- *Column level* : if the weighting is at column level, we can accentuate (weight) a column giving it an extra power which means in physical term that we have seen several times a given column.

This weighting should not disturb neither the Baum-Welch training mechanism nor the Viterbi search. That means if such weighting is applied the Markov constraints should be satisfied [Rab89]. In order to satisfy such constraints a normalization process is necessary. As we are multiplying or raising to a power the pixel or column probability with a weight, we are still in the probability domain. These new combined/reinforced observations can also be used as observation probabilities for the new NSHP-HMM model called structural NSHP-HMM.

Let denotes generally by w^{inf} this structural information considered as a weight measure. This weight can be interpreted according to the meaning of this information. This weight can

be calculated at pixel level or at column level. In function of this structural information the modified overall column observation probability can be described as follows :

1. If the structural weight is global for the column j, we propose to transform the equation (4.3) into :

$$\overline{P_j} = \left(\prod_{i=1}^{m} p_{ij} \right) \times w_j^{inf} \tag{4.5}$$

where w_j^{inf} is considered as being the weight calculated for the column j considering all the pixels (i, j) and their structural properties.

2. If the structural weight is local for the pixel (i, j), we propose to transform the equation (4.3) into :

$$\overline{P_j} = \prod_{i=1}^{m} (p_{ij} \times w_{ij}^{inf}) \tag{4.6}$$

where w_{ij}^{inf} is considered as being the weight calculated for the pixel (i, j) belonging to the column.

In the same manner, we can establish two other equations :

3. If the structural weight is global for the column j, we propose to transform the equation (4.3) into :

$$\overline{P_j} = \left(\prod_{i=1}^{m} p_{ij} \right)^{w_j^{inf}} \tag{4.7}$$

where w_j^{inf} is considered as being the weight calculated for the column j considering all the pixels (i, j) and their structural properties.

4. If the structural weight is local for the pixel (i, j), we propose to transform the equation (4.3) into :

$$\overline{P_j} = \prod_{i=1}^{m} (p_{ij})^{w_{ij}^{inf}} \tag{4.8}$$

where w_{ij}^{inf} is considered as being the weight calculated for the pixel (i, j) belonging to the column.

4.3.3 Local weight and global weight

Considering the weighting mechanism proposed by the equations (4.5 - 4.8), we can distinguish two cases. The weight w^{inf} can be local (4.6, 4.8) or global (4.5, 4.7). In this section we discuss these aspects.

If the weight is local at pixel level, instead of weighting the column, each pixel is weighted separately. Hence, a more precise estimation is performed. In that case the weight can be established according to the quality of the information. If a given pixel (i, j) has no structural meaning,

the $w_{ij}^{inf} = 1$ should be satisfied, otherwise w_{ij}^{inf} is calculated in function of the weight meaning. In that case the weight w_{ij}^{inf} can be interpreted as the influence (power) of a pixel according to its structural nature among the pixels belonging to column j.

If the weight is global for the column, the weight mechanism is applied for the joint pixel probability observation considered by the HMM. The weighting is applied for the whole column obviously derived from the nature of the pixels composing the column. While in case of local weighting we give importance of some pixels in the column, here the whole column is considered. This can be translated as the importance (power) of this column j among the other columns composing the image X.

4.3.4 The nature of the weight

An important aspect in our reflection was to find an adequate physical explanation assigned to each weight measure. Considering the general w^{inf} two different meanings can be assigned to this measure.
The first is qualitative , while the second is quantitative. By quality we mean that we can generate a rank based on the importance between the different type of information based on their discriminative power and derive the numerical weight values based on this assumption.
The second solution does not make any difference between the different type of high-level information, each type is treated as same, assigning the same importance to them.

We decided to discard the first option as in order to establish a rank concerning the importance of the information based on its type is going further than the objectives of this thesis. Such a rank can be established just with the help of psychologists studying such kind of problem based on solid experimental results.

In the second solution, the quantitative one is easier to implement and fits properly in the formalism described above. The basic idea of the NSHP is to consider the contribution of each pixel equally. In that case, no distinction is made at pixel level.

4.3.5 The source of the weight

To extract the weight from the analyzed image, we should define the weight itself. To integrate the high-level information in the model, the weight should also be reduced to pixel level.
Definition 2. Structural point :

In an analyzed shape, a pixel (i, j) is considered as being a structural point if the given point belongs to a given ν set, $(i, j) \in \nu$, where ν denotes different structural feature extractable from the word shape.

Considering the definition, once a structural feature is extracted from the analyzed word shape all the pixel components belonging to the feature are considered as structural pixels carrying extra

information.

4.3.6 The weight calculus

Such preprocessing is necessary for the information extracted from the word shape to allow a strict probabilistic framework. The normalization can be performed based on the nature of the information. If the nature of the weight is qualitative, the weight mechanism is complicated and it is based on a priori knowledge of the human reading perception not always scientifically argued.

If the weight is based on the quantity of the information without considering its quality, a more interesting normalization can be developed. In that case each structural information is considered as being equal. In the next sections different normalization will be developed and discussed, based on the quantity of the perceptual information and based on their nature.

Considering the weighting mechanism, an important issue has been raised.
Taking into account the different implant mechanism proposed for the NSHP-HMM is it possible to consider the weighting as an external one, instead of the internal proposed ? Are these techniques equal ?

However, if the weight is calculated outside the NSHP (i.e. projecting the analyzed words shape in a probability feature space, where each axis is a possible feature which can be extracted) the column observation and the structural weight are independent. Therefore, it is not necessary to introduce the weighting mechanism inside the model. The a posteriori probability given by the NSHP-HMM (denoted by event A) can be multiplied with the feature weight denoted by event B) through the following rule :

$$P(A \cap B) = P(A) \times P(B) \tag{4.9}$$

considering the probability of the co-occurrence.

Are these techniques equal ? The raised question is quite interesting, as if we can proof that the weighting of the columns of the image X according to the structural information in the given column learn by the NSHP-HMM can be replaced with the a posteriori probability performed by the NSHP-HMM on the simple image columns weighted by a simple global weight. In that case it is not necessary to encapsulate inside the model the different pixel or column-wise probability measures.

In formal terms speaking, let us have $0 < w_0, w_1 \ldots, w_F \leq 1$ the weights calculated for a given word for each column and an overall weight w for the same word. F denotes the number of weights to be extracted (the number of columns). Creating the observation with and without the weight, we can have different observation sequences :

1. $O^1 = (O_1 \times w_1)(O_2 \times w_2) \ldots (O_T \times w_T)$ where the information at each column is weighted by a measure based on the quality or the quantity of the information in the given column, with the constraint that $w_1 + w_2 + \ldots + w_T = 1$
2. $O^2 = O_1 O_2 \ldots O_T$ where an observation is based just on the mass of the pixel for the given column and we know that the probability calculated through the feature space is w. The weight w is calculated in global terms considering the whole word. A possibility to calculate this term is to average the column wise weights calculated for each pixel column.

As the features and their probability (local or global) are extracted independently from the image X, the question is :

$$P(O^1 \mid \lambda) \stackrel{?}{=} w \times P(O^2 \mid \lambda) \tag{4.10}$$

If there is any kind of difference to add inside the model the structural information or this can be added from outside. The question is : does it worth to design a complex model by inserting the structural information in the model while it can be added from outside by multiplying the a posteri probability performed by the model λ and the global weight w extracted from the shape ?. The equality or not equality is based on the assumption than the a posteriori probabilities are calculated separately.

Proof :

Let's suppose we have extracted the structural weights (column wise) of a given word $0 < w_0 \leq w_1 \leq \ldots \leq w_F$ and suppose the global weight extracted from the same word $w = 1$.

In the Viterbi algorithm, we have the recursive formula :

$$\delta_t(j) = max_{1 \leq i \leq N} \left[\delta_{t-1}(j) \times a_{ij}\right] b_j(O_t) \tag{4.11}$$

Considering the formula described by Equation 4.11, our $\delta_t(j)$ values for the observation sequences O^1 and O^2 respectively, can be considered as follows :

$$\begin{aligned} \delta_t^1(j) &= max_{1 \leq i \leq N} \left[\delta_{t-1}^1(j) \times a_{ij}\right] (b_j(O_t) \times w_t) \\ \delta_t^2(j) &= max_{1 \leq i \leq N} \left[\delta_{t-1}^2(j) \times a_{ij}\right] b_j(O_t) \end{aligned} \tag{4.12}$$

In the first case O^2 is considered as being the classical observation, while O^1 is the extended version, considering the corresponding structural weights.

We can suppose that the state transition probability is similar (see a_{ij}) in these two cases. So the term is depending just on the observation probability which is similar in these two cases. Hence, we have shown that the numerical value is not similar in the two cases.

$$P(O^1 \mid \lambda) \neq w \times P(O^2 \mid \lambda) \tag{4.13}$$

As we have considered as starting condition $w = 1$, the left term of the equation 4.10 can be simplified $P(O^2 \mid \lambda)$, so :

$$P(O^1 \mid \lambda) \leq P(O^2 \mid \lambda) \tag{4.14}$$

Conclusions :

As the observation probability meaning has changed in the first case (O^1) in comparison with the baseline observation sequence (O^2), based just on the joint conditional pixel probabilities along the column, the two systems are totally different. No exact comparison is possible.

As there is no formal proof available to show the superiority of an observation against the other, some test should be performed in order to measure the discriminative power of the model, where the structural information is implanted in the model and for the model where the structural information is added just at the end as simple product rule.

Considering the two approaches aligned, we consider that it is better to insert the weight in the model giving to the NSHP-HMM a supplementary information, used in the re-estimation. While in the second case just a static information is provided by the product. In our further research we have considered just the implant mechanism as such an embedded system can better re-estimate the NSHP-HMM parameters.

4.3.7 The weight normalization

As the extensions proposed by the equations (4.6) and (4.8) have some technical limitations concerning the value representation. If we calculate the weights foar each pixel, the observation considered by the NSHP-HMM will be quite a small value which does not fit anymore in the baseline system. We limit the further discussions to the equations (4.5) and (4.7).

To obey the Markov constraints [Rab89], a normalization process is necessary for the weight calculus. As the structural information is extracted from the height normalized image and the observations are column-wise, the normalization is ensured, if the information quantity is considered for this weight purpose.

To distinguish between a column observation where no structural information is present and a column where are pixels carrying out structural information, the weight, w_j^{inf} in the equation (4.5) is calculated as follows :

$$w_j^{inf} = \frac{1}{nbFeature + 1} \tag{4.15}$$

where *nbFeature* denotes the number of pixels having structural property in the column j. We can observe than this kind of weight calculus assures the $0 < wjinf \leq 1$ condition.
If there is no structural point in the column, the observation is similar as in the former system, otherwise the w_j^{inf} will weight the column j.

Finally, considering the weight described by equation (4.15), the column observation can be transcribed as follows :

$$\overline{P_j} = \frac{1}{nbFeature + 1} \times \left(\prod_{i=1}^{m} p_{ij} \right) \tag{4.16}$$

For the equation (4.7) the weight w_j^{inf} is calculated as follows :

$$w_j^{inf} = \begin{cases} \eta & if\ nbFeatures > \kappa \\ 1 & \text{otherwise} \end{cases} \tag{4.17}$$

where η and κ are some parameters set to suitable values based on trial runs. In that case the extended column based observation is described by the following equation

$$\overline{P_j} = \begin{cases} \left(\prod_{i=1}^{m} p_{ij} \right)^{\eta} & if\ nbFeatures > \kappa \\ \\ \prod_{i=1}^{m} p_{ij} & \text{otherwise} \end{cases} \tag{4.18}$$

Once the observation defined by the equations (4.16) and (4.18) we can use the same train and test mechanism as described in [Sao97, SB97, Cho02].

We have used as extra information, the structural information as we consider that these high-level perceptual features are sufficiently descriptive for handwriting characterization. Moreover, many HWR system use such features to discriminate the different handwritten word shapes [dOJdCdAFS02, GS00, dAFBS01].

As the method is generic, any other kind of information can be used instead of the perceptual information selected by us. Considering other type of information, some minor changes at normalization level may be invoked, but the general framework is invariant.

4.3.8 Model complexity

Concerning the model complexity, the memory complexity of the new model will be similar to the case of the former system $\mathcal{O}[N(N + 2^V Y]$ while the computational complexity will grow in function of the features which will be extracted. For the calculus we have considered a model having N states, analyzing Y pixels in each column using a neighborhood of order V.

4.4 Experiments and results

In this section we describe the different experiments and the results obtained by the NSHP-HMM. We describe the different databases used in the experiments and the different pre-

processing actions preceding the recognition. Finally, we will discuss the results in the framework of the application area of the system.

4.4.1 Databases

SRTP

The tests were performed on two different handwritten word datasets. The Roman one is the SRTP dataset containing handwritten French bank check amounts. The 7031 images are distributed not uniformly in 26 classes. The 26 classes correspond to the different French words describing the different legal amounts : *un, deux, trois, quatre, cinq, six, sept, huit, neuf, dix, onze, douze, treize, quatorze, quinze, seize, ving, trente, quarante, cinquante, soixante, cent, mille, franc, et, centimes*. A more detailed description of the dataset can be found in Section A.3.

BANGLA

The second dataset is a Bangla city name database containing Indian city names written in Bangla script, collected in Kolkata, West Bengal, India. The dataset contains 7500 postal documents and we have used just the different Bangla city names extracted manually. We have identified 76 different city names : *Dhanekhali, Chandannagar, Bagnan, Srirampore, Bankura, Tarokeswar, Bishnupur, Uluberia, Rayganj, Dhaniakhali, Bardhaman, Gangarampur, Raina, Islampur, Kalna, Karandighi, Durgapur, Patrasayar, Seuri, Asansole, Kantoa, Rampurhat, Memari, Chittaranjan, Bolepur, Santiniketan, Nalhati, Murshidabad, Beldanga, Rajnagar, Basirhat, Barasat, Kasba, Jalangi, Jalongi, Sodepore, Panskura, Jangipore, Farakka, Tomlook, Bongao, Malda, Englishpore, Harischandrapore, Kanshipore, Dimonharber, Purulia, Manbazer, Namkhana, Ranghunatpore, Sonarpore, Darjiling, Kalimpong, Alipurduwar, Alipur, Ranaghat, Coachbihar, Chakda, Shantipur, Bali, Mathabhanga, Nabadwip, Kalighat, Kakdwip, Arambag, Jhagram, Kanthi, Barrackpore, Jalpaiguri, Karsiang, Dhupguri, Nakhshalbari, Tuphangange, Kalyani, Churchura, Howrah.*

In order to have a uniform distribution of city names (100 images/class) some extra images were necessary. For both datasets (SRTP, BANGLA) the image acquisition was off-line at 300 dpi. A more detailed description on the Bangla dataset can be found in Section A.2.
In all our experiments we have used 2/3 of the images to train the systems and the 1/3 remaining images were used to test the system.

4.4.2 Image preprocessing

The NSHP-HMM has the advantages to require just a few preprocessing steps like :

– Skew correction

– Slant correction

– Differential image normalization

As stated by Vinciarelli [Vin00], the ideal model in handwriting the word supposed to be written horizontally with ascenders and descenders aligned along the vertical direction. Unfortunately, in real data such conditions are rarely respected. Slope (the angle between the horizontal direction and the direction of the implicit line on which the word is aigned) and slant (the angle between the vertical direction and the direction of strokes supposed to be vertical) are often different from 0 and must be eliminated.

The slope correction is a classical pre-processing phase in handwriting recognition and it is imperative for our system as considering the formal description of the NSHP-HMM we can deduce that such an approach is not invariant to affine transformations. We suppose that process was already considered as we are working on isolated on separated words, extracted manually.

The slant correction is also a classical pre-processing and it is also necessary to consider it as it can have deep impact on the NSHP-HMM during the analysis. As the model observes pixel columns, the slant correction becomes very important issue. Saon performs a global slant correction giving promising results. To refine such an approach different works can be found in the literature [BS89, UTS01] but the major part of the approaches are based on local estimation which we would like to avoid.

In [dSBJSL$^+$00] the authors have mentioned that for connected digit strings recognition the slant estimated from the whole word in unsatisfactory, since they have individual components (digits and segments) each of which has its own slant. They have also concluded that an independent (local) correction of each compoenent is also not viable, since this may produce distortions when broken digits are present in the numeral string.

The global slope is calculated by counting the horizontal transitions around the current pixel, considering the whole word entity. The image analysis is performed in a global manner with a certain precision. Such approach seems to be a robust one and it can be considered also a reliable one for our purpose.

For complexity reasons, we have considered to use fixed size NSHP-HMM, so the image analyzed by the model should have the same size for each word model. For that reason, Saon is considering just a simple height normalization in [Sao97] to fit the image into the required size, and a proportional width normalization.

Considering the different experimental results leads by Saon to establish appropriate model height, we have concluded that the best trade-off between the model complexity and the amount of information to be considered is to use models analyzing 20 lines.

To conserve the redundant information in the different words belonging to the same class, here a differential height normalization was considered based on the middle zone of the writing.

The busy-zone of the writing was considered using the product of the horizontal projection histogram and the horizontal transition histogram. After refining the product histogram, the busy zone is determined by some threshold values set-up empirically. This threshold mechanism has strong dependencies on the scripts which is applied for.

While this method is just an adaptation of the busy-zone finding used by Choisy for Roman script, we developed a special technique to find the middle part of the Bangla writing. The basic idea of the algorithm is based on the water reservoir concept and the relative position of these reservoirs in the word shape. A more detailed description of the algorithm can be found in [RVP+05b].

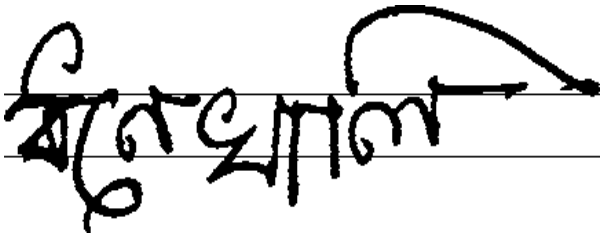

FIGURE 4.3 – Busy-zone finding for the Bangla word Dhanekhali using projection profiles

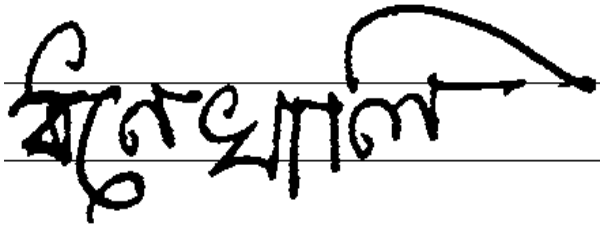

FIGURE 4.4 – Busy-zone finding fr the Bangla word Dhanekhali using water reservoir based features

Taking into account the performances of these two busy-zone finding algorithm, we can state that there is no much differences between the two separate methods. While the first one, using just projection profiles, the second algorithm has a much more increased running complexity considering the search of reservoirs in the same. So this is the reason why the projection profile based system was used for further investigations.

In the Fig 4.5 and Fig. 4.6 we can see the differential height normalization applied to a French word four "quatre" and the Bangla word Dhaniekhali where ascenders and descenders can be detected.

To discard the images where the middle band is not considerable in comparison with the image height, a threshold value was also introduced.

While the SRTP database is a clean one, in the BANGLA dataset some images (1.5% of the total image set) have been discarded based on this threshold described above. The reason in that cases is the excessive size of ascenders and descenders occuring often in the database.

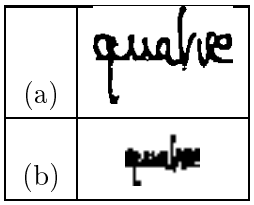

FIGURE 4.5 – (a) Original image and (b) Normalized image of the word "four" in French

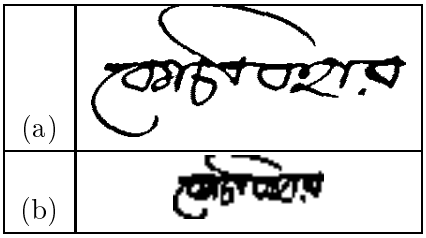

FIGURE 4.6 – (a) Original image and (b) Normalized image of the Bangla word Dhaniekhali

As stated also by Choisy [Cho02], some difficulties can be encountered to find the busy zone of short words, where there is no sufficient information.

4.4.3 Perceptual feature extraction

To test the implant of structural information in the system, some feature extraction was necessary. As our goal was to propose and design a generic method to implant structural information in the NSHP-HMM system, we limited our feature extraction to ascenders and descenders considering them the most powerful features considering their discriminative power in different Roman scripts.

The normalization of the images is based on middle zone of the writing as described above. We used this information to extract the ascenders and descenders. As the upper zone, the middle zone and the lower zone were mapped equally in the normalized image, the upper line and the baseline of the writing can be found in the normalized image.

We considered all the pixels in the upper zone (the horizontal strip above the middle (busy) zone)as being a part of an ascender and in the same manner, all the pixels in the upper zone (the horizontal strip below the middle zone) are considered as being part of a descender. In order to really detect the ascenders and descender, the length and the height of the connected component have been considered for this purpose. To avoid the occurred noises and the error coming from

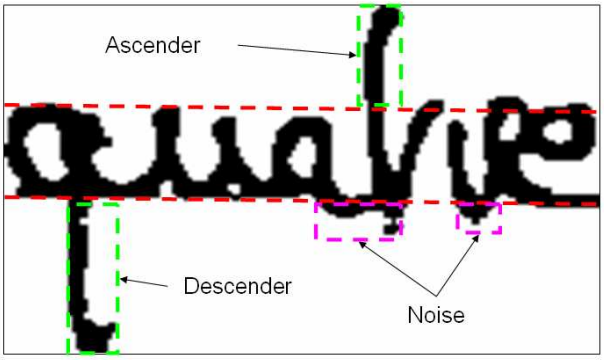

FIGURE 4.7 – Ascender and descender extraction based on the middle zone of writing

the precision of the busy-zone finding algorithm, some threshold values were introduced to allow to discard the non significant information. The threshold value is calculated based on the height and width of the ascender/descender candidate.

4.4.4 The structural NSHP-HMM parameters

In the preceding chapters, sections we presented the formal description of the model, while in this section our aim is to give some hints on the parameters of the system.

In general, the researchers introduce some a priori knowledge to guide, to force the model to react in a way or another. In [BR99] the authors consecrate an entire chapter to this issue in order to cover the different aspects of the initialization.

The Baum-Welch training procedure is an MLE procedure that converges to a local minimum and therefore is sensitive to the model initialization. The initialization of the transition matrix A is less critical, also because the transition probabilities have a limited impact on the recognition performances. In essence, transition probabilities relative to the same state or to contiguous states can be initialized to uniform values, while transition probabilities to non-contiguous states are generally set to zero.

Taking into account the general consideration for the model initialization, the parameters of the system are :

- the height of the column which should be analyzed
- the order of the NSHP
- the number of states to be considered for each model
- the topology of the model described by the allowed transition probabilities among the states

Considering the "height" of the model as described in the section concerning the normalization (Section 4.4.2) is fixed to 20 lines. Other experiments giving less results lead us to use this value. This conclusion confirms the affirmations of Saon and Choisy considering this height but does

not exclude the possibility to explore some new values taking into account the complexity aspect considered as heavy for a more considerable height.

We suggest to analyze separately each word model, and maybe in the future some model based height can be set-up considering the graphical shape of the model. In case of words where there is no presence of perceptual features, a possibility is to allow to analyze much more pixels increasing the model "height".

The order of the NSHP-HMM is considered in function of the neighborhood Θ_{ij} used by the NSHP for the estimation of the joint pixel probability for a given pixel (i, j). In our work we have used third order neighborhood which reduces considerably the complexity of the model depending exponentially on this parameter. Even if Saon has reached higher performances for a 4th order model [Sao97], we have considered this is a convenient trade-off between the complexity and accuracy. The neighborhood covers the top-left corner of the pixel.

Such a complexity reduction can be evaluated as one of the principal gain of the analytical model proposed by Choisy. The reduction can be explained by the analytical aspect where a letter does not need as much information as a word does.

The number of states in the models is a hard issue nowadays in the scientific community. There is no rule, there is no scientific evidence for the best number of states.

In the last few years different works have been lead to find such a solution [BMF03, LKK02, KCGM93, dSBJSBS01, KCMT01] for this challenge but the most part of the works use some constraints or a priori knowledge base on the nature of the data which does not allow to use it in a generic context.

Our state number estimation is also based such kind of heuristics. We consider than 2 pixel columns can be *"read"* by the same state. For that reason after the differential height normalization described in Section 4.4.2 an estimated average width value for each letter component is calculated and the number of states in the letter model is fixed as half of the letter width. The estimation developed is based on different trial runs, considering different formula. This estimation procedure is based on the work described in details in [Cho97].

In other works, the authors used letter models composed by 9,10,14 states without giving any particular explanation of these "magic" numbers [GB04, TLK+01, Koe02].

In HMM parameter estimation work proposed by Günter and Bunke [GB03b] the authors examined two approaches :

- *Constant number :* The number of states of each HMM is set to a constant value s. The best s value is determined using the validation set. For their system they have found 14 as being the optimal solution. In the HTK (Hidden Markov Model Toolkit) originally developed for speech [YJO+95] but used with success in other fields like character recognition, DNA sequencing, the number of optimal states is 16 as two non-emitting states are included.
- *Flexible number :* In that case the number of states of each HMM is set to the average

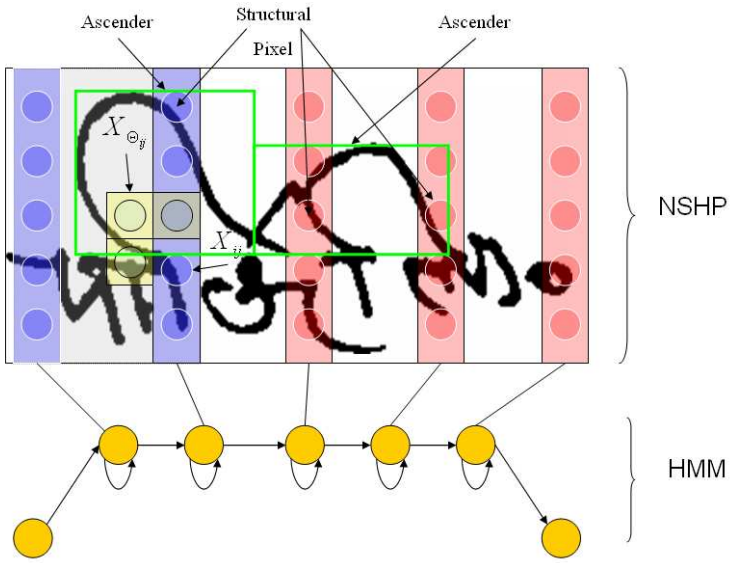

FIGURE 4.8 – The structural NSHP-HMM analyzing the word Darjiling considering the structural information extracted from the word shape

length of the corresponding sequence of features vectors times a constant f. They have also considered the average letter length for this purpose. The average length is estimated by running the HMM recognizer is forced alignment for the training set.

We can argue such a high state number with the fact that in that case sliding window techniques have been used for feature extraction with a little displacement, so many observations have been created so more states are necessary to "read" them.

Because of the left to right direction of writing, a linear structure has to be adopted for the NSHP-HMM. From each state just the state itself and the following state can be reached. Self loops are permitted. Such an approach can be considered as a constraint but considering the analysis of the NSHP by column another sort of transition can disturb the probability distributions. Disturbing the probability distributions this can raise in the same time problems in the information distributions in the different states limiting the discriminative power of the system. A graphical representation of the structural NSHP-HMM is shown is Fig. 4.8.

4.4.5 Results concerning the classical NSHP-HMM

As discussed before, we have applied the classical NSHP-HMM and the structural NSHP-HMM for different dataset written in French and Bengali. In Table 4.1 the Top1 results of the NSHP-HMM are given without considering any rejection criteria.

We can observe that the system was not sensitive to the vocabulary opening. The model gives more or less the same accuracy for the SRTP(26 classes) and the Bangla word dataset (76 classes) which is a considerable results for such a holistic system.

Method	SRTP	Bangla
NSHP-HMM	85.92%	86.40%

TABLE 4.1 – The classical NSHP-HMM results concerning the SRTP and the Bangla city name dataset

As highlighted in the literature such an extension can already reduce the recognition accuracy of the system but in our case the implicit segmentation considered in the analytical approach allowed not to consider the word as a whole entity, but rather a letter chain allowing a better modeling. This letter segmentation conserves the discriminative power of the model.

While in other HWR systems there is a difference between uppercase and lowercase letters, here just the lowercase aspect is considered. For the SRTP dataset this choice is motivated by the low occurrence of uppercase letters in the dataset, while for the BANGLA dataset this notion of uppercase-lowercase does not exist, so just one letter is considered. The difficulties occur in the Bangla alphabet where instead of the well-know 26 letters used in Roman 350 characters should be considered. In our dataset the number of letters is 103 which extends considerably the letter models and the different context where these letters can be found. Considering the data amount available for training purpose we can conclude that our analytical approach based on implicit segmentation is a robust and reliable recognition system.

In the next tables we give more details on the results obtained for the different databases (SRTP : Table 4.2 and BANGLA Table 4.3).

Class	**un**	**deux**	**trois**	**quatre**	**cinq**	**six**	**sept**
Result	10.00%	83.45%	79.01%	89.83%	93.18%	55.88%	72.22
Class	**huit**	**neuf**	**dix**	**onze**	**douze**	**treize**	**quatorze**
Result	65.00%	75.00%	90.24%	100.00%	92.31	57.14%	62.50%
Class	**quinze**	**seize**	**vingt**	**trente**	**quarante**	**cinquante**	**soixante**
Result	83.36%	66.67%	95.86%	85.00%	76.74	90.54%	91.14%
Class	**cent**	**mille**	**centimes**	**francs**	**et**		
Result	87.19%	67.09%	67.74%	91.14%	71.43%		

TABLE 4.2 – The results of the classical NSHP-HMM for the SRTP dataset

4.4.6 Results using the structural NSHP-HMM

In that section we will discuss the results given by the structural NSHP-HMM, considering the different extensions of the pixel column observation described by the different equations (Eq.4.16, Eq.4.18) in the Section 4.3.7. A general overview of the results is shown in the Table 4.4.
Improvement1 is the method where the equation (4.16) is used, while *Improvement2* is the me-

Class	Acc.(%)	Class	Acc.(%)	Class	Acc. (%)
dhanekhali	82.35	**chandannagar**	47.06	**bagnan**	91.18
srirampore	91.18	**bankura**	84.85	**tarokeswar**	91.18
bishnupur	93.94	**uluberia**	100.00	**rayganj**	97.06
dhaniakhali	94.12	**bardhaman**	87.88	**gangarampur**	97.06
raina	88.24	**islampur**	91.18	**kalna**	67.65
karandighi	97.06	**durgapur**	94.12	**patrasayar**	94.12
seuri	87.10	**asansole**	100.00	**kantoa**	91.18
rampurhat	100.00	**memari**	93.75	**chittaranjan**	96.97
bolepur	88.24	**santiniketan**	97.06	**nalhati**	87.88
murshidabad	96.88	**beldanga**	82.35	**rajnagar**	82.35
basirhat	94.12	**barasat**	97.06	**kasba**	88.24
jalangi	73.53	**jalongi**	85.29	**sodepore**	94.12
panskura	76.47	**jangipore**	73.53	**farakka**	73.53
tomlook	79.41	**bongao**	79.41	**malda**	82.35
englishpore	94.12	**harischandrapore**	91.18	**kanshipore**	91.18
dimonharber	88.24	**purulia**	85.29	**manbazer**	90.62
namkhana	85.29	**raghunathpore**	88.24	**sonarpore**	85.29
darjiling	85.29	**kalimpong**	67.65	**alipurduwar**	85.29
alipur	94.12	**ranaghat**	85.29	**coachbihar**	91.18
chakda	85.29	**shantipur**	94.12	**bali**	93.94
mathabhanga	67.65	**nabadwip**	94.12	**kalighat**	85.29
kakdwip	73.53	**arambag**	97.06	**jhargram**	70.59
kanthi	85.29	**barrackpore**	91.18	**jalpaiguri**	91.18
karsiang	85.29	**dhupguri**	87.88	**nakshalbari**	91.18
tuphangange	67.65	**kalyani**	84.85	**chuchura**	70.00
howrah	61.76				

TABLE 4.3 – The results of the classical NSHP-HMM on the Bangla city name dataset

Method	SRTP	Bangla
Classical NSHP-HMM	85.92%	86.40%
Improvement1	**87.52%**	**86.80%**
Improvement2	86.39%	86.52%

TABLE 4.4 – Improvement results concerning the SRTP and the Bangla city name dataset

thod where the equation (4.18) is used. We can observe that the results given by the observations described by the equations (4.16) are better than the results given by the equation (4.18).

Considering the Eq. (4.15) and Eq. (4.17) the results given by these equations are much less significant, so we are not mentioning them here. The poor results can be explained by the fact that in these cases the observation probabilities calculated become small values introducing precision lost in the corresponding calculus.

In the next tables we give more details on the results obtained for the different databases using the proposed observation described by the equation (4.16). The first value is the ancient value obtained by the classical NSHP-HMM, while the second value is the results achieved by the structural NSHP-HMM described above.

Class	**un**	**deux**	**trois**	**quatre**	**cinq**	**six**
Result	10.00/10.58	83.45/85.19	79.01/80.21	89.83/90.13	93.18/93.58	55.88/55.88
Class	**sept**	**huit**	**neuf**	**dix**	**onze**	**douze**
Result	72.22/72.57	65.00/67.31	75.00/75.87	90.24/90.87	100.00/100.00	92.31/94.01
Class	**treize**	**quatorze**	**quinze**	**seize**	**vingt**	**trente**
Result	57.14/56.02	62.50/64.39	83.36/83.57	66.67/66.69	95.86/96.72	85.00/86.35
Class	**quarante**	**cinquante**	**soixante**	**cent**	**mille**	**centimes**
Result	76.74/78.49	90.54/92.13	91.14/91.77	87.19/89.41	67.09/69.63	67.74/68.91
Class	**francs**	**et**				
Result	91.14/91.85	71.43/73.21				

TABLE 4.5 – The detailed results of the structural NSHP-HMM for the SRTP dataset

4.4.7 Discussions

We can observe that in case when we are multiplying the column probability with the weight calculated from the structural nature of the pixels along the column it is better than raising to a given power the conditional column probability. While in first case this weighting can be integrated in the NSHP-HMM framework as it just accentuate the importance of the column, in the second case we have changed the nature of the observation.

The achieved improvement considering the structural NSHP-HMM is much more considerable in case of the SRTP dataset (1.57%), while in case of the Bangla dataset the gain is just 0.4%.

The difference is due to the nature of the scripts and the used structural features. While in the case of the SRTP bank check dataset, the words are Roman words, so the notion of ascender/descender is clearly distinguishable ; the same notion has not the same signification in the case of Bangla. In order to reach higher results for Bangla, some other kind of structural features should be extracted as water reservoir features [PBC03] which can better describes the

Class	Ac. (%)	Class	Ac. (%)	Class	Ac. (%)
dhanekhali	82.35/74.03	**chandannagar**	47.06/42.68	**bagnan**	91.18/89.74
srirampore	91.18/92.93	**bankura**	84/85/89.28	**tarokeswar**	91.18/92.68
bishnupur	93.94/93.94	**uluberia**	100.00/97.67	**rayganj**	97.06/98.76
dhaniakhali	94.12/95.82	**bardhaman**	87.88/95.66	**gangarampur**	97.06/95.82
raina	88.24/89.94	**islampur**	91.18/92.67	**kalna**	67.65/63.45
karandighi	97.06/95.96	**durgapur**	94.12/94.28	**patrasayar**	94.12/97.23
seuri	87.10/89.10	**asansole**	100.00/98.18	**kantoa**	91.18/89.98
rampurhat	100.00/100.00	**memari**	100.00/100.00	**chittaranjan**	96.97/96.97
bolepur	88.24/92.88	**santiniketan**	97.06/95.62	**nalhati**	87.88/86.55
murshidabad	96.88/96.88	**beldanga**	82.35/84.19	**rajnagar**	82.35/86.99
basirhat	94.12/92.68	**barasat**	97.06/95.88	**kasba**	88.24/92.88
jalangi	73.53/63.46	**jalongi**	85.29/75.23	**sodepore**	94.12/89.94
panskura	76.47/75.23	**jangipore**	73.53/60.52	**farakka**	73.53/81.11
tomlook	79.41/75.23	**bongao**	79.41/78.17	**malda**	82.35/84.05
englishpore	92.12/92.88	**harischandrapore**	91.18/92.88	**kanshipore**	91.18/84.05
dimonharber	88.24/89.13	**purulia**	85.29/95.82	**manbazer**	90.62/95.45
namkhana	85.29/95.82	**raghunathpore**	88.24/90.03	**sonarpore**	85.29/85.29
darjiling	85.29/86.79	**kalimpong**	67.65/72.19	**alipurduwar**	85.29/89.94
alipur	94.12/89.94	**ranaghat**	85.29/89.94	**coachbihar**	91.18/91.18
chakda	85.29/85.29	**shantipur**	94.12/85.29	**bali**	100.00/100.00
mathabhanga	67.65/72.19	**nabadwip**	94.12/92.88	**kalighat**	85.29/88.99
kakdwip	73.53/68.32	**arambag**	97/06/95.82	**jhargram**	70.59/78.17
kanthi	85.29/75.13	**barrackpore**	91.18/95.82	**jalpaiguri**	91.18/86.99
karsiang	85.29/84.05	**dhupguri**	87.88/68.37	**nakshalbari**	91.18/98.08
tuphangange	67.65/57.58	**kalyani**	84.85/87.80	**chuchura**	70.00/82.21
howrah	61.76/69.25				

TABLE 4.6 – The detailed results of the structural NSHP-HMM for the Bangla city name dataset. The first value is the result obtained by the classical NSHP-HMM while the second value is the recognition accuracy achieved by the structural NSHP-HMM

Bangla script.

Analyzing the feature extraction mechanism proposed by us in Section 4.4.3 we can assume that a more precise technique would be necessary. Especially the threshold mechanism should be improved. Otherwise, some noise can be considered as perceptual features introducing uninvited data in the system, which can negatively influence the system.

Another amelioration aspect could be to extract much more structural features like convex and concave sectors, cross points, cutting points, etc. which can better describe the different characteristics of the scripts.

For example, the NSHP-HMM uses a word meta-model composed of letter models. Hence, it is obvious than if we add information concerning the letter limits, the system system should improve its modeling capacities. Using the knowledge of explicit letter limits, we can drive the Baum-Welch training to estimate the state transitions in function of these limits. Such kind of extra structural information can drive the re-estimation calculus by forcing the state transitions to adapt themselves to the real letter limits in a word.

Extracting a huge variety of features, the normalization process can be also refined as different weights can be assigned to the different features in function of their discriminating power. In that case instead of using the same weight for each structural feature, the weight can be assigned based on the nature of the weight considering the discriminating power of each of the features in a general learning context.

4.4.8 Comparison study with the state of the art

Often the results comparison is quite a difficult task as the performed experiments are not performed in the same conditions, the size and the quality of the analyzed dataset varies, etc.

However, to get an idea about the results obtained by us using the structural NSHP-HMM, some results based on handwritten bank check amounts is given in the Table 4.7, mainly concerning the SRTP dataset. For the comparison purpose, the database, the used approach, the image resolution and the recognition accuracy in Top1 is considered.

In the same manner, we present a comparison for the reduced size vocabularies considering other data than legal bank check amounts. As our results concerning the Bangla script is unique, hence it is not possible to have a direct cross-checking. In the Table 4.8 we have considered the latest results, using different stochastic approaches. For that purpose the source of the work, the database nature, the vocabulary size and the recognition accuracy will be considered in the comparison. No rejection criteria was used to report this results.

Considering the results on the SRTP dataset in Table 4.7, we can conclude than our analytical approach can be considered as one of the top solutions in the matter as it outperforms almost every results given by the different techniques, based on analytical or global approaches.

Ref.	**Dataset**	**Method**	**dpi**	**Vocabulary**	**Rec. rate**
[SBG93]	SRTP	analytical	300	26	76.9%
[SBG93]	SRTP	global	300	26	71.8%
[Gui95]	LA (ENG)	global	300	32	72.60%
[Gui95]	LA (FR)	global	300	25	83.10%
[GS95]	LA(FR)	analytical	300	25	76.90%
[GS95]	LA(FR)	global	300	25	78.30%
[SKX+00]	LA(ENG)	analytical	-	32	82.00
[LLGL97]	SRTP	analytical	300	26	74.00%
[LLGL97]	SRTP	analytical	300	26	77.00%
[ABPK98]	LA(FR)	analytical	300	28	92.90%
[PAO99]	SRTP	global	300	27	58.70%
[PAO99]	SRTP	analytical	300	27	57.88%
[KKS00]	LA (ENG)	analytical	-	32	82.00%
[TLK+01]	SRTP	analytical	300	26	80.02
[TLK+01]	SRTP	analytical	300	26	94.60
[GLL95]	LA(ENG)	analytical	300	30	73.10%
[GLL95]	LA(ENG)	analytical	300	30	83.70%
[FYBS00]	LA(POR)	global	300	39	67.70%
[dAFBS01]	LA(POR)	global	300	39	77.00%
[GAA+01]	LA(ENG)	analytical	-	38	81.00%
[TAS+04]	LA(CHI)	analytical	-	20	60.00%
[KFS04]	LA(POR)	global	-	39	81.70%
[SBG95]	SRTP	analytical	300	26	82.80%
[Sao97]	SRTP	global	300	26	90.01%
[Sao99]	SRTP	global	300	26	82.5%
[Cho02]	SRTP	analytical	300	26	86.20%
[NGM05]	SRTP	global	300	27	80.75%
Personal (Thesis)	SRTP	analytical classic	300	26	**85.92**%
Personal (Thesis)	SRTP	analytical structural	300	26	**87.52**%

TABLE 4.7 – Comparison of different results considering bank check amount recognition

The results reported by Saon in [Sao97] are much sound than our results but we can not compare really the two systems. Saon has considered a holistic approach, while ours is analytical one. Considering the size of the database, such a global approach gives excellent results but a possible extension of the dictionary will reduce substantially the accuracy of such a system as the discrimination is global without considering the local aspects of the word shapes. In comparison, our results are much more stable as we can see, the behavior of the system. Considering the SRTP dataset containing just 26 words, the 87.52% is a good results.When an extended vocabulary has been used, containing 76 word entries, the result of 86.80% is also remarkable achievement taking into account the complexity of the script and the increased number of letters. The robustness can be invoked as one of the major quality of our system besides the analitic approach allowing to model the letters in the words.

The performances achieved by the system of Tay et al. [TLK+01] can be explained with the fact that the HWR system proposed by the authors is a hybrid one, combining a NN with a HMM. The NN is in charge to compute the observation probabilities for each letter hypothesis, while the HMM computes the likelihood for each word model. Such a combination seems to be more efficient than the NSHP and the HMM pair, where instead of a training, a more statical approach has been used to estimate the observation probabilities. In contrary, while they should assume a good segmentation into letters, in our case such hard presumption is not necessary as we work on pixel level observations. Another drawback of this system is the estimation of the observation probability given by the different segments. Increasing the number of segments can also drive the system to failure.

Comparing the results of Choisy discussed in detail in [Cho02], our results outperforms the results of the baseline system, which demonstrates the power of the structural NSHP-HMM model. The improvements achieved by the implant mechanism can be considered encouraging and it can be a future research field to explore the different features and their combination with the low-level information coming as the output of the NSHP.

Concerning the Bangla city name recognition, the achievement is duplex. Firstly, in our best knowledge, this work is unique as no work has been done yet to recognize handwritten Bangla words. Such a segmentation-free, analytical approach was necessary to do this task as the Bangla being a much more complex script in shape terms, a segmentation process is not available now and even in the future a good segmentation algorithm it will be a thoughtful challenge for the future research.

Secondly, the recognition accuracy achieved for the handwritten Bangla city names being a much more difficult task than the similar one for Roman script can be compared even with the state of the art works considering the same size vocabulary, containing more less letters and ligature variations as it can be observed for Bangla.

A comparative table considering the different results reported by different authors using the

Ref.	Script	Method	Vocabulary	Accuracy
[KPH04]	French	analytical.	100	75.00%
[KPH04]	French	analytical + context	100	83.00%
[KPH04]	French	analytical + knowledge	100	84.40%
[KPH04]	French	analytical + knowledge + context	100	90.00%
[TNEBA04]	Arabic	analytical	25	73.00%
[KG95]	English	analytical	100	87.40%
Personal (Thesis)	Bangla	analytical classic	76	**86.40**%
Personal (Thesis)	Bangla	analytical structural	76	**86.80**%

TABLE 4.8 – Comparison of different recent results concerning reduced size vocabularies using stochastic approaches

NSHP-HMM as basic model can be found in Table 4.9.

Ref.	Dataset	Method	Vocabulary	Accuracy
[SBG95]	SRTP	analytical	26	82.80%
[Sao97]	SRTP	global	26	90.01%
[Sao99]	SRTP	global	26	82.5%
[Cho02]	SRTP	analytical	26	86.20%
Personal (Thesis)	SRTP	analytical classic	26	**85.92**%
Personal (Thesis)	SRTP	analytical structural	26	**87.52**%
Personal (Thesis)	BANGLA	analytical classic	76	**86.40**%
Personal (Thesis)	BANGLA	analytical structural	76	**86.80**%

TABLE 4.9 – Different results obtained by different NSHP-HMM system

4.5 General conclusions

We can note than the adaptation of the baseline NSHP-HMM conceived initially for Roman scripts (predominantly French) was a success even for a totally different scripts as Bangla is, considering the huge number of letters and the inter and intra-letter variation which occurs in the script. The different modification in the data representation, optimization, etc. have shown their efficiency in the model.

However, the main work presented here describes in details a generic formal mechanism to introduce extra information in the NSHP-HMM, without considering the nature of the information. Such an implant mechanism allows to gather different kind of information possibly extractable from the word shape and integrate them in the stochastic NSHP-HMM framework.

Some important ideas should be highlighted to show the strength of the system :

- The bi-dimensional model (NSHP-HMM) used for handwriting modeling and recognition allows us to take into account the 2D model of handwriting, without imposing any constraints concerning the complexity of the model. The MRF and the HMM is a good combination to analyze locally the word shape and to exploit the regularities encountered in different words, considering just pixel information.
- The analytical extension of the system allows to better distribute the information in the model states, especially the letter context is considered for this purpose.
- The extension of the baseline system was necessary as the NSHP pixel re-estimation impose limits as the low-level information cannot carry any semantical meaning for the considered information. While the raw image guarantees sufficient information for entities like separated digits, in a more wide context like word recognition, a more precise feature description is necessary.
- The implant mechanism proposed by us is a new concept and opens a way to insert high-level information in the baseline system without any hard modification of the former model causing no kind of extra complexity issues.
- Taking into account the performance of the structural NSHP-HMM we can conclude that is much more performant than the former systems proposed by Choisy, so the impact of the perceptual feature implant is real.
- Considering the results obtained on the Bangla dataset, it shows that the system is able to adapt itself to different scripts such as Bengali, so it is script independent thanks mainly to the analytical extension of the system.

We achieved the goals proposed in the first part of the chapter by going further as the current limits of the system. We have introduced a generic mechanism for extra information implant in the system taking into account the formal framework defined by Saon and Choisy and the perceptual information implant (ascenders/descenders) has shown their importance in the accuracy analysis, outperforming the result of different former systems referenced by the literature.

In the mean time we should also consider the inconveniences derived directly from the nature of the system. Applying random field in the observation probability estimation leads to an exponential memory usage and the memory is calculated in function of the neighborhood order.

To ameliorate the time complexity, in the next chapter we propose an original technique to reduce the time factor using flat lexicon representation.

5

Time complexity reduction in the Viterbi decoding

5.1 Objectives

After a substantial analysis of the NSHP-HMM we can conclude the same as the HMM community does : the tool is highly appropriate for modeling purpose for different kind of signals like speech, handwriting, DNS sequences, etc. but the recognition mechanism embodied usually by the Viterbi algorithm is sub-optimal. The algorithm has an increased time cost considering the underlaying dynamic programming allowing to find the optimal alignment between each lexicon entries and the analyzed word shape. Basically, the current works in the field use flat representations for the lexicon entries, while lately for large vocabularies a prefix-tree representation has been proposed with success. The later one allows to exploit the common prefix parts of the words (see Fig. 2.14) allowing to diminish the memory requirement for the lexicon representation as shown in Table 2.1 and Table 2.2.

Nowadays, in natural language processing, a more complicated organization procedure based on the *Directed Acyclic Word Graph* (DAWG) has been designed, which can be considered as an extension of the prefix tree. Here not just the common prefix parts are share but the other common parts too, such as : terminations or suffixes [GFK02]. However, it is not clear how such a method could work in handwriting as there are no experimental result available yet.

In our case, we should also consider the non-symmetric aspect due to the NSHP as instead of using one HMM for each word model in the model discriminant strategy, for each model 4 sub-models are generated as shown in Fig. 5.1, multiplying by 4 the training and test processes.

Our idea is to apply in the flat representation mechanism some pruning strategies in order to reduce the Viterbi decoding without loss of accuracy. Instead of representing the lexicon in a more compact manner, like prefix-tree [Koe02], where the common prefix parts are considered

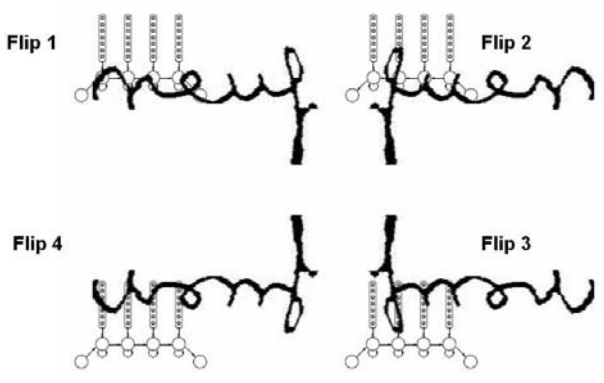

FIGURE 5.1 – The considered symmetric aspects in the NSHP-HMM

or DAWG [Bal02], we consider the flat representation and we exploit the analytical aspect of the model. We will establish some rules, more exactly a pruning strategy to be able to perform partial Viterbi matching. We plan to stop the high complexity mechanism (search)in cases where we are certainly sure that there is no meaning to continue the decoding process.

Why not a prefix-tree representation ? Such a data structure can be possible in cases where an implicit segmentation precedes the recognition. We should be certain that we have passed the letter limits otherwise such a search space representation cannot work efficiently. Koerich introduced a measure to characterize the reduction in a lexical tree [Koe02]. Such a representation can be viewed as a reduction of the average word length that is given as :

$$L' = \frac{L}{rf} \tag{5.1}$$

where L' is the new average word length and rf is the reduction factor that is given as :

$$rf = \frac{Ncl}{Nct} \tag{5.2}$$

where Ncl is the total number of characters in the flat lexicon. Nct is the total number of characters in the tree structured lexicon. Considering the equation (5.1), it is clear that the more words in the lexicon share common prefixes, the more advantageous it will be to use a lexical tree representation.

In that case, techniques like level building algorithms used by Koerich [KSSEY00, Koe02] can be successful. Such a representation supposes a considerable data redundancy at prefix level which is not our case taking into account the SRTP and the BANGLA datasets.

5.2 General description of the reduction process

In the system architecture this reduction process can be considered as a plugin module, allowing to reduce considerably the recognition time and preserving the accuracy of the system.

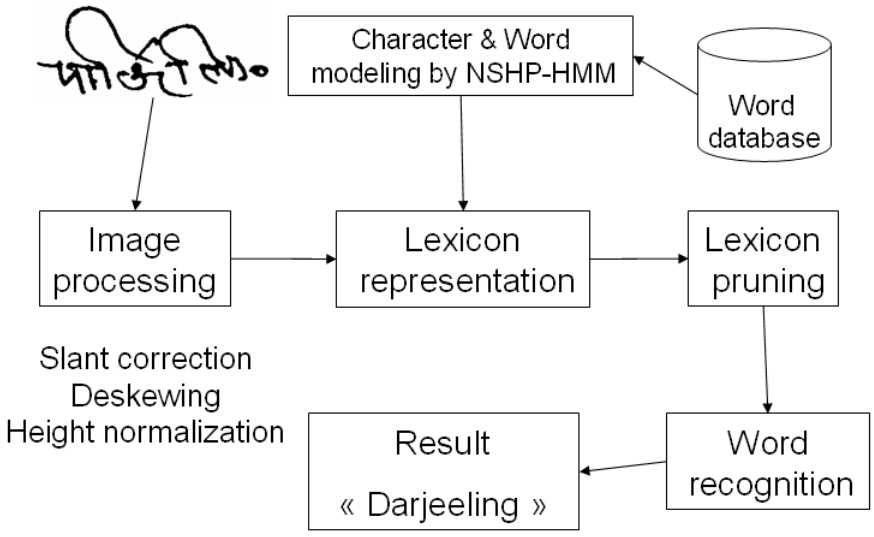

FIGURE 5.2 – General system overview for lexicon reduction

A general system overview concerning the lexicon reduction is given in Fig. 5.2.

The reduction process proposed by us can be separated in two parts. In the first part an interference in the Viterbi algorithm is proposed to prune the search, while secondly a natural length estimation is proposed to help to reduce the decoding performed in a flat lexicon representation.

For the pruning mechanism we start from the idea that if the first part of the analyzed word does not match with a model, we can decide to continue for the further Viterbi analysis or just stop and look for another. This can be interpreted as a partial matching instead of a complete one. Such a presumption is possible as if in the first part of the analyzed word the likelihood is weak the possibility to reach a good matching score at the end becomes unlikely.

For the natural length estimation the average number of white-black and black-white transitions found in the middle-zone of the writing have been considered. For this purpose the middle zone finding algorithm previously described has been applied.

5.3 Formal description of the reduction

To introduce the pruning strategy in the decoding process, we will give some details concerning the Viterbi algorithm is the NSHP-HMM framework.

The Viterbi algorithm, named after its developer Andrew Viterbi, is a dynamic programming algorithm for finding the most likely sequence of hidden states known as the Viterbi path, that results in a sequence of observed events, especially in the context of the hidden Markov models.

For the recognition purpose, besides the Viterbi algorithm, we can use also the Baum-Welch algorithm being more optimal but in that case we cannot track information about the obtained

information limits being essential for our purpose. We want to exploit the capability of the system to find more or less with a precision the limits imposed by the meta-models to perform a pruning strategy based on these letter limits.

5.3.1 The Viterbi algorithm

Here we are considering the algorithm for the maximum likelihood calculus and the corresponding state sequence producing this likelihood. The algorithm, so called Viterbi algorithm [Rab89] allows to find the best state sequence $Q = \{q_1, q_2, \cdots, q_T\}$, for a given observation sequence $O = \{O_1 O_2 \cdots O_T\}$ and the corresponding model λ. For that reason, Rabiner has introduced a measure :

$$\delta_t(j) = max_{q_1,q_2,\cdots,q_{t-1}} P\left[q_1, q_2, \cdots, q_t = j,\ O_1 O_2 \cdots O_t \mid \lambda\right] \tag{5.3}$$

where this $\delta_t(j)$ is considered the best score along a single path, at time t, which accounts for the first t observations and ends in state s_i. By mathematical induction we have :

$$\delta_{t+1}(j) = [max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}) \tag{5.4}$$

In order to retrieve the state sequence producing the highest probability, we need to keep track of the arguments (states) which maximized the equation 5.4, for each t and j. For that purpose a two dimensional variable $\psi_t(j)$ is considered.
The complete *Viterbi algorithm* is as follows :

1. *Initialization :*

$$\delta_t(j) = \pi_i b_j(O_t)\ ; 1 \leq j \leq N \tag{5.5}$$

$$\psi_1(j) = 0\ ; 1 \leq j \leq N \tag{5.6}$$

2. *Recursion :*

$$\delta_t(j) = [max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij}] \cdot b_j(O_t)\ where\ 2 \leq t \leq T\ ; 1 \leq j \leq N \tag{5.7}$$

$$\psi_t(j) = argmax_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right]\ where\ 2 \leq t \leq T\ ; 1 \leq j \leq N \tag{5.8}$$

3. *Termination :*

$$P^* = max_{1 \leq j \leq N} \left[\delta_T(j)\right] \tag{5.9}$$

$$q_T^* = argmax_{1 \leq j \leq N} \left[\delta_T(j)\right] \tag{5.10}$$

4. *Path sequence (backtracking) :*

$$q_t^* = \psi_{t+1}(q_{t+1}^*)\ t = T-1, T-2, \cdots, 1 \tag{5.11}$$

Considering the algorithm, we can note that the Viterbi algorithm is similar as the $\alpha - \beta$ calculus excepting the backtracking step.

This recall of the algorithm is necessary to introduce the threshold mechanism which is also based on the δ introduced here. The cumulated threshold is the partial maximum likelihood given by the Viterbi algorithm.

5.3.2 Threshold mechanism

The calculation of these threshold values is based on the Viterbi decoding. Once the training process has been finished, performed by the classical Baum-Welch algorithm, we perform resemblance estimation on the learning corpus for each general word model. In the figure above Fig. 5.3, you can find an NSHP-HMM scheme of such a general word-model. Let suppose we have a general word model M containing two letters (ex. or, in, at, to, etc.)

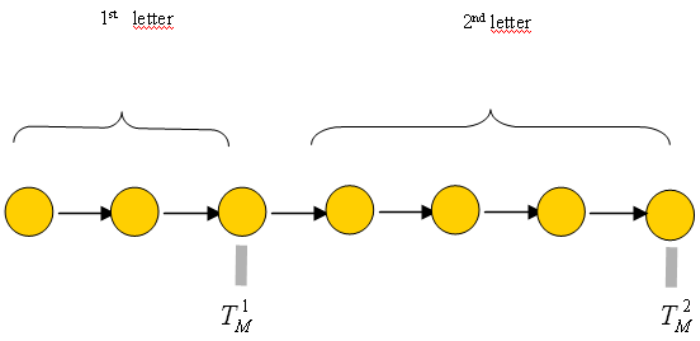

FIGURE 5.3 – The NSHP-HMM with the different threshold values fixed at each letter limit. The letter limits are known as the general word NSHP-HMM are built considering the word meta-models and the letter models.

The calculated T_M^k threshold value means : which is the resemblance till the letter k (letter k included) of the general word-model M taking into account the previous letters composing the model and the unknown word shape. This kind of threshold value calculation has a sense as it can be imagined that we perform in run-time manner a matching with a word segment each time. Building step-by-step the general word-model (at letter level), allows us to estimate properly the resemblance of the unknown word shape with the built model. In that kind of estimation, we are taking into account the different letters contexts as due this context-parameter inter-letter variations occurs. Once the threshold calculation is performed for the entire training set belonging to the model M, a mean value for each threshold is calculated :

$$\overline{T_M^k} = \frac{1}{N} \times \sum_{i=1}^{N} T_M^k(pattern_i) \tag{5.12}$$

where N is the number of patterns in the learning corpus and $T_M^k(pattern_i)$ is the i-th threshold value for the pattern belonging to the model M.

As the NSHP-HMM is a discriminative model, this threshold values should be calculated once in order to get a threshold set for each model (see Fig. 5.4), where $M_n^k = \{T_M^1, T_M^2, \cdots, T_M^k\}$ and L is the size of the lexicon.

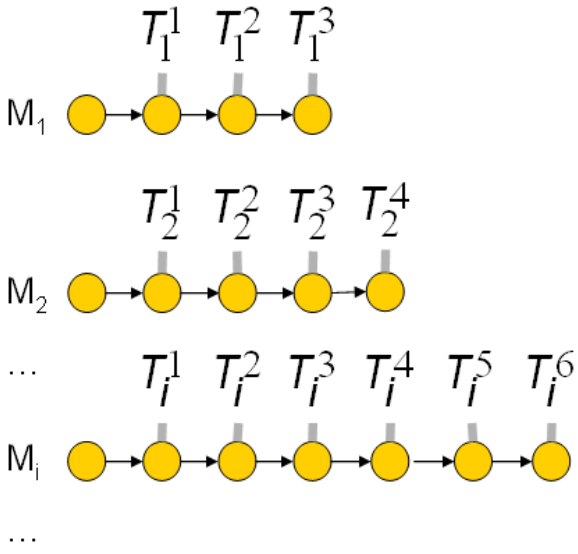

FIGURE 5.4 – The Viterbi pruning considered for a flat lexicon

The proposed level-based pruning mechanism is based on the comparison of the calculated resemblance value of a given observation sequence (height normalized word shape) in an instant t reaching the state j, where j is the final state for a letter in the general word-model. The stopping criterion is based on the number of the letters composing the general word model and the thresholds. If during the decomposition process the run-time calculated values are not higher than the thresholds, the Viterbi mechanism can be stopped.

Given an unknown word shape (observation sequence) and the n-th general word-model M_n with a pre-calculated threshold-set $M_n^k = \{T_{M_n}^1, T_{M_n}^1, \cdots, T_{M_n}^k\}$, containing k characters, the resemblance estimation through the pruning Viterbi algorithm is as follows :

Viterbi pruning algorithm :

Length :=k*l ;

i :=1 ;

ThresholdCounter :=0 ;

t :=1 ;

for each j :=1..N
    Calculate $\delta_t(j)$;
while(t<T) ***do***
{
repeat
    t++ ;
    for each j :=1..N
        Calculate $\delta_t(j)$;
until((t<T) and (argmax($\delta_{t-1}(j)$) = $argmax(\delta_t(j))$))
if $(argmax(\delta_{t-1}(j)) == LastStateOfLetter(i))$
    {
    if $(\delta_{t-1}(j) \geq T^i_{M_n})$ ***then*** ThresholdCounter++ ;
    if((i=Length) ***and*** (ThresholdCounter<C)) ***then*** STOP ;
    i++ ;
    }
}

where $l \in [0, 1]$ is a constant established based on trial runs and the *pattern* is the current image which is analyzed. C is also a constant value based on the number of letters considered by the model M_n. The function LastStateOfLetter(i) returns the last state of the letter i of a given model cumulating the previous states of the previous letters. This value is based on the number of states in the letter and the position of the letter in the meta-model.

The algorithm is adapted as it is based on the number of letter components for each model M. The Viterbi pruning algorithm is based on the classical one, but it has the possibility to stop it before analyze throughout the whole word shape. During the classical Viterbi algorithm, it's calculating the resemblance for the observation sequence with the model M and when it reaches the state j of the model the probability is compared to the corresponding T^k_M threshold. This kind of comparison is performed in a part (see *Length* in the algorithm) of the model. If during this analysis the calculated resemblances are higher than the pre-calculated thresholds, the decoding can continue in the classical manner, otherwise in function of the number of thresholds attempted (see *ThresholdCounter* in the algorithm) the algorithm can be stopped and the current model discarded.

The C value is necessary to control the possible variations of the model in length. The *Length* parameter controls the depth of the search. We are continuing the resemblance calculus till we attend the *Length* which means the number of letters which were analyzed. This parameter is one of the most important, as if this value is small the algorithm has no possibility the compare the calculated resemblance value with the established threshold. While, if this parameter is set

to the number of letters encountered in the word, the complexity of the algorithm grows. A good trade-off should be found between precision and time complexity.

5.3.3 Natural length estimation

Short words can be easily distinguished from long words by comparing their lengths. So the length is a very simple criterion for lexicon reduction. The length of the observation sequence (or feature vector) extracted from the input image has a hint about the length of the word from which the sequence was extracted.

Many lexicon reduction methods use such extra information to reduce the number of entries to be matched during the recognition phase [DG00, KCGM93, KBH97].

Kaufmann et al. [KBH97] use a length classifier to eliminate from the lexicon the models that differ significantly from the unknown pattern in the number of symbols. For each model, a minimal and a maximal length is estimated. Based on this range, a distance between word and a model class is defined and used during the recognition process to select only the pertinent models.

Kaltenmeier et al. [KCGM93] use the word length information given by the statistical classifier adapted to features derived from Fourier descriptors for the outer contours to reduce the number of entries in the vocabulary of city names.

Other methods do not rely on the feature vector to estimate the length of words but on particular techniques. Kimura et al. [FK93b] estimate the word length of a possible word candidates using the segments resulting from the segmentation of the word image. Such estimation provides a confidence interval for the candidate words and the entries outside of such an interval are eliminated from the lexicon. An overestimation or an underestimation of the interval leads to errors. Furthermore, the estimation of length requires a reliable segmentation of words, what still is an ill-posed problem.

Powalka et al. [PSW97] estimate length of cursive words based on the number of times an imaginary horizontal line drawn through the middle of the word intersects the trace of the pen in its densest area.

A similar approach is considered by Guillevic et al. [DG00] to estimate word length and reduce the lexicon size. The number of characters is estimated using the counts of stroke crossings within the main body of a word.

In our case we are using a similar approach. The *natural length* is defined as the average number of black and white transitions encountered in the middle zone of the writing. This approach has been used with success in other holistic approaches too. This measure seems to be an adequate and reliable feature to reduce the number of entries in a dictionary [GSS94, MGR+95]. Considering this natural length estimation, there is an increase of 4.5% in the accuracy, which

is a considerable gain considering the time costs of such estimation.

5.4 Experiments and results

5.4.1 Results concerning the symmetry in the NSHP-HMM

In that section we will show the impact of the symmetry considering the different NSHP-HMM flips contributing to the final response. As the neighborhood is not symmetric 4 NSHP-HMM contribute at the final scores, summing the partial results as discussed in the previous chapters.

Number of flips	Algorithm	Word/Sec	Score
1+2+3+4	Viterbi classic	3,7	86.45%
1	Viterbi classic	15	75.68%
2	Viterbi classic	15,2	72.30%
3	Viterbi classic	15,2	4.64%
4	Viterbi classic	15	2.59%
1+2	Viterbi classic	7,4	80.94%

TABLE 5.1 – The impact of the flips considered by the NSHP-HMM for the Bangla dataset considering 76 word classes

As it has been shown in the Table 5.1 the different flips have different influences in the final scores. While the first and the second flip gives a considerable score of the final result, the remaining two flips have just minor contributions.

Considering just the flip 1 and flip 2 the results of 80.94% it is an acceptable results in comparison with the 86.40% obtained with all the 4 flips. The system accuracy decreases but the recognition time gain has also a 2 factor.

This can be explained with the fact that the upside-down images do not give sufficient information even for a machine reading system which should not consider the left to right sense of writing. Such a conclusion can be also interpreted as : the NSHP-HMM handwriting recognition model is much more closer to the human reading mechanism where the upside-down word images cannot be read easily not even by the human readers.

5.4.2 The Viterbi pruning results

In that section we will give the results achieved by the Viterbi pruning algorithm described in the previous section. The recognition time as the flips have been also considered.

For the further results analysis, considering the minimal addition of the flip 3 and 4, we are considering just the first and the second flip discarding the remaining two flips. To show general

Lexicon	Flip	Algorithm	Word/Sec	Recognition
Bangla(30)	1	Viterbi pruning	100	82.63%
Bangla(30)	1+2	Viterbi pruning	48	88.00%
Bangla(65)	1	Viterbi pruning	27.81	75.01%
Bangla(65)	1+2	Viterbi pruning	15	76.75%
Bangla(76)	1	Viterbi pruning	24.36	72.38%
Bangla(76)	1+2	Viterbi pruning	12.12	74.66%

TABLE 5.2 – Results reported on different Bangla lexicon sizes using the Viterbi pruning algorithm

results some basic interpolation mechanism has been considered. The reported results are based on the cumulus of the flip 1 and flip 2 respectively.

The recognition time changes linearly in the different vocabulary entries but comparing to the classical amelioration, we can observe a speed up factor ≈ 2 which can be considered a success. The results reported here are less than for the Viterbi classic, but we should consider the fact that in this case just partial NSHP-HMM flips are considered. The stopping criterion invoked before is based on the length of the word model in letter terms speaking, which gives a certain rigidity to the system.

To decrease much more the time complexity of the Viterbi algorithm, a dynamic pruning should be proposed, where the stopping criteria based on the covered word shape length should be established based on a training process performed on the training dataset. That means, instead of using just a the number of letters as baseline criteria, we should extend it considering a neural system. This system will be in charge to learn for each word class which is the optimal path which should be followed in order to be able to decide at the end if the resemblance estimation should continue or not. Such approach has been used also with success to establish the parameters of certain rejection criteria used in digit recognition.

5.4.3 Natural length estimation results

Considering the natural length estimation, we have focused our experiments on two different things, related to the number of models which were pruned during the Viterbi search and the accuracy achieved by the system. In Tab 5.3 the results are related to the entire Bangla dataset, considering the 76 city name classes and the NSHP-HMM with the four flips.
where ϵ is the Euclidean distance between the natural length given by our algorithm and the models length considered for the Viterbi search.

The achieved results are not so sound as the results using the classical Viterbi algorithm for the flat representation, but we have shown the importance of the natural length estimation

ϵ	**Pruned models**	**Word/Sec**	**Accuracy**
1	54.96%	7.1	62.19%
2	31.84%	5.3	78.38%
3	16.25%	5.0	82.62%
4	7.19%	4.8	84.51%

TABLE 5.3 – The impact of the natural length estimation in the Bangla city name recognition

without loosing considerably the accuracy factor of the system and decreasing by a factor ≈ 2 the time complexity of the algorithm.

Comparing the results with the classical Viterbi algorithm, we can conclude that we are loosing 1.79% of accuracy but in the mean time we have a considerable speed gain. Instead of recognizing 3.7 word/sec, using the pruning, we can recognize 4.8 word/sec. We have a gain of 1 word/sec fact which shows the interest of our pruning mechanism.

The results obtained by us cannot be directly compared with the results of Saon or Choisy as they were not interested in the time complexity aspect of the system. The systems proposed by them do not need such kind of pruning as the number of entries considered is composed just by 26 words which is much less than the Bangla lexicon, which has 76 entries. The width of image samples is much considerable as in case of the SRTP dataset.

5.5 Conclusions

In that chapter we described a pruning mechanism based on cumulative threshold calculus at letter level considering the threshold at letter level given by the meta-model and the general word model. Instead of using a prefix tree representation where the common prefix part are shared, we developed this pruning mechanism in a flat lexicon representation. This is because the composition of our lexicon there are not so many common parts to be exploited. Another fact why such approach can not work properly : in our system there is no segmentation, so the letter limits are not well defined which is a must in these kind of representations.

The cumulative thresholds are calculated based on the letter models and word meta-models. At each letter ending in the model, a resemblance value is calculated based on the Viterbi algorithm. This threshold value means the resemblance of the word shape with the model analyzing the first k letters in the model. The thresholds are estimated through the training patterns giving a certain statistical power to the model.

The technique is new as it is working on an implicit segmentation, while the current vocabulary reduction strategies assume an explicit segmentation, where the level-building algorithms based on letters can find an optimal solution for the Viterbi decoding. To improve the stopping

criteria proposed by us, we should train the different models to decide themselves about the limit where the stopping in the Viterbi algorithm should occur.

The natural length estimation proposed as second reduction strategy is also an interesting issue as it allows to reduce the search space considered by the Viterbi algorithm, but in our case just 7.19% of the models have been reduced. This is due to the fact that the distribution of word based on the number of letters is not uniform. The most part of the lexicon entries are composed by 5 up to 8 letters. For more details concerning the Bangla city name dataset, please refer to Section A.2.1.

6

Neural and stochastic methods in handwritten digit recognition

The aim of this chapter is to show the differences between the neural and stochastic methods and their strength and weakness throughout the results obtained for the different handwritten digit datasets. In order to exploit the complementarity between the different approaches, some combination schemes will be also proposed. The results achieved by the different combination schemes drive us towards such fusion techniques as being the best solutions.

6.1 Introduction

In this section, we will discuss different personal contributions based on separated digit recognition.

The first work deals with a specific multi-layer perceptron proposed by us to recognize separated handwritten digits. An MLP type neural network will be described in details. In the meantime, a specific training mechanism will be presented allowing to train the network. The fast training mechanism is based on error minimization selecting some hard patterns creating a set of support vectors in SVM terminology. These hard patterns will constitute the frontiers of the decision surfaces adjusted by the network. The training process is a generic one, allowing to apply it for different task without loss of accuracy and generalization.

The second issue concerns the usage of the NSHP-HMM in digit recognition. Nowadays the scientific community use different neural approaches for this task. Our aim is to show the importance of the HMM based models highlighting a different classification paradigm which seems to be complementary with the other classification techniques. Another type of vision, another type of learning can relieve these differences.

The last issue discussed here is the combination of different classifiers. Such a classification

is necessary in order to get high accuracy classification schemes. The only prerequisite is the complementarity allowing to help the different classification schemes to merge their results for a better final solution.

Finally, some results will be given and discussed in details. To orientate and to measure our contribution to the field some comparison study will also be given.

6.2 Proposed neural and stochastic strategies in digit recognition

6.2.1 The multi-layer perceptron : *ReadNet*

We used a Multi-layer Perceptron (MLP) Neural Network based scheme [BVM+04, RVP+05a, RVP+05b, VB05a] for the recognition of English and Bangla numerals. Instead of using time consuming feature extraction methods or data dimensionality reduction strategies, we have preserved the whole information by feeding the network with the raw images. The other systems use feature extraction methods [Guy91] considering moments, gravity center and other statistical features. The drawback of such system is their leak of dependency of features with special consideration to the bi-dimensional aspect of the pattern considered as being the most important one by the human vision.

The raw image (pixel level) conserves the most the information. Our reflection in word recognition (see the NSHP-HMM) as well in digit recognition is guided also by this fact. We have considered the pixel level information. To exploit totally the information given by the raw image, we have considered a global dependency between the pixels. Some other methods presented in [LBBH01, CVB05a] presume just a local dependency using different size windows to analyze the image. As one of the constraints of the NN is the fixed input size, some size normalization was considered to fit the images into a 28×28 shape matrix. The choice of this pattern size is inspired by the MNIST separated digit dataset.(see details in A.1)

The network topology is constituted by : the number of processing units in the layers (input, output, hidden layers), the number of hidden layers intercalated between the input and the output layer, the nature of the activation function and the links between the different processing units. Concerning the topology of the network there is no rigorous scientific protocol to design a neural network for a given pattern recognition task. The topology settings are often based on empirical presumptions.

Instead of using any kind of presumption on the considered data, our MLP is a fully connected one. That means each processing unit in a layer is interconnected with all the units of the following layer. Our topology is driven by the data. As the image original MNIST images were size normalized to 28×28, our input layer contains 784 inputs while the output has 10 units, each unit corresponds to a digit class to be identified. We have considered one hidden layer to

be able to model a complex decision surface as in case of digits, where different digits can be confused. The number of the units in the hidden layer was fixed to 555 based on trial runs.

For training purpose the well known error back propagation has been considered. The learning rate and the momentum are set to suitable values based also on trial runs. The stopping criteria of back propagation algorithm selected for the present work is that the sum of the squared errors for all training patterns will be less than a certain limit. This allows a proper generalization. Some early stopping criteria was also tested.

We have also designed a more complex neural network where the number of classes was 16. Although because of bi-lingual (English and local language Bangla) nature of the Indian postal documents, the number of numeral class is supposed to be 20. However, we have mapped only 16-classes in the output layer of the MLP. This is because English and Bangla "zero" are (historically the Arabs borrowed the zero from India and transported to the west) the same and we consider these two as a single class. Also, English "eight" and Bangla "four" are same in the shape. Moreover English and Bangla "two" looks sometimes very similar. English "nine" and Bangla "seven" are also similar. Example of some handwritten Bangla numerals is shown in Fig. 6.1. To get an idea of some similar numerals in Bangla and English see Fig. 6.2.

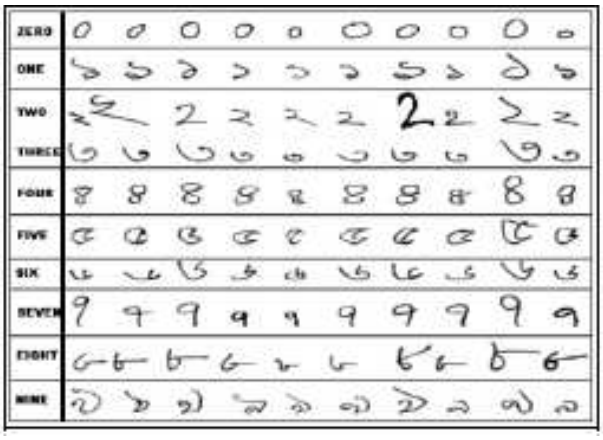

FIGURE 6.1 – Samples of Bangla handwritten numerals.

FIGURE 6.2 – (a) English Nine and Bangla Seven, (b) English and Bangla Two.

In the proposed recognition system we used three classifiers for the recognition. The first classifier deals with 16-class problem for simultaneous recognition of Bangla and English numerals. The other two classifiers are for recognition of Bangla and English numerals, separately for

10 numerals. Based on the output of the 16-class classifier we decide the language in which pin-code is written. As mentioned earlier, Indian pin-code contains six digits. If majority of these six numerals are recognized as Bangla by the 16-class classifier then we use again Bangla classifier on this pin-code to get higher recognition rate. Similarly, if the majority of the numerals are recognized as English by the 16-class classifier then we use English classifier for final recognition of pin-code digits.

Fast Data Driven Learning Corpus Building (FDDLCB) algorithm description

Our method is based on incremental learning using error selection criteria. The approach is based on an MLP type classifier with one hidden layer.

The main idea of the FDDLCB algorithm is to build-up in run-time a data driven minimal learning-corpus based on the LMS by adding additional patterns to the training corpus at each training level in order to cover maximally the different variations of the patterns and reducing the recognition error.

Let us denote by *GlobalLearningCorpus* (GLC) the whole set of patterns which can be used during the training procedure, by *GlobalValidationCorpus* (GVC) the pattern set which helps to quide the training, by *GlobalTestingCorpus* (GTC) the whole set of patterns which can be used for the test and by *DynamicLearningCorpus* (DLC) the minimal set of patterns which can serve to train the network. Let's also denote by NN the neural network and by N the iterator, which provides the number of new patterns to be considered at each learning level and M denotes the number of classes to be separated.

Algorithm description :

Initialization :
$DLC = \{x_i \in GLC \mid i = \overline{1, M}\}$
$GLC = GLC - DLC$
Database Building :
repeat
{
repeat
{
TrainNetwork(NN,DLC)
}until(NetworkError(NN,DLC,ALL)> $Threshold_1$)
TestNetwork(NN,GVC)
if (NetworkError(NN,GVC,ALL)$\prec$ *Threshold*$_2$*) then STOP*
else
{
TestNetwork(NN,GLC)
if (NetworkError(NN,GLC,ALL)$\prec$ *Threshold*$_1$*) then STOP*
else
{
$DLC = DLC \cup \{y_i \in GLC \mid i = \overline{1, N}\}$
$GLC = GLC - DLC$
}
}
}until($\mid GLC \mid > 0$)
Results :
NN contains the modified weight set
DLC contains the minimal number of patterns which is sufficient to train the NN
where :

- *TrainNetwork(NN,DATASET)* will train the NN with the given DATASET using classical LMS error minimization and error backpropagation
- *TestNetwork(NN,DATASET)* will test the NN with the given DATASET
- *NetworkError(NN,DATASET,SAMPLES_NUMBER)* calculates the error given by NN using SAMPLES_NUMBER of patterns from the DATASET using the LMS criterion
- y_i denotes the pattern from the GLC giving the i-th highest error during the test
- $\mid DATASET \mid$ denotes the cardinality of the DATASET

The algorithm is starting with an initialized DLC set, where we have selected for each class one random representative pattern (x_i) in order to not favor one or another class initially. The algorithm performs the network training with these samples. Once the training error is less than an empirical threshold value, the training process stops and we test our network with the samples

belonging to the GVC. If the error criterion is satisfied, the algorithm stops as the training was successful and we test the network considering the GTC. Otherwise, we should continue by adding new samples to our DLC set. To do this we are looking from the GLC for the N samples (y_i) giving the highest error in the classification. If this error is less than a threshold value we are stopping the algorithm, as we cannot add extra helpful information to the network. Otherwise we are picking these N elements from GLC and move them in the DLC and restart the training process on this new extended dataset, considering as initial parameter state the parameters learn in the previous step. The algorithm stops when the error criterion is satisfied or there are no more available patterns in GLC set. In the second case we are in the classical training as finally we are using the whole dataset. So there is no restriction in the algorithm. In the worst case we have almost the same results as in case of using the whole dataset using classical learning strategies.

A modified version of the FDDLCB algorithm consists to feed the network with class samples having the same distribution. This precaution is necessary as stated by [Jap00], in order to not influence the system in a way or another. For that reason we modified the conditions of the DLC set creation. Now, at each iteration we add N samples for each pattern class based on their highest error contribution in their class instead of using the first N samples of the dataset giving the highest error rate. Using this selection process we can guarantee the distribution uniformity for each pattern class.

Experiments and results

The experiments performed by the *ReadNet* neural network and the FDDLCB algorithm used as input data the MNIST reference database (see A.1). This dataset contains 60.000 samples for learning and 10.000 samples for test. The training corpus was split in a training corpus (50,000 patterns) and a validation corpus (10,000 patterns) in case of the FDDLCB algorithm. The 28x28 normalized gray-scale images contain separated handwritten digits from 0 to 9. The tests were performed with a fully connected MLP with one hidden layer as described above. Additionally, some real data coming from Indian postal document was also considered. Here we can distinguish between Bangla digits and English digits. A complete description of the dataset can be found in Section A.2.

The results of the *ReadNet* neural network concerning the MNIST benchmark dataset and a Bangla digit dataset is shown firstly, while in the second part the results of the FDDLCB algorithm will be discussed.

The results shown in Table 6.1 give the evolution of the recognition rate in function of the number of hidden units and free parameters used in the hidden layer. We can observe a semi-linear relation between the number of units and the accuracy. A more considerable number of

Number of units	Number of free parameters	Recognition accuracy
10	7,940	90.66%
50	39,700	96.25%
100	79,400	97.45%
200	158,800	97.75%
300	238,200	98.10%
400	317,600	98.30%
500	397,000	98.41%
555	440,670	**98.59**%

TABLE 6.1 – The ReadNet results considering different number of hidden units

hidden units allows to create a more complex decision surface but starting from 300 units, a stabilization can be observed.

We can confirm the hypothesis that after a certain number of hidden units the network cannot learn new specificities or such a considerable number of neurons implies an increasing number of free parameters which cannot be trained with such a number of samples.
The recognition by class is given in Table 6.2

Class	Recognition accuracy	Class	Recognition accuracy
0	99.59%	1	99.38%
2	98.45%	3	98.81%
4	98.07%	5	98.21%
6	98.85%	7	98.15%
8	98.36%	9	97.92%

TABLE 6.2 – The recognition of the MNIST dataset

We can observe that the network can handle the digit 0 but he can not recognize the digit 9. This fact can be explained with the complexity of the shapes and the physical patterns available for train and test the network. In Table 6.3 the confusion matrix of the classifier is given.

The most common confusion is between the digit 7 and 2 which is due the similar shape of these patterns as well as the digit 7 is written in American style omitting the line. Similarly, the digit 4 is recognized 9 times as being digit 9 which is also due to the slant and the shape similarity. In order to analyze in detail the different type of confusion please refer to Fig.6.3.

In Fig. 6.3 all the mis-recognized images from the MNIST dataset are listed.

In order to get a comparative idea, considering more or less the same test conditions, in the Table 6.4 some MNIST related results are presented.

We can observe that our achievement overcomes all the results mentioned in the Table 6.4.

Confusion	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	976	0	1	0	0	1	1	1	0	0
1	0	1128	2	1	0	1	1	1	1	0
2	4	0	1016	2	1	0	1	5	3	0
3	0	0	4	998	0	2	0	2	3	1
4	1	1	2	1	963	0	3	0	1	10
5	3	0	0	6	0	876	2	2	2	1
6	4	2	0	0	2	2	947	0	1	0
7	0	3	8	2	0	0	0	1009	2	4
8	1	0	1	3	3	1	1	4	958	2
9	2	2	0	3	5	2	1	5	1	988

TABLE 6.3 – The confusion matrix for the MNIST dataset

FIGURE 6.3 – The 141 test patterns misclassified by ReadNet. Below each image is displayed the correct answer (left side) and the corresponding network answer (right side). These errors are mostly caused either by the genuinely ambiguous patterns or by digit written in a style that are underrepresented in the training set.

Ref.	Method	Rec. rate
[LBBH01]	28x28-300-10	95.30%
[LBBH01]	28x28-1000-10	95.50%
[LBBH01]	28x28-300-100-10	96.95%
[LBBH01]	28x28-500-150-10	97.05%
Thesis (personal)	28x28-555-10	**98.59**%

TABLE 6.4 – A comparative result set on the MNIST dataset

This can be explained with a proper weight initialization before the training procedure, a good choice of the hidden units, adapted to the recognition task and a decreasing learning rate in order to avoid the possible local minimum in the error descent. Comparing the results of the networks containing 300 units in the hidden layer we can observe a 2.80% of net amelioration.

It remains somewhat of a mystery that networks with such a large number of free-parameters manage to achieve reasonably low testing errors. We conjecture that the dynamics of gradient descent learning in multiple layer nets has a "self-regularization" effect. Because the origin of weight space is a saddle point that is attractive in almost every direction. The weights invariably shrink during the first few epochs. Small weights cause the sigmoid to operate in the quasi-linear region, making the network essentially equivalent to a low-capacity, single layer network. As the learning proceeds the weights grow, which progressively increases the effective capacity of the network. A better theoretical understanding of these phenomena and more empirical evidence, are definitely needed.

Considering the digit recognition for Indian postal documents, namely separated pin-codes, the overall recognition accuracy of the proposed 16-class classifier and the individual Bangla and English classifier on the above data set are given in Table 6.5. From the results we note that in Bangla classifiers we obtained 2.03% better accuracy than the 16-class classifier. This is due to decrease in the number of classes and also decrease in the shape similarity between English and Bangla numerals. The confusion matrix of the tree classifiers is shown in Fig. 6.4 , Fig. 6.5 and Fig. 6.6.

Classifier	Training set accuracy	Test set accuracy
16-class classifier	98.31%	92.10%
Bangla classifier	98.71%	94.13%
English classifier	98.50%	93.00%

TABLE 6.5 – Overall numeral recognition accuracy

In order to allow a comparison of our result concerning the Bengali digit dataset in the Table 6.6 we are presenting some other results based on other Bengali digit datasets. As stated by Pal

Numeral	Classified as --->															
(data size)	০	১	২	৩	৪	৫	৬	৭	৮	৯	1	3	4	5	6	7
০ (1226)	1169	2	2	14	5	8	4	5	1	1	0	0	4	8	3	0
১ (433)	1	399	4	1	2	0	5	0	0	17	1	0	0	2	0	1
২ (759)	4	8	689	1	20	6	0	3	3	2	6	3	8	1	3	2
৩ (303)	2	3	0	269	2	10	9	0	0	5	0	0	0	0	3	0
৪ (507)	3	0	2	0	476	1	0	8	0	0	4	4	1	4	3	1
৫ (246)	4	2	1	1	4	233	0	0	1	0	0	0	0	0	0	0
৬ (211)	1	2	0	9	0	3	191	0	2	1	0	0	1	0	1	0
৭ (655)	1	0	4	1	5	0	0	622	0	0	1	1	11	0	0	9
৮ (206)	1	0	0	0	0	0	0	0	203	1	1	0	0	0	0	0
৯ (206)	1	13	2	0	3	0	2	0	0	184	0	0	0	0	0	1
1 (418)	0	1	14	0	1	0	0	9	4	0	375	0	1	10	2	1
3 (226)	1	1	2	1	7	0	0	7	0	0	5	189	0	9	0	4
4 (216)	0	0	0	1	0	0	0	2	0	1	2	0	207	1	0	2
5 (289)	6	0	1	0	10	0	0	2	1	1	7	9	8	239	4	1
6 (280)	0	0	1	3	3	3	3	0	2	2	2	0	1	2	258	0
7 (225)	1	0	1	0	0	0	0	12	1	0	2	0	9	2	0	197

FIGURE 6.4 – Confusion for 16-class classifier (Bangla and English)

Numeral	Classified as --->									
(data size)	০	১	২	৩	৪	৫	৬	৭	৮	৯
০ (1226)	1175	3	1	14	4	10	1	10	0	8
১ (433)	2	394	4	1	2	0	6	0	0	24
২ (759)	5	6	705	2	20	7	1	3	6	4
৩ (303)	3	4	0	270	1	10	:0	0	0	5
৪ (507)	5	1	6	0	480	4	0	10	0	1
৫ (246)	5	2	2	0	1	232	2	0	1	1
৬ (211)	1	2	1	11	0	3	189	0	2	2
৭ (655)	1	0	6	1	5	2	0	640	0	0
৮ (206)	1	0	0	0	0	0	0	0	202	3
৯ (206)	2	:1	1	0	3	0	3	0	0	186

FIGURE 6.5 – Confusion matrix for 10-class Bangla classifier

Numeral	Classified as --->									
(data size)	0	1	2	3	4	5	6	7	8	9
0 (1226)	1197	0	0	2	7	7	3	0	3	7
1 (418)	0	369	16	1	3	12	3	4	0	10
2 (759)	7	6	701	7	8	0	6	2	17	5
3 (226)	4	5	4	192	1	4	0	5	6	5
4 (216)	0	1	0	0	208	1	1	3	0	2
5 (289)	7	8	3	14	6	234	4	1	11	1
6 (280)	2	1	4	0	1	1	268	0	3	0
7 (225)	0	3	1	0	9	1	0	200	0	11
8 (507)	4	4	5	4	2	4	3	1	475	5
9 (655)	2	3	4	1	14	0	1	7	2	621

FIGURE 6.6 – Confusion matrix for 10-class English classifier for digits coming from Indian Postal documents

Ref.	Method	Rec. rate
[WLS07]	PCA	95.05%
[PBC06]	Binary tree classifier	92.80%
[aKDTDP02]	SOM and MLP	93.26%
Thesis (personal)	ReadNET	**94.13**%

TABLE 6.6 – A comparative result set on different Bengali digit datasets

in [PBC06] just a few works have been done yet in this domain of handwritten Bangla digit recognition. However, considering the comparison we can note that our results are among the best results.

In order to achieve a recognition rate like 98.59% (for MNIST), using the whole learning corpus we need at least 30 learning epochs. That means we should present at least $30 \times 60.000 = 1.800.000$ patterns to our network. In Table 6.7 we show different constructed datasets, the number of patterns presented to the system and the obtained results by the FDDLCB algorithm on the test set in function of N parameter.

So we can achieve comparable result (98.36% accuracy), even with just 1960 samples from the possible 60,000 that means, the other patterns can be considered redundant information, so it's not necessary to use them. The learning process can be reduced substantially as it is possible to achieve almost similar results presenting just 93,180 patterns to the network. So we can speed up the learning process 14 times that is a considerable gain even for a high-tech computer.

N	Generated learning set (DLC)	Patterns presented	Recognition rate
50	1,260	64,390	98.30%
100	1,310	63,400	98.29%
150	1,960	93,180	**98.36**%
200	2,810	121,410	98.32%
250	3,010	130,900	98.21%
300	4,810	176,670	98.15%
350	3,510	147,720	98.20%
400	3,210	128,120	98.22%
450	3,160	128,270	98.23%

TABLE 6.7 – Results obtained with different datasets constructed by FDDLCB algorithm

As in [SC03] the authors provide results of their pattern selection method on MNIST benchmark dataset, a comparison study can be performed.

Nine SVM type binary classifiers were used : class 8 is paired with each of the rest. The reported recognition error in average over nine classifiers is 0.28% using all the available patterns and 0.38% for the pattern selection based technique. The lost of accuracy is similar as in our case. Unfortunately there is no results reported concerning the real recognition accuracy for each separated digit class so a direct comparison cannot be performed with our method.

The time factor is reduced with a factor of 11.8 which is much less as in our case where a 14 speed factor was established.

Similarly, the number of used patterns (16.76%) serving as support vector is much more considerably than our 3.20% selected patterns to train the system.

The modified FDDLCB algorithm result 98.01% is near to the result produced by the original algorithm but it needs much more iterations and samples (9,000 different samples were selected while 864,600 patterns were presented to the network).

In Figure 6.7 we present the class distribution for the different datasets built by FDDLCB in function of the N parameter. The x-axis means the different classes, and the y-axis means the distribution percentage of the different classes. We can see than the element distribution variance is not significant for the different datasets so the N parameter can control just the learning convergence speed and the size of the built dataset.

The empirical value $N = 50$ was established after some trial runs performed with different N values. We found this is the optimal value which should be used in order to achieve a considerable speed gain.

Similarly, the results presented in Table 6.7 prove than the changing of parameter N has no

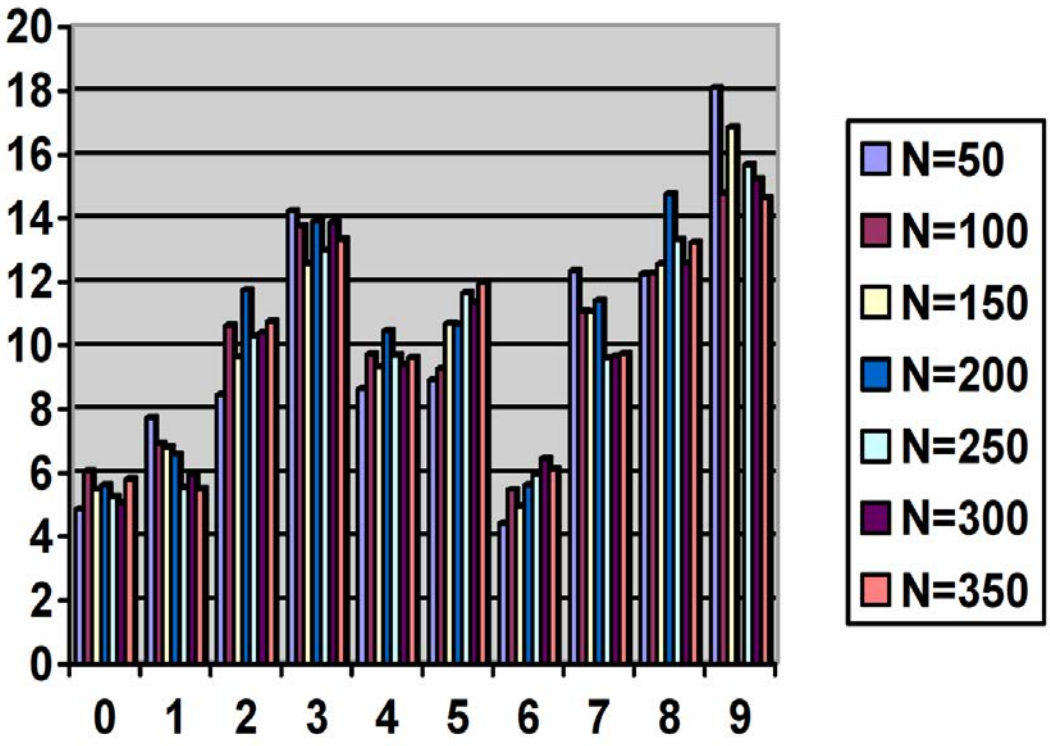

FIGURE 6.7 – The samples distribution in the classes for the different constructed datasets

major influence on the results. It can influence just the size of the built dataset and the speed of the building process.

Using the same pattern distribution as in Fig. 6.7 using random choice for the patterns selection for the dataset creation, the recognition accuracy cannot achieve higher average recognition scores than 91.01%. For this test purpose 1,000 random dataset was generated and considered.

Analyzing the dynamic learning corpus we can pronounce also in the matter of the intra-class and inter-class variance. In the MNIST database the class "0" contains the fewest variance and the class "9" contains the most variation, so we need much more samples belonging to class "9" in order to achieve a good recognition score.

In pattern complexity terms speaking, the class "0", "1", "6" are the classes which are the simplest and the classes "3", "4", "8", "9" are the more complex ones, which is natural as they can be confused.

6.2.2 Conclusions

Considering the proposed ReadNET network, we can conclude that the network even if it is a fully-connected multi layer perceptron, gives satisfactory results in comparison with the best state of art works in this subject. If just the multi-layer perceptron like works are considered [LBBH01], our results outperform them.

The same network has been used for two different recognition tasks : English digit recognition and Bangla digit recognition with the same success. The increased number of units in the hidden layer allow to differentiate between the small variation of the inter-class patterns and to distinguish between the intra-class patterns. The number of units in the hidden layer invoke an increased number of free-parameters which requisite much more patterns for the training process. That is the reason why the results on the MNIST database are much more higher than for the digits collected from the Indian postal documents. In this case, the number of samples is less and the distribution of the samples is not uniform as for the MNIST data. This kind of distribution can invoke the problem of the imbalanced dataset problem discussed in detail by Japkowicz in [Jap00].

The FDDLCB algorithm is based on a dual LMS error estimation, which can guarantee the convergence of the algorithm. The first LMS minimization is used in the training process for the error back propagation. The second one is used when we are calculating the LMS error for the samples during the recognition. The misclassified patterns should be added to the DLC set in order to minimize the recognition error by learning these new items which have contributed to the error accumulation. The method reduces substantially the learning period and discards the redundant information in order to avoid the over fitting.

The performed tests on MNIST showed that is possible to achieve 98,36% recognition accuracy using just 1,960 different samples and the learning time can be reduced by a factor of 14, a time gain which is also substantial, considering the algorithm complexity.

The mechanism cannot function for the improvement of the system presented in [SSP03] which is based on the data redundancy.

The algorithm tries to enlarge the different class boundaries using in learning the "extreme" patterns. The algorithm increases the number of forward steps (propagation) but decreases substantially the number of backward steps (error back propagation) which are much more costly in calculus.

The FDDLCB algorithm can also be used to solve the challenge proposed by Japkowicz in [Jap00] in order to deal with the class imbalance problem, which often occurs in the real world applications. Many times we deal with learning corpuses where the distribution of the samples for the different classes is not uniform. There are under represented classes and respectively over represented classes. The methods presented in [Jap00] based on down-sizing and re-sampling are restrictive as there is no rigorous selection criteria to choose which elements should be discarded or re-sampled.

The FDDLCB can avoid the over fitting effect caused by the presented methods using a rigorous selection criterion and does not allow to specialize the network in all the variation of the different pattern. Selecting the hard patterns located at the decision surface borders the classifier learn just the relevant features of the pattern without considering the little variations inside the

"cluster".

6.2.3 The NSHP-HMM in digit recognition

Considering the general NSHP-HMM model described in Chapter 3 we describe here just the test conditions to perform recognition of separated handwritten digit recognition using this stochastic model. A general scheme is shown in Fig 6.8.

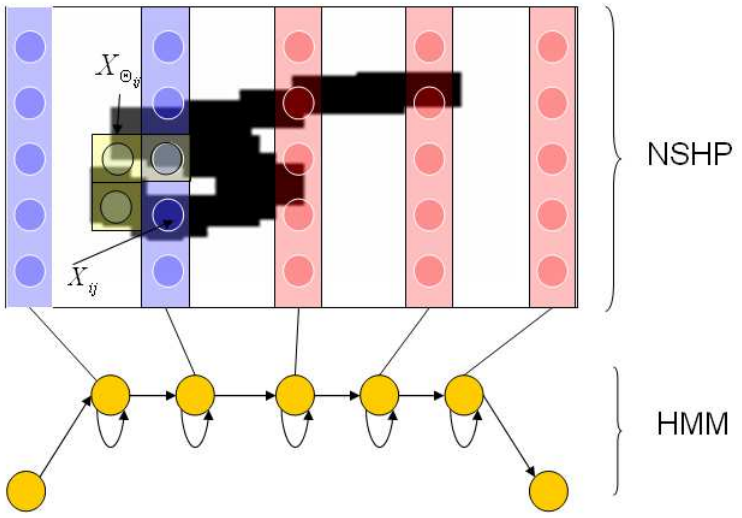

FIGURE 6.8 – The NSHP-HMM scheme for separated handwritten digit recognition

As we have tested the model on the MNIST separated digit dataset, which is size normalized to 28×28, no kind of normalization has been performed for the input images. The number of states for the 10 models considered on this experiment are established based on trial runs and the state estimation procedure described in Chapter 4. We have found as many author did : 14 states corresponds to all the models without considering their shape. This presumption can hold as the digit in this dataset were centered and normalized. The order of the NSHP-HMM is fixed. We have considered as in case of words a 3^{rd} order model which is a compromise between the information quantity and algorithm complexity.

6.2.4 Experiments and results

In this section we will show the results of the NSHP-HMM on the MNIST dataset and in the mean time a comparative table will be given in order to get an idea about the impact of the model in this reduced recognition area of the digit recognition considering $2D$ HMM based models. In the Table 6.8 we can observe the recognition accuracy of the system through the different iterations. We can observe than in case of the NSHP-HMM the training procedure is

much more faster than in the case of the neural models where the training samples should be presented many times. This can be considered as a main advantage of the Hidden Markov Model.

Training iteration	**Training set**	**Test set**
1	96.54%	90.64%
5	97.25%	95.90%
10	97.81%	96.03%
15	98.21%	96.30%
20	98.52%	**96,45%**

TABLE 6.8 – The NSHP-HMM results on MNIST dataset considering the training and the test set

In Table 6.9 a detailed class by class result is given.

Class	**Recognition accuracy**	**Class**	**Recognition accuracy**
0	97.96%	1	99.03%
2	96.32%	3	97.62%
4	97.56%	5	96.52%
6	96.45%	7	95.62%
8	92.92%	9	94.15%

TABLE 6.9 – Recognition scores by class on the MNIST dataset

The confusion matrix for the best result can be observed in Table 6.10.

In Table 6.11 some other results are reported in order to measure the importance of the model. Besides the accuracy we also listed the recognition time which is also an important factor.

6.2.5 Conclusions

In this section we have presented different stochastic methods based on Bayesian network, HMMs and NSHP-HMM in order to show the ability of such models in handwritten digit recognition too. Despite of their success in handwritten word recognition, we have demonstrated they are powerful tools for digit recognition too. The weaker results obtained by the HMM based approaches can be explained with the fact than in digit recognition the temporal aspect is minuscule as the digit images are not huge, so the information repartition in the different states is minimal. Such drawback cannot be observed in case of the handwritten words, where the succession of the letters in the word is used and exploited by the model.

Comparing the NSHP-HMM results with the recent results published by other scientists using Markov model based approaches, we can conclude that our method is one of the best one in the

Confusion	0	1	2	3	4	5	6	7	8	9
0	961	0	0	0	5	2	2	2	8	0
1	1	1124	5	0	0	2	0	1	3	0
2	8	3	992	7	3	0	3	10	6	0
3	1	0	3	988	1	5	0	6	5	1
4	0	9	0	0	956	0	2	2	3	10
5	2	1	1	17	0	864	2	1	4	0
6	6	9	0	0	5	13	922	0	3	0
7	0	6	15	0	7	0	0	984	1	15
8	29	0	6	11	5	5	1	1	907	9
9	8	7	8	6	8	1	0	11	13	947

TABLE 6.10 – The corresponding confusion matrix

Ref.	**Method**	**Dataset**	**Recognition (digit/sec)**	**Accuracy**
[Sao97]	NSHP-HMM	UNIPEN	-	90.00%
[HLS04]	BN+HMM	MNIST	0.13	94.42%
[Che04]	2D HMM	MNIST	3	97.10%
[CGPL05]	2D HMM	MNIST	3	97.68%
Personal (thesis)	NSHP-HMM	MNIST	**25**	**96.42%**

TABLE 6.11 – Comparative results on MNIST using recent HMM based approaches

field. Considering the time complexity of the different algorithms, our method outperforms all the possible methods. While in case of ReadNet we can recognize 150 digit/sec, in this case just 25 digit/sec can be recognized but this time factor is due to the complexity of the Viterbi algorithm and the 4 flips calculated for the final recognition.

6.3 Classifiers combination in a digit recognition framework

In the recognition of handwritten characters and words, there has been a recent movement towards combining the decision of several classifiers in order to arrive at improved recognition results. This is due to a number of reasons. Among these are the demands imposed by real-life applications and the availability of a wide variety of algorithms. Practical applications demands highly reliable classification, which is extremely difficult for a particular algorithm to achieve. Since many algorithm are available for these tasks, it is logical to consider the use of several classifiers to achieve higher reliability.

The combination can be implemented using different strategies. In [KSC97] the combined decision is obtained by majority vote of the individual classifiers and different variations of the method, so called majority voting have been implemented in [BVM+04] with success. When the individual classifiers output ranked list of decisions, these rankings can be used to derive combined decisions by the highest rank, Borda count, and logistic regression methods [HHS94]. From these ranked lists, a nonparametric procedure can be used to combine the classification results and a measure of confidence assigned to that decision. Further developments in obtaining a combined decision include statistical approaches, formulations based on Bayesian and Dempster-Shafer theories of evidence and neural networks [LS93]. Other authors use polynomial classifiers to combine the results of multiple classifiers, using the output of the individual classifiers as features. In all these cases, it was found that using a combination of classifiers can result in remarkable improvements in the recognition performances. This is true regardless of whether the classifiers are independent or not [LHS97].

In general, the methods of combining multiple classifiers decisions depend mainly on the type of information produced by the individual classifiers. We consider hereinafter different combination methods that can be applied at both the abstract level and measurement level. In former case, each classifier outputs a unique label or class for each input pattern, while in the latter instance, the classifier produces a measurement value for each label.

In the following sections we will discuss different combination schemes, followed by some experimental results on separated digit recognition.

6.3.1 Combination rules

In the case of several classifiers, the combination of D different classifiers denoted $e_k, k \in \{1, .., D\}$ is defined as E. Each classifier assigns to a pattern x a decision j_k denoted by $e_k(x) = j_k$. The final solution j for the sample x is given by E. Let $v_i^{(k)}(x)$ be the real value computed by the classifier number k for the sample x and the class C_i. This value can represent a probability, a confidence value. It means the degree of membership to one class. In this work we will only discuss about a particular architecture : the horizontal combination scheme. It corresponds to a topology where classifiers are performed in parallel. The classifiers work independently and concurrently and a fusion module combines their results.

Combination strategies

The outputs of each classifier can be combined by simple rules. These rules merge the outputs value of all the classifier for one class.

- Selection of the maximum result : $\forall i \in \{1, .., N\}, v'_i(x) = max_{k=1,..,D} v_i^k(x)$
- Sum of the results : $\forall i \in \{1, .., N\}, v'_i(x) = \sum_{k=1}^{k=D} v_i^k(x)$
- Median of the results : $\forall i \in \{1, .., N\}, v'_i(x) = median_{(k=1,..,D)} v_i^k(x)$

$$E(x) = \begin{cases} i & if\ v'_i(x) = max_{k=1,..,D} v_j^{',k}(x)\ \ and\ v'_i(x) \geq \alpha \\ N+1 & otherwise \end{cases} \tag{6.1}$$

Where $\alpha \in [0; 1]$ is a threshold value. These methods allow to merge the results of each classifier but none of them extract knowledge concerning each classifier strength.

Majority Voting

The majority voting is an easy method to implement and it has shown good results in the literature [Alp94, LS97]. For a multi-classifier system E, the majority voting can be expressed as follows :

$$(E(x) = i) \Leftrightarrow (|\{k \in \{1..D\}, e_k(x) = i\}| \geq ((D/2) + d)), 1 \leq d \leq (D/2) \tag{6.2}$$

If $d = D + 2$ then the voting corresponds to a consensus : all the classifier agree to the same solution.

Behavior Knowledge Space

A behavior knowledge space is a D-dimensional space, each dimension corresponding to the decision of one classifier [HS95]. Each classifier has as decision values the total number of classes N. Let $x \in C_i$ be the character to be recognized belonging to the class C_i. Let $s_k = j_k, k = 1..D$

be the k^{st} classifier among D and j_k its answer for the current character x. The probability that $x \in C_i$ is defined by the following formula :

$$Belief(C_i) = \frac{P(s_1(x) = j_1, .., s_D(x) = j_D, x \in C_i)}{P(s_1(x) = j_1, .., s_D(x) = j_D)} \tag{6.3}$$

A cell of the BKS corresponds to the intersection of the individual classifiers decisions. Each point of the BKS is noted by $BKS(j_1, .., j_D), j_i = 1..N$; and contains a vector of size N : $bks(j_1, .., j_D)(i), i = 1..N$.

Let $bks(j_1, ..j_D)(i)$ be the total number of characters x such that $s_1(x) = j_1, .., s_D(x) = j_D$ and $x \in C_i, i = 1..N$. Let $T(j_1, .., j_D)$ be the total number of characters x such that $s_1(x) = j_1, .., s_D(x) = j_D$. The best representative class of $BKS(j_1, .., j_D)$: R is defined by :

$$R = argmax(bks(j_1, .., j_D)(i)), i = 1..N \tag{6.4}$$

If one cell of the BKS is empty then the pattern is naturally rejected. A small database could be a problem to obtain a good generalization. Many empty cells may occur if the database is not representative. As the BKS size increases exponentially with the number of classifiers, the data sets has to increase in the same way [RR03]. For BKS cells where the most representative class is defined by a low probability, meaning ambiguous cases, characters are rejected. R is rejected if $Belief(C_i) \leq \alpha$ where α is a threshold representing the desired recognition quality.

6.3.2 Experiments and results

The system has been tested on the MNIST database described in details above.

The first objective is to show the behavior of the different combination methods for these classifiers. Let $NSHP_1$, $NSHP_2$, $NSHP_3$, $NSHP_4$ be the 4 flip of the NSHP-HMM described in detail in Section 6.2.3. Let NN_1 be the neural network with the fully connected topology. NN_2 and NN_3 are convolutional neural networks described in details by Cecotti and Vajda in [CVB05a]. The $NN3$ neural network has been trained with both the initial MNIST database and the MNIST database with inverted colors.

TABLE 6.12 – Recognition rate for each classifier.

	$NSHP_1$	$NSHP_2$	$NSHP_3$	$NSHP_4$	NN_1	NN_2	NN_3
Train	93.69	95.00	94.42	95.22	99.72	99.71	99.57
Test	93.44	94.91	94.00	95.25	98.54	98.73	98.41

The Table 6.12 shows the results obtained for each classifier for the test database. The different parts of the NSHP-HMM model obtain the lowest recognition rates whereas the different neural networks give the best results. The best classifier in the system is the convolutional neural

network (NN_2). While many classifiers process the images by extracting time-costly features, in our work the considered classifiers are based on pixel level (i.e raw images). The recognition rate of all these classifiers is still low, compared to the actual best results reported in the literature [BdSBJOM05, LBBH01, LF05]. However, some of the top recognition percentages on the MNIST database have been achieved by using different expansion of the initial MNIST training database [SSP03], or using SVM, which still suffers of memory space and computational speed issues for classification [LF05]. In our tests, only the initial training database has been used for all of the 7 classifiers presented.

TABLE 6.13 – Strength of each classifier.

	$NSHP_1$	$NSHP_2$	$NSHP_3$	$NSHP_4$	NN_1	NN_2	NN_3
$NSHP_1$	0	411	304	404	576	587	575
$NSHP_2$	264	0	274	221	436	440	424
$NSHP_3$	248	365	0	378	516	533	520
$NSHP_4$	223	187	253	0	393	414	400
NN_1	66	73	62	64	0	98	85
NN_2	58	58	60	66	79	0	63
NN_3	78	74	79	84	98	95	0

The strength of each classifier is exposed in the Table 6.13. A cell (i, j) of the table corresponds to the number of pattern recognized by the classifier j and not recognized by the classifier i. It exhibits the strength and the weakness of each classifier versus the others in the test database.

Firstly, there is a strong complementarity between the different flips of the NSHP-HMM. Each flip of the NSHP-HMM can contribute with about more than 200 patterns to the other flips. In this case, we have clearly a proof that results must be combined. Moreover, the 4 classifiers extracted from the NSHP-HMM method come from the same method. A little difference between those classifier, even coming from the same algorithm, leads to obtain a high complementarity. Secondly, in spite of the strength of the different neural networks, they can be completed by all the classifiers. The contribution is not as significant as between the the NSHP-HMM flips but they can be combined as they all give different results. These results display that any classifier makes the same mistake as the others. The results can be combined in order to extract their local strengths.

Without searching the forces and different relationships between classifiers, their results can be fused as described in 6.3.1. Classifiers have been clustered in two groups. The first group contains the 4 $NSHP - HMM$ classifiers and the second group is composed of the 3 neural networks. The different fusing method presented have been tested. The triplet (τ_r, τ_s, τ_q) of

TABLE 6.14 – Combination Results.

	Test (all classifiers)	**Test (NSHP-HPP 4 flips)**	**Test (3 NN)**
Consensus	87.21/12.75/0.04	87.89/11.37/0.74	97.09/2.65/0.26
Majority Voting	97.93/0.77/1.30	93.97/4.15/1.88	98.91/0.13/0.96
Oracle	99.89/0.00/0.11	98.61/0.00/1.39	99.68/0.00/0.32
Maximum rule	98.54/0.00/1.46	95.66/0.00/4.34	99.03/0.00/0.97
Sum rule	96.76/0.00/3.24	96.44/0.00/3.56	99.03/0.00/0.97
Median rule	95.66/0.00/4.34	96.09/0.00/3.91	98.96/0.00/1.04
BKS	97.94/1.34/0.72	96.11/0.31/3.58	98.42/0.59/0.99

each voting method is shown in the Table 6.14. Each rows gives for each classifiers cluster the triplet (τ_r, τ_s, τ_q) for one fusing techniques. The oracle method simulates the results that could be obtained with an optimal vote : if one of the classifier finds the good class then this class is selected. It allows estimating limits for the voting methods. In the BKS case with just the 4 $NSHP-HMM$ flips and with just the 3 neural networks, the recognition rate did not increase but the error has decreased. It has though improved the relevance of the global results. The best score is obtained by the maximum rule with the combination of the 3 neural networks : 99.03%.

6.3.3 Conclusions

We have presented the combination of different kinds of classifier for handwritten digits recognition. These classifiers come from two different approaches : a stochastic model $NSHP-HMM$ and a neural network model. They have been combined using different rules. Their strength and weakness have been highlighted. Thanks to the combination, we have obtained good results.

Multi-classifier systems can always improve a recognition system even in a case where the complementarity between classifiers is low. When the ensembles of classifiers may not always directly improve the recognition rate, they can improve the reliability of the results by qualifying the rejection.

6.4 General conclusions

In this chapter we have discussed different statistical and stochastic methods used for separated digit recognition.

Instead of the system propsed in [LBBH01], we have presented a personal contribution in this field by considering a multi-layer perceptron (ReadNet) where the main interest was to learn

all the possible variations of the network. Despite of its architecture, counting 555 hidden units in the hidden layer, the method is fast and the results performed on the MNIST dataset show its interests for such kind of recognition task. The results can even be compared with the state of the art results.

Similarly for Bangla and English digits collected from Indian documents, the results are encouraging. The weaker results obtained for these kind of digits can be explained with the bad quality of the images and the reduced number of training patterns, which is vital for a neural based approach, where the weights are tunned thanks to the different intra-class and inter-class variations of the input patterns.

The proposed FDDLCB algorithm proposed in this chapter allows a quick selection of the most representative patterns from the database and reduce considerably the time factor for such a model without loss of accuracy. The method outperforms the other similar strategies where SVM has been considered in speed as well in number of used patterns for training.

The experiments performed by the NSHP-HMM adapted to digit recognition have shown the interest of such stochastic tool for this purpose. Comparing the results, our result is one of the best one using such kind of Markov model based technique and it outperforms in recognition time the most recent methods published in this field.

Following the new trends in the classification, we have tested different combination schemes in order to get higher recognition scores. Working with different statistical and stochastic methods those allowed us to combine the outputs of the separate classifiers in order to reach 99.03% of good recognition score, wich is a sound achievement in this particular dataset. In the same time we have shown than there is a strong complementary between the different flips of the NSHP-HMM. This complementarity confirms the necessity of the 4 flips considered also for word recognition.

This chapter has shown the superiority of the neural approaches in separated handwritten digits, but we can conclude than the 2D HMM based approaches are also powerful tools. While in case of the neural approaches the recognition is fast, for the Markov model based techniques the time complexity is due to the dynamic programing applied to 2D signals.

Another aspect is the temporal factor which has been considered in case of the HMM based approaches. While in case of digits this aspect cannot be exploited due to the size of the images, this aspect is one of the main advantages of the HMM models against the neural models. Even if in case of Time Delayed Neural Networks the temporal aspect is considered, the horizontal elasticity provided by the NSHP-HMM cannot be overtaken.

7

Conclusion

This thesis has focused its attention toward word recognition and digit recognition in the framework of postal address automation system concerning Indian postal documents.

The research is proposing different improvements of a baseline recognition system, originally designed for small size Latin (French) script based vocabularies. The model and the corresponding program prototype has been implemented and tested with success on an unconstrained handwritten Bangla city names database. The Bangla script in the second most used script in India and its increased number of letters allows a huge variability in writing, raising a challenging issue to recognize such words. The results show that it is possible to recognize writer independent, unconstrained handwritten words in reasonable time using a segmentation-free analytical technique as there is no available segmentation mechanism for such a complex script. These achievements were only possible because of the strategies that have been proposed and developed in this thesis. The work can be considered a pioneering achievement in Bangla script recognition.

Improvements in the recognition accuracy are achieved by inserting in the system perceptual features which combined with the low-level pixel informations leads to a robust recognition system. The high-level features implant mechanism can be considered as a weight in the column observations given more or less importance to a column observed by the NSHP-HMM. The proposed implant mechanism is a generic one as it not depends on the nature of information which should be considered to upgrade the recognition capabilities of the system.

Improvements in the recognition speed are achieved using an original technique using a pruning mechanism in the Viterbi algorithm. As it was not possible to use a level building matching strategy, we preferred to develop a decoding process based on early stop at letter level, based on the letter model and the meta model strategy adopted by us in the NSHP-HMM.

Improvements were also achieved by our neural network designed for digit recognition purposes. The so called ReadNET network and the FDDLCB algorithm used for pattern selection decrease considerably the training process in such a network without loss of accuracy.

The main results using the novel recognition strategy presented in this thesis are the 2 speed factor and about 1.5% improvement in accuracy based on the baseline NSHP-HMM for Latin script. We have used the baseline NSHP-HMM systems described in details by Saon and Choisy respectively in [Sao97, Cho02] as benchmarks for recognition purposes. On a 76-word Bangla vocabulary task the baseline recognition system requires 0,33 seconds to perform the recognition of a single word with recognition rates of about 85.95%. Using the strategies that have been developed both the recognition accuracy and the recognition speed it is possible to achive recognition rates up to 87.52%.

Similarly for the digit recognition, the 14 speed factor achieved by the FDDLCB algorithm and the 99.03% of good recognition score obtained by our combination scheme can be considered as majore realizations in the field.

It is also one of the first system in terms of script as it was already mentioned, there is no available recognition system for handwritten Bangla words recognition. The developed system is considered as an important module in the Indian Postal Automation, developed in straight collaboration between the Indian CVPR Unit of the Indian Statistical Institute headed by Prof. B. B. Chaudhury and the French group READ from the Loria Research Center lead by Prof. A. Belaïd, in the framework of an Indo-French collaboration proposed by IFCPAR.

The recognition accuracies and the recognition times obtained in this thesis may not meet all the throughput requirements of many real-life applications, however, they are very encouraging and hopefully, this work will stimulate other researchers to pursue interesting research into this subject, since Bangla is a complex Asian scripts and many applications will be necessary to read such a kind of handwriting.

7.1 Summary of results

The results in this thesis are based on the test carried out on the SRTP bank check database and the Bangla city name and digit dataset and the well known MNIST separated digit dataset. The SRTP bank check dataset consist is unconstrained handwritten French legal amounts (hand printed, cursive and mixed) distributed in 26 classes.
The Bangla city names dataset consist in unconstrained handwritten Bangla city names distributed in 76 word classes coming from different writers, belonging to different social categories which implies a complex dataset with many intra-class variations.

The experiments were conducted using different subsets of the Bangla lexicon containing 30,40,50,60,76 words, the SRTP bank check amount and the MNIST dataset. In the next few lines we will summarize the main results achieved in this thesis :

– The new system developed for Bangla script recognition can be considered as a robust and reliable one with special consideration to the explicit segmentation mechanism performed

by the NSHP-HMM.

- The middle zone finding algorithm proposed for the Bangla writing gives excellent results using the profiles based approach or the more complex water reservoir based mechanism.
- Even if in our case, the vocabulary opening was not so important (we extended the vocabulary from 26 entries to 76), the system is stable taking into account the recognition accuracy which outperforms the result of the same system for Roman script. This result can be explained by the fact that the Bangla script is much more complex and more pixel based information can be extracted. In the mean time the natural length of the word shape considered in the experiments is much considerable as in case of the bank check amounts.
- It is possible to improve the results of the baseline NSHP-HMM performing recognition on low-level pixel observations by implanting in the system high-level perceptual knowledge derived from ascender and descender information extracted from the shape. The improvement of 1.57% obtained for the SRTP bank check dataset shows the importance of the implant mechanism in the baseline system. The minor improvement (0.4%) reached for the Bangla city name cannot be considered as an important success, but the implant mechanism proved its interest.
- It is possible to improve the recognition performances of the system considering a pruning mechanism in the Viterbi decoding, reducing to half the recognition time.
- Considering a special multi-layer perceptron type neural network called ReadNET, we achieved a 98.59% good recognition score for the MNIST dataset, while for the Bangla digits the score is 94.13%
- Considering the same ReadNET classifier, we can reach 98.36% good recognition score using just 1,960 patterns from the possible 60,000 reachng 14 time factor gain, using the FDDLCB algorithm.
- Considering the multi classifier scheme, with the contribution of different type of neural networks, we reach 99.03% good recognition score on the MNIST data.

In summary, we have developed a new handwriting recognition system which can recognize Roman script with an accuracy of about 86.80% and Bangla script with an accuracy of about 87.52% and processing time about 0,34 second on a conventional computer hardware [4]. It is important to notice here that we did not attempt to modify the baseline system proposed by the former researchers of our group, but to extend the system limits. There was no kind of optimization on the re-estimation process or the training mechanism.

For digit recognition, we have developed two different systems based on different action mechanisms and a combination scheme. Taking into account the results achieved by the ReadNET (98.59%) and the 14 speed factor achieved by the FDDLCBA, we can conclude that we have built a reliable, robust and fast recognition tool. The combination scheme applied allows us to

4. Intel Celeron 1.5GHz with 1024 MB of RAM

position our results (99.03%) among the best results ever obtained.

7.2 Contributions

One of the main contribution of this thesis is in extending the limits of an off-line handwriting recognition system by implanting high level perceptual features in the pixel observations performed by the former model.

So far, the researchers have been concentrated their effort to segment the words into parts as graphemes and after that to perform a dynamic programming based matching for the final hypothesis considering the observations as unidimensional signals. Many of the current handwriting recognition systems consider as possible observations high-level features and low-level features as being separate issues without considering any kind of relation which can link them.

We have brought to attention the importance of the combination of these low-level information and high-level informations in the current framework considering a generic methodology for the implant strategy. Here all the low-level pixel information are considered with their possible perceptual quality enriching the quantity and the quality of the information.

We have demonstrated that by using the methods and strategies proposed by this thesis it is possible to design a reliable and robust handwriting recognition system which can achieve high accuracy and time gain. Particularly, the implant mechanism seems to be very promising approach to improve the recognition accuracy of current systems considering the different type of informations as one, as in human perception.

We have started with a baseline recognition system which performance was limited to small size vocabularies of no more than 26 entries and we ended up with a 76-word recognition system that delivers high recognition performances for a complex script as Bangla is.

We have demonstrated that our pattern selection method is very fast in comparison with the traditional methods and this selection does not affect the recognition accuracy of the system. We can also mention than the HMM tools are powerful classifiers even when there is no sufficient observation as in case of digits.

Finally, we have demonstrated that the different combination schemes for the multi-classifier systems can give better results than the single classifiers on their own, using the complementarity between them.

7.3 Future work

During the development of this thesis, we did not have the opportunity to address all the problems related to this postal address recognition issue due to the time constraints, however some important aspect have been overlooked.

We believe that the performances of the proposed address recognition system may be further improved by developing a number of points such as :

- *Precise feature extraction :* We assumed during the implant mechanism that the perceptual feature extraction module is reliable which is not totally true as the extraction mechanism proposed by us it is not precise and the number of perceptual feature points extracted from the shapes is not sufficient. A diversification of the perceptual feature extraction (for i.e. extracting loops, line segments, cross-points, possible cutting points) allows to improve the system by adding quantity and quality in the implant process. Considering a large high-level feature set extractable from the word shape can improve the recognition scores of the system and can also precise the weighting mechanism assigning different kind of importance of the features, based on their descriptive capacity.
- *Baum-Welch re-estimation process modification :* During the re-estimation of the pixel probabilities, just the pixel color and its position is considered without taking into account the nature of the pixel. If the re-estimation process, the nature of the pixel should be also considered for a more precise evaluation. This kind of modification allows to integrate dynamically the perceptual information which right now can be considered a static one.
- *Pruning mechanism improvement :* It is necessary as the results are promising but the complexity of the system is still high due the non-symmetric aspect of the NSHP-HMM model. While in the actual model, the stopping criteria (number of letter) is assigned on trial-runs, a more sophisticated decision rule should be developed based on a training mechanism guiding the system for each word model. Such a training will be based on the training set allowing a better generalization for the system.
- *Final hypothesis selection with context :* While actually the system is based on a *soft max* calculus for the final hypothesis, this process can use the considerable context provided by the pin code recognition module which is much more reliable as the word recognition. Such an information can also reduce the research space of the words as knowing the pin code or just a fragment of it helps to discard some word models during the Viterbi process leading to a speed up process.
- *ReadNet improvement :* Instead of using a fully-connected architecture some convolutional layers should be considered as hidden layers with weight sharing property in order to reduce the number of free-parameters of the system.

Furthermore, another important aspect that may be further investigated is the extension of the system to large vocabulary entries where the flat lexicon representation should be replaced a more sophisticated one as trie or even a complex one the DAWG where the different prefix and suffix part are also considered. Such a representation will raise new scientific challenges as the training in such a data structure where the common parts as well the ligatures are shared throughout a dictionary represents a modern challenge.

A

Databases description

A.1 Modified NIST database

The Modified NIST database contains handwritten separated digits. The database was constructed from the NIST's Special Database 3 and Special database 1 containing binary images of handwritten digits. NIST originally designated SD-3 as being a training set while SD-1 as being their test set. The SD-3 set is much more cleaner and easier to recognize than the SD-1. The difference between the datasets can be explained with the fact than the SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. In order to get something independent and descriptive the two datasets were mixed. SD-1 contains 58.527 images written by 500 writers. In contrast to SD-3, where blocks of data form each writer appeared in sequence, the data in SD-1 is scrambled. Writer identities for SD-1 are available and it was used this information to scramble the writers. The SD-1 was split in two : characters written by the first 250 writers went into the new training set while the remaining 250 writers were used to build the test set. In consequence, each set contains nearly 30.000 digit samples. The new training set was completed with enough examples from SD-3, starting at pattern #0, to make a full set of 60.000 training patterns. Similarly, the test set was completed with SD-3 samples starting at pattern #35.000, to make a full set with 60.000 patterns. In the reality just a subset of this dataset is used. 5.000 patterns from SD-1 and 5.000 patterns form SD-3. This databased is the so-called Modified NIST, or just simply MNIST dataset.

The original black and white images were size normalized to fit in a 20x20 pixel box while preserving their aspect ration. The resulting images contain gray levels as result of anti-aliasing (image interpolation) technique used by the normalization algorithm. Three versions of the database were used. In the first version the images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field. In the second version of the database, the character images were deslanted and

FIGURE A.1 – Digit samples from the MNIST dataset

cropped down to 20x20 pixels images. The deslanting computes the second moments of inertia of the pixels (counting a foreground pixel as 1 and a background pixel as 0), and shares the image by horizontal shifting the lines so that the principal axis is vertical. This version of the database will be refereed as the deslanted database. I the third version of the database (used earlier), the images were reduced to 16x16 pixels.

The regular database (used also in our experiments, see Fig. A.1) containing 60.000 training examples and 10.000 test examples normalized to 20x20 and centered by center of mass in 28x28 fields is available at http ://yann.lecun.com/exdb/mnist/. More details can be found in [LBBH01].

A.2 Bangla digit and city name database

The Indian postal documents come from real life data collected from a post-office (Cossipore Post Office of North Kolkata circle, West Bengale, India). An A4 flatbed scanner (manufactured bu UMAX, model AstraSlim) was used to digitize the postal documents. 7500 postal documents were collected from the post office to realize the experiments. The original images are in gray tone with 300 dpi and stored in TIF (Tagged Information Format) files. A two-stage approach was used to convert them into two-tone (0 and 1) images. In a first stage a pre-binarization was

done using a local window based algorithm in order to get an idea of different regions of interest. On the pre-binarized image, RLSA 5Run Length Smoothing Algorithm) is applied to overcome the limitations of the local binarized method. After component labeling of smoothed image, each component was matched in the original image and the final binarized image is obtained using a histogram based global binarizing algorithm on the components. As sometimes the documents are skewed a Hough-transform was used to deskew them. The digitized image may contain spurious noise pixels and irregularities on the boundary of the character, leading to undesired effects on the system. A smoothing technique due to Chaudhuri and Pal was used to correct the noise. In order order to better characterize the dataset some statistics were made :

- 65.69% of the postal documents contain pin-code box
- 73.59% of people write pin-code within the pin-code box
- 63.8% people write all the digits of the pin-code (irrespective of pin-code box)
- 13.49% writers evendo not mention pin-code on postal documents
- 10.02% touching characters are present in the pin-code numbers
- 5.83% of the documents is printed and the rest is handwritten
- 24.62% of the addresses are written in Bangla, 65.37% in English, and 22.04% address is written in two language scripts (English and local state language)
- the address is started in 87.6% cases at the bottommost, 72.3% at the rightmost and 70.06% at right bottommost position
- among the collected postal documents 13.41% are envelopes, 31.09% postcards and 15.76% inland letters (a kind of letter that can be sent anywhere in India)

The Fig. A.2 shows different city name samples from the Bangla dataset. For the separated digit database 15096 numerals were collected, where 80% of the data is coming from real postal documents while the rest was collected from individual writings of non-postal documents. Among these numerals 8690 (4690 of Bangla and 4000 of English) were selected for training while the remaining 6406 (3197 of Bangla and 3227 English) was used to test. For experiments on English and Bangla individual digits two datasets were collected containing 10677 respectively 11042 numerals. We have considered 5876 (6290) data for training and 4801 (4752) data for testing the (English,Bangla) digit recognition. More details can be found in [RVP+05a].

A.2.1 Statistics concerning the Bangla vocabulary

In order to analyze the different reduction aspects in the Bangla city name dataset, we performed some basic statistics concerning the composition of the vocabulary.

In Fig. A.3 a word distribution has been considered, where the x axis designates the number of letters in the words, while the y axis designates the number of samples in which these number of letter has been encountered in the Bangla city name dataset.

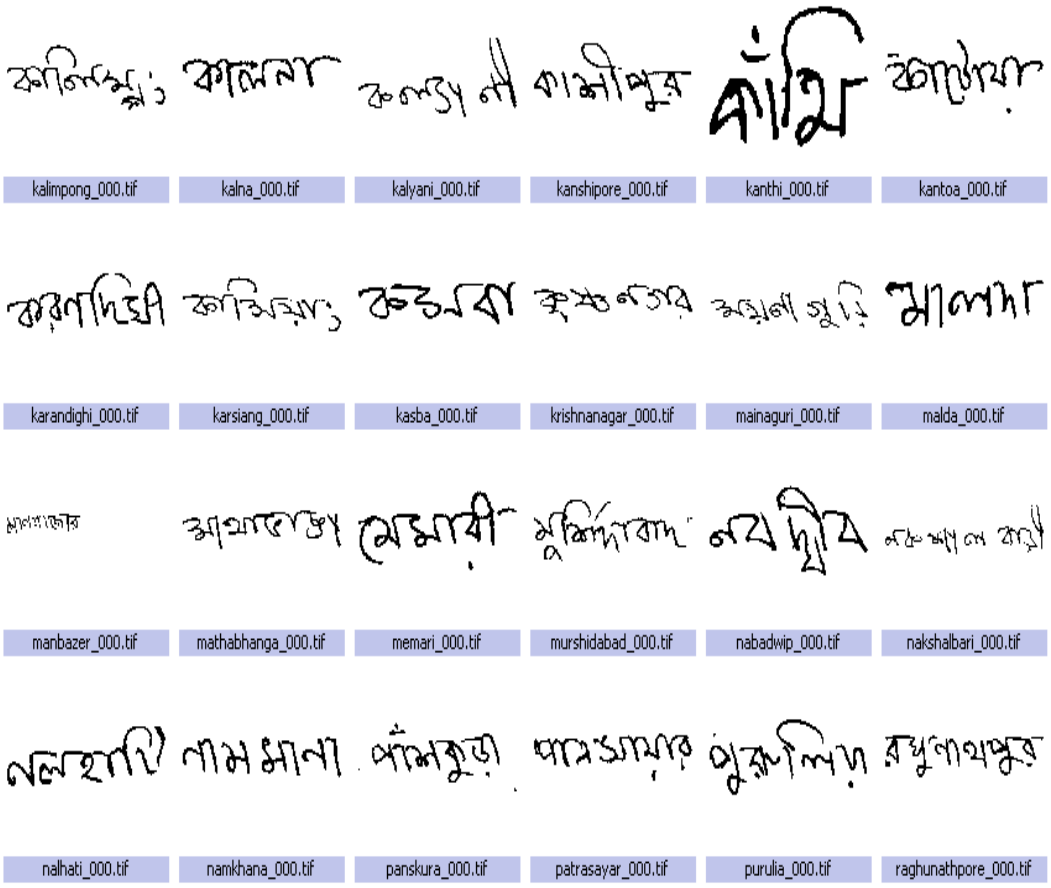

FIGURE A.2 – Some word city name samples for the Bagla city name dataset

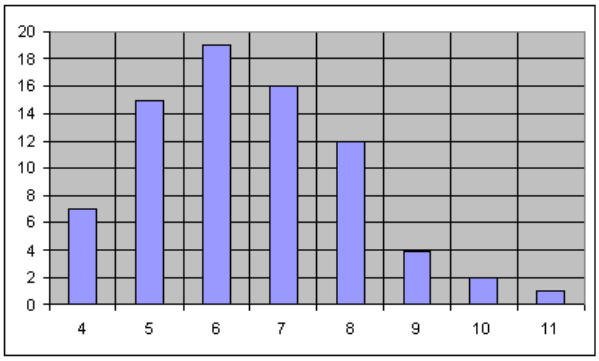

FIGURE A.3 – The distribution of the word entries in the Bangla vocabulary based on the number of letter in the words

The minimal word length is 4 ("kasba", "bagnan", etc.), while the maximum letter length occurs for the word "dimonharber" counting 11 letters. The average letter length is : 6,47.

In summary, considering the word distribution in Fig. A.3 we can pronounce us also in the matter of the natural length estimation. As the natural length estimation algorithm can have an error of $+/-1$ character due the error of the natural length estimation, the major part of the words cannot be separated based on this kind of criteria.

Based on the analysis of the confusion we can declare that the major confusions occur when the number of letters in the word is similar and the word shapes are identical.

A.3 SRTP French bank check database

This dataset is composed by extracted French back check amounts provided by the SRTP [5]. It is composed by 7031 images not unformly distributed in 26 classes as mentioned in the Tab. A.1. The image acguisition has been done at 300 dpi (dot per inch). It is a clean and good quality database but unfortunately containg just a small data amount. However, it can be considered as a benchmark dataset and allows to consider it for comparison purposes with some other systems.

Word class	**un**	**deux**	**troi**	**quatre**	**cing**	**six**	**sept**
Samples No.	28	425	238	519	256	99	104
Word class	**huit**	**neuf**	**dix**	**onze**	**douze**	**treize**	**quatorz**
Samples No.	115	128	239	14	37	18	21
Word class	**quinze**	**seize**	**vingt**	**trente**	**quarante**	**cinquante**	**soixante**
Samples No.	63	16	496	175	126	154	232
Word class	**cent**	**mille**	**francs**	**et**	**centimes**		
Samples No.	1422	230	1726	59	91		

TABLE A.1 – SRTP dataset : The distribution of literal amounts

5. Service de Recherche Technique de la Poste

B

The Bengali script

B.1 Origins

Bangla Script grew out of Kutila, which was a reformed version of Brahmi. Although the Brahmi script is believed to have evolved in the ancient past, its earliest specimens are two inscriptions, dating from the 5th century BC, discovered at Pipraba and Bali. From 350-100 BC the Brahmi script, now known as Ashoka or Maurya script, underwent certain transformations. Asoka script or Maurya script can be divided into two stages : ancient and modern. Ancient Maurya script had two forms : uttari and daksini. Modern script evolved through seven stages.

The second stage in the evolution of the Brahmi script is into the Kushan script, named after the Kushan royal dynasty and in use upto 100-300 AD. The third stage of its evolution was into the Gupta script, named after the Gupta royal dynasty, and current between the 4th and 5th centuries AD. During this period, some letters of the Gupta script took the shape of modern Bangla letters. For instance, in Maharaja Jayanatha's grant, B and M are similar to the Bangla letters today.

The next stage in the evolution of the Brahmi script was into the Kutila script, current between the 6th to 9th centuries. The name perhaps comes from the fact that Kutila letters and vowel symbols are rather complex (Kutila, meaning complicated). Almost all modern scripts of India have grown out of the two main forms of the Kutila script. Devanagari evolved from the west regional form of north-Indian Kutila, while Bangla evolved from its eastern or Magadha form. The transformation of eastern Kutila script began in the 6th century AD. Some time during the reign of the Gurjara kings, most possibly during the reign of Mahendrapala I, son of Bhoja, Kutila script entered Bengal. The copperplate inscriptions of his son Vinayakapala, dating from the 10th century AD, are in the Kutila script. Kutila script evolved further, finally developing into the basic Bangla script towards the end of the 10th century AD. Specimens of this writing are to be found in the Bangad grant of King Mahipala I (980-1036) and the Irdar grant of King

Nayapaladeva (1036-1053).

An improved form of Bangla script is seen in vijayasena's (1098-1160) Deopada inscription. By the end of the 12th century, the script had almost assumed its present form, as may be seen in laksmanasena's Anuliya grant and the Sundarban grant of 1196. The Muslim conquest of Bengal in 1204 AD briefly halted the development of bangla literature and culture, as well as further evolution of the Bangla script. However, under the patronage of the independent sultans, bangla language and literature were revived in the 15th century. Under the influence of Sri chaitanya's vaisnavism, the six Goswamins, 64 Mohantas and many other Vaisnavas wrote innumerable books in sanskrit and Bangla using the Bangla script. In srikrishnakirtan (14th century) and Vodhicharyavatar (15th century), Bangla script had more or less attained its present form.

Between the 16th-18th centuries, some Bangla letters underwent a few insignificant changes. In 1778 Charles Wilkins established the first Bangla printing press at Hughli with letters modelled after the handwritten letters used in old Bangla books of verses. The first Bangla book to be printed was nathaniel brassey halhed's A Grammar of the Bengal Language (1778). Letters made by Wilkins were used for the Bangla text in the book. During the 19th century, numerous printing presses were established, leading to a reduction in the production of manuscript books. Printing ended the further evolution of the Bangla script. As long as books were written by hand, there were variations in the shapes of the letters. The introduction of printing put an end to these variations, and Bangla script assumed its present form. Current technology has provided various fonts for Bangla script, but its basic form remains unaltered.

The Bangla alphabet consists of both vowels and consonants. There are eleven vowels and 39 consonants , making a total of 50 letters. The vowels can be pronounced independently, but the consonants need the support of vowels to be pronounced. Unlike English, Bangla vowels are not always written in full, being replaced by their signs. The vowel A is considered to be part of every consonant if there is no other vowel or vowel sign. However, other vowels are necessary, appearing in their complete forms at the beginning of a word and represented by their signs thereafter

B.2 Notable features

The Bengali alphabet is a syllabic alphabet in which consonants all have an inherent vowel which has two different pronunciations, the choice of which is not always easy to determine and which is sometimes not pronounced at all.

Vowels can be written as independent letters, or by using a variety of diacritical marks which are written above, below, before or after the consonant they belong to.

When consonants occur together in clusters, special conjunct letters are used. The letters for the consonants other than the final one in the group are reduced. The inherent vowel only

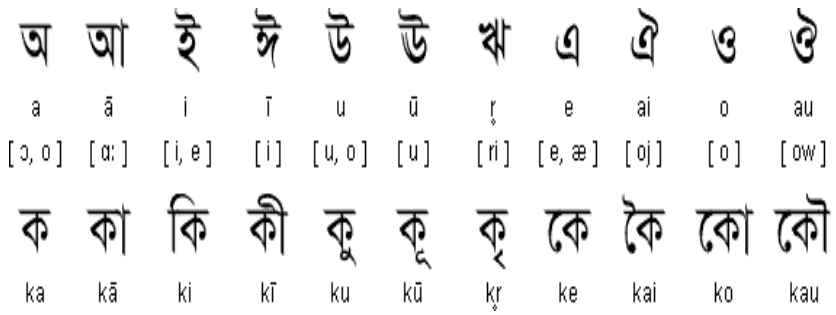

FIGURE B.1 – Bengali vowels and vowel diacritics

applies to the final consonant.

B.3 Used to write

Bengali, is an eastern Indo-Aryan language with around 211 million speakers in Bangladesh, the Indian state of West Bengal and also in Malawi, Nepal, Saudi Arabia, Singapore, Australia, the UAE, UK and USA.

Assamese, is an eastern Indo-Aryan language spoken by about 15 million people in the Indian states of Assam, Meghalaya and Arunachal Pradesh, and also spoken in Bangladesh and Bhutan.

Manipuri, is one of the official languages of the Indian state of Manipur in north-east India and has about 1.1 million speakers. It is a member of the Sino-Tibetan language family. Also has it's own alphabet

Garo, is a Sino-Tibetan language spoken by about 500,000 people in the Brahmaputra valley in the Indian state of Assam.

Mundari, is a Munda language with about 850,000 speakers in eastern India, mainly in the Indian state of Bihar. Also written with the Devanagari, Bengali, Oriya and Roman alphabets.

B.4 The Bengali alphabet

In this section some samples concerning the Bengali alphabet and script are given.
The translation of the text shown in the Fig. B.5 is :

"All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood."

ক	ka	[kɔ]	খ	kha	[kʰɔ]	গ	ga	[gɔ]	ঘ	gha	[gʰɔ]	ঙ	ṅa	[ŋɔ]
চ	ca	[ʧɔ]	ছ	cha	[ʧʰɔ]	জ	ja	[ʤɔ]	ঝ	jha	[ʤʰɔ]	ঞ	ña	[nɔ]
ট	ṭa	[ʈɔ]	ঠ	ṭha	[ʈʰɔ]	ড	ḍa	[ɖɔ]	ঢ	ḍha	[ɖʰɔ]	ণ	ṇa	[ɳɔ]
ত	ta	[t̪ɔ]	থ	tha	[t̪ʰɔ]	দ	da	[d̪ɔ]	ধ	dha	[d̪ʰɔ]	ন	na	[nɔ]
প	pa	[pɔ]	ফ	pha	[pʰɔ]	ব	ba	[bɔ]	ভ	bha	[bʰɔ]	ম	ma	[mɔ]
য	ẏa	[ʤɔ]	র	ra	[rɔ]	ল	la	[lɔ]						
শ	śa	[ʃɔ/sɔ]	ষ	ṣa	[ʃɔ]	স	sa	[ʃɔ/sɔ]	হ	ha	[ɦɔ]			

FIGURE B.2 – Bengali consonants

ক্ক	kka	ক্ট	kṭa	ক্ত	kta	ক্ব	kba	ক্ম	kma	ক্র	kra	ক্ল	kla	ক্ষ	kṣa	ক্ষ্ম	kṣma
ক্স	ksa	গ্ধ	gdha	গ্ন	gna	গ্ব	gba	গ্ম	gma	গ্র	gra	গ্ল	gla	ঘ্ন	ghna	ঙ্ক	ṅka	ঙ্ক্ষ	ṅkṣa
ঙ্খ	ṅtha	ঙ্গ	ṅga	ঙ্ঘ	ṅgha	ঙ্ম	ṅma	চ্ছ	ccha	চ্ছ্ব	cchba	ঞ্চ	cña	জ্জ	jja	জ্জ্ব	jjba
জ্ঝ	jjha	ঞ্ঞ	jña	জ্ব	jba	ঞ্চ	ñca	ঞ্ছ	ñcha	ঞ্জ	ñjha	ট্ট	ṭṭa	ট্ব	ṭba	ণ্ট	ṇṭa
ণ্ঠ	ṇṭha	ণ্ড	ṇḍa	ণ্ণ	ṇṇa	ণ্ম	ṇma	ত্ত	tta	ৎত্ব	ttba	ৎথ	ttha	ত্ন	tna	ত্ব	tba
ত্ম	tma	ত্র	tra	দ্দ	dda	দ্ধ	ddha	দ্ব	dba	দ্ভ্র	dbhra	ন্ট	nṭa	ন্ড	nḍa	ন্ত	nta
ন্ত্ব	ntba	ন্ত্র	ntra	ন্দ	nda	ন্ধ	ndha	ন্ন	nna	ন্ব	nba	ন্স	nsa	প্ট	pṭa	প্ত	pta
প্ন	pna	প্প	ppa	প্ল	pla	প্স	psa	ফ্ল	phla	ভ্র	bhra	ৎল	bhla	ম্ন	mna	ম্ফ	mpha
ম্ব	mba	ম্ল	mla	ল্ট	lṭa	ল্ড	lḍa	ল্ব	lba	ল্ল	lla	শ্চ	shcha	ষ্ক	ṣka	ষ্ট	ṣṭa
ষ্ণ	ṣṇa	স্ক্র	skra	স্ত	sta	স্ত্র	stra	স্ব	sba	হ্ন	hna	হ্ম	hma	হ্ব	hba	হ্ল	hla

FIGURE B.3 – A selection of conjunct consonants in Bengali

০	১	২	৩	৪	৫	৬	৭	৮	৯	১০	
sunna	ek	dui	tin	cār	pāṅc	chay	sāt	āṭ	nay	daś	(Bengali)
	ek	dui	tini	sāri	pās	say	khāt	āth	na	dah	(Assamese)
0	1	2	3	4	5	6	7	8	9	10	

FIGURE B.4 – Bengali numerals

সমস্ত মানুষ স্বাধীনভাবে সমান মর্যাদা এবং অধিকার নিয়ে জন্মগ্রহণ করে। তাঁদের বিবেক এবং বুদ্ধি আছে; সুতরাং সকলেরই একে অপরের প্রতি ভ্রাতৃত্বসুলভ মনোভাব নিয়ে আচরণ করা উচিৎ।

FIGURE B.5 – Article 1 of the Universal Declaration of Human Rights in Bengali

Bibliographie

[AB03] N. E. Ben Amara and F. Bouslama. Classification of arabic script using multiple sources of information : State of the art and perspectives. *International Journal of Document Analysis and Recognition*, 5(4) :195–212, 2003.

[ABE98] N. E. Ben Amara, A. Belaïd, and N. Ellouze. Modélisation pseudo-bidimensionelle pour la reconaissance de chaines de charactéres arabes imprimées. In *CIFED*, pages 131–140, 1998.

[ABPK98] E. Augustin, O. Baret, D. Price, and S. Knerr. Legal amount recognition on french bank checks using a neural etwork-hidden markov model hybrid. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 45–54, 1998.

[ACRS01] N.-E Ayat, M. Cheriet, L. Remaki, and C. Y. Suen. Kmod - a new support vector machine kernel with moderate decreasing for pattern recognition. application to digit image recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 1215–, 2001.

[aKDTDP02] U. Bhattacharya a K. D. Tanmoy, A. Datta, and S. K. Parui. A hybrid scheme for handprinted numeral recognition based on a self-organizing network and mlp classifiers. *IJDAR*, 16(7) :845–864, 2002.

[Alp94] E. Alpayadin. Improved classification accuracy by training multiple models and taking a vote. In *6th Italian Workshop on Neural Nets*, pages 180–185, 1994.

[AM04] H. R. Arabnia and Y. Mun, editors. *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Volume 2 & Proceedings of the International Conference on Machine Learning ; Models, Technologies & Applications, MLMTA '04, June 21-24, 2004, Las Vegas, Nevada, USA*. CSREA Press, 2004.

[AOC+99] S. Adam, J. M. Ogier, C. Cariou, R. Mullot, J. Gardes, and Y. Lecourtier. Multi-scaled and multi oriented character recognition : An original strategy. In *Proc. International Conference on Document Analysis and Recognition*, pages 45–48, 1999.

[AS00] A. S. Atukorale and P. N. Suganthan. Hierarchical overlapped neural gas network with application to pattern classification. *Neurocomputing*, 35(1-4) :165–176, 2000.

[AYV02] N. Arica and F. T. Yarman-Vural. Optical character recognition for cursive handwriting. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 24(6) :801–813, 2002.

[Bal02] M. Balík. Dawg versus suffix array. In *CIAA*, pages 233–238, 2002.

[BBD+93] C. J. C. Burges, J. Ben, J. S. Denker, Y. LeCun, and C. R. Nohl. Off line recognition of handwritten postal words using neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4) :689–704, 1993.

[BdSBJOM05] F. Bortolozzi, A. de S. Britto Jr., L. O. Oliveira, and M. Morita. Recent advances in handwritten recognition. In *1st International Worskshop on Document Analysis*, pages 1–30, 2005.

[Bel96] B. Belkacem. Une application industrielle de reconnaissance d'adreses. In *CIFED*, pages 93–100, 1996.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[BM99] R. Bippus and V. Margner. Script recognition using inhomogeneous p2dhmm and hierarchical search space reduction. In *Proc. International Conference on Document Analysis and Recognition*, pages 773–776, 1999.

[BMF03] M. Bicego, V. Murino, and M&#225 ;rio A. T. Figueiredo. A sequential pruning strategy for the selection of the number of states in hidden markov models. *Pattern Recognition Letters*, 24(9-10) :1395–1407, 2003.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 24(4) :509–522, 2002.

[BR99] C. Becchetti and L. P. Ricotti. *Speech Recognition : Theory and C++ Implementation.* John Wiley & Sons Inc., 1999.

[BRKR00] A. Brakensiek, J. Rottland, A. Kosmala, and G. Rigoll. Off-line handwritingrecognition using various hybrid modeling techniques and character n-grams. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 343–552, 2000.

[BRST95] H. Bunke, M. Roth, and E.G. Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern Recognition*, 28(9) :1399–1413, 1995.

[BS89] R. M. Bozinovic and S. N. Srihari. Off-line cursive script word recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 11(1) :68–83, 1989.

[BS97] C.J.C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector learning machines. volume 9, pages 375–381, Cambridge, MA, 1997. MIT Press.

[BSM99] I. Bazzi, R. M. Schwartz, and J. Makhoul. An omnifont open-vocabulary ocr system for english and arabic. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 21(6) :495–504, 1999.

[BT04] D. Bouchaffra and J. Tang. Introduction to the concept of structural hhm : Application to mining customers' preferences in automotive design. In *Proc. International Conference on Pattern Recognition*, 2004.

[Bun03] H. Bunke. Recognition of cursive roman handwriting - past, present and future. In *Proc. International Conference on Document Analysis and Recognition*, pages 448–, 2003.

[Bur93] J-C. Burbaud. Applied development of advanced techniques in post offices. *Pattern Recognition Letters*, 14(9-10) :259–265, 1993.

[BV97] M. Blumenstein and B. Verma. A segmentation algorithm used in conjunction with artificial neural networks for the recognition of real-world postal addresses, 1997.

[BVM+04] U. Bhattacharya, S. Vajda, A. Mallick, B. B. Chaudhuri, and A. Belaïd. On the choice of training set, architecture and combination rule of multiple mlp classifiers for multiresolution recognition of handwritten characters. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 419–424, Tokyo, Japan, 2004. IEEE Computer Society.

[BW00] Z. Boger and R. Weber. Finding an optimal artificial neural network topology in real-life modeling. In *The ICSC Symposium on Neural Computation*, 2000.

[CA04] S. Carbonnel and E. Anquetil. Lexicon organization and string edit distance learning for lexical post processing in handwriting recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 99–103, 2004.

[CB02] Ch. Choisy and A. Belaïd. Cross-learning in analytic word recognition without segmentation. *International Journal of Document Analysis and Recognition*, 4(4) :281–289, 2002.

[CB03] Christophe Choisy and Abdel Belaïd. Coupling of a local vision by markov field and a global vision by neural network for the recognition of handwritten words. In *ICDAR*, pages 849–853, 2003.

[CCB04] Ch. Choisy, H. Cecotti, and A. Belaïd. Character rotation absorption using a dynamic neural network topology : Comparison with invariant features. In *PRIS*, pages 90–97, 2004.

[CDS90] Y. Le Cun, J. S. Denker, and S. A. Solla. Optimal brain damage. *Advances in Neural Information Processing Systems 2 (NIPS*89)*, 1990.

[CGPL05] S. Chavalier, E. Geoffrois, F. Preteux, and M. Lemaitre. A generic 2d approach of handwriting recognition. *Proc. International Conference on Document Analysis and Recognition*, pages 489–493, 2005.

[CGS99] W.-T. Chen, P. D. Gader, and H. Shi. Lexicon-driven handwritten word recognition using optimal linear combinations of order statistics. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 21(1) :77–82, 1999.

[Che04] S. Chevalier. *Reconnaissance d'écriture manuscrite par des techniques markoviennes : une approche bidimensionelle et générique.* PhD thesis, University René Descartes - Paris 5, 2004.

[Cho97] Ch. Choisy. Utilisation de champs de markov en reconnaissance analytique de l'écriture manuscrite. Master's thesis, University Henri Poincaré - Nancy II, 1997.

[Cho02] Ch. Choisy. *Modèlisation analytique de l'écriture manuscrite par une approache optimale sans segmentation basée sur des champ de Markov.* PhD thesis, University Henri Poincaré - Nancy II, 2002.

[CK94] W. Cho and J. H. Kim. Off-line recognition of cursive words with network of hidden markov models. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 410–417, 1994.

[CK00] W. F. Clocksin and M. S. Khorsheed. Word recognition in arabic handwriting. In *8th International Conference on Artificial Intelligence Application*, volume 1, 2000.

[CK03] S.-J. Cho and J. H. Kim. Bayesian network modeling of hangul characters for on-line handwriting recognition. In *ICDAR*, pages 207–211, 2003.

[CKK00] S. Cho, J. Kim, and J. Kim. Verification of graphemes using neural networks in an hmm based on-line korean handwritting recognition system. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 219–228, 2000.

[CKZ94] M-Y. Chen, A. Kundu, and J. Zhou. Off-line habdwritten word recognition using a hidden markov model type stochastic network. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 16(5) :481–495, 1994.

[CL96] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 18(7) :690–706, 1996.

[CL99] J. Cai and Z.-Q. Liu. Integration of structural and statistical information for unconstrained handwritten numeral recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 21(3) :263–270, 1999.

[CLR95] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* MIT Press, 1995.

[CS00] F. Camastra C. Scagliola, G. Nicchiotti. Enhancing cursive word recognition performance by the integration of all the available information. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 363–372, 2000.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines : and other kernel-based learning methods.* Cambridge University Press, New York, NY, USA, 2000.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3) :273–297, 1995.

[CVB05a] H. Cecotti, S. Vajda, and A. Belaïd. High performance classifiers combination for handwritten digit recognition. In *3rd International Conference on Advances in Handwriting Recognition.*, volume 3686 of *LNCS*, pages 619–626, 2005.

[CVB05b] H. Cecotti, S. Vajda, and A. Belaïd. Hmm based viterbi paths for rejection correction in a convolusional neural network classifiers. *Proc. International Workshop on Neural Networks and Learning in Document Analysis*, pages 23–26, 2005.

[dAFBS01] C. O. de A. Freitas, F. Bortolozzi, and R. Sabourin. Handwritten isolated word recognition : An approach based on mutual information for feature set validation. In *Proc. International Conference on Document Analysis and Recognition*, pages 665–669, 2001.

[DE87] H. Derin and H. Eliott. Modeling and segmentation of noisy and textured images using gibbs random field. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 9(1) :39–55, 1987.

[DFV97] G. Dzuba, A. Filatov, and A. Volgunin. Handwritten zip code recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 766–770, 1997.

[DG00] K. Yamada D. Guillevic, D. Nishiwaki. Word lexicon reduction by character spotting. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 373–383, 2000.

[DK82] P. A. Devijver and J. Kittler. *Pattern Recognition : A statistical approach*. Prentice-Hall, 1982.

[dOJdCdAFS02] J. J. de Oliveira Jr., J. M. de Carvalho, C. O. de A. Freitas, and R. Sabourin. Evaluating nn and hmm classifiers for handwritten word recognition. In *SIBGRAPI*, pages 210–217, 2002.

[DS98] V. Delevski and S. Stankovic. Recognition of handwritten digits based on their topological and morphological properties. *Lecture Notes in Computer Science*, 1451 :516–523, 1998.

[DS02] D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3) :161–190, 2002.

[dSBJSBS01] A. de S. Britto Jr., R. Sabourin, F. Bortolozzi, and C. Y. Suen. An enhanced hmm topology in an lba framework for the recognition of handwritten numeral strings. In *International Conference on Advances in Pattern Recognition*, pages 105–114, 2001.

[dSBJSBS04] A. de S. Britto Jr., R. Sabourin, F. Bortolozzi, and C. Y. Suen. Foregrond and background information in a hmm-based method for recognition of isolated characters and numeral strings. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 371–376, 2004.

[dSBJSL$^+$00] A. de S. Britto Jr., R. Sabourin, E. Lethelier, F. Bortolozzi, and C. Y. Suen. Improvement in a handwritten numeral string recognition by slant normalization and contextual information. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 323–332, 2000.

[EG95] A. Eliaz and D. Geiger. Word-level recognition of small sets of hand-written words. *Pattern Recognition Letters*, 16(10) :999–1009, 1995.

[Eng01] A. P. Engelbrecht. Selective learning for multilayer feedforward neural networks. *Lecture Notes in Computer Science*, 2084 :386–393, 2001.

[EPI98] A. J. Elms, S. Procter, and J. Illingworth. The advantage of using an hmm-based approach for faxed word recognition. *International Journal of Document Analysis and Recognition*, 1(1) :18–36, 1998.

[FGB98] C. Farouz, M. Gilloux, and JM. Bertille. Handwritten word recognition with contextual hidden markov models. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 133–142, 1998.

[FK93a] N. Narasimhamurthi F. Kimura, M. Shridhar. Lexicon driven segmentation - recognition procedure for unconstrained handwritten words. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 121–131, 1993.

[FK93b] Z. Chen F. Kimura, M. Shridhar. Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words. In *Proc. International Conference on Document Analysis and Recognition*, pages 18–22, 1993.

[FNVZ98] A. Filatov, V. Nikitin, A. Volgunin, and P. Zelinsky. The *ddress*script$^{\mathrm{tm}}$ recognition system for handwritten envelopes. In *Document Analysis Systems*, pages 157–171, 1998.

[FYBS00] C. Freitas, A. El Yacoubi, F. Bortolozzi, and R. Sabourin. Isolated word recognition in brazilian bank check legal amounts. In *International Workshop on Document Analysis and Systems*, 2000.

[GAA+01] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximov. Industrial bank check processing : the *a2ia checkreader*^{TM}. *International Journal of Document Analysis and Recognition*, 3(3) :196–206, 2001.

[GB03a] S. Günter and H. Bunke. Fast feature selection in an hmm-based multiple classifier system for handwriting recognition. In *DAGM-Symposium*, pages 289–296, 2003.

[GB03b] S. Günter and H. Bunke. Optimizing the number of states, training iterations and gaussians in an hmm-based handwritten word recognizer. In *Proc. International Conference on Document Analysis and Recognition*, pages 472–476, 2003.

[GB04] S. Günter and H. Bunke. Optimization of weights in a multiple classifier handwritten word recognition system using genetic algorithms. *Electronic Letters on Computer Vision and Image Analysis.*, 31(1) :25–41, 2004.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal Machine Learning Research*, 3 :1157–1182, 2003.

[GFK02] K. Georgila, N. Fakotakis, and G. Kokkinakis. Large vocabulary search space reduction employing directed acyclic word graphs ans phonological rules. *International Journal of Speech Technology*, 5(5) :335–370, 2002.

[Gil93] M. Gilloux. Research into the new generation of character and mailing address recognition systems at the french post office research center. *Pattern Recognition Letters*, 14(9-10) :267–276, 1993.

[Gil94] M. Gilloux. Reconnaissance de chiffres manuscrits par modèles de markov pseudo-2d. In *CIFED*, pages 11–17, 1994.

[Gil00] M. Gilloux. Réduction dynamique du lexique par la méthode tabu. In *CIFED*, pages 319–324, 2000.

[GLL95] M. Gilloux, B. Lemarié, and M. Leroux. A hybrid radial basis function network/hidden markov model handwritten word recognition system. In *Proc. International Conference on Document Analysis and Recognition*, pages 394–397, 1995.

[GM97] G. Lorette G. Menier. Lexical analyzer based on self organizing feature maps. In *Proc. International Conference on Document Analysis and Recognition*, pages 1067–1071, 1997.

[Gos84] A. Goshtasby. Description and discrimination of planar shapes using shapes matrices. *IEEE Transactions of Pattern Recognition and Machine Intelligence*, 7(6) :738–743, 1984.

[GS95] D. Guillevic and C. Y. Suen. Cursive script recognition applied to the processing of bank cheques. In *Proc. International Conference on Document Analysis and Recognition*, pages 11–14, 1995.

[GS98] D. Guillevic and C. Y. Suen. Hmm-knn word recognition engine for bank cheque processing. In *Proc. International Conference on Pattern Recognition*, pages 1526–1529, 1998.

[GS00] F. Grandidier and R. Sabourin. A new strategy for improving feature sets in a discrete hmm-based handwriting recognition system. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 113–122, 2000.

[GSEY+99] F. Grandidier, R. Sabourin, M. A. El-Yacoubi, M. Gilloux, and C. Y. Suen. Influence of word length on handwriting recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 777–780, 1999.

[GSS94] V. Govindaraju, R. Shrihari, and S. Srihari. Handwritten text recognition. In *International Workshop on Document Analysis and Systems*, pages 157–171, 1994.

[GSX02] V. Govindaraju, P. Slavík, and H. Xue. Use of lexicon density in evaluating word recognizers. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 24(6) :789–800, 2002.

[GT03] V. Govindaraju and S. Tulyakov. Postal address block location by contour clustering. In *ICDAR*, pages 429–432, 2003.

[Gui95] D. Guillevic. *Unconstrained handwriting recognition applied to the processing of bank cheques*. PhD thesis, Concordia University, 1995.

[Guy91] I. Guyon. Application of neural networks to character recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1) :353–382, 1991.

[GWGH95] P. Gader, M. Whalen, M. Ganzberger, and D. Hepp. Handprinted word recognition on a NIST data set. *Machine Vision Applications*, 8(1) :31–40, 1995.

[HB93] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(12) :1217–1232, December 1993.

[HF98a] S. Hanafi and A. Freville. An efficient tabu search approach for the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 106(2-3) :663–697, 1998.

[HF98b] M. S. Hoque and M. C. Fairhurst. A moving window classifier for off-line character recognition. In *Proceedings of 7th International Workshop on Frontiers in Handwriting Recognition*, pages 595–600, 1998.

[HHS94] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 16(1) :66–75, 1994.

[HL93] T. Hildebrandt and W. Liu. Optical recognition of handwritten chinese characters : Advances since 1980. *Pattern Recognition*, 26(2), 1993.

[HLS04] K. Hallouli and L. Likforman-Sulem. Reconnaissance de caractéres manuscrits par réseaux bayésiens dynamiques. In *Proc. Colloque International Francophone sur l'Écrit et le Document*, pages 37–41, 2004.

[How87] D. Howard. Reading without letters. In M. Coltheart, G. Sartori, and R. Rob, editors, *The Cognitive Neuropsychology of Language*. 1987.

[HS95] Y.S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 17(1) :90–94, 1995.

[HZB98] T. M. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31(3) :257–272, 1998.

[Jap00] N. Japkowicz. The class imbalance problem : significance and strategies. In *Proceedings of International Conference on Artificial Intelligence 2000 (IC-AI2000)*, 2000.

[JW87] F.-C. Jeng and J. W. Woods. On the relationship of the markov mesh to the nshp markov chain. *Pattern Recognition Letters*, 5(4) :273–279, 1987.

[KA94] S. S. Kuo and O. E. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 16(8) :842–848, 1994.

[KBH97] G. Kaufmann, H. Bunke, and M. Hadorn. Lexicon reduction in a hmm-framework based on quantized feature vectors. In *Proc. International Conference on Document Analysis and Recognition*, 1997.

[KC02] Y. Kundu and M. Chen. Alternatives to a variable duration hmm in handwriting recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 20(1) :842–848, 2002.

[KCGM93] A. Kaltenmeier, T. Caesar, J.M. Gloger, and E. Mandler. Sophisticated topology of hidden markov models for cursive script recognition. In *Proc. International Conference on Document Analysis and Recognition*, 1993.

[KCMT01] S. Kwong, C. W. Chau, K.-F. Man, and K.-S. Tang. Optimisation of hmm topology and its model parameters by genetic algorithms. *Pattern Recognition*, 34(2) :509–522, 2001.

[KFK02] E. Kavallieratou, N. Fakotakis, and G. K. Kokkinakis. An unconstrained handwriting recognition system. *International Journal of Document Analysis and Recognition*, 4(4) :226–242, 2002.

[KFS04] M. N. Kapp, C. Freitas, and R. Sabourin. Handwritten brazilian month recognition : An analysys of two nn architecture and a rejection mechanism. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 209–214, 2004.

[KG82] F. P. Kuhl and C. R. Giardina. Elliptic fourier features of a closed contour. *Computer Vision, Graphics and Image Processing*, 18(3) :236–258, March 1982.

[KG95] G. Kim and V. Govindaraju. Handwritten word recognition for real-time applications. *ICDAR*, 1 :24, 1995.

[KG97] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time application. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 19(4) :366–379, 1997.

[KH04] R. Koggalage and S. Halgamuge. Reducing the number of training samples for support vector machine classification. *Neural Information Processing - Letters and Reviews*, 2(3) :57–65, 2004.

[KHB89] A. Kundu, Y. He, and P. Bahl. Recognition of handwritten word : first and second order hidden markov model based approach. *Pattern Recognition*, 22(3) :283–297, 1989.

[Kho03] M. S. Khorsheed. Recognising handwritten arabic manuscripts using a single hidden markov model. *Pattern Recognition Letters*, 24(14) :2235–2242, 2003.

[KKS00] J. H. Kim, K. K. Kim, and C. Y. Suen. Hybrid schemes of homogeneous and heterogeneous classifiers for cursive word recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 433–442, 2000.

[KKS02] K. K. Kim, J. H. Kim, and C. Y. Suen. Segmentation-based recognition of handwritten touching pairs of digits using structural features. *Pattern Recognition Letters*, 23(1-3) :13–24, 2002.

[KLSS02] A. L. Koerich, Y. Leydier, R. Sabourin, and C. Y. Suen. A hybrid large vocabulary handwritten word recognition system using neural networks with hidden markov models. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 99–103, 2002.

[Koe02] A. L. Koerich. *Large vocabulary off-line handwritten word recognition.* PhD thesis, Ecole de Technologie Supérieur, Université du Québec, 2002.

[Kor97] A. Kornai. Experimental hmm-based postal ocr system. In *Int. Conf. Acoustics, Speech, Signal Processing*, pages 3177–3180, 1997.

[KPH04] G. Koch, T. Paquet, and L. Heutte. Combination of contextual information for handwritten word recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, 2004.

[KSC97] J. Kim, K. Seo, and K. Chung. A systematic approach to classifier selection on combining multiple classifiers for handwritten digit recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 459–462, 1997.

[KSS03] A. L. Koerich, R. Sabourin, and C. Y. Suen. Large vocabulary off–line handwriting recognition : A survey. *Pattern Analysis and Applications*, 6(2) :97–121, 2003.

[KSS04] A. L. Koerich, R. Sabourin, and C. Y. Suen. Fast two-level hmm decoding algorithm for large vocabulary handwriting recogition. In *International Workshop on Frontiers in Handwriting Recognition*, pages 232–238, 2004.

[KSSEY00] A. L. Koerich, R. Sabourin, C. Suen, and A. El-Yacoubi. A syntax-directed level building algorithm for large vocabulary handwritten word recognition. In *International Workshop on Document Analysis and Systems*, 2000.

[KY00] Y. Kato and M. Yasuhara. Recovery of drawing order from single-stroke handwriting images. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 22(9) :938–949, 2000.

[LBBH01] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

[LBK+99] Z. Lu, I. Bazzi, A. Kornai, J. Makhoul, P. Natarajan, and R. Schwartz. A robust, language-independent ocr system. In *27th AIPR Workshop : Advances in Computer-Assisted Recognition*, 1999.

[LDG+00] V. Di Lecce, G. Dimauro, A. Guerriero, G. Pirlo, and A. Salzo. A new hybrid approach for legal amount recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 199–208, 2000.

[LF05] C.-L. Liu and H. Fujisawa. Classification and learning for character recognition : Comparision of methods and remaining problems. *Proc. International Workshop on Neural Networks and Learning in Document Analysis*, pages 1–7, 2005.

[LHS97] L. Lam, Y.S. Huang, and C. Y. Suen. Combination of multiple classifiers decision for optical character recognition. In H. Bunke and P. S. P. Wang, editors, *Handbook on Optical Character Recognition and Document Image Analysis*, chapter 17, pages 79–98. World Scientific, 1997.

[LKK02] J. J. Lee, J. Kim, and J. H. Kim. *Data-driven design of HMM topology for online handwriting recognition.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002.

[LLGL97] M. Leroux, E. Lethelier, M. Gilloux, and B. Lemarié. Automatic reading of handwritten amounts on french checks. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(4) :619–638, 1997.

[LM00] A. Lifchitz and F. Maire. A fast lexically constrained viterbi algorithm for on-line handwriting recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 313–321, 2000.

[LNG00] J. Li, A. Najmi, and R. M. Gray. Image classification based on a multiresolution two dimensional hidden markov model. *IEEE Transactions on Signal Processing*, 48(2) :517–533, 2000.

[Lon98] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8) :983–1001, 1998.

[LP92] E. Levin and R. Pierracini. Dynamic planar warping for optical character recognition. In *ICASSP*, pages 149–152. IEEE, 1992.

[LPD03] S. M. Lucas, G. Patoulas, and A. C. Downton. Fast lexicon-based word recognition in noisy index card images. In *Proc. International Conference on Document Analysis and Recognition*, pages 462–466, 2003.

[LPT91] S.-W. Lee, H.-S. Park, and Y. Y. Tang. Translation-, scale-, and rotation-invariant recognition of hangul characters with ring projection. In *Proc. International Conference on Document Analysis and Recognition*, pages 829–836, 1991.

[LS93] D.-S. Lee and S. N. Srihari. Handprinted digit recognition : a comparison of algorithms. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 153–164, 1993.

[LS95] J. Lii and S. N. Srihari. Location of name and address on fax cover pages. In *ICDAR*, pages 756–759, 1995.

[LS97] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition : an analysis of its behavior and performance. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 27(5) :553–568, 1997.

[MA01] H. Miled and N. E. Ben Amara. Planar markov modeling for arabic writing recognition : Advancement state. In *Proc. International Conference on Document Analysis and Recognition*, pages 69–73, 2001.

[MB00] U.-V. Marti and H. Bunke. Handwritten sentence recognition. In *Proc. International Conference on Pattern Recognition*, pages 3467–3470, 2000.

[McC76] J. L. McClelland. Letter identification in the presentation of words and non-words. *J. Experimental Psychology : Human Perception and Performance*, 2 :80–91, 1976.

[MFW96] S. Manke, M. Finke, and A. Waibel. A fast search technique for large vocabulary on-line handwriting recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 183–188, 1996.

[MG96] M. Mohamed and P. Gader. Handwritten word recognition using segmentatio-free hidden markov modeling and segmentation-based dynamic programming techniques. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 18(5) :548–554, 1996.

[MG01a] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 23(2) :149–164, 2001.

[MG01b] S. Madhvanath and V. Govindaraju. Using holistic features in handwritten word recognition. In *Proceedings of U.S. Postal Service 5th advanced Technology Conference*, pages 149–164, 2001.

[MGR+95] U. Madhvanath, V. Govindaraju, V. Ramanaprasad, D. S. Lee, and S. N. Srihari. Reading handwritten us census forms. In *Proc. International Conference on Document Analysis and Recognition*, pages 82–85, 1995.

[MK97] S. Madhvanath and V. Krpasundar. Pruning large lexicons using generalized word shape descriptors. In *Proc. International Conference on Document Analysis and Recognition*, pages 552–555, 1997.

[MKG01] S. Madhvanath, V. Krpasundar, and V. Govindaraju. Syntactic methodology of pruning large lexicons in cursive script recognition. *Pattern Recognition*, 34(2) :37–46, 2001.

[MS99] U. Madhvanath and S. N. Srihari. Parsing and recognition of city, state and zip codes in handwritten addresses. In *Proc. International Conference on Document Analysis and Recognition*, pages 325–328, 1999.

[MSM98] J. Mao, P. Sinha, and K. Mohiuddin. A system for cursive handwritten address recognition. In *Proc. International Conference on Pattern Recognition*, pages 1285–1287, Washington, DC, USA, 1998. IEEE Computer Society.

[NCA+03] K. Nitz, W. Cruz, H. Aradhye, T. Shaham, and G. Myers. An image-based mail facing and orientation system for enhanced postal automation. In *ICDAR*, page 694, Washington, DC, USA, 2003. IEEE Computer Society.

[NGM05] A. Namane, A. Guessoum, and P. Meyrueis. New holistic handwritten word recognition and its application to french legal amount. In *3rd International Conference on Advances in Handwriting Recognition.*, volume 3686 of *LNCS*, pages 654–663, 2005.

[PAO99] T. Paquet, M. Avila, and C. Olivier. Word modeling for handwritten word recognition. In *Vision Interface*, pages 49–56, 1999.

[Par96] C. Parisse. Global word shape processing in off-line recognition of handwriting. *PAMI*, 18(4) :460–464, April 1996.

[PBC03] U. Pal, A. Belaïd, and Ch. Choisy. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24(1-3) :261–272, 2003.

[PBC06] U. Pal, A. Belaid, and B. B. Chaudhuri. A system for bangla handwritten numeral recognition. In *IETE Journal*, 2006.

[PD03] U. Pal and S. Datta. Segmentation of bangla unconstrained handwritten text. In *Proc. International Conference on Document Analysis and Recognition*, pages 1128–1132, 2003.

[PL95] H-S. Park and S-W. Lee. Hidden markov mesh random field : Theory and its application to handwritten character recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 409–412, 1995.

[PL96] H-S. Park and S-W. Lee. Off-line recognition of large-set handwritten characters with multiple hidden markov models. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 29(2) :231–244, 1996.

[PS00] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition : A comprehensive survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 11(1) :68–89, 2000.

[PSW97] R. K. Powalka, N. Sherkat, and R. J. Whitrow. Word shape analysis for a hybrid recognition system. *Pattern Recognition*, 30(3) :412–445, 1997.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77 :257–286, 1989.

[RB05] Y. Rangoni and A. Belaïd. Data categorization for a context return applied to logical document structure recognition. *Proc. International Conference on Document Analysis and Recognition*, pages 297–301, 2005.

[RBTT95] R. Romero, R. Berger, R. Thibadeau, and D. Touretzky. Neural network classifiers for optical chinese character recognition. *Proceedings of the 4th annual Symposium on Document Analysis and Information Retrieval*, pages 385–389, 1995.

[RF03] A. F. R. Rahman and M. C. Fairhurst. Multiple classifier decision combination strategies for character recognition : A review. *International Journal of Document Analysis and Recognition*, 5(4) :166–194, 2003.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. pages 318–362, 1986.

[RJ93] L.R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* PTR Prentice-Hall, Inc., 1993.

[RL01] N. G. Rodriguez and L. L. Ling. Feature extraction based on fuzzy set theory for handwriting recognition. In *Proc. International Conference on Document Analysis and Recognition*, 2001.

[RPH01] S. Udpa R. Polikar, L. Udpa and V. Honavar. Learn++ : An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man and Cybernetics - Part C : Application and Reviews*, 31(4) :497–508, 2001.

[RR03] S. Raudys and F. Roli. The behavior knowledge space fusion method : Analysis of generalization error and strategies for performance improvement. In *Multiple Classifier Systems*, pages 55–64, 2003.

[RVP+05a] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, and A. Belaïd. A system for indian postal automation. In *International Workshop on Document Analysis*, pages 249–274, Calcutta, India, 2005.

[RVP+05b] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, and A. Belaïd. A system for indian postal automation. In *Proc. International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005.

[RVPC04] K. Roy, S. Vajda, U. Pal, and B.B. Chaudhuri. A system towards indian postal automation. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, Tokyo, Japan, 2004.

[Sao97] G. Saon. *Modèles markoviens uni- et bidimensionel pour la reconnaisance de l'écriture manuscrit hors-ligne*. PhD thesis, University Henri Poincaré - Nancy I, 1997.

[Sao99] G. Saon. Cursive word recognition using a random field based hidden markov model. *International Journal of Document Analysis and Recognition*, 1(4) :199–208, 1999.

[Say73] M. Sayre. Machine recognition of handwritten words. a project report. *Pattern Recognition*, 5 :213–228, 1973.

[SB97] G. Saon and A. Belaïd. High performance unconstrained word recognition system combining hmms and markov random fields. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(5) :771–788, 1997.

[SBG93] J. C. Simon, O. Baret, and N. Gorski. A system for the recognition of handwritten literal amounts of checks. In *Workshop on Document Analysis Systems*, pages 135–155, Kaiserslautern, 1993.

[SBG95] G. Saon, A. Belaïd, and Y. Gong. Stochastic trajectory modeling for recognition of unconstrained handwritten words. In *Proc. International Conference on Document Analysis and Recognition*, pages 508–511, 1995.

[SBM04] Z. Sun, G. Bebis, and R. Miller. Object detection using feature subset selection. *Pattern Recognition*, 37(11) :2165–2176, 2004.

[SBS99] B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in kernel methods : support vector learning*. MIT Press, Cambridge, MA, USA, 1999.

[SC03] H. Shin and S. Cho. Fast pattern selection for support vector classifier. *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNCS*, (2637), 2003.

[Sch78] J. Schurmann. A multifont word recognition system for postal address reading. *IEEE Transaction on Computers*, 27(8) :721–732, 1978.

[Sen94] A. W. Senior. *Off-line Cursive Handwriting Recognition using Recurrent Neural Networks*. PhD thesis, Trinity Hall Cambridge, 1994.

[SGH94] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden markov models. In *Proc. ICASSP '94*, pages 637–640, Adelaide, Austrailia, 1994.

[SHK97] M. Shridhar, G. F. Houle, and F. Kimura. Handwritten word recognition using lexicon free and lexicon directed word. In *Proc. International Conference on Document Analysis and Recognition*, pages 861–865, 1997.

[Sin97] P. Sinha. Cursive script postal address recognition. Master's thesis, Michigan State University, 1997.

[SKTH02] M. Shridhar, F. Kimura, B. Truijen, and G. F. Houle. Impact of lexicon completeness on city name recognition. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 513–518, 2002.

[SKX+00] C. Y. Suen, K. Kim, Q. Xu, J. Kim, and L. Lam. Handwriting recognition - the last frontiers. In *ICPR '00 : Proceedings of the International Conference on Pattern Recognition (ICPR'00)-Volume 4*, page 4001, Washington, DC, USA, 2000. IEEE Computer Society.

[SLD94] P. Simard, Y. LeCun, and J. Denker. Memory based character recognition using a transformation invariant metric. In IAPR, editor, *Proc. of the International Conference on Pattern Recognition*, volume II, pages 262–267, Jerusalem, October 1994. IEEE.

[SLGS02] S. Setlur, A. Lawson, V. Govindaraju, and S. N. Srihari. Large scale address recognition systems truthing, testing, tools, and other evaluation issues. *IJDAR*, 4(3) :154–169, 2002.

[SLS99] C. Y. Suen, K. Liu, and N. W. Strathy. Sorting and recognizing cheques and financial documents. In *DAS '98 : Selected Papers from the Third IAPR Workshop on Document Analysis Systems*, pages 173–187, London, UK, 1999. Springer-Verlag.

[SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Proceeding of 5th Annual ACM Workshop on Computational Learning Theory*, pages 287–299, 1992.

[SR92] L. Spirkovska and M. B. Reid. Robust position, scale and rotation invariant object recognition using higher-order neural networks. *Pattern Recognition*, 25(9) :975–985, 1992.

[Sri00] S. N. Srihari. Handwritten address interpretation : A task of many pattern recognition problems. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(5) :663–674, 2000.

[SSP03] P. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proc. International Conference on Document Analysis and Recognition*, pages 958–962, 2003.

[TAS+04] H. Tang, E. Augustin, C. Y. Suen, O. Baret, and M. Cheriet. Spiral recognition methodology and its application for recognition of chinese bank checks. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 234–240, 2004.

[TJT96] Ø.D. Trier, A.K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4) :641–662, 1996.

[TKL+01] Y. H. Tay, M. Khalid, P.-M. Lallican, S. Knerr, and C. Viard-Gaudin. An analytical handwritten word recognition system with word-level discriminant training. In *Proc. International Conference on Document Analysis and Recognition*, pages 726–730, 2001.

[TL02] L.-N. Teow and K.-F. Loe. Robust vision-based features and classification schemes for off-line handwritten digit recognition. *Pattern Recognition*, 35(11) :2355–2364, 2002.

[TLK+01] Y. Tay, P.-M. Lallican, M. Khalid, C. Viard-Gaudin, and S. Knerr. An offline cursive handwritten word recognition system. In *IEEE Region10 Conference*, 2001.

[TNEBA04] S. M. Touj and H. Amiri N. E. Ben Amara. Modélisation markovienne planaire pour la reconnaissance de l'écriture arabe. In *Proc. Colloque International Francophone sur l'Écrit et le Document*, pages 104–109, 2004.

[Tok93] Y. Tokunaga. History and current status of postal mechanization in japan. *Pattern Recognition Letters*, 14(9-10) :277–280, 1993.

[Tom93] H. Tominaga. Special issue on postal processing and character recognition. *PRL*, 14(4) :257–354, April 1993.

[UTS01] S. Uchida, E. Taira, and H. Sakoe. Nonuniform slant correction using dynamic programming. In *Proc. International Conference on Document Analysis and Recognition*, pages 434–438, 2001.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA, 1995.

[VB05a] S. Vajda and A. Belaïd. How to speed up the learning mechanism in a connectionist model. *Proc. International Workshop on Neural Networks and Learning in Document Analysis*, pages 13–17, 2005.

[VB05b] S. Vajda and A. Belaïd. Structural information implant in a context based segmentation-free hmm handwritten word recognition system for latin and bangla script. *Proc. International Conference on Document Analysis and Recognition*, pages 1126–1130, 2005.

[VdC99] L. R. Veloso and J. M. de Carvalho. Neural versus syntactic recognition of handwritten numerals. In *ICDAR*, pages 233–236, 1999.

[Vin00] A. Vinciarelli. A survey on off-line cursive word recognition. IDIAP-RR 43, IDIAP, 2000.

[Wad33] M. Wada. Proposal for fully automated mail processing system for the 21st century. *Pattern Recognition Letters*, 14(9-10) :281–290, 19933.

[WBKR00] W. Wang, A. Brakensiek, A. Kosmala, and G. Rigoll. Hmm based high accuracy off-line cursive handwriting recognition by a baseline detection error tolerant feature extraction approach. In *Proc. International Workshop on Frontiers in Handwriting Recognition*, pages 209–218, 2000.

[Whe70] D. D. Wheeler. Word recognition processes. *Cognitive Psychology*, 1 :59–85, 1970.

[WLS07] Y. Wen, Y. Lu, and P. Shi. Handwritten bangla numeral recognition system and its application to postal automation. *Pattern Recognition*, 40(1) :99–107, 2007.

[WNC05] J. Wang, P. Neskovic, and L. N. Cooper. Training data selection for support vector machines. *International Conference on Neural Computation*, 2005.

[WP93] L. Wang and T. Pavlidis. Direct gray-scale extraction of features for character recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 15(10) :1053–1067, 1993.

[WP94] R. Wolf and J. C. Platt. Postal address block location using a convolusional locator network. *Advances in Neural Information Processing*, (6), 1994.

[XG01] H. Xue and V. Govindaraju. Building skeletal graphs for structural feature extraction on handwriting images. In *Proc. International Conference on Document Analysis and Recognition*, pages 96–100, 2001.

[YBG95] A. El Yacoubi, J.-M. Bertille, and M. Gilloux. Conjoined location and recognition of street names within a postal address delivery line. In *ICDAR*, page 1024, Washington, DC, USA, 1995. IEEE Computer Society.

[YGSS99] A. El Yacoubi, M. Gilloux, R. Sabourin, and C.Y. Suen. An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 21(8) :752–760, August 1999.

[YH98] J. Yang and V. Honavar. Feature subset selection using genetic algorithms. *IEEE Transactions on Intelligent Systems*, (3-4) :45–49, 1998.

[YJO+95] S. J. Young, J. Jansen, J. J. Odell, D. Olasson, and P. C. Woodland, editors. *The HTK Hidden Markov Model Toolkit Book*. Entropic Cambridge Research Laboratory, 1995.

[ZBS05] P. Zhang, T. D. Bui, and C. Y. Suen. Hybrid feature extraction and feature selection for improving recognition accuracy of handwritten numerals. In *ICDAR '05 : Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 136–140, Washington, DC, USA, 2005. IEEE Computer Society.

[ZM99] M. Zimmermann and J. Mao. Lexicon reduction using key characters in cursive handwritten words. *Pattern Recognition Letters*, 20(11-13) :1297–1304, 1999.

Abstract

Large variations in writing styles and difficulties in segmenting cursive words are the main reasons for handwritten cursive words recognition for being such a challenging task. An Indian postal document reading system based on a segmentation-free context based stochastic model is presented. The originality of the work resides on a combination of high-level perceptual features with the low-level pixel information considered by the former model and a pruning strategy in the Viterbi decoding to reduce the recognition time.

While the low-level information can be easily extracted from the analyzed form, the discriminative power of such information has some limits as describes the shape with less precision. For that reason, we have considered in the framework of an analytical approach, using an implicit segmentation, the implant of high-level information reduced to a lower level. This enrichment can be perceived as a weight at pixel level, assigning an importance to each analyzed pixel based on their perceptual properties. The challenge is to combine the different type of features considering a certain dependence between them.

To reduce the decoding time in the Viterbi search, a cumulative threshold mechanism is proposed in a flat lexicon representation. Instead of using a trie representation where the common prefix parts are shared we propose a threshold mechanism in the flat lexicon where based just on a partial Viterbi analysis, we can prune a model and stop the further processing. The cumulative thresholds are based on matching scores calculated at each letter level, allowing a certain dynamic and elasticity to the model.

As we are interested in a complete postal address recognition system, we have also focused our attention on digit recognition, proposing different neural and stochastic solutions. To increase the accuracy and robustness of the classifiers a combination scheme is also proposed.

The results obtained on different datasets written on Latin and Bengali scripts have shown the interest of the method and the recognition module developed will be integrated in a generic system for the Indian postal automation.

Keywords: postal address automation, handwritten word recognition, features combination, lexicon reduction, flat lexicon, HMM, Bengali script, Roman script