OPTICAL CHARACTER RECOGNITION OF PRINTED ARABIC TEXT (OCRA)

Ahmed Abdelfattah Hassan, Ali Mohamed Mostafa Abbas, Eslam Mohamed Abdelreheem
Department of Computer Engineering
Cairo University, Egypt
Email: a.abfattah@gmail.com
Email: islam_eng47@yahoo.com
Email: eng_3li@ymail.com

INTRODUCTION

Motivation

Since the invention of writing it has been the tool that enabled humanity to preserve their history and transfer their knowledge. In our modern world electronic storage means are gradually replacing papers, hence comes the important of OCR that translates the knowledge from human-domain to computer-domain.

Although OCR is a very important topic and has wide range of applications (e.g.: Conversion of old archives into digital format, generating searchable pdfs …etc.) , there are few OCR software tools which support Arabic : ABBYY[1], IRIS[2] and SAKHR [3]are examples of commercial solutions and Tesseract[4] is a FOSS (Free and Open Source Software) that added support to Arabic language since v3.0+ it doesn't support Arabic natively .

We propose implementing an Arabic OCR algorithm that's proven to achieve accurate recognition results and release it as free and open source software for the aid of Arabic researchers and users.

Justification

- OCR is widely used as a data entry method from original source paper.
- fIt is commonly used to digitize printed text to make it more compact, searchable, editable and usable by the computer.

Problem Definition

OCR is a computer vision research field that involves pattern recognition and artificial intelligence, its aim is to extract characters and text from images enabling computers to do further processing on the information stored in papers.

Challenges [11]

1. The connectivity and cursively:

Graphemes are connected to one another within the same word with this connection interrupted a few certain characters or at the end of the word. This necessitates any Arabic OCR to do not only the separate graphemes recognition task, but also another may be tougher graphemes segmentation. To make things even harder, both of these tasks are mutually dependent and must hence be done simultaneously.

2. The dotting:

Dotting is extensively used to differentiate characters sharing similar graphemes. It is apparent that the digital differences between the members of the same set are small. Whether the dots are eliminated

[1] *http://finereader.abbyy.com/about_ocr/whatis_ocr/*
[2] *http://www.irislink.com/c2-2888-189/I-R-I-S----OCR-Technology-and-Document-Management-Solutions.aspx*
[3] *http://www.sakhr.com/index.php/en/solutions/ocr*
[4] *https://code.google.com/p/tesseract-ocr/*

before the recognition process, or recognition features are extracted from the dotted script, dotting is a significant source of confusion – hence recognition errors – in Arabic OCR systems especially when run on noisy documents; e.g. those reproduced by photocopiers.

3. The multiple grapheme cases:

Due to the mandatory connectivity in Arabic orthography; the same grapheme representing the same character can have multiple variants according to its relative position within the Arabic word segment {Starting, Middle, Ending, Separate}

4. The ligatures:

To make things even more complex, certain compounds of characters at certain positions of the Arabic word segments are represented by single atomic graphemes called ligatures. Ligatures are found in almost all the Arabic fonts, but their number depends on the involvement of the specific font in use.

5. The diacritics:

Arabic diacritics are used in practice only when they help in resolving linguistic ambiguity of the text. The problem of diacritics with Arabic OCR is that their direction of flow is vertical while the main writing direction of the body Arabic text is horizontal from right to left. Like dots; diacritics – when existent - are a source of confusion of font written OCR systems especially when run on noisy documents, but due to their relatively larger size they are usually preprocessed.

6. The size variation:

Different Arabic graphemes do not have a fixed height nor a fixed width. Moreover, neither the different nominal sizes of the same font scale linearly with their actual line heights, nor the different fonts with the same nominal size have a fixed line height.

Benchmarking and APTI Database

An important issue that faces Arabic Recognition systems is the performance analysis. Though these systems have shown improvements, still most of them were benchmarked against a small and a private database which makes the comparison process between different system cumbersome and unfair. A quality document corpus would typically consist of a large set of document images accompanied with their *ground truth*. The ground truth includes the document's source text and other related information [1].

SAIC collected the DARPA[2] document corpus which consisted of 297 images scanned from different sources. Fouad Slimane *et. al* [3] initiated an effort to collect a large database of Arabic Printed Text (APTI) which aims to large-scale benchmarking of open-vocabulary, multi-font, multi-size and multi-style text recognition systems in Arabic. APTI was recently used by research groups in the Arabic Recognition Competition: Multi-font Multi-size Digitally Represented Text held in the context of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)[4], APTI is mainly characterized by : very large set of images (> 45 million images), large lexicon of 113’248 words, multi-font, multi-size and single word images.

RELATED WORK

Arabic Recognition Systems Classification

Recognition of printed Arabic text has been an important ongoing research topic since the 1980's [5]. Many approaches have been adopted to tackle this problem achieving various results, yet there is no universally accepted technique that achieves best results.

These approaches can be generally classified into segmentation-based(analytical) and segmentation-free(global). In segmentation-based approaches, text should be partitioned into characters in a separate phase and the output is fed to the classifiers to perform the recognition, but in case of noisy scripts their performance degrades [6, 7] while in segmentation-free approaches no partitioning work is required on the text. In [6] another class was introduced: implicit segmentation approaches, where in them characters are segmented during recognition.

Also, OCR approaches can be classified into online and offline systems. An online system recognizes systems as they are drawn while offline recognition is performed on data after it has been written.

State of The Art

Hidden Markov Models (HMM) approaches [7, 8, 9, 10, 11, and 12] are widely adopted by many researches in the field. In the second place comes ANN-based approaches [13,14 and 15], other approaches like graph-based segmentation [6] and Generalized Hough Transform [8] but they are being investigated less frequently.

HMM

HMM were originally applied to speech recognition systems. Many tools were released to aid the researchers like HMM Toolkit (HTK) and since they achieved success, researchers have used them for both Printed and Handwritten Arabic Text recognition due to its advantages: no need for segmentation, robust against variations, immune to noise and its tools are freely available.
Khorsheed and Alfaifi proposed IPSAR system [7] which doesn't require segmentation of characters but instead extracts a set of features extracted from overlapping vertical windows along the line image, then clustered into discrete symbols. Alkhoury *et al.* presented a system based on a variant of HMM, Bernoulli HMMs (BHMMs), that is, HMMs in which conventional Gaussian mixture density functions are replaced with Bernoulli mixture probability functions [9].
Al-Muhtaseb *et al.* [10] made their HMM-OCR system based on Novel hierarchical sliding window technique (this technique was implemented to extract text features). They represent each sliding strip by 16 features from one type of simple features for each sliding window, while 80 features of four types of features are used. Their technique considers each shape of an Arabic character as a separate class (not combining multiple shapes in one class as is done by other researchers). The number of classes thus becomes 126 compared with 40 classes as usual, if all the shapes of a character are considered as separate classes. Rashwan *et al.* [11] drew an analogy between ASR and OCR proposing the approach shown in fig. 1.

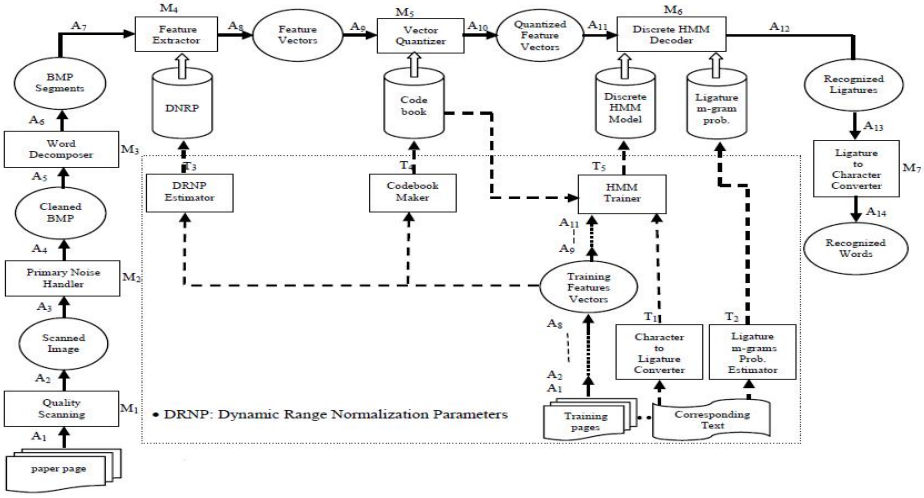

Fig.1 HMM-based OCR analog to HMM-based ASR

HMM-based systems can also be designed to be language-independent [12]; the key to this is extracting features from thin slices of the image rather than relying on segmentation of characters.

ANN

Nawaz *et al* proposed a segmentation-based system in [13] that relies on RBF ANN network for recognition. Text is preprocessed, segmented into individual characters then passed to the classifier.
Amin and Murshed proposed another variant in [14] that avoids segmentation; global features (like number of subwords, number of peaks within them and position of the complementary character, etc.) are fed to a Fuzzy ARTMAP neural network.
A novel approached was introduced by Rashid *et al*[15] which combined ANNs and HMMs, a trained multi-layer perceptron network is used to scan the text and extract features and trained HMM is used to classify these features.

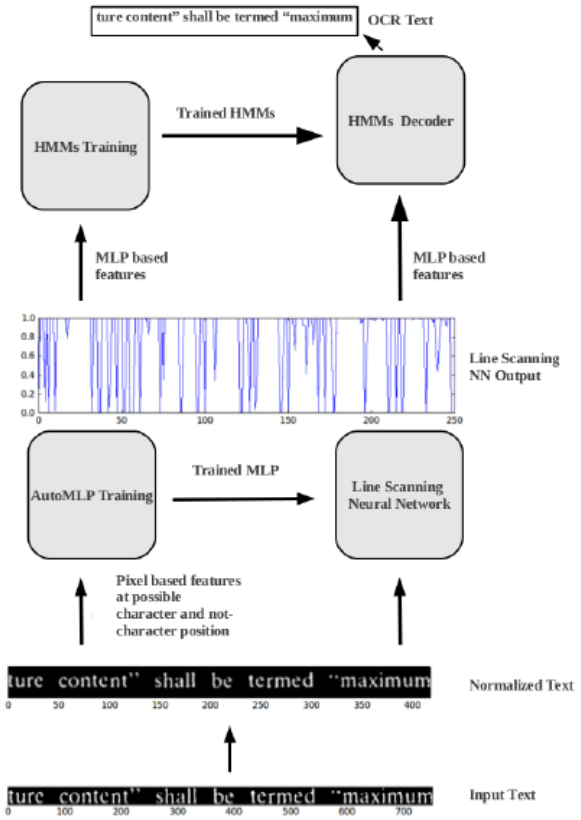

Fig.2 Line Scanning Neural Network Architecture

Others

Another paper proposed an approach for Arabic Character recognition based on the use of a Generalized Hough Transform [8].The GHT can extract and localize characters from any target image. It can detect objects in different scales and orientations which may resolve many problems related to the recognition of Arabic printed document without any constraints.

The major contribution of [6] is a new graph-based structural segmentation approach based on the topological relation between the baseline and the line adjacency graph (LAG) representation of the text. The graph representing the text is used to extract structural shape features such as strokes, loops and feature points that are used in the recognition. Two different classifiers are used. A classifier for the scripts and a classifier for the dots and diacritic signs.The results of the two classifiers are combined together using some linguistic rules and the final recognition result are obtained using a regular grammar describing the formation of the characters from the basic scripts.

Results Comparison

Table 1 illustrates a table of results obtained by the aforementioned approaches.

Description	Technique	Database	Results
ARABIC OCR SYSTEM ANALOGOUS TO HMM-BASED ASR SYSTEMS	HMM	540 pages	WER of 0.18% for Mudir font
Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)	HMM	600 pages	92.4% for Andalus font
Recognition of Off-line Printed Arabic Text using HMM	HMM	2788 lines from Saheh Al-Bukhari and Saheh Muslem	average of 99%
Generalized Hough Transform for Arabic Optical Character Recognition	Hough Transform	166 873 samples of characters in Arabic Transparent font	average of 93%
A Graph-Based Segmentation and Feature Extraction Framework for Arabic Text Recognition	Graph based	31 + 15 pages from Arabic magazines in Naskh font	94.80%
IPSAR System (Same as 2 but on different db)	HMM	APTI Database	WRR of 65.3 , CRR of 89.7
UPV-PRHLT-REC1	BHMM	APTI Database	WRR of 91.7 , CRR of 98.3
UPV-PRHLT-REC2	BHMM	APTI Database	WRR of 91.7 , CRR of 98.3
Recognition of printed Arabic words with fuzzy ARTMAP neural network	ANN	217 words with different fonts (each word has 15 samples)	a mean of 95.25%
A New Recognition Scheme for Machine-Printed Arabic Texts based on Neural Networks	ANN	100 Arabic text images	98%
An approach to offline Arabic character recognition using neural networks	ANN	?	76%
Scanning Neural Network for Text Line Recognition	ANN	1060 text lines, having 51,261 characters	98.40%
Segmentation-Free Word Recognition with Application to Arabic	Morphological	42000 words	94% for noise free , 73% for scanned

Table1. A comparison of different Arabic OCR approaches

PROPOSED APPROACH

Due to the aforementioned advantages of Hidden Markov Models (HMM) many researchers have used them for Arabic text recognition (there is no need for segmentation phase, immune to noise, robust against variations, and the HMM tools are freely available), some researchers used HMM for handwriting word recognition [16, 17, 18, and 19] while others used it for text recognition [7, 8, 9, 10, 11, and 12]. Research work on Arabic text recognition using HMM is going solid, the results are getting better (check Table.1) and it's gaining wider acknowledgment by researchers in the field.

Hence, we decided to adopt the HMM approach. Originally, HMMs were applied to speech recognition problems where a sliding window was applied to the speech signal; features were collected for each window and then were used to train the model. In the same sense, HMM can be used for text recognition with the difference of the sliding window being applied to a text line,

The algorithm can be as following:

- text images should be preprocessed to remove noise and adjust skew… etc. ;
- then features are calculated for each window;
- input-output pairs are then fed to the HMM in order to train the signal;
- After training is complete new inputs can be tested on the HMM for classification.

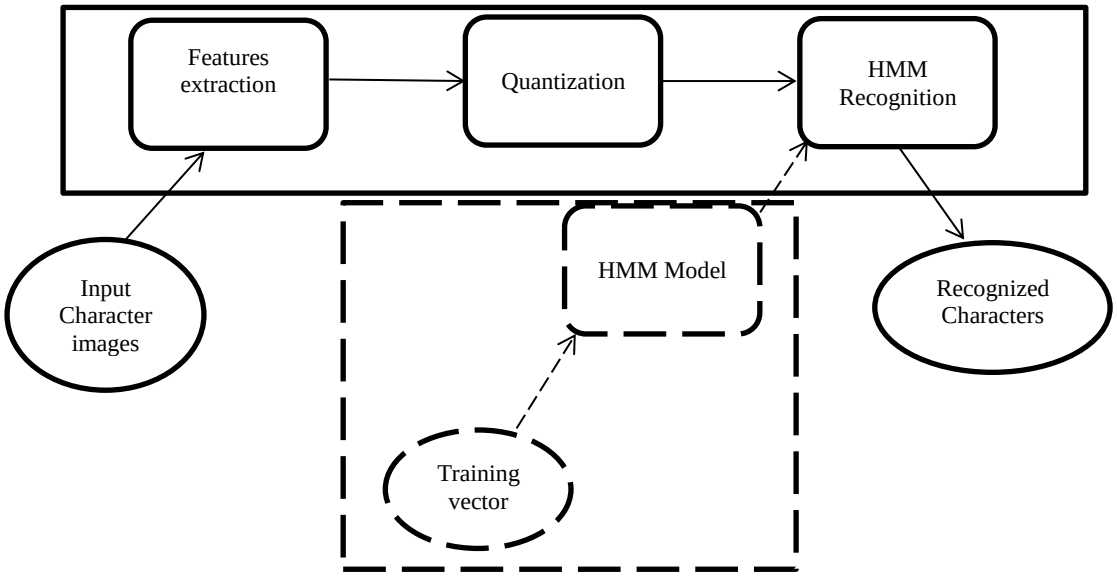

Fig.3 HMM-based Proposed Approach

Problem Definition

For complete word recognition, researchers suggested building a hierarchal network of HMMs which consists of a set of nodes which are connected by arcs. Each node is represented by a HMM which is itself a network of states connected by arcs [21]. The lower level of the hierarchy detects characters while the higher level recognizes whole words. A language model is needed to be built to define the transitions between character nodes.
In this work we will tackle only the characters recognition problem, our aim is to detect handwritten Arabic characters and later a full system to recognize full words and full text will be built.

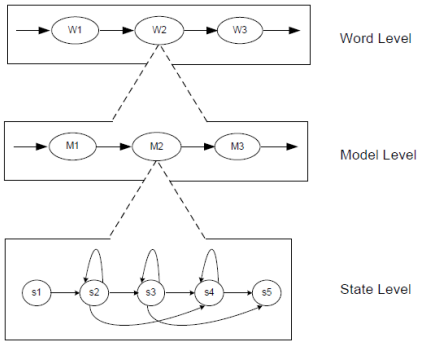

Fig.4 Recognition Network Hierarchy

Theory[20]

Let each character be represented by a sequence of vectors or *observations O*, defined as

$$O = o_1, o_2, o_3 \ldots\ldots o_T$$

where o_t is the vector observed at time *t*. The isolated word recognition problem can then be regarded as that of computing

$$argmax\{P(c_i|O)\}$$

where $w_i$is the *i*'th vocabulary word. Using Bayes' Rule gives

$$P(c_i|O) = \frac{P(O|c_i)\,P(c_i)}{P(O)}$$

Thus, since prior probabilities $P(c_i)$ are given, the most probable character depends only on the likelihood $P(O|c_i)$. Given the dimensionality of the observation sequence O, the direct estimation of the joint conditional probability $P(o_1, o_2, o_3 \ldots\ldots o_j|c_i)$ from training example is not practicable. However if Markov Model is assumed for word production, we can simplify the problem and calculate $P(O|M_i)$ instead of $P(O|c_i)$ and achieve the same results

$$P(O|c_i) = P(O|M_i)$$

where for each training set $P(O|M_i)$ of the i'th vocabulary word can be calculated.

Database Collection

To collect a database of characters, an Android application was implemented and a touch screen pen tablet was used to collect data from 7 different persons. Each person wrote an average of 5 instances per character.

Fig.5 Characters Database

Feature Extraction and Quantization

In this project we used the sequence of directions that the shape of letter took during writing. We calculated the angle θ between each 2 points in the course of the letter shape, and then the angle was quantized into 16 levels. Each letter is represented by an observation vector of the directions taken by the letter as shown in figures 6, 7, and 8.

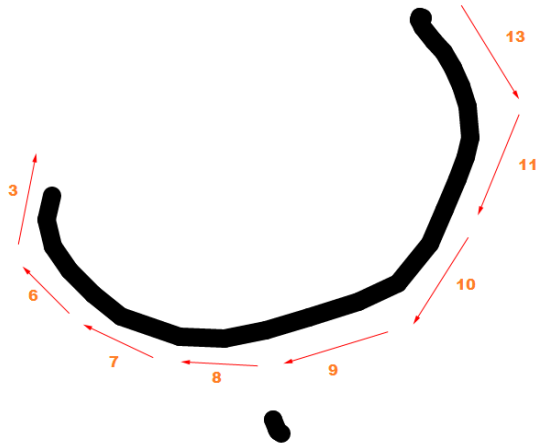

Fig.6 Sequence Vector of Letter Baaa

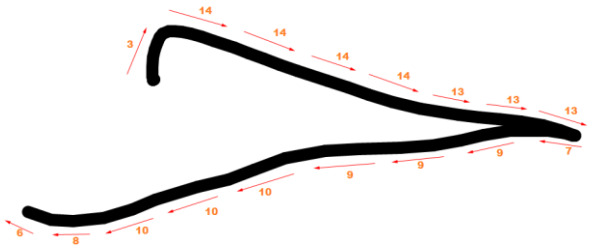

Fig.7 Sequence Vector of Letter Haaa

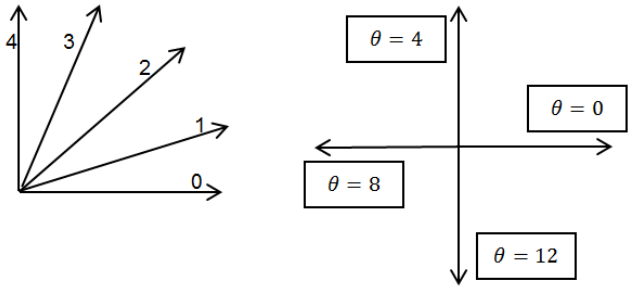

Fig.8 Quantization levels of Theta

HMM

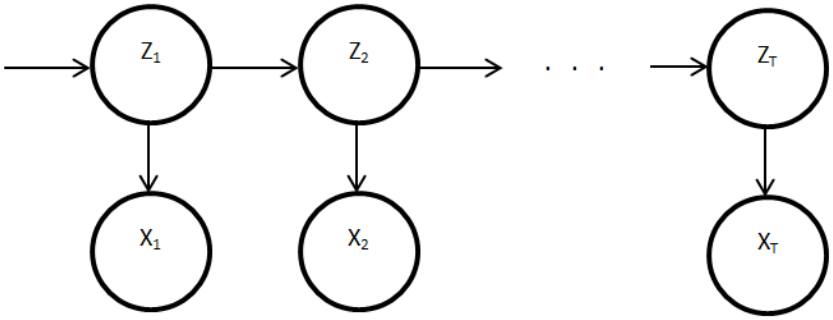

Fig.9 HMM

HMM Terminology
A HMM Model is specified by:

- The set of states: $S = \{ S_1, S_2, \ldots, S_{ns} \}$,
- and a set of parameters $\Theta = \{ \pi$,A,B$\}$:
  - The prior probabilities $\pi_i = P(q_1 = S_i)$ are the probabilities of S_i being the first state of a state sequence. Collected in a vector π.
  - The transition probabilities are the probabilities to go from state i to state j: $A_{i,j} = P(q_n + 1 = S_j \mid q_n = S_i)$. They are collected in the matrix A.
  - The emission probabilities characterize the likelihood of a certain observation x, if the model is in state S_i.

    Depending on the kind of observation x we have:
    - for discrete observations,

$x_n \in \{V_1, \ldots, V_k\}$: $b_{i,k} = P(X_n = V_k \mid q_n = S_i)$, the probabilities to observe V_k if the current state is $q_n = S_i$. The numbers $b_{i,k}$ can be collected in a matrix B.

- for continuous valued observations, e.g., $x_n \in R^D$:

  A set of functions $b_i(X_n) = p(X_n \mid q_n = S_i)$ describing the probability densities (probability density functions, pdfs) over the observation space for the system being in state S_i. Collected in the vector B(x) of functions. Emission pdfs are often parameterized, e.g, by mixtures of Gaussians. The operation of a HMM is characterized by

- The (hidden) state sequence $Q = \{q_1, q_2, \ldots, q_n\}, q_n \in S$,
- The observation sequence $X = \{x_1, x_2, \ldots, x_n\}$.

Useful formula:

- Probability of a state sequence: the probability of a state sequence $Q = \{q_1, q_2, \ldots, q_n\}$ coming from a HMM with parameters Θ corresponds to the product of the transition probabilities from one state to the following:

$$P\left(Q \middle| \Theta\right) = \pi . q_1 . \prod_{n=1}^{n-1} A_{q_n, q_{n+1}} = A_{q_1,q_2} . A_{q_2,q_3} \ldots . A_{q_{n-2},q_{n-1}}$$

- Likelihood of an observation sequence given a state sequence, or likelihood of an observation sequence along a single path: given an observation sequence $X = \{x_1, x_2, \ldots, x_n\}$ and a state sequence (of the same length) $Q = \{q_1, q_2, \ldots, q_n\}$ determined from a HMM with parameters Θ, the likelihood of X along the path Q is equal to:

$$P\left(X \middle| Q, \Theta\right) = \prod_{n=1}^{n-1} P\left(X_n \middle| Q_n, \Theta\right) = b_{q_1,x_1} . b_{q_2,x_2} \ldots . b_{q_n,x_n}$$

  i.e., it is the product of the emission probabilities computed along the considered path.

- Joint likelihood of an observation sequence X and a path Q: it is the probability that X and Q occur simultaneously, $P\left(X \middle| Q, \Theta\right)$, and decomposes into a product of the two quantities defined previously:

$$P\left(X, Q \middle| \Theta\right) = P\left(X \middle| Q, \Theta\right) . P\left(Q \middle| \Theta\right) \quad [Bayes]$$

- Likelihood of a sequence with respect to a HMM: the likelihood of an observation sequence $X = \{x_1, x_2, \ldots, x_n\}$ with respect to a Hidden Markov Model with parameters Θ expands as follows:

$$P\left(X \middle| \Theta\right) = \sum_{all\ Q} P\left(X, Q \middle| \Theta\right)$$

i.e., it is the sum of the joint likelihoods of the sequence over all possible state sequences Q allowed by the model.

HMM Training

- Given: HMM structure (Ns states, K observation symbols)
- Given: Training sequence X = { $x_1, x_2, \ldots, x_n$ }
- Wanted: optimal parameter values Θ = { π ,A,B}

$$P\left(X\middle|\hat{\Theta}\right) = \max_{\Theta} P\left(X\middle|\Theta\right) = \max_{\Theta} \sum_{Q \in Q^N} P\left(X,Q\middle|\hat{\Theta}\right)$$

MATLAB provides a toolbox for HMM, with function for training which estimates a parameter model `ESTTR,ESTEMIT` for a matrix of sequence vectors `seq` using Baum-Welch or Viterbi algorithm:

```
[ESTTR, ESTEMIT] = hmmtrain(seq,TRGUESS,EMITGUESS)
```

HMM Recognition

- Given: HMM parameters Θ = { π ,A,B}
- Given: Observed sequence X = { $x_1, x_2, \ldots, x_n$ }
- Wanted: Probability P(X | Θ), for X being produced by Θ

$$\mathrm{P}\left(X\middle|\Theta\right) = \sum_{all\ Q} \mathrm{P}\left(X,Q\middle|\Theta\right)$$

MATLAB provides a toolbox for HMM, with function for recognition which calculates the log of likelihood of an observation sequence `seq` to be generated by a model `TRANS, EMIS`:

```
[PSTATES, logpseq] = hmmdecode(seq,TRANS,EMIS)
```

EXPERIMENTAL RESULTS

Error Calculation

To calculate the recognition rate:

- An Evaluation set is composed of all handwritten letters (28 characters per sample).
- For each set element:
  - The likelihood of each evaluation sample element (letter *Alef* for example) is calculated for the 28 character models that were previously trained, and the model that produced the maximum likelihood is regarded as the appropriate model for this sample.
    - If the sample was classified correctly nothing is done.
    - If the sample was not classified correctly, then error = error + 1;
  - Total error was calculated by averaging the above error

Results

Using Directions Sequence approach

Runs	**n-States HMM**	**n-training Samples**	**Evaluation Data Source**	**n-Recognition sets**	**Error**	**Recognition Time(ms)**
10	3	3	Training Set	1	**12.50%**	31
10	5	3	Training Set	1	**8.12%**	43
10	10	3	Training Set	1	**1.43%**	70

10	3	5	Training Set	2	**26.61%**	57
10	5	5	Training Set	2	**15.71%**	61
10	10	5	Training Set	2	**7.50%**	126
10	3	5	Different Set	2	**84.29%**	41
10	5	5	Different Set	2	**82.86%**	64
10	10	5	Different Set	2	**88.21%**	130
10	3	10	Training Set	2	**45.36%**	40
10	5	10	Training Set	2	**39.29%**	66
10	10	10	Training Set	2	**24.82%**	110
10	20	10	Training Set	2	**3.57%**	294
10	3	10	Different Set	2	**81.07%**	35
10	5	10	Different Set	2	**79.82%**	46
10	10	10	Different Set	2	**78.57%**	78
10	20	10	Different Set	2	**80%**	190

Table2. Results for Directions Sequence Approach

Using Sliding Window approach

Runs	**n-States HMM**	**n-training Samples**	**Evaluation Data Source**	**n-Recognition sets**	**Error**	**Recognition Time(ms)**
10	3	20	Different Set	2	**85.57%**	47
10	5	20	Different Set	2	**83.39%**	33
10	10	20	Different Set	2	**86.07%**	84
10	3	20	Training Set	2	**41.07%**	26
10	5	20	Training Set	2	**30.54%**	42

Table3. Results for Sliding Window Approach

CONCLUSION

In this work, we presented an HMM model implemented using MATLAB to recognize Arabic letters, although our work didn't achieve better than state of the art results but some important conclusions can be drawn from our work:

- More methods to extract features needs to be tested.
- While increasing the number of states of the HMM model decrease the recognition error, on the other hand the training time increases and the divergence rate of the algorithm increases.
- Increasing the number of training samples improves the accuracy of the model.
- Recognition run time shows that using HMM is feasible for online recognition.

FUTURE WORK

In this work, we presented an HMM model implemented using MATLAB to recognize Arabic letters, although our work didn't achieve better than state of the art results but some important conclusions can be drawn from our work:

- More classes needs to be introduced in addition to the original 28 classes to represent the character in different positions of the word.
- Compile a larger database.
- Releasing the current software and database as GPL free software to aid other researchers.
- Hierarchical models need to be built to recognize whole words.
- After prototyping phase on MATLAB is done, the application should be implemented on a suitable platform like Android.
- Combine the words recognizer with a dictionary to improve words recognition rate.

BIBLIOGRAPHY

[1] Zavorin, Ilya, and Eugene Borovikov. "Data collection and annotation for Arabic document analysis." *Guide to OCR for Arabic Scripts*. Springer London, 2012. 375-394.

[2] Davidson, R., and R. Hopely. "Arabic and Persian OCR training and test data sets." *Proc. of Symp. on Document Image Understanding Technology, April*. Vol. 30. 1997.

[3] Slimane, Fouad, et al. "A new arabic printed text image database and evaluation protocols." *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009.

[4] Slimane, Fouad, et al. "ICDAR 2011-arabic recognition competition: Multi-font multi-size digitally represented text." *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011.

[5] Parhami, Behrooz, and M. Taraghi. "Automatic recognition of printed Farsi texts." *Pattern Recognition* 14.1 (1981): 395-403.

[6] Elgammal, Ahmed M., and Mohamed A. Ismail. "A graph-based segmentation and feature extraction framework for Arabic text recognition." *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001.

[7] Khorsheed, Mohammad S. "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)." *Pattern Recognition Letters* 28.12 (2007): 1563-1571.

[8] Touj, Sofien, Najoua Essoukri Ben Amara, and Hamid Amiri. "Generalized hough transform for arabic optical character recognition." *Proceedings of the Seventh International Conference on Document Analysis and Recognition-Volume 2*. IEEE Computer Society, 2003

[9] A. Gime andnez, I. Khoury, and A. Juan, "Windowed bernoulli mixture hmms for arabic handwritten word recognition," in 2010 International Conference on Frontiers in Handwriting Recognition (ICFHR), 2010, pp. 533 –538.

[10] Al-Muhtaseb, Husni A., Sabri A. Mahmoud, and Rami S. Qahwaji. "Recognition of off line printed Arabic text using Hidden Markov Models." *Signal Processing*88.12 (2008): 2902-2912

[11] Rashwan, M. A., et al. "Arabic OCR system analogous to HMM-based ASR systems; Implementation and evaluation." *Journal of Engineering and Applied Science Cairo*- 54.6 (2007): 653.

[12] Lu, Zhidong A., et al. "Robust language-independent OCR system." *The 27th AIPR Workshop: Advances in Computer-Assisted Recognition*. International Society for Optics and Photonics, 1999.

[13] Nawaz, S. N., et al. "An approach to offline Arabic character recognition using neural networks." *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on*. Vol. 3. IEEE, 2003.

[14] Amin, Adnan, and Nabeel Murshed. "Recognition of printed Arabic words with fuzzy ARTMAP neural network." *Neural Networks, 1999. IJCNN'99. International Joint Conference on*. Vol. 4. IEEE, 1999.

[15] Rashid, Sheikh Faisal, Faisal Shafait, and Thomas M. Breuel. "Scanning Neural Network for Text Line Recognition." *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012.

[16] Pechwitz, Mario, and Volker Maergner. "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database." *ICDAR*. Vol. 3. 2003.
[17] Arica, Nafiz, and Fatos T. Yarman-Vural. "Optical character recognition for cursive handwriting." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.6 (2002): 801-813.
[18] Kundu, Amlan, et al. "Arabic handwriting recognition using variable duration HMM." *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. Vol. 2. IEEE, 2007.
[19] Al-Hajj Mohamad, Ramy, Laurence Likforman-Sulem, and Chafic Mokbel. "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.7 (2009): 1165-1177.
[20] Evermann, Gunnar, et al. The HTK book. Vol. 2. Cambridge: Entropic Cambridge Research Laboratory, 1997.
[21] Al-Sulaiman, Mansoor. "RECOGNIZING CURSIVE ARABIC SCRIPT USING HIDDEN MARKOV MODELS." PhD diss., College of Computer and Information Sciences Department of Computer Engineering RECOGNIZING CURSIVE ARABIC SCRIPT USING HIDDEN MARKOV MODELS Supervisor Dr. Mansoor Al-Sulaiman Co–Supervisor Dr. Mohammad S. Khorsheed Submitted By Saad Ali Hussien Al-Qahtani 420020521 Submitted in Partial Fulfilment of the Requirements for the Master's Degree in the Department of Computer Engineering at the College of Computer and Information Sciences, King Saud University, 2004.