WestVirginiaUniversity
THE RESEARCH REPOSITORY @ WVU

Graduate Theses, Dissertations, and Problem Reports

2012

Recognition of Arabic handwritten words

Omar Alhmouz
West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

RECOGNITION OF ARABIC HANDWRITTEN WORDS

by

Omar Alhmouz

**Problem Report submitted to the Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of**

**Master of Science
in
Electrical Engineering**

Approved by

**Xin Li, Committee Chairperson
Hanny Ammar
Guodong Guo**

Lane Department of Computer Science and Electrical Engineering

**Morgantown, West Virginia
2012**

Abstract

**RECOGNITION OF ARABIC HANDWRITTEN
WORDS**

by Omar Alhmouz

Recognizing Arabic handwritten words is a difficult problem due to the deformations of different writing styles. Moreover, the cursive nature of the Arabic writing makes correct segmentation of characters an almost impossible task. While there are many sub systems in an Arabic words recognition system, in this work we develop a sub system to recognize Part of Arabic Words (PAW). We try to solve this problem using three different approaches, implicit segmentation and two variants of holistic approach. While Rothacker found similar conclusions while this work is being prepared, we report the difficulty in locating characters in PAW using Scale Invariant Feature Transforms under the first approach. In the second and third approaches, we use holistic approach to recognize PAW using Support Vector Machine (SVM) and Active Shape Models (ASM). While there are few works that use SVM to recognize PAW, they use a small dataset; we use a large dataset and a different set of features. We also explain the errors SVM and ASM make and propose some remedies to these errors as future work.

DEDICATION

The author wishes to Author would like to thank West Virginia University for making this research possible. Also, thanks to Dr. Xin Li for proposing this research subject, to Dr. Hanny Ammar for his support and Dr. Guodang Guo for his valuable recommendations. Author would like to acknowledge support and funding of Dr. Daryl Reynolds.

ACKNOWLEDGMENTS

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / NOMENCLATURE

C H A P T E R 1 : I N T R O D U C T I O N

1. Background

In a recent survey paper, the work by Almuallim and Yamaguchi was referenced as "The(first) method to focus on text that was not pre-segmented"(Lorigo, & Govindaraju, 2006).While Arabic language is spoken by more than 400 million people (world fact book,2011), the subject of Arabic character recognition is not meeting enough credit; the above mentioned work (Almuallim, & Yamaguchi, 1987) has a modest number of citations , i.e. 156 citations, (Google scholar, retrieved on February, 10th , 2012). The reason for lack of interest was justified by Amin, "This is the result of lack of adequate support in term of funding, and other utilities such as Arabic text database, dictionaries, etc.. and of course of the cursive nature of its writing rules"(1997).

In the year 1996 a system for sorting handwritten addresses were introduced at the USPS service. The latter is the largest consumer of patron recognition system; USPS uses Remote Computer Units (RCR) to decode the handwritten address found on the postal envelops. This services was successful such that "from the year 1996 to 2004, the encode rate for RCRs has increased from 35% to 90%" (Srihari, & Setlur, 2006). While research on recognition of handwritten English text have helped to produce commercial system like the one used at USPS, research on recognition of Arabic handwritten text have a long way before postal offices in Arabic speaking countries can automate mail sorting.

Postal services and mail sorting is an important application for Arabic handwritten text recognition but there are many other application; we will mention some of them for example: Bank can make advantage of an automated system to recognize handwritten check amount fields and compare it to written digits in the check(Maddouri, Amiri, 2002). Seven million Arabic historical documents written in science, philosophy and arts between 7 and 15 centuries; these documents can be found in museums, libraries, and private collections (Saabni, El-Sana, 2011). Researchers need to indexed and archive these documents automatically based on an Arabic word recognition system or Arabic word spotting system.

Arabic handwritten text recognition approaches can be divided according to weather they need segmentation or not into: segmentation based recognition and holistic recognition approaches (Larigo, Govindarau, 2006). Segmentation-based approach suffers from the dilemma of the dependence of recognition results on the segmentation results; any recognition will fail unless the word has been correctly segmented to smaller units. Implicit segmentation tries to elude the segmentation problem by virtually segmenting the word into smaller units and then reconstructing the word again such that the highest recognition accuracy is achieved. Unfortunately this approach suffers if the rejection rate is high; rejecting a correctly classified sample might reduce the recognition rate (Leondes, 1998). Thus rejection rate must be kept low in order to insure high recognition rate. Holistic approach does not suffer from such problems but depends heavily on the recognition engine to achieve good recognition rate. While some recognition engines use features to describe the word, other types of engines build models of words and output a similarity measure between the test images and the training model.

2. Problem Statement

Recognizing Arabic words through recognizing its characters is a challenging problem mainly due to the boundary problem. The later problem renders correct segmentation as almost impossible. Thus anther approach that avoids segmentation and recognizes words or parts of words is possible yet, the deformations found in different people writing styles makes the classifier task a challenging one. Moreover, dots and diacritic add more confusion to the classifier. In this work we address the problem of recognizing Arabic Handwritten Sub Words. With reference to the system in Figure 1.1, we focus on second sub system in the figure, specifically PAW classifier.
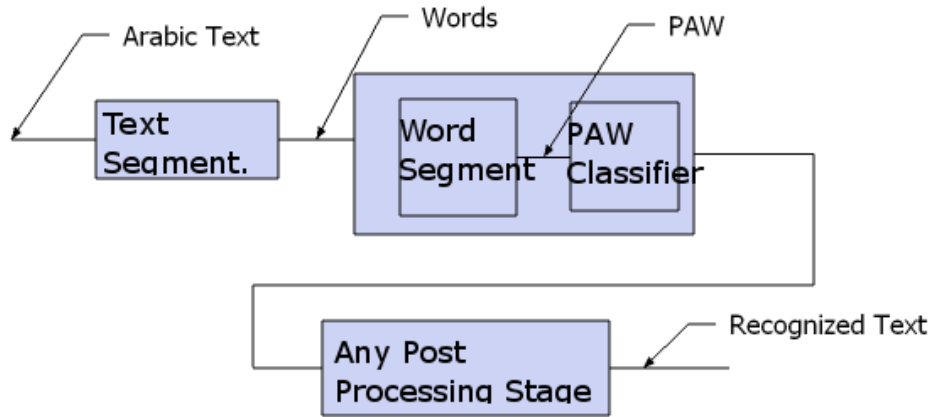
Figure 1.1 Arabic Words Recognition System.

3. Research Objective

Obtaining recognition rate of 75% or higher for sub words based recognition of handwritten Arabic words.

Before we describe approaches taken in this work, some properties of Arabic language must be considered. While Arabic writing has the same punctuation marks as Latin, it has many different properties including writing in the direction from right to left. Arabic writing is cursive; the word has many characters connected to each other from one side or both sides such that the line connecting them, called the baseline, has the lowest number of pixels at the connection point. Although Arabic letters can be connected from one or two sides, their form can vary depending on their position in the word and some forms of letters cannot be connected from the left. This separation allows the word to be divided in to smaller units called sub-words. While English letters consist of smaller set called vowels, Arabic letters have no vowels but have a set of short vowels, called diacritical markings, used to distinguish between different words in addition to some grammatical use. Furthermore, letters have complementary shapes connected to them. These are dots in different numbers, one, two and three and with positions above, below and in the middle of the letter. There are also zig zag complementary shapes. While diacritical markings are not always added and words can be distinguished from context, dots and zig-zag shapes are essential to distinguish different letters from each other. While dots can be removed during processing, some Arabic letters have complex shapes, like concavities and holes that cause a

2

single letter to be divided in multiple parts during segmentations. Furthermore, sometimes writing has the problem of touching characters, where a character touches the next character at points below or above the base line. Segmentation of touching characters results in shapes that cannot be recognized as part of any character. Finally, ligature is another issue similar to touching characters but different in a sense that the characters are vertically concatenated; this issue appears with some styles of writing only. Not only does character form depend on its position in the word, but also handwritten characters have variability in character shape and this result in large number of classes.

4. Report Organization

The rest of this report is organized as follows: in chapter two, we present a literature review of recent articles related to the problem discussed in this report specifically implicit segmentation and holistic approach. Furthermore, we present details of the Arabic handwritten words database used to test our algorithms. In chapter three, we present different approaches used to achieve the objective of this work. Specifically, we propose a system based on implicit segmentation and show difficulties with such approach. Then we present two methods based on holistic approach: sub words recognition using Active Shape Models and sub words recognition using Support Vector Machines. In chapter four, we show some results for the holistic approach and discuss the results. Finally, we discuss conclusions and future work in chapter five.

C H A P T E R 2 : R E L A T E D W O R K

In this chapter we discuss some related work to the problem at hand. In section one, we discuss recent works using implicit segmentation and cite some problems facing this approach. In section two, holistic approach based work is discussed. Specifically, recent research on recognizing PAW using SVM is explained along with some other works based on ASM, and Hidden Markov Models.

1. Implicit Segmentation

Rothacker, Vajda and Fink used Scale Invariant Feature Transforms (SIFT) without scale space analysis. Since the number of interest points detected by SIFT does not suffice to recognize words, others have used a dense grid of interest points for document spotting (Rusiñol, Aldavert, Toledo,& Lladós,2011). Rothacker, Vajda and Fink choose to use Harris corner detector at multiple spatial sizes of the descriptor in the image. Unlike other works (Azad, Asfour & Dillmann, 2009), here multiple scales refer to descriptor covering multiple resolutions, i.e. a scale magnification factor is used to control the size of the descriptor in the image. Thus the scale is increase according to:

$$S_i = S_{i-1}^{1.5}$$

the value 1.5 is chosen experimentally and S_o is chosen as the mean scale of all interest points scales detected by a SIFT detector ,i.e. by Scale-Space analysis (Rothacker, Vajda and Fink, 2012). In his thesis, Rothacker indicated that character recognition based on part_level, character level, approach would "only work on a small number of categories" (Rothacker, 2011). Thus Rothacker choose to work on whole words, Arabic and English words. A bag of features representation based on the descriptor described above resulted in a recognition rates of 69.85% with k-NN classifier on set d of IFN/ENIT database (Pechwitz et al., 2002). Never the less, when he used an HMM based system with bag of features statistics recognition rates were 96.2%, 92.9%, and 90.2% on sets d,e, and f respectively. Here a bag of features refer to visual word (batches in the image described by SIFT) part of visual vocabulary. In the first approach document retrieval style was adopted; k-NN was used to search through a categorized database of bag of features of images. K most similar visual words were retrieved and the category that is most frequent in the list is the winner. Rothacker used a constraint on the location of visual word to filter the results before ranking them based on cosine distance to reach the recognition rate reported above, 69.85%. He indicated that increasing k would increase the rate until k is equal to the number of words in each category, then no further increase in performance is noticed. In the second approach, two variants of Hidden Markov Models (HMM) are used to recognize words. While Margner, ElAbed, and Pechwitz used Semi-Continues HMM to achieve a rate of 74.69% on set d of IFN/ENIT database (Pechwitz et al., 2002)(2006), Rothacker achieved 93.39% recognition rate on same set using bag of features representation and a Continues HMM(Rothacker, Vajda and Fink, 2012). Nevertheless, he reports better performance using a variant of HMM, he called pseudo discrete HMM. In this system, the probability of observing a distribution of visual words in a state is given by (Rothacker, Vajda & Fink,2012):

$$b_j(f) = \sum_{k=1}^{v} c_{jk} f_k$$

where f_k is the visual words probabilities in the observation sequence. This variant gives better results, i.e., the results reported in the beginning of this paragraph. Rothacker noted that SIFT is used in this context, recognizing Arabic words, for the first time. Other applications of SIFT are word spotting for historical documents (Rusiñol, Aldavert, Toledo,& Lladós,2011), character recognition/text detection in natural scenes (Coates, et. al. 2011) and writer identification (Woodard, et al., 2010). While all these applications are not relevant to our application, we review

the last paper due to its similarity to Rothacker's work. Woodard et al. compared multiple detectors in a writer identification framework (2010). Similar to Rothacker's approach, SIFT descriptors are clustered and quantized to create a code book. SIFT features for unknown image are quantized to create a co-occurrence table n(d=quary, w), where w is an index for the quantized SIFT descriptor; this table is bag of features statistics. While Woodard et al. compared 5 similarity measures, the best performance was obtained for a similarity measure based on co-occurrence table (2010):

$$\log(P_{di}) = \sum_{j=1}^{L} n(d = \text{quary}, w_j) \log \sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid d_i)$$

here z is the latent semantic. They reported correct writer identification rate of 98%. Parvez, and Sabri segmented PAW into characters and recombined the characters and ill-segmented words into PAW Hypothesis (2012). The later results from matching character models with segmented characters and ill-segmented words. These hypotheses are verified by consulting a lexicon for the hypothesis. This lexicon is created from recording number of dots and their location training PAW. While dynamic programming is used to combine segmented characters, a rejection option is used to ignore non valid character combinations. Never the less, reject rule and the rejection rate are not given.

2. Holistic Approach

P Lanities, Taylor and Cootes used active shape models (ASM) to represent letters, numerals, and Alpha numeric. While considered to be an advantage of using ASM, models were not able represent characters outside point distribution model (PDM). The latter are constructed from 20 control points (landmarks) on the character's skeleton. Numerals had a recognition rate of 93% were as letters had a recognition rate of 90%. Finally, the alphanumeric had a rate of 86%. If the correct answer is considered from the best 3 matches then rates would go up to 97%. While recognition rates are comparable to other classifier performance in the case of a single classifier case, classifier combination techniques achieved better classification rates at the time of publication (1993). Furthermore, PDM was initialized to be as close as possible to the test image to ensure algorithm convergence. Two measures: Support measure, and explained evidence measure were essential to classify the characters. While Active shape Models (ASM) and K-Nearest Neighbor (K-NN) provide a fast and approximate approach for searching a PAW in a data base, the first approach, ASM, needs a classifier to remove ambiguity and the second approach, K-NN, cannot find the exact solution in a reasonable time. Saabni and Bronstein combine these two approach by proposing to use a signature like feature vector in a data base search based on K-NN and a generative classifier, Active Dynamic Time Warp (ADTW) (2011). The former provide a short list of possible matches to the unknown PAW and then ASM is used to search within all possible deformations of each PAW in the short list. Finally DTW distance is used to decide upon the best match for the input image; the combination of ASM and DTW is referred to as ADTW (Roy, Maghvanath, & Sharam, 2009). A previous publication for the author proposed an algorithm to cluster and label PAWs. Furthermore, ADTW showed a reduction in the recognition rate compared to using DTW alone. There was 50% reduction in processing time using ADTW and K-NN was needed to give reasonable search time.

Khalifa and BingRu used Discrete Cosine Transforms (DCT) to extract features from Arabic handwritten Pieces of Arabic Words (PAW) of the IFN/ENIT database (Pechwitz et al., 2002) (2011). Khalifa and BingRu used DCT to represent the image in the frequency domain; more specifically, they record the Euclidean distances between maximum and minimum of every column of DCT coefficient matrix into a vector. Then local minima and local maxima points of this sequence are represented using Principle Component Analysis (PCA) after passing them through a feature selection stage called Recursive Feature Elimination (RFE) (Lei, & Govindaraju, 2005).

This stage reduce features during training and build PCA representation for them; during training of each class, RFE saves the best features and the corresponding PCA representation for them to be used at the testing stage. Recognition engine used is Support Vector Machine; recognition rates are 89% without RFE and increase to 91.7% after RFE for 56 classes of PAW. Never the less, authors report high dissimilarity between feature vectors of the same class and "best results were produced for mono font for all writers". Unfortunately, no further details were given on what is mono font. Hence we are not able to replicate the results. Furthermore authors reported results on PATS_A01 database from (Al-Muhtaseb, Mahmoud, & Qahwaji, 2008) for Arabic typewritten text. Recognition rates are 92-93% without and with RFE for 980 typewritten words divided for training/testing. In a later paper, Khlifa, BingRu and Mohammed used Scale Invariant Feature Transforms (SIFT) to transform the image into a matrix of SIFT descriptors (2012). Authors use Scale-Space analysis on the resultant matrix to find interest points in this matrix. Now these interest points are the input to the SVM machine to be recognized. While in a previous work they use one-against one setup, they use one-against-all setup for this multiclass problem. Number of words/shapes is 78 taken from IFN/ENIT database (Pechwitz et al., 2002); no details are given on the dataset used. Khalifa, BingRu and Mohammed reported recognition rates of 94-99% "depending on the kind and number of (words) classes used". Authors compare this work to (Khlifa and BingRu, 2011); they use typewritten Arabic words from PATS_A01; now, 392 words are used and divided into training and testing. Recognition rate for typewritten words is 98.4% and it turn out that scale-space analysis help reduce feature dimensionality without affecting performance. Dreuw, Heigold and Ney used a two-pass Hidden Markov Models system where the output of the first pass forms a training data for the 2nd pass system. Using the complete output of the first pass as an input to the second pass would not result in an increase in the performance. Hence a confidence term is added in the training data of both passes. Results show that confidence based system that reject states with low confidence in the first pass performs better than removing word parts based their confidence in the first pass. Although explained by a diagram and an example, word segment/word frame alignment to reject word/states is not well detailed. Furthermore, Authors used empirical data to choose the threshold used to remove words or states in first pass such that a rejection rate of 5% is satisfied. They combine this training approach with their earlier work to obtain an error rate of 14.55%, a recognition rate 85.45%, on set e of IFN/ENIT database (ICDAR2005) (2009). While Particle Swarm Optimization (PSO) is able to optimize the parameters of HMM engine: transition probabilities distribution, and observation transition probability, this heuristics is not robust enough to guarantee best performance in all cases. The system proposed by Hamdani, Hamdani, Alimi, Abed, and Margner achieves a recognition rate of 81.02% on set e (ICDAR2005) (2009). Although the system performs better on set f in ICDAR2010, this rate is lower than the work of dreuw et.al. for the same year. Multiple models are optimized to give the best number of states and best transition matrix for 6 HMM engines. While the output of these 6 engines is combined as post processing stage and achieve a higher rate, authors did not give specific details for the implementation of combination.

3. Arabic Handwritten Words Database

Finally we give some details about the IFN/ENIT database. As the name of the database refers, it is the results of cooperation of the Institute for Communications Technology, Technical University, Germany and the Ecole Nationale d'Ingénieur de Tunis, Tunis. Database images represent the names of Tunisian cities, 937 city/village names, collected from handwritten forms written by 411different writers. Total number of handwritten city names is 26000 divided into 4 sets a-d. New sets e,f, and s (collected later) are not publically available; these sets are used to test recognition algorithms participating in the ICDAR context for recognition of Arabic handwritten text (Märgner, & El Abed, 2011). Each year on March this contest is held, except for the year 2012, and the year after the context is held a data set that was used in the context is made public; set e is now public, set s and f were provided to Rothacker, Vajda and Fink to test their algorithm (2009). In the beginning of announcing this database, it was decided that baseline information is

provided with ground truth. Never the less, currently only set (a) has baseline information. Ground truth and zip code information is given for all sets with some flags to indicate size of image and other information, refer to (Pechwitz et al., 2002) for further details on the database. Abandah and Khedher performed statistical measurements, mean and covariance, on a large set of features: main body features, skeleton features and boundary features (2009). The study showed that bounding energy, orientation, perimeter to diagonal ratio and pixel density has lowest variance among features tested on a database of 102 characters classes. This database was collected at Jordan University, it has manually segmented characters in addition to a 5696 PAWs (Abandah, & Khedher, 2009).Database images where resampled to smaller size, which reduces the image quality once resized to original size.

C H A P T E R 3 : W O R D R E C O G N I T I O N

In this chapter we present two approaches that we have implemented to achieve the research objective. First approach uses implicit segmentation which did not prove feasible thus we did not pursue further. In section 3.2 we present the second approach based on holistic recognition. Results for holistic recognition can be found in chapter 4.

1. **Implicit Segmentation**

At the start of this work, matching of characters to words was supposed to recognized characters and their locations in the word. In addition to many other problems, we show here that finding location of a character in the word is an error prone process. While we are not aware that a similar work is being prepared, Rothacker reported in his thesis the same conclusions about our problem (2011).

Scale Invariant Features Transforms (SIFT) can be used to describe characters; some advantages of using SIFT are: it can handle occlusions in the character due to segmentation faults, it can handle rotation to some degree (when there is to baseline rotation) and it can handle scaling issues. This means no need to perform preprocessing on the written words to remove baseline rotation or to do normalization to the character height.

Instead of segmenting words into characters at the recognition stage, we use sub words, called Pieces of Arabic Words (PAW). The reason is two folds: first it is better to deal with sub words as it eludes the need to develop a segmentation algorithm, second, recognizing PAW avoids the problem of having similar characters; using PAW increases possibility of having unique characters in a PAW. Note that matching a character to a PAW does not reveal how many duplication of this character in the PAW. Thus using PAW would make recognition task easier by one unknown; only the location of the matching character have to be found. Never the less, we note that the writing style for different writers results in deformations for characters with circles, and ellipses. The former case gives more confusion to the recognizer; the circles in the characters are always not correctly recognized. Usually the curves in the character have more discriminative features and result in correct recognition of the character (Rothacker, Vajda, & Fink, 2012). Since confusion comes from having a feature match that is an outlier matching PAW with characters is the most important step. We choose to use the function VL_UBCMATCH in VL_Feat (Vedaldi, & Fulkerson, 2008) that is written in C thus allowing fast matching for large number of descriptors.

The approach suggested by (Lowe, 1999) to recognize a correct match is to calculate the second closest descriptor to first closest descriptor distance ratio; if the ratio is larger than a user defined threshold the first match is considered to be a correct match. Once a character has some feature points match with PAW image, we use a simple approach to filter candidate characters. We calculate the probability of a character as the ratio between the numbers of feature points in the target image matched to the template character in consideration to the number feature points in the template character; if it is larger than some ratio the character is declared as a possible character hypothesis.

Post-processing steps is required to choose from many character hypothesis produced when matching template images to the PAW's image. The estimated positions of characters help to re-construct a word hypothesis and to reject the outliers. While Lowe have used Hough transforms to cluster SIFT points and then used RANSAC to verify the objects' pose (1999), we use k-means clustering to cluster feature points according to their location and orientation. The number of clusters should be initialized; we have assumed a good estimate for k to be the number of recognized characters. Clustering results help to reject character hypothesis based on the following rule: if there are feature points scattered on more than one cluster, the distance between the clusters is measured. Clusters are joined when they are close and they are less than two clusters otherwise character hypothesis is rejected.

The estimated positions of the characters are clustered using k-means clustering again. The number of clusters in initialized to 3, and reduced to 2 and 1 if the algorithm fails. This assumption helps in the next post processing step. Some authors used a set of rules related to the shape and position of the character in a PAW (Touj, Ben Ammara, Amiri, 2005). Base on Arabic language writing rules, we construct a tree of three levels, or less if kmeans cannot find such tree. Each level corresponds to beginning, middle and end position of a character in a PAW. We use the tree to construct a hypothesis for a PAW by organizing the characters' hypothesizes according to their location. Tree branches pruning is done by removing character hypothesis that does not fit the location they are found in ,i.e., a letter in the middle form cannot be found in the start position. After tree pruning, tree leafs correspond to PAW hypothesis and have a probability which is the product of its character hypothesis. We had two problems with this approach, first number of interest points detected by SIFT is small, second the rejection rate was high even after increasing the SIFT feature points. We describe next how to increase SIFT feature points.

When constructing a training set of characters, we have noticed that SIFT gives very less number of descriptor points. This is due to the small size of segmented characters and the fact that there is white background (Rothacker, Vajda, & Fink, 2012). Thus we use the descriptor given in (Azad, Asfour, & Dillmann, 2009). Azad, Asfour and Dillmann proposed to omit the scale-space step in SIFT and replace it with calculating Harris corner detector interest points (Harris, & Stephens,1989) at specific scales levels separated by a scale difference more than one octave (2009). While we have intended to deal with the problem of low number of feature points, using this descriptor provides an extra advantage; it has less computational complexity than SIFT. Here we give a brief introduction to the descriptor. The descriptor main steps are: after finding the key points using Harris corner detector, the orientation for each key point is calculated from the gradient. The scale and orientation information is used to produce the SIFT descriptor for the input key points. For each level the image is rescaled by a factor of delta S using linear interpolation. The former number, delta S, is taken here 0.75 as the author recommends to compromise between the speed and the coverage of the key points to current scale. Other approaches to increasing number of interest points can be found in (Rothacker, Vajda, & Fink, 2012) and (Zhang, Jin, Ding, & Gao, 2009). As explained in the literature review section, Rothacker, Vajda and Fink report problems with SIFT when used to recognize characters "(Approach) only work on a small number of categories" (Rothacker, 2011). Unfortunately, we note anther problem: high rejection rate.

Rejection of a sample when it has low confidence is undesirable behavior; it might reduce the recognition rate (Leondes, 1998). We note that there are many sources of rejection in the procedure we described above: The threshold for rejecting SIFT descriptors, the clustering of SIFT features based on their locations is based on distance threshold, clustering characters based on their location might reject characters in same position based on their probability. All

these thresholds can be optimized but the effort is optimizing their values, choosing the correct distance criteria to cluster them is error prone. Even if recognition was relatively correct, these problems might make such approach undesirable as we will see in the next approach.

2. Holistic Approach

Previous section showed main difficulty with using a feature based approach for recognizing characters in a PAW, in this section we take another approach to recognizing PAW, holistic approach. We discuss two ways to describe a PAW as a whole: either using a model or using features.

a. Statistical View

While constructing a module for a PAW seems an attractive option, the number of parameters of the module, the optimization algorithm needed to fit the model to the image and the number of choices of similarity function makes this approach very time consuming.

We choose to use Active Shape Modules (ASM), due to its ability to deform. Since PAW can have rigid deformation due to slant, rotation, writing stroke size and non-rigid deformations due to writing style, ASM ability is an important aspect that affected our choice. Before we explain problems with this model, some definitions, background on ASM module construction and optimization algorithm used to fit the module should be explained.

Modeling possible variations in characters or Pieces of Arabic Words (PAW) is possible. Suppose we describe the outer contour of a PAW using a set of points, called landmarks; all writing style variants of the same PAW should have a boundary having similar shape. Assume the position of points corresponding to same landmark position is a cloud of points. In ASM we describe the variance of these points around a mean point using Principle Component Analysis. The eigen values of the clouds for a set of landmarks (one PAW) represents the modes of a model that describe the PAW. Optimization of the ASM will give the closest possible deformation of the model in accordance with the input image. While we would like to capture the maximum range of variations seen in the training set, the number of Eigen values should be compromised with the range of possible variations (Cootes, Taylor, Cooper, & Graham, 1994). ASM allow capturing the variations of Arabic hand writing style based on training samples such that any new variant of a PAW can be described by (Cootes, Taylor, Cooper, & Graham, 1994):

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}\mathbf{b}$$

$$\mathbf{P} = (p_1 p_2 \ldots p_{t-1} p_t)$$

$$\mathbf{b} = (b_1 b_2 \ldots b_{t-1} b_t)^t$$

where P is a matrix of first t eigen vectors and b is a vector of weights.

Aligning a model towards an unknown image requires finding the best set of rotation θ, scale s and best position change for the points in the model dx. Many methods can be used to align two images, or two set of points (Cootes, Taylor, Cooper, & Graham, 1994; Cootes, 2000; Learned-Miller, 2006). Although initial placement of the model over the image affects the convergence of the algorithm, we initialize the location of a model at a weighted center of the locations of test image ON pixels. Furthermore, the rotation is initialized at an estimate of the rotation found in the test image (Lanities, Taylor, & Cootes, 1993).

Since ASM is an optimization problem the initial guess for rotation, scale and shape parameters is refined. ASM starts with a guess, as explained above, for θ, s, dx, b and creates a profile for region around each landmark in the model. The algorithm places the model on the unknown image and searches for a profile that is most similar to the profile of the training image. Note that the profile is a statistical description of number of sample points around each landmark. Samples are collected in a direction normal to the model boundary and samples are assumed to follow a multivariate Gaussian distribution. Thus ASM estimates the mean and covariance of the profile of each landmark and optimizes the following cost function (Cootes, 2000):

$$\min f(\mathbf{g_s}) = (\mathbf{g_s} - \overline{\mathbf{g}})^T \mathbf{S_g}^{-1} (\mathbf{g_s} - \overline{\mathbf{g}})$$

where the g_s is the normalized gradient of the profile of a sample from the unknown image, g⁻ is the mean of the profile at the landmark and S_g is the covariance of the profile at the landmark. Thus ASM optimizes the mahalanobis distance of the sample from the model mean. It is possible to use a Gaussian pyramid and repeat the optimization procedure above for each level of the pyramid going from coarse to fine levels (Cootes, 2000, Ginneken, et al., 2002). Kroon built an implementation of ASM including a modification to the above cost function (Kroon, 2011). Instead of using the mahalanobis distance expression above, Kroon suggested to minimize the PCA representation of the profile after subtracting the mean and normalizing by the variance. Search stopping criteria, the model parameters and the search parameters for ASM details can be found in (Cootes, 2000).

The main requirement of using ASM is the training data in the form of a set of points describing each object, in our case a PAW's contour. These points, called landmarks, have to be ordered and identically placed in each of the training images. Furthermore, landmarks can be specified manually, or can be specified automatically by an algorithm (Hill, & Taylor, 1994). Best location to place the landmarks on an image is on the contour of a PAW or contour of a character (Lanities, Taylor, & Cootes, 1993; Saabni, & Bronstein, 2011). Restrictions on landmark placement are: point should be placed on the same location and in the same order for every image.

Roy, Madhvanath, Anand, and Sharma defined free samples "Training samples that do not form prominent clusters typically corresponds to rarely occurring writing styles within the training set" (2009). Active shape models perform poorly when the number of samples is small. Remember that ASM cannot deform to shapes outside the training samples. This explains the need to have free samples and explain the need to have a large data set for training; Roy, Madhvanath, Anand, and Sharma clustered a minimum of 6 samples per training set cluster and adaptively increased the number of samples in the cluster during testing such that number of samples is >9 (2009). Furthermore, Saabni, and Bronstein noted that warping of contours of distant shapes from a mean shape will cause much damage (2011). We use 3-9 samples per class, never the less the results were not satisfying. Furthermore, we found that the search parameters used to fit the model have one parameter that affects the results, n_s, it is defined as: the number of sample points to search on either side of landmark. It is used to move the model's point to a new location during the optimization of the model fit to the image. However, we followed some steps to choose other parameters as well. We explain these steps in details due to their importance to optimizing any statistical model (Woodard et al., 2010).

Model parameters are detailed in (Cootes, 2000, Ginneken, et al., 2002), but their choice is not clearly justified. As the complexity of the model would be increased if we add a search algorithm to find optimum parameters, we try to optimize these parameters offline. First we fit a linear equation relating one or two parameters in the model, and search algorithm to the similarity measure results. An iterative procedure is used to optimize the weights of the equation based on steepest decent. It showed that n_s and n_i (the number of points in the model) have the highest weights in the linear function. Since this is linear function, we only used this

knowledge in interpreting one-against-all (OAA) or leave-one-out validation approach explained next. Cootes used "leave one out" to estimate how good the model will perform (2000); for each PAW there are m results, one for each sample taken as the test image and other samples used to build the model.

While it is preferred to optimize the choice for similarity measures, we choose to use the measures proposed by Lanities, Taylor, and Cootes (1993). This choice is justified by application used in Lanities, Taylor, and Cootes, which is digit recognition problem, and considering the complexity of the measure. We found many examples in the literature for similarity measure and its importance in finding correct matches. We refer to work by (Woodard et al., 2010 and Zhang, & Srihari, 2003) due to their relation to our problem. Their work shows that different similarity measures can be combined according to their discriminative power. Thus only experiments can validate the fit of a measure to the problem at hand. Due to the time constraint we did not evaluate different choices. Furthermore, some similarity measures have weights that should be optimized or have computational complexity that limit their search ability. An example of such similarity measure is given in (Saabni, & Bronstein, 2011). We refer to literature review section for more details on Saabni, and Bronstein's work.

In the next section we show how Support Vectors reduce recognition complexity, allow the use of simple distance metric and give good results in the same time.

b. Maximizing the Margin

The basic requirement of any classification algorithm is to define a plane to separate the class. Once this plan is found, the classification problem is transformed to finding the distance of the unknown sample to this plan. This approach is used in Support Vector Machines (SVM) to classify samples and requires only calculating the dot product between the unknown sample and the support vectors, thus it is complexity is reduced to a set of dot product operations. Thus our objective function is a linear function of parameters found through finding the maximum margin. The latter is the distance between the closest training positive or negative example, called support vector, to the separating plan.

Since finding the parameters of this plane is a regression problem over all the support vectors, then it can be learned as follows (Bishop, 2006):

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_i w_i x_i + b = <\mathbf{w}, \mathbf{x}> + b = \sum_{l=1}^{M} \alpha_l <\mathbf{x}_l, \mathbf{x}>$$

the last equality is due to the fact that weights of the learned function lie in the training samples space and the last summation is over all training samples. The Lagrange =1 for the support vectors and it is found by solving the following optimization problem (Bishop, 2006):

$$\max \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{i=1}^{M} \sum_{k=1}^{M} \alpha_i \alpha_k t_i t_k <\mathbf{x}_i, \mathbf{x}_k>$$

subject to the constraint

$$\sum_{l=1}^{M} \alpha_l t_l = 0$$

$$\alpha_l \geq 0$$

Since this is a regression problem we can work in a higher virtual space by assuming the following feature space transformation φ(x) and replace the dot product in the above equation $<\mathbf{x}_l, \mathbf{x}>$ by $<\phi(\mathbf{x}_l), \phi(\mathbf{x})> = \kappa(\mathbf{x}_l, \mathbf{x})$.

While we have transformed features into a higher space, mapping to higher space is transparent to the optimization problem and the dot operation advantage still exists. Now we want to learn the function (Bishop, 2006):

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

The above formulation considers the case of linearly separable data. Soft margins are considered into case of non-linearly separable data. In this case minimum errors are allowed to happen in classification and slack variables are introduced in the optimization problem to account for error. SVM with soft margins solves the following optimization problem (Bishop, 2006):

$$\min_{\mathbf{w},b} \ \mathbf{w}^T\mathbf{w} + C\sum_{l=1}^{M} \zeta_l$$

where ξ_n is a slack variable for each training sample M, it is >1 if there is an error in the classification. C is used to trade off maximization of margin with minimization of errors. After introducing the Lagrange multipliers, derivation with respect ξ and equating to zero, we can eliminate w, b and ξ in the minimization equation and obtain the following minimization (Bishope, 2006):

$$\min \sum_{l=1}^{M} a_l - \frac{1}{2}\sum_{i=1}^{M}\sum_{k=1}^{M} a_i a_k t_i t_k \kappa(\mathbf{x}_i, \mathbf{x}_k)$$

and the following constraints apply (Bishop, 2006):

$$0 \le a_l \le C$$

$$\sum_{l=1}^{M} a_l t_l = 0$$

After solving the above for langrage multipliers we have to find b to evaluate y(x) for an unknown sample.Due to its wide use and its ability to represent nonlinear data, we have chosen in this work to use Gaussian Radial Basis function (RBF) as a kernel. This Gaussian kernel maps the feature space to infinite dimensions and can be described by (Bishop, 2006):

$$\kappa(\mathbf{x}_l, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{2\sigma^2}\right)$$

where σ is the kernel width and its choice affects the behavior of kernel. The later parameter with the tradeoff parameter C should be optimized based on the training data by cross validation. Next we show how to convert binary classification problem solved by SVM to a multi class classification problem and perform cross validation to obtain RBF kernel parameters. This can be done by testing each unknown sample x with (Bishop, 2006):

$$y(\mathbf{x}) = \sum_{l=1}^{M} a_l t_l \kappa(\mathbf{x}_l, \mathbf{x}) + b$$

Finally we elaborate on the set of features used for classification and pre-processing steps (Alhmouz, Hassouneh, Guodang, & Ammar, 2012).

i. Features

1. Shape Context: Shape context is a statistical feature that explains for each point the distribution of angle and distance towards all points; this information is provided as a log-polar histogram for each point in the shape (Belongie, Mori, & Malik, 2006).

2. Distribution of Pixels: Other statistical features describing shape can be the distribution of white and black pixels: the clipped image is divided into upper, lower, left and right half and area of black pixels to white pixels is calculated (Abandah, & Khedher, 2009).

3. Boundary Descriptors: The object's perimeter length can be described by (Abandah, & Khedher, 2009). If f_i is the i-th chain code and N is total number of chain code elements, then perimeter length is(Abandah, & Khedher, 2009):

$$T = \sum_{i=1}^{N} L(f_i)$$

$$L(f_i) = \left(\left\{ \begin{matrix} 1, f_i \text{ is even} \\ \sqrt{2}, f_i \text{ is odd} \end{matrix} \right\} \right)$$

4. Perimeter length to diagonal ratio is a scale invariant feature

$$\frac{T}{2D} = \frac{T}{2\sqrt{W^2 + H^2}}$$

5. Bending Energy: Bending energy E is a measure of the curvature of the boundary. It can be found from chain code as (Young, Walker, & Bowie, 1974)

$$D(i) = f_i - f_{i-1}$$

$$\Delta p(i) = L(f_i) + L(f_{i-1})$$

$$E = \frac{1}{T} \sum_{i=1}^{N} (\frac{D(i)}{\Delta p(i)})$$

ii. Preprocessing

While not all classifiers need this step, pre-processing is an important step. Based on previous work in other languages (Jaeger, Manke, Reichert, & Waibel, 2001) for online handwritten English language and current research for Arabic handwritten words (Farooq, Govindaraju, & Perrone, 2005), preprocessing is required before recognition task for Arabic handwritten words. Specifically, Baseline detection, rotation correction, slant correction and size normalization for characters are preprocessing steps in most handwritten text recognition systems.

1. Base line Detection: One property of Arabic writing is a strong baseline. It is a virtual line connecting Arabic characters. Using this line we can perform rotation estimation and size normalization. There are many approach suggested for baseline estimation for Arabic words, we reference some works (Maddouri,

et. al., 2009), (Pechwitz, & Margner, 2002), (Farooq, & Govindaraju, 2005. While it was supposed to be a property of IFN/ENIT database (Pechwitz et al., 2002), baseline information is provided only with set (a) of this database. Thus we made advantage of this information in our experiments.

2. Rotation Detection and Correction: Baseline detection can help estimate rotations in a word. Never the less, other approaches can be used to detect the rotation without requiring baseline estimation. Note that Tilt in the baseline affects the segmentation of Pieces of Arabic words (PAW). In our work we used baseline information to estimate the rotation in the word and fixed the word before segmenting it to PAW.

3. Slant Detection and correction: People sometimes write characters with different tilt for each character; straight lines in characters are tilted in different angles for each character. Note this is independent of the baseline tilt. A simple shear transforms can be used to correct the slant angle once detected.

4. Size Normalization: The objective of this step is to normalize characters to the same size, thus improving the recognition task and making scale invariant features redundant. Information provided with IFN/ENIT includes top line information. Top line is a line touching top of characters and usually passes through diacritical markings in the word.

5. Removing Diacritical Markings: While diacritical markings provide valuable information once detected (Abandah, & Khedher, 2009), some of them are difficult to detect due to touching some characters, being misplaced, or being larger than some separate characters in the word. Simple heuristics are used to remove these markings based on the database in consideration (Mozaffari, Faez, M¨argner, & El-Abed, 2008). While not being very accurate approach but sufficient, we assume that the top line goes through the top diacritical markings (Mozaffari, Faez, M¨argner, & El-Abed, 2008). A threshold is imposed on the size of any single connected component touching the top line in order to remove upper diacritical markings. On the other hand lower diacritical marking are easier to remove, they are below the baseline and rarely touch characters.

CHAPTER 4: RESULTS

Here we show selected Results of the second and third approaches discussed earlier ASM and SVM classifiers: first, we show results for SVM and discuss the result, second, we conclude with the results on ASM and give some comments on the results

1. Support Vector Machines

SVM is a binary classifier; to convert this classifier to a multi class classifier several approaches can be used. The most effective approach is to decompose multi class problem into a many binary classification tasks; composition can take two approaches: one to many and one to one. The later approach is a popular choice for decomposition (Hsu and Lin, 2002). In the first approach, sometimes called one versus all, k binary classifiers are constructed for a k class problem. To train each classifier, all training samples from the i-th class are considered positives examples and all other training examples are considered negative examples. An unknown sample, a test sample, is tested on all the classifiers and the highest output classifier is considered to be the winner; the class that this classifier was trained for is announced as the wining class. Thus it is important to normalize the outputs of all classifiers in this approach. In the later approach, for a k class problem k(k-1)/2 binary classifiers are constructed; each pair of classes train a classifier. An unknown sample will be tested on all classifiers and the class with majority of votes is the winner; votes are calculated in different ways. For example when a classifier outputs a decision, either the winning class score is increase or the other class score decreased.

We use set (a) of IFN/ENIT database after performing preprocessing steps described earlier. Segmenting words into PAWs and then clustering similar PAWs resulted in 10 classes of PAWs; total number of PAWs was 1588 PAW. As mentioned earlier, we use bending energy, pixel distribution and shape context features to describe PAW. The number of bins in the log-polar histogram of shape context features is set to 12 bins for angle and distance; this number is recommended based on some characteristics of Arabic text (Fakir, Hassani, & Sodeyama, 2000). We divide the dataset into 70% and 30% subsets for training and testing purposes respectively. Märgner and El Abed noted that errors in systems participating in ICDAR2005 follow the frequency of PAWs in the dataset (Märgner, & El Abed, 2008); we used different number of training and testing samples for different classes but follow the 70%-30% rule for all classes. To solve a multi class problem, we choose one to one approach in decomposing the problem into 90 binary classifiers. For this propose, LibSVM tool (Chang, Lin, 2001) is used to construct and train classifiers. As we use Gaussian RBF kernel, we need to optimize the parameters of the kernel, C and σ using cross validation. While the tool's output for each unknown sample is its prediction for class membership, it provides an option to map the output of the classifiers to probability of each class; more details on mapping distance from the hyper plan to probability can be found in (Wu, Lin & Weng, 2004). The recognition rate for this experiment was 76.99% and the ROC curve for the classifier is shown below (Alhmouz, Hassouneh, Guodang, & Ammar, 2012). As we note from the curve the performance of the classifier is good; the curve is far to the left for most classes; this shows that the classifier is able to disambiguate the unknown samples.
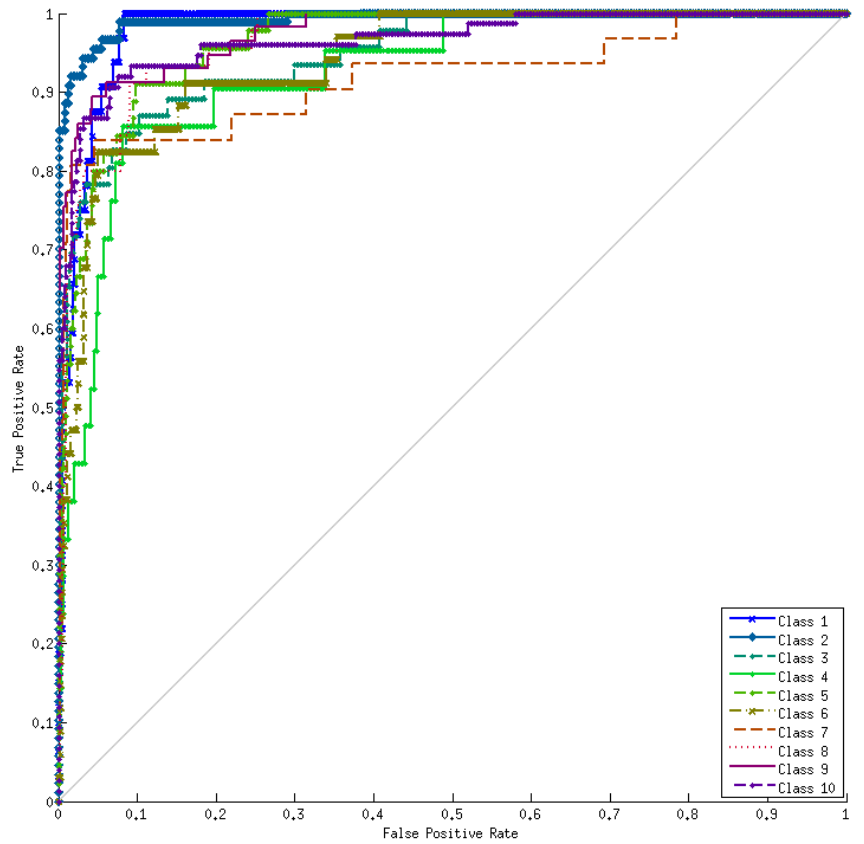
Figure 4.1 ROC for SVM classifier.

Next, we analyze the confusion matrix of the classifier to find the source off errors. As we see from the confusion matrix below errors are few for each class; there is no single class with high percentage of errors. However, confusion of a class of A as class B occur in bulks; we find few samples misclassified as class B.Even though the number of samples is not equal through the set, we do not note a relation between the error frequency and the frequency of samples in the set. Never the less, class number 4 has the lowest number of correctly classified samples.

17

	1	2	3	4	5	6	7	8	9	10	
1	20 4.2%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	4 0.8%	1 0.2%	0 0.0%	2 0.4%	0 0.0%	71.4% 28.6%
2	2 0.4%	81 17.1%	2 0.4%	3 0.6%	1 0.2%	0 0.0%	3 0.6%	1 0.2%	3 0.6%	0 0.0%	84.4% 15.6%
3	0 0.0%	1 0.2%	32 6.8%	4 0.8%	4 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	74.4% 25.6%
4	0 0.0%	0 0.0%	0 0.0%	7 1.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	77.8% 22.2%
5	0 0.0%	0 0.0%	3 0.6%	2 0.4%	32 6.8%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	5 1.1%	74.4% 25.6%
6	9 1.9%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	24 5.1%	1 0.2%	3 0.6%	1 0.2%	0 0.0%	61.5% 38.5%
7	0 0.0%	0 0.0%	0 0.0%	3 0.6%	1 0.2%	1 0.2%	25 5.3%	2 0.4%	0 0.0%	1 0.2%	75.8% 24.2%
8	0 0.0%	1 0.2%	3 0.6%	0 0.0%	1 0.2%	2 0.4%	1 0.2%	33 7.0%	3 0.6%	1 0.2%	73.3% 26.7%
9	1 0.2%	2 0.4%	1 0.2%	1 0.2%	0 0.0%	2 0.4%	0 0.0%	4 0.8%	48 10.1%	2 0.4%	78.7% 21.3%
10	0 0.0%	1 0.2%	5 1.1%	1 0.2%	5 1.1%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	62 13.1%	81.6% 18.4%
	62.5% 37.5%	93.1% 6.9%	69.6% 30.4%	33.3% 66.7%	71.1% 28.9%	70.6% 29.4%	80.6% 19.4%	73.3% 26.7%	84.2% 15.8%	82.7% 17.3%	77.0% 23.0%

Output Class (vertical axis) / Target Class (horizontal axis)

Figure 4.2 Confusion matrix for SVM

2. Active Shape Models

To be able to use ASM in character recognition (Lanities, Taylor, & Cootes, 1993; Saabni, & Bronstein, 2011) post processing stage is needed to use its output to classify objects or detect objects in an image. After convergence, a measurement is needed to specify how close input image to the deformed model. Lanities, Taylor, and Cootes have used simple measures, support measure and an explained evidence measure, never the less other have used Dynamic Time Warping (DTW) distance to measure the distance between the input image and deformed model (Saabni, and Bronstein, 2011). we have chosen to use the measures proposed by Lanities, Taylor, and Cootes (1993). These measures are as follows:

Similarity Measure: Xs =(1/ Model Points) *(Model Points ∩ Dilated skeleton of Image)

Evidence Measure: Xe =(1/ skeleton of Image) *(skeleton of Image ∩ Dilated Model Points)

Number of training samples for each class is 3-9; we have chosen to include free samples in some of the training data. There is 1588 PAW that represents 10 PAW classes. Pre-processing has been applied to PAWs as given in the pre-processing section. Parameters of the model are number of landmarks n=47, Coarsest level of Gaussian pyramid to search Lmax=2, bounds on eigen values is 2, and they explain f_v=98% of the model, search length for profile on

either sides ns=4 and profile length is k=7. The recognition rate obtained for this classifier is 20% and this rate is very low. As explained above, we optimize these parameters; in the next experiment we show the result after optimizing parameters of some models.

While we did not obtain an acceptable recognition rate, we can analyze the confusion matrix and the ROC curve to get an insight of problems statistical classifiers like the one in hand might face. Below we show ROC curve for the classifier; some of the classes have their curve close to the no-discrimination line.
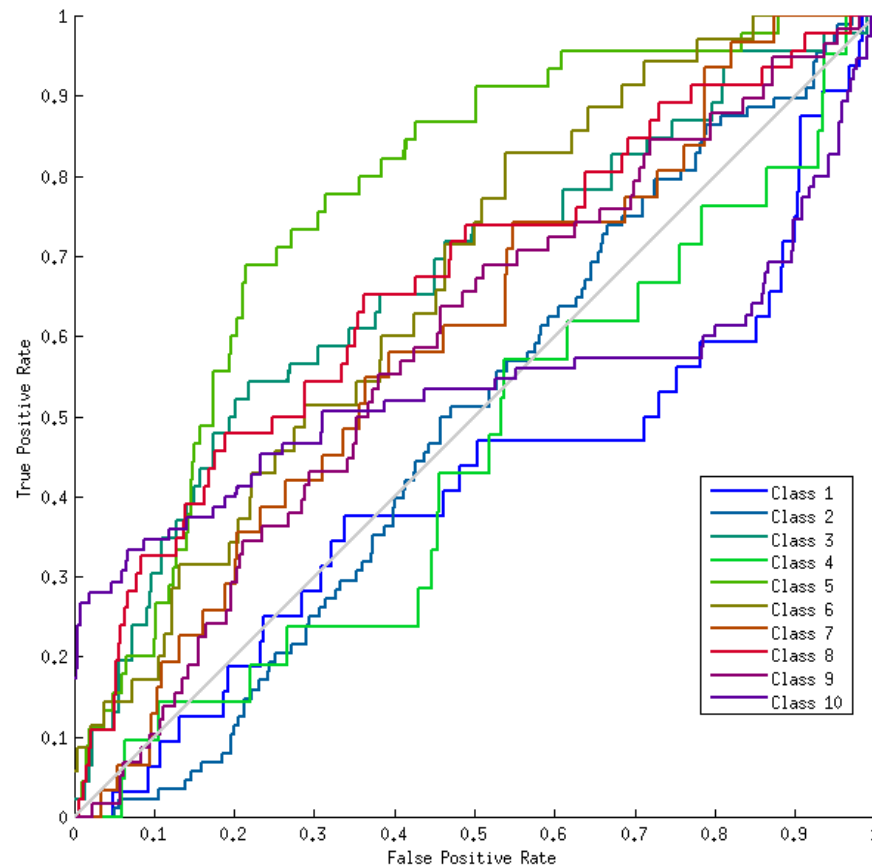
Figure 4.3 ROC for ASM classifier.

Thus we show the confusion matrix to obtain the classes with most number of errors. Generally, errors are spread out; number of errors is large and distributed on all classes. As can be seen from confusion matrix class number 4 is showing 0 samples recognized. This class has 70 samples, 30% are used for testing, and that is about 21 samples. Thus the classifier missed 21 samples, this behavior shows that either the similarity cost is not useful or the parameters used for searching model's fit is not set properly.

One way to optimize the model's parameters is to use cross validation over some samples of the same model. As explained in chapter 3, we use leave-one-out to find how well the model fit the test data; we choose m=6-9 for each model in this experiment. In Figure 4.5, we show confusion matrix after optimizing parameters of model 1,3,5,6 and increasing the number of training samples for model 1, 3, 8. We note from confusion matrix that model 1,3,8 discriminative ability has increase. Yet, discriminative ability for model 5, 6 decreased. Even though we did not change model 2, 7, 8, 9, 10 parameters, they have lower number of samples detected. This can be explained as follows: the fit probability of these models to their test samples appears lower compared to other models fit probability. Thus recognition rate has now decreased to 19%. Finally, we note for each class, less than half of the test samples is correctly recognized. Consequently, we think that the discriminative ability of the models is very low.

Figure 4.4 Confusion matrix for ASM

Figure 4.5 Confusion matrix for ASM for second expriment

CHAPTER 5: CONCLUSION AND FUTURE WORK

Recognition of Arabic words have many applications in the field of writer identification, word spotting,…, etc. The segmentation problem with Arabic writing imposes constraints on the way we can solve the Arabic handwriting recognition problem. Never the less, the best way to attack the problem is not clear tell now.

We have used SIFT to elude the segmentation problem but it turns out that locating characters in a PAW is a major problem that might increase the rejection rate. On the other hand, statistical modules we used are not discriminative. Finally we used SVM and argued to for its reduced complexity and its good performance. Part of SVM good performance was due to the correct set off features borrowed from Latin language recognition systems.

Using statistical modules is an interesting approach but requires optimization of model's parameters and a good choice for similarity measure. Yet, a robust less computationally demanding classifier that achieves good results is based on SVM.

While the results obtained were supported by similar results from the literature (specifically for SIFT), conclusions and performance limits found here are not conclusive; parameters of ASM can be optimized to achieve better performance and a bad choice of features can make SVM induce low recognition results.

As future work we have the following recommendations:

1. As noted in results section, providing a reject option for SVM is a very tempting option. This is due to the nature of errors the SVM classifier made; errors of type classify class A as class B occurs in bulk but are rear. Thus, a test on the confidence of SVM's decision can be made. If the confidence is below a certain threshold, then ASM can compare the unknown image to the model SVM has predicted. While this option is similar to classifier fusion, its complexity is reduced; only some unknown samples will be tested against a single model via ASM. Thus, it is possible to use anther algorithm to optimize ASM parameters to maximize the similarity cost function.

2. If the ASM classifier results are improved then classifier fusion is an interesting option. Both classifiers are able to out decision probability as an option, thus fusion techniques like sum and averages of decision probability (Kuncheva, 2004) becomes low complexity solution to increase the recognition rate.

3. Other statistical classifiers can be used, such as Hidden Markov Models. Lessons learned from ASM classifier encourage using other distributions to model the writing style deformations. Classifiers like HMM are based on modeling the distribution of individual states. Thus, we have more freedom in this case to choose models other than Gaussian.

BIBLIOGRAPHY

Srihari, S., and Setlur, S., (2006). Pattern Recognition at the US Postal Service: A Decade of Achievement. International Association for Pattern Recognition IAPR Newsletter, pp.3, 28(2). Retrieved from http://www.iapr.org/publications/

Saabni, R., El-Sana, J., (2011). Word spotting for handwritten documents using Chamfer Distance and Dynamic Time Warping. Document Recognition and Retrieval XVIII, 78740J.

Maddouri, S.S.; Amiri, H.; (2002).Combination of local and global vision modelling for Arabic handwritten words recognition. Workshop on Frontiers in Handwriting Recognition, 2002., pp. 128- 135.

Leondes, C.T. (Ed.). (1998). Volume 5: Image Processing and Pattern Recognition. Waltham, Massachusetts: Academic Press.

Rothacker, L., (2011). Learning Bag-of – Features Representations for Handwriting Recognition. (Diploma Thesis). Technische Universität Dortmund. Retrieved from http://patrec.cs.tu-dortmund.de/pubs/theses/da_rothacker.pdf

Mahmoud, Ahmad ,Alshayeb, and Al-Khatib. "A DataBase for Arabic Handwritten Text Recognition." Image Analysis and Recognition. Ed. Mohamed Kamel, Aurélio Campilho. ICIAR 2011, Springer LNCS, pp. 163-171, 2011.

Almuallim, H. and Yamaguchi, S., (1987). A method of Recognition of Arabic Cursive Handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(5).

Amin, A., (1997). Arabic Character Recognition. In Bunke, H. And Wang, P.S.P., (Eds.), Handbook of character recognition and document image analysis(397-420), River Edge, NJ: World Scientific Publishing Company.

L. Lorigo and V. Govindaraju, (2006). Arabic Handwritten Word Recognition: A Survey. IEEE Transaction on Pattern Analysis and Machine Intelligence,28(5), pp.712-724.

Lowe,D.G., (1999). Object recognition from local scale-invariant features. International Conference on Computer Vision, pp. 1150-1157

Vedaldi A., and Fulkerson, B., (2008). VLFeat: An Open and Portable Library of Computer Vision Algorithms. Retrieved on, April, 2012 from http://www.vlfeat.org.

Azad, P., Asfour, T. and Dillmann, R., (2009). Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition. IEEE RSJ International Conference on Intelligent Robots and Systems, pp. 4275-4280.

Rath, T., and Manmatha, R., (2003). Word image matching using dynamic time warping. Computer Vision and Pattern Recognition.

Saabni, R., and Bronstein, A., (2011). Fast Key-Word Searching via Embedding and Active-DTW. International Conference on Document Analysis and Recognition, pp 68-72.

Sridha, M, Mandalapu, D and Patel, M., (1999). Active-dtw: A generative classifier that combines elastic matching with active shape modeling for online handwritten character recognition. International Conference on Frontiers in Handwriting Recognition.

Woodard, J., Lancaster, M., Kundu, A., Ruiz, D., and Ryan, J. (2010) Writer Recognition of Arabic Text by Generative Local Features. 4th IEEE International Conference on Biometrics.

Rusiñol, M., Aldavert, D., Toledo, R. and Lladós,J., (2011). Browsing heterogeneous document collections by a segmentation-free word spotting method. International Conference on Document Analysis and Recognition, pp. 63–67.

Rothacker, L., Vajda, S., and Fink, G.A., (2012). Bag-of-Features Representations for Offline Handwriting Recognition Applied to Arabic Script. International Conference on Frontiers in Handwriting Recognition, pp.149-154.

Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T, Wu, D., and Ng, A. , (2011). Text detection and character recognition in scene images with unsupervised feature learning. International Conference on Document Analysis and Recognition, pp. 440–445.

Abandah, G. and Khedher, M.Z., (2009). Analysis of Handwritten Arabic Letters Using Selected Feature Extraction Techniques. International Journal of Computer Processing Of Languages, 22(1), pp. 1–25.

Harris, C. and Stephens, M., (1989). A combined corner and edge detector. 4th Alvey Vision Conference, pp. 147–151.

Belongie, S., Mori, G., and Malik, J., (2006). Matching with Shape Contexts. In Krim, H. and Yezzi, A. (Eds), Statistics and Analysis of Shapes, pp. 81-105, Birkhäuser :Boston.

Khalifa, M., BingRu, Y. and Mohammed, A., (2012). A robust SIFT feature for fast offline arabic words classification", IEEE International Conference on Computer Science and Automation Engineering, pp. 83-86.

Touj, S., Ben Amara, N.E., Amiri, H., (2007).Two Approaches for Arabic Script Recognition-Based Segmentation Using the Hough Transform. 9th International Conference on Document Analysis and Recognition, 2, pp. 654–658.

Margner, V., ElAbed, H., Pechwitz, M., (2006). Offline handwritten Arabic word recognition usingHMM—a character based approach without explicit segmentation. 9thColloqueInternationalFrancophonesurl'Ecritetle Document.

Zhang, Z., Jin, L., Ding, K., and Gao, X., (2009) Character-sift: A novel feature for offline handwritten chinese character recognition. International Conference on Document Analysis and Recognition, pp. 763–767.

Roy, V., Madhvanath, S, Anand, S., Sharma, R.R.,(2009). A Framework for Adaptation of the Active-DTW Classifier for Online Handwritten Character Recognition. 10th International Conference on Document Analysis and Recognition, ICDAR '09., pp.401-405.

Cootes, T., (2000). Image Processing and Analysis. (Ed.) Baldock, R. and Graham, J., Oxford: Oxford University Press.

Jaeger, S., Manke, S., Reichert, J., Waibel, A., (2001). On-Line Handwriting Recognition: The Npen++ Recognizer. International Journal on Document Analysis and Recognition, 3(3), pp. 169-181.

Lanities, A., Taylor, C.J., and Cootes, T.F., (1993). A Generic System for Classifying Variable Objects Using Flexible Template Matching. British Machine Vision Conference (BMVC'92), pp. 329-338.

Hill, A. and Taylor, C.J.,(1994). Automatic Landmark Generation for Point Distribution Models. Conference on British machine vision, 2, pp. 429 – 438.

Cootes, T.F., Taylor, C.J., Cooper, D.H., and Graham, J., (1994). Active Shape Models- Their Training and Application. Computer Vision and Image Understanding, 61(1), pp 38-59.

Bishop, C.M., (2006). Pattern Recognition and Machine Learning. New York, NY: Springer Science+Business Media.

Young, I.T., Walker, J.E., and Bowie, J.E., (1974). An Analysis Technique for Biological Shape. Information and Control, 25, pp. 357-370.

Farooq, F., Govindaraju, V., Perrone, M., (2005). Pre-processing Methods for Handwritten Arabic Documents. International Conference on Document analysis and Recognition.

Khalifa, M., and Yang BingRu, (2011). A Novel Word Based Arabic Handwritten Recognition System Using SVM Classifier. In Shen, Gang, and Huang, Xiong (Eds), Advanced Research on Electronic Commerce, Web Application, and Communication (pp. 163-171), Springer: Berlin Heidelberg.

Märgner, V., El Abed, H., (2011). ICDAR 2011 – Arabic Handwriting Recognition Competition. International Conference on Document analysis and Recognition.

Märgner, V., El Abed, H., (2008). Databases and Competitions: Strategies to Improve Arabic Recognition Systems. In Doermann, D. and Jaeger, S. (Eds), Arabic and Chinese Handwriting Recognition (pp., 82-103), Springer: Berlin Heidelberg.

Al-Muhtaseb, H.A., Mahmoud, S.A., Qahwaji, R.S., (2008). Recognition of offline printed Arabic text using Hidden Markov Models. Signal Processing Journal, 88(12), pp.2902–2912.

Märgner, V., El Abed, H., (2009). Comparison of Different Preprocessing and Feature Extraction Methods for Offline Recognition of Handwritten Arabic Words. International Conference on Document Analysis and Recognition, 2, pp., 974-978.

Maddouri, S.S., Samoud, F.B., Bouriel, K., Ellou, N., El-Abed, H., (2009). Baseline Extraction of Six Methods on IFN.ENIT Database. International Conference on Frontiers in Handwriting Recognition.

Pechwitz, M., Maddouri, S.S., Margner, V., Ellouze, N., Amiri, H., (2002). IFN/ENIT database of handwritten Arabic words. Colloque International Francophone sur l'ecrit et le document, pp.127-136

Lei, H., Govindaraju, V.,(2005). Speeding Up Multi-class SVM by PCA and Feature Selection. 5th SIAM International Conference on Data Mining,

Mozaffari, S. , Faez, K., M¨argner, V., and El-Abed, H., (2008). Lexicon reduction using dots for off-line Farsi/Arabic hand written word recognition. Pattern Recognition Letters 29(6).

Kroon, D. J., (March, 2011). Shape Context Based Corresponding. Retrieved from URL http://www.mathworks.com/matlabcentral/fileexchange/30845-shape-context-based-corresponding-point-models

Chang, Chih-Chung, Lin, Chih-Jen (2001). Libsvm: a Library for Support Vector Machines. Department of Computer Science, National Taiwan University, Taiwan. Retrieved from URL www.csje.ntu.edu.ta/~cjlin.

Kuncheva, L.I, (2004). Combining Pattern Classifiers: Methods and Algorithms. Hoboken, New Jersey: Wiley Inter Science.

Alhmouz, O., Hassouneh, Y., Guodang, G. and Ammar, H., (2012). Arabic Handwritten Recognition. To appear in International Computing Conference in Arabic.

Parvez, M.T., Sabri, A.M. (2012). Arabic handwriting recognition using structural and syntactic Pattern attributes. Pattern Recognition, 46(1), pp. 141-154.

APPENDIX

[Click and type text]

INDEX

CURRICULUM VITAE

[Click and type text]