Handré: An Experimental Approach to Intelligent Character/Word Recognition using Support Vector Machines & Dynamic Time Warping

Sang Woo Jun
wjun@mit.edu

Chong-U Lim
culim@mit.edu

December 6, 2011

Abstract

Optical Character Recognition (OCR) is the term used to describe the process of recognizing scanned images of text from documents, which may be handwritten manuscripts or printed text. An extension to OCR is Intelligent Character Recognition (ICR), which involves additional processing and recognition techniques in order to improve the accuracy of translating such documents by performing recognition on the level of words as opposed to individual characters. In this paper, we present an experimental system to outline the process of performing recognition of an entire handwritten document. Our approach outlines the feasibility of performing word recognition without having to collect hand-written samples for training. We make use of a windowed time-series method and pixel analysis to perform segmentation of the document invididual words and characters respectively. To perform individual character recognition, we used 2 different approaches – a kernel-based support vector machine (SVM) classifier and a dynamic time-warping (DTW) method which were both trained using a database of TrueType fonts. We then perform a committee based process of combining results from both models together with a probablistic spelling checker in order to perform the word recognition.

Contents

1 Introduction

1.1 Contributions

full-system approach

feature-dtw

2 Segmentation

2.1 Word Segmentation

2.1.1 Time-series Segmentation

A real-time edge detection method for times series was used to extract individual word images from a scanned image of a document. The image is first scanned vertically, constructing a time series consisting of the number of black pixels per vertical position. This information is analyzed to separate each lines into individual images. Each segmented line image is then scanned horizontally, constructing a time series. This is analyzed to segment individual words.

In order to accurately segment lines and words, we adapted a time series edge detection method proposed to segment heartbeats to identify separating spaces between lines and words. This is done by first differentiating the time series and then doing a moving window integration, to discover the transition point separating the dense and sparsely colored regions. This generally resulted in more accurate results than simple pixel density analysis employed by most work.

Figure **??** shows the plot of a processed time series derived from pixel density of a single line image and its resulting word segmentation. This image was taken from a screenshot of one of professor Leslie Kaelbling's lecture notes.

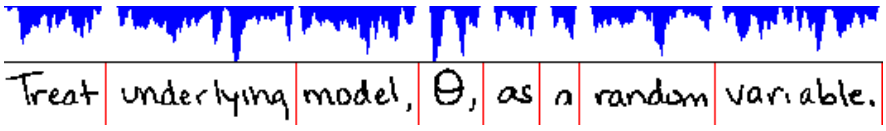

Figure 1: Word segmentation of an example line.

2.1.2 Post-processing

Because scanned handwritten documents contain many types of noise information such as underlines and shadows, a certain amount of post-processing on the segmented word image to obtain relatively clean character data. First, single pixel noise was eliminated by removing pixel 'islands', that existed by itself with only one or no neighboring pixels. Second, underlines were removed by picking the top one or two horizontal lines with a very high pixel density and removing each pixel from it, unless it had pixels above and below it. Third, strokes spilled over from lines above one in interest was removed, by scanning all pixels on the first row of the image and recursively filling all adjacent black pixels with white.

2.2 Character Segmentation

Because individual characters in a handwritten word often have overlapping widths or just plainly connected, the same method of using pixel density cannot be used to segment characters in a word. Furthermore, it is provably impossible to perfectly segment characters in a word without symantic context. For example, it is just as likely for 'm' to be segmented into 'm', as it is to be segmented into 'rn' or any other subsection of 'm' that may not exist in the english alphabet.

However, because this research is a preliminary feasibility search that does not heavily focus on character recognition, we employed some preprocessing by hand and simple pixel analysis to perfectly segment characters in a word. First, word images were preprocessed by hand, by deleting some pixels so that no characters are touching each other. Then, simple pixel analysis was used to exploit this feature and correctly segment characters. This method was useful because it was much more productive than hand segmenting every character manually.

3 Recognition

In this section, we outline the techniques which we focused on in order to address recoginition task of our project. In Section 3.1, we describe desribe support vector machines (SVM) and how they can be used for classification of the different alpha-numerical characer by training on a dataset in order to learn a model which generalizes well for all characters.. Section 3.2 covers dynamic time warping, and describes our approach to using it to classify alpha-numerical characters by minimising the distance of difference between the pixel-representations of the different characters. Section 3.3 explains how a spelling checker is implemented in order to act as a type of regularization of the combined results from the SVM and DTW systems.

3.1 Classification using Support Vector Machines

Support Vector Machines (SVMs) are a form of supervized learning methods which can be used for classificaiton or regression problems. In a binary classification example, we would train the SVM on a labelled dataset and if they they are linearly separable, the SVM will find a unique separation boundary in the form of a hyperplane with points falling on each side having different classifications. The separation boundary would be one in which the margin is maximised.

In general, not all data points will be linearly separable often as a result of overlapping class-conditional probabilities. Also, there is a chance of overfitting on the training data points which might negatively affect the generalizability of the classifier for future points. As such, we adopt the use of *slack variables* which results in a 'soft margin', in which we allow some data points to be incorrectly misclassified with a certain penalty with the aim of overcome overfitting. The general formulation of SVMs as constained quadratic programming problem is as follows

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & C\sum_{i=0}^{n}\xi_i + \frac{1}{2} \parallel \theta \parallel^2 \\ \text{subject to} \quad & y_i(\theta \cdot \mathbf{x}_n + \theta_0) \geq 1 - \xi_i,\ i = 1, \ldots, m \end{aligned}$$

where x_i represents each training data point, with y_i being its corresponding target classification. θ is the model, or parameter, of the classifier with offset θ_0, while C and ξ represent the penalty and slack variables respectively.

3.1.1 Supporting Multiple Classes

In our project, we are interested in the use of SVMs to classify individual characters based the a given vector input of pixel values. Extending SVMs to support multiple target classifications requires us to train multiple binary classifiers, one for each individual target character. Given an input vector, each classifier would give a possible classification, and we then employ **voting** as a way of deciding the best classification result for our data. The common voting strategies to decide on a classification described as follows.

One-versus-one In one-versus-one voting, the idea is to fit a classifier for each pair of classes, and when it comes to making the prediction, we select the class with the most number of votes.

One-versus-rest In one-versus-rest voting, each character has a classifier which is fit to it. When making the predicition, the class with the highest classification output is chosen in a *winner-takes-all* strategy.

3.2 K-means Clustering using Dynamic Time-Warping

The second method used for character recognition is K-means clustering, using modified dynamic time warping as a distance metric. K-means clustering is an efficient method to use when classifying an input to one of the K possible clusters, which is a good fit to character recognition. Dynamic time warping (DTW) is a dynamic programming method to calculate the similarity between two time series by warping each time series into various ways to discover the warp configuration that makes the two series most similar. This is also a good fit for character recognition because characters are generally similarly shaped, but it is stretched or warped in various ways.

Modified Dynamic Time Warping We were unable to use multi-dimentional dynamic time warping for our goals, which is a common form of modified dynamic time warping used for image and texture analysis. Multi-dimentional dynamic time warping is done by taking the euclidian distance between each possible set of two rows, from both images and using dynamic time warping on it. However, this method could not be used because of two reasons. First, character sizes were often very different, making euclidian distance between rows not a very accurate measure. Second, it took a prohibitely long time.

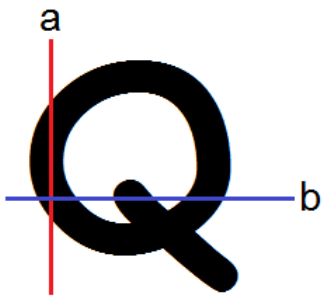

Figure 2: x and y features at position a and b

Therefore, we developed a novel method of character distance metric called feature-dtw, which has not been used in previous works, to the best of our knowledge. This method first generates two feature time series per character image, *xfeature* and *yfeature*, which are time series consisting of the number of pixel lumps that exists on that particular row or column. For example, in Figure **??**, the item in xfeature at index a would be 1, and the item in yfeature at index b would be 3. This feature series can be expected to have enough information in them, so when dynamic time warping is used on them, they can give a close approximation to the level of difference between two characters.

Because feature-dtw is not completely accurate, it is only used as the first tree in a forest of classifiers, to first prune the problem space by taking the top 10 or so of most likely characters. Feature-dtw can often mistake some different characters to have the same features, depending on the tilt or shape of the characters. So after taking the top 10 or so of most likely characters

from feature-dtw, traditional multi-dimentional dynamic time warping is used to actually classify the character.

Modified K-means clustering Because the level of difference between characters differ by lot, the default method of K-means clustering yielded multiple clusters with different characters in the same cluster, or with the same character split among different clusters. Therefore, we modified the K-means method to fit better into our ICR interface by picking character instances from different computer fonts that approximately separated the characters as much as possible, and using it as a 62-means cluster (26 + 26 upper- and lowercase letters and 10 digits).

3.3 Combining Approaches for Word Recognition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

3.4 Spelling Corrector

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

4 Experimental Design

In this section, we desribe the experimental design which we undertook in order to apply the segmentation and recognition techniques from Sections 2 and 3. In Section 4.2, we describe the prodedure in which we trained the multi-class SVMs on training data which consisted of different images of each alphabet character for a variety of fonts. Section 4.3 outlines the application of DTW onto a single font set and how it was used to perform its classification. How the results from both experiments were combined together to make committee-based decision is covered in Section 4.4. Finally, Section 4.5 explains the use of the dictionary spelling checker to improve the result and come up with a final decision.

4.1 Training & Testing Data

For the purpose of our project, we tried to locate a resource of training data for alphabets which was commonly available. Several repositories containing data were either unsuitable or was paid-only. More importantly, we wanted to show-case that system could make use of a generic database for its training data, and not have to resort in having a particular target domain to have to first provide us with both a vast number of examples, but also go through the tedious task of hand-labelling them though it given the resources and time, it might be a possible way to improve the system (Section 6). As such, we decided to using readily available TrueType fonts as our training data.

4.1.1 Using Font Images as Training Sets

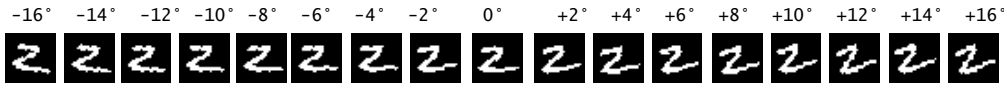

Figure 3: 17 different rotations for each alphabet character

We made use of fonts for our training data by randomly picking a set of handwritten fonts which are freely available on the web (see Appendix A.1 for the list and samples.) In order to normalize the characters, we resized each character to fit as a 28x28 white-on-black square image[1]. The character is then set to black-and-white mode, in which the pixel values are either 0 (black) or 255 (white). For each character in the font, we also created additional samples of the character image by rotating the characters by an angle between [-16°, +16°]. Thus, each training sample can be viewed as a 1x784 vector of pixel values. Figure 1 shows an example of a font character which has been normalized and its various rotations.

[1]We referenced the dimensions used by the MNIST database, and serves as a reference point for common handwritten character sizes.

4.1.2 Testing on Handwritten Text

Figure 4: Segmenting and normalizing a scanned image into its individual characters

Image segments containing words which were obtained from the word-segmentation step (Section 2) were used for tests. The images were then cleaned and individual characters were extracted for the recognition process. We perform similar normalization steps in order to make the test images have the same dimensions and format as our training data – images have their colors inverted, are cropped and resized to 28x28 pixels. Using our optimal performing classifier, we then attempt to classify each individual character in the word of l characters. An example of a test image is shown in Figure 2.

4.2 Recognizing Alphabet Characters using Multi-Class SVMs

In order to be train the best classifier for our recognition task, we varied the parameters and kernels used and performed cross-validation in order to select the best performing one. The following list describes the various parameters and choices we used

- Radial Basis Kernel
  - C = [1, 10, 100, 1000]
  - γ = [0.01, 0.001, 0.0001]
- Linear Kernel
  - C = [0.001, 0.01, 0.01, 1, 10, 100, 1000]

We optimised the parameters based on the precision score and recall score, and the results of which are shown in Section 5.1. Additionally, we made use of both the 1-versus-1 strategy and 1-versus-rest voting schemes to perform the multi-class classification. We also obtain the probability scores of each class (signifying how confident we are of our classification) to obtain a table containing l rows, and each row contains a vector of 52 probabilities – one each for the alphabets (26 upper-case and 26 lower-case characters.) As covered later in Section 4.4, we will make use of this probability table to make a combined decision, together with the DTW method, in order to improve our word recognition results.

4.3 K-means Clustering using DTW

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

4.4 Combined Character recognition

With the resulting probability tables from the trained SVMs (Section 4.2) and the DTWs (Section 4.3), we then developed a policy to select the resulting individual characters. First, each row in the probability was ordered from high to low – with the most likely prediction in the front. Next, each character to be classified in the target string, we selected a subset from both the tables as a combined prediction by taking the intersection of the two sets. In the event that no characters are common between the two sets, we picked the most likely members from each set.

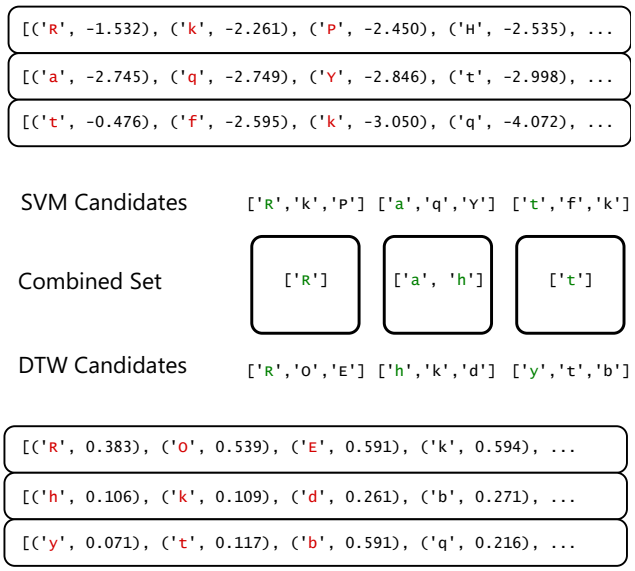

Figure 5: Illustrating how predictions from both systems are combined.

This is illustrated in Figure 3. In this example, the scanned image contains the word ‘Rat’.

The prediction table for the SVM system is shown at the top, while the table for the DTW system is at the bottom. For each character to be guessed, a subset of the top 3 predicted characters are chosen from both tables, so for the first character, we have the set $['R','k','p']$ and $['R','O','E']$ from the SVM system and DTW system respectively. Then, the intersection of the subets are assigned for first character of 'Rat', which is 'R'. For the second character, as there are no intersecting characters, we pick the top two predictions to form the subset $['a','h']$. The third character 't' follows the same discussion as the first character 'R'.

After some experimentation, we emperically decided that selecting subset of 3 characters from each prediction table row was the best policy due to potentially large number of permutations that might result from taking more. In the case where the intersecting set is empty, we picked the top predicted character from each of the tables.

4.5 Combined Word recognition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

5 Results & Analysis

In this section, we provide the results from the experiments conducted for the project which were explained in Section 4.

5.1 Alphabet Character Recognition using SVMs

5.1.1 Optimal Tuning Parameters

5.2 K-means Clustering using DTW

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra

purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

5.3 Combined Character recognition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

5.4 Combined Word recognition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

6 Conclusions & Future Work

6.1 Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

6.2 Future Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque posuere molestie metus. Suspendisse tellus urna, porta sit amet rutrum eu, tristique quis urna. Donec varius pharetra purus, eget mollis tortor ornare vel. Nullam sagittis tellus id dui placerat eget congue libero facilisis. Donec mattis sagittis lectus, eget porta quam facilisis vel. Vestibulum non urna ante, nec mattis mauris. Nulla sit amet interdum eros. Nam congue lacinia nulla, vitae aliquet nisl tincidunt vel. Morbi gravida bibendum ipsum, at accumsan nisl suscipit ac. Sed accumsan cursus tortor a faucibus. Phasellus tempus, orci ac lacinia hendrerit, dui justo accumsan mi, congue dapibus massa turpis at lectus. Cras a tellus nisi. Aliquam vitae dolor id nunc lacinia fermentum et sit amet metus. Nullam viverra ante eu mauris ultrices nec adipiscing lectus dapibus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam mollis commodo lacus, eget bibendum risus lobortis nec.

7 Acknowledgements

We would like to thank Dahua Lin for reviewing our initial project proposal, and subsequently taking time out of his busy schedule to meet with us and provide guidance, ideas and tips on approaches tp the project. Also, we thank Prof. Leslie Kaelbling for imparting her wisdom and stimulating our ethusiasm in the topics of machine learning.

A Data & Results

A.1 Font Samples

B Project Timeline

Week 1: 10/28 – 11/5

- Submit project proposal
- Narrow down choices to one for the project
- Read up on related papers/books

Week 2: 11/6 – 11/12

- Continue reading related works
- Word Segmentation investigation
- Numerical digit classification investigation

Week 3: 11/13 – 11/19

- Character segmentation
- Dynamic time-warping experiments
- Alphabet characters classification investigation

Week 4: 11/20 – 11/26

- Font database
- Dynamic time-warping for classification
- SVMs for training on fonts and classification

Week 5: 11/27 – 12/5

- Improving individual results
- Heuristics for combining both predictors
- Dictionary spell-correction
- Report writing

C Division of Labor