Restaurant Menu Expert

A digital image processing pipeline to increase the accuracy of the state-of-the-art OCR algorithm

Chung Yu Wang (chungyuw@stanford.edu)
Yu-Sheng Chen (yusheng@stanford.edu)
Ziyu Lang (zylang@stanford.edu)

***Abstract*—We have developed a user-friendly system for non-English speakers that translates the text in English restaurant menus into pictures of the dishes. The system uses the photo of a menu that users take with a camera as an input, processes it then retrieve the corresponding pictures of the dishes in database, and finally displays the pictures to users in order to help them get a clearer idea of what they are looking at. To build a robust menu recognition engine, we adopt several pre and post processing techniques to increase recognition accuracy. We correct the input image's rotation by performing featureless rotation and reduce search space and noise by identifying the ROI. Since text recognition is inherently also a language model based problem, we further provide a post-processing edit-distance error improve accuracy from the Tesseract OCR engine. We use three sample menus to test the effectiveness and robustness of our system. Our experimental results show that our system can attain 50% higher in accuracy rate than the plain OCR recognition without our pipeline. With some tolerance of edit distance (i.e., 9 characters), the match success rate could reach up to 90%.**

Keywords—Character Recognition; Featureless Rotation; Segmentation; ROI; minimum edit distance

I. INTRODUCTION

One of the biggest challenges when traveling is the language barrier. This is especially a big problem when ordering dishes in a restaurant. Because the uniqueness of the names of the dishes, they usually have specific meanings and correspond to distinct dishes, which are hard to imagine by only reading the text on menu. Even though people may have some basic understanding of the literal meaning of theses names, the dishes can be totally different from what they think of due to cultural differences. In these scenarios, presenting the pictures of dishes instead of presenting the names of the dishes might be more helpful for people to make good decisions on what they would like to order.

Inspired by the idea of helping people lower such hurdle, in this project, we tackled the problem of providing a pipeline that automatically display the dish images beside the queried dish names. With this application, people just need to simply put a menu in front of the camera, take a photo of the menu, let the program process the image and recognize the characters in the menu and finally present the images of the dishes on the side of the original dish names on the menu in order to help the users have more straightforward understanding of the dishes they are interested.

II. RELATED WORKS

There are many implementations related to this topic, which gave us a great insight and inspiration when forming our idea. A. Heng described an iPhone application which is designed to quickly and easily split a restaurant bill amongst a group of people in his paper [1]. The application uses the Tesseract OCR engine to extract words from the receipt, then performs text processing to define individual items on the receipt. This application can effectively reduce the time of figuring out how much one has to pay from a group check. Based on the observation that paper receipt is still irreplaceable and there is no easy mean to convert it into an electronic format despite many advanced electronic payment systems exist, C. N. Nshuti discussed in his paper how to realize the digitalization of paper receipts and developed a pipeline for performing OCR on the picture of a document taken by a mobile phone [2]. Also, there are many applications aim at recognizing the character from foreign languages. Base on the problem that many existing OCR approaches don't work well on some character based languages such as Chinese and Japanese, Zhang et all studied the algorithm to realize the robust and efficient Chinese character recognition using SIFT feature and RANSAC method, and then applied it to Chinese restaurant menus to create a new mobile application that translates pictures of restaurant menu items in Chinese into photos of the entrees in real time [3]. The results of their experiments are very inspiring: the method is able to perform fast Chinese character recognition and find a matching entree picture within 5 to 6 seconds with a success rate of about 91 percent for clear and focused camera inputs.

Our study focuses on developing different methods to increase the accuracy of the OCR algorithm, and then building a robust and real-time English menu translating system for non-English speakers. We built a database for a predefined set of dishes that are commonly seen from various parts of the world. The system is robust against noise, rotation, and different fonts. We implemented several techniques we learned in class, including thresholding, dilation, erosion, segmentation, merging, etc. The technical approach pipeline is described in section III. We are concerned about the effect of different preprocessing techniques. As such, we performed detailed comparison of the results, which is described in section IV. In section V, we discussed the local and overall performance of our menu recognition system in terms of several system parameters. In section VI, we compared our system with other implementations in related work and discussed the advantages and shortcomings of our system. In section VII, we concluded

our valuable experience with this project and gave an outlook to future work.

III. Technical Approach

The application pipeline contains 6 main procedures: inputting the menu image (photo of the menu), preprocessing on the input image, which contains two parts: featureless rotation and string segmentation, performing optical character recognition (OCR) on menu texts, string matching with the database, and finally displaying the results.

A. Image Input

The input phase is quite straightforward: a user takes a photo of the menu with a camera. After user captures the image, we provided an interface where the user may optionally drag the desired region of interest (ROI) to specifically find out the dish images of certain dishes on the menu.

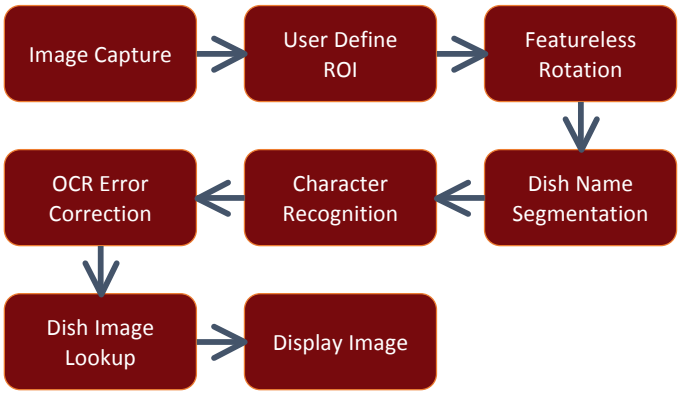

Fig. 1. Menu Recognition Pipeline

Photos taken in the natural scene have a great diversity and significant uncertainty, which leads to difficulty of recognizing characters from them. For example, characters in different images may have different sizes, fonts, colors, rotations and so on. There are other variables that may result from the environment such as motion blur and out out-of-focus image. These are definitely challenging problems, but we assumed that the user would have full control over the image quality at the time of image capturing; hence not part of our focus of the project.

B. Featureless Rotation

After user defined the rectangular ROI, we automatically find out the rotation angle based on the inherent structure of menu where most texts are horizontally aligned. To do so, we first applied global Otsu thresholding and applied dilation with a small disk of 5 pixel in diameter. The dilated text will form connected components between characters and as such we can easily identify the connected components and find the smallest rectangular region that bounds the connected component. With these bounding boxes of the connected components, we calculate the mean aspect ratios between the width and the height of the bounding boxes at every 10° from -90° to 90°, with subsequent refinement of every 1° to pinpoint the angle of rotation. From these set of rotations, we automatically find the largest mean aspect ratios, as that suggests that most bounding boxes are horizontal at that particular angle. We limited our rotation correction to be between -90° to 90°, as we assume that the user will operate our menu detection pipeline within reason.

C. Dish Name Segmentation

To facilitate and improve the accuracy from subsequent Optical character recognition (OCR) engine, we preprocessed the image by first identifying the bounding box for each dish name in ROI and segmenting out the dish names and pass it into the OCR engine.

For our purpose, we kept the entire dish name to be within one bounding box, such that the subsequent OCR pipeline may potentially utilize such information. To do so, we created the connected component by dilating the texts with a horizontal line structure on the Otsu binarized image. With such structure, all texts within one line will form one single connected component and hence within one single bounding box.

To allow minor variations in text alignment and be more robust to noises, we applied thresholds on the bounding box aspect ratios and areas and merged nearby bounding boxes together to form one single bounding box. These thresholding and merging techniques are similar to the underlying implementation of the Canny Edge detection where non-max suppression and edge linking are applied to reduce noise.

D. Optical character recognition (OCR)

After we get segmented dish name texts within one single bounding box per line, we adopt MATLAB's implementation of Tesseract algorithm, an open source OCR engine initially developed at HP Labs and currently managed by Google [4], in our project to perform character recognition. The basic principles of Tesseract OCR are as follows: First, character outlines are extracted and assembled together into Blobs by performing connected component analysis. Then the lines of text, which are formed by blobs, are split up into separate words depending on the spacing between each character. The following stage is known as word recognition, which is a two-pass process. In the first pass each word is recognized in turn. Once a word is recognized, it will be stored in an adaptive classifier and be used as training data. In the second pass the words that weren't successfully recognized are recognized again by using the training data obtained from the first pass. Finally, a string of words is given as an output.

E. Dish name matching with database

Dish name matching is performed after OCR string results are obtained. At this stage we may expect some misspellings coming from the OCR results, and we must correct them in order to find a match in our name list of image database. Here we utilize the algorithm of finding minimum edit distance. Given acceptable number of character mismatch, our system is able to correct the results from the OCR engine, which increases the overall success rate of image lookup. But high complexity of searching minimum edit distance is the main problem we should cope with. The related analysis and improvement is covered in section IV.

F. Display the final result

After looking up all recognized dish images from the database, we resize the dish images according to the bounding box positions and its width/height. And paste them on the observed menu image ROI. For those OCR strings that cannot match any dish name, no images are pasted. This also can reduce the chance of showing up some unreasonable images due to unpredictable OCR errors or menu input

IV. Results

In this section, we present our experimental results of our pipeline and discuss the strengths and possible improvements to improve the performance.

A. Results of Entire Pipeline

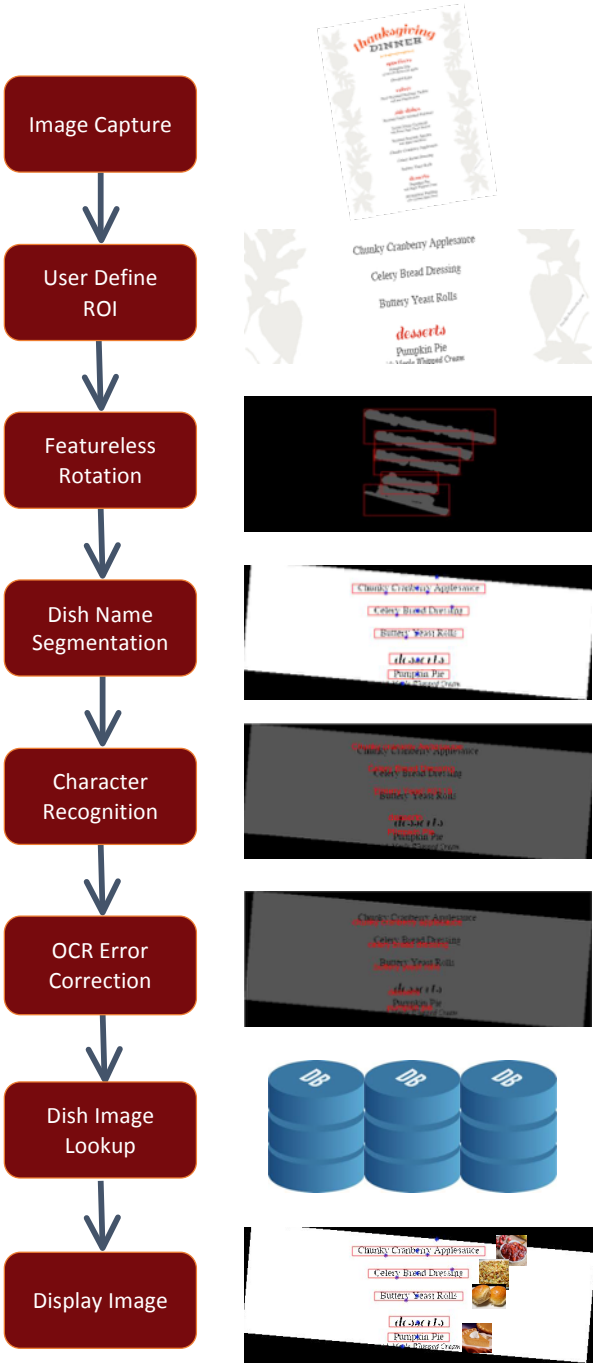

Fig. 2. Menu Recognition Step-by-step Results

Above figure shows the results of each step in our pipeline, incrementally adjusting the inputs into the OCR engine and fixing the typos from the OCR result to displaying the final images beside the dish names.

B. Without Rotation Correction vs Rotation Correction

We compared the results between passing into OCR engine the segmented dish names with various original angle of rotation without and with proposed rotation corrections.

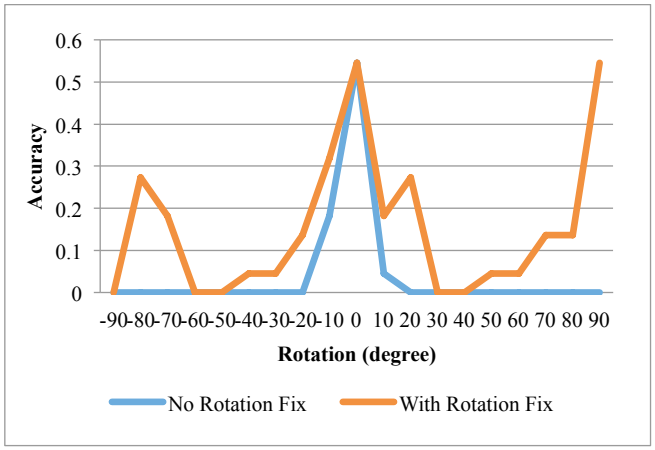

Fig. 3. Rotation Correction Comparison on Thanksgiving Dinner Menu

From the above figure, we see that the rotation correction technique consistently outperforms the technique without rotation correction. At first, we expected that the accuracy should remain about the same within +/- 20° and with a steady decrease in accuracy as the magnitude of the rotation increases. However, this was not the case and the main reason was due to the drop in resolution of the image as we conduct rotation correction. At certain angles, the rotation is a lossy operation, and hence the subsequent OCR engine cannot correctly identify the words.

C. Passing Full image vs Segmented Image into OCR

We compared the results between passing full image to the OCR engine and segmented dish names into the OCR engine and see the resulting accuracy.

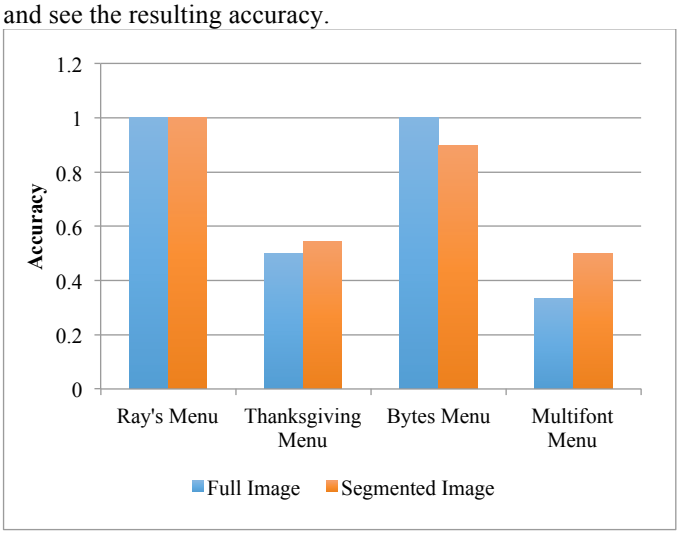

Fig. 4. Image Segmentation for OCR Comparison

From the above figure, we see that passing in the segmented images to the OCR engine performs better than passing in the full image, except for the Bytes Café menu. The decrease in accuracy in Bytes Café menu might be attributed to the MATLAB's OCR engine, which is built on top of the Google's open source Tesseract algorithm. Since the Tesseract algorithm would run through the query image twice, the first time for adaptive classification and the second time using the information from original database and the training data from all the words found in the first round, by segmenting the dish names, we have lost such benefits of using adaptive classification from other words. Nonetheless, we have shown that in other cases, such loss of information from the first

round of adaptive classification for segmentation may boost the accuracy of the OCR.

D. OCR Correction by Minimum Edit Distance

After OCR for each bounding box of image, the recognition result may contain some misspelling due to image noise, motion blurring, wrong estimate of bounding box, and false positive of character recognitions. To maximize the success rate of matching the image database, we perform OCR correction using minimum edit distance technique.

To correct misspellings, we find the closest matching dish name based on Levenshtein's edit distance for each dish name identified by the OCR engine. We implemented such algorithm with the parameter of maximum tolerance of edit distance, aka cutoff distance. As it turns out, edit distance measurement is the part with the highest complexity. So it encourages us to seek any possibility to reduce the complexity of OCR correction, without loss of the success rate of OCR correction.

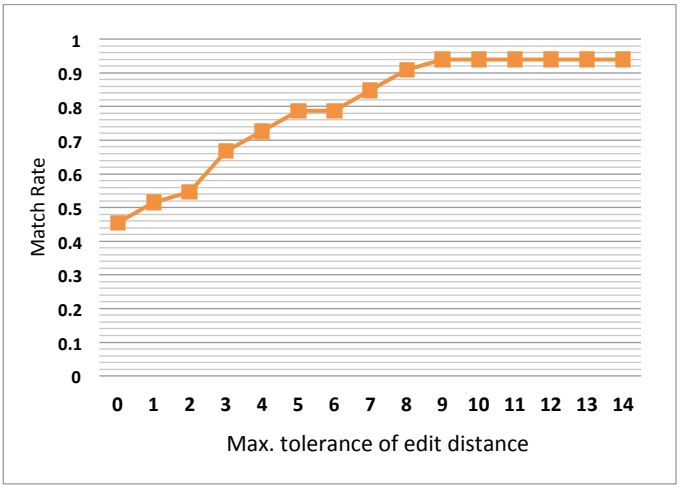

Fig. 5. OCR correction by minimum edit distance

As shown in the figure above, the match success rate is lower than 50% with exact string lookup in the name list of image database. But with little tolerance of edit distance (i.e., 5 characters), the match success rate rises rapidly to 80%, and finally saturates around 90%, the reason of saturation is that even when we increase more tolerance of edit distance for string comparisons, it's very likely the recognized strings with min edit distance > 9 are totally wrong and not able to match the correct names of images regardless of the max tolerance.

Based on our observation, the best cutoff for the max tolerance is 9 characters, the recursion complexity of the algorithm of finding minimum edit distance can be limited by this distance range, any recursive calls with distance > 9 will be terminated, which greatly saves us total complexity of OCR correction, especially for the match failure cases.

E. Time Complexity

From the figure below, we can see the total time complexity of the overall system. The most time consuming part is the OCR correction. The rest of the processes only accounts for less than 15% of the total execution time. To further improve the time performance for real-time demonstration purpose, we came up with several enhancement methods to save time complexity of OCR correction.

The first method is to modify the min-edit-distance function, rather than creating recursion for comparison of each character, we iterate through the character comparisons until encountering mismatch, then split into 3 recursion calls (delete, insert, replace). This reduces complexity significantly as the saving is magnified by the length of dish name list in the database, the number of recognized strings from OCR, and even the average length of the dish names.

The second improvement concept comes from the system view. Once we find a name on the dish name list which is perfectly matched with the recognized string from OCR, then we can stop going through the rest of the dish name list. In general, this strategy cannot benefit most of the challenging cases, but it greatly helps the average cases and makes the total execution time short enough for real-time demonstration.

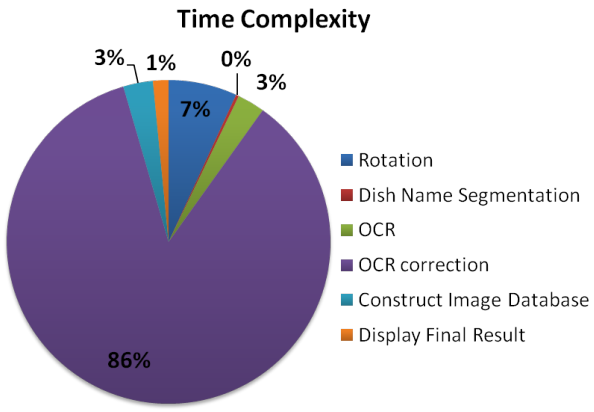

Fig. 6. Time complexity analysis of the overall system

V. Evaluation of System Parameters

In this section we discuss the impacts of several system parameters on the local and overall performance of our menu recognition system. In particular, we discuss the impacts of selecting different sizes of the structural elements in the rotation correction and dish name segmentation parts of the pipeline.

A. Changing Dilation Size for Rotation Correction

For the rotation correction, we first performed dilation on the image in order to connect characters to form words while preserving the general contour of the words. In this process, we want to find the angle of rotation with the maximum mean aspect ratio on the bounding box of each connected component, as these connected components are English texts that are horizontally aligned. Hence, we used a relatively small disk element of 5 pixels in diameter to perform such task. For checking the structural element size sensitivity, we passed in a menu that is rotated at 10° and analyze the effect of different structural element size.

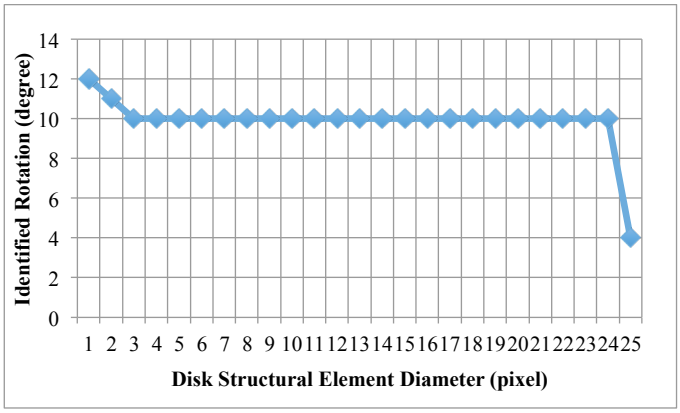

Fig. 7. Rotation Correction Sensitivity to Structural Element Diameter

In the above figure, we can see that without any dilation at all, the identified angle of rotation may be slightly off from the actual angle of rotation, this is because many characters are now considered as one connected component; hence the assumption that aspect ratio is the largest when rotated the image back to the actual angle is no longer valid.

We also see that when the structural element diameter is too big (above 25 pixel in this case), the identified angle of rotation is incorrect as well. This is because all the words in the menu and the border of the menu is grouped into one single connected component and hence the aspect ratio does not represent the bounding box around a single English word, and hence our assumption also fails in this case.

B. Changing Dilation Structure Size for Name Segmentation

Similar to the rotation correction, dish name segmentation also heavily relies on the size of the structural element. In this case, we opted horizontal structural line element to group all the words in a dish name into one single connected component. We experimented various line widths and present findings in the following figure.

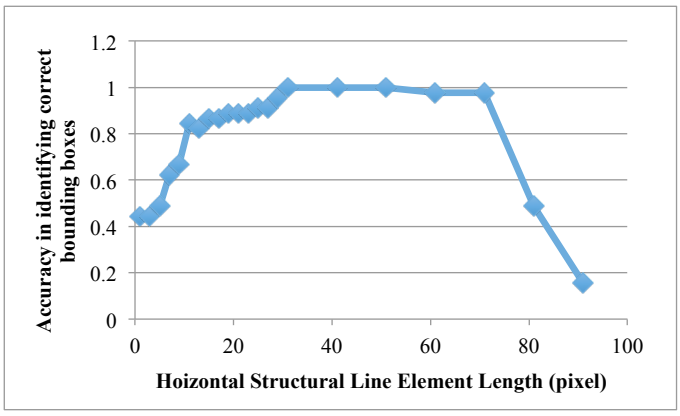

Fig. 8. Segmentation Sensitivity to Structural Element Line width

In the above figure, we can see that we only obtain 100% accuracy in identifying correct bounding boxes around dish names for structural element lengths between 31 and 51 pixels. The main reason for lower accuracy at a smaller element lengths is because of over segmentation, specifically words from a single dish are not in the same bounding box. On the other hand, when the line structural element is too long, then the dilated objects encroach the borders, making the border and the dilated text one single connected component, resulting in incorrect bounding boxes.

VI. Comparison to Alternative Approaches

It can be seen from the experiment results that the pipeline of our project have certain advantages compared to other existing implementations. Most importantly, it shows good robustness, which is very important in practice, against noise, rotation, different fonts, etc. by adopting Tesseract OCR and performing several pre and post processing techniques, including featureless rotation, text segmentation, and minimum edit distance correction. OCR is a more accurate faster way to realize character recognition than feature matching methods like SIFT because there are too many features in characters. We developed featureless rotation and text segmentation and use them before character recognition in our project, which have been shown to be able to improve the overall recognition accuracy effectively. Furthermore, traditional ways of recognition lack flexibility to incorporate meaning of individual characters. We improved this by performing minimum edit distance algorithm after character recognition to make correction. After setting the max tolerance of edit distance to 9, the match rate could reach above 90%. An ROI not only enables users to choose the dishes they are interested then get their translation instead of translating the whole menu, which accordingly boosts user experience, but also reduces the search space and noise, thus lead to better results.

Yet, it still has some aspects that need to be improved. A big challenge is how to reduce the runtime. We know from the results section that the most time consuming part is OCR correction. We need to be faster in matching the dish name with its corresponding image in database to meet real-time demonstration requirement. This is especially the case when the system grows and has a huge database. Another problem is how to increase the accuracy rate. The results of rotation correction and name segmentation are related to the size of the structural element used in dilation processes to a large extent. Both too small and too large structure elements will lead to misrecognition. Yet we have no clear idea which size yields the best results.

In the next section, we will discuss possible improvements that can be made in further studies to reduce these weaknesses and enhance the overall performance of our system.

VII. Further Discussions and Future Work

In this project, we have successfully developed an automatic English menu translation system to help non-English speakers overcome the difficulties of ordering meals in a foreign restaurant, with emphasis on studying different methods to increase the accuracy of the state-of-the-art OCR algorithm and finally increase the rate of correct recognition. The system is robust, fast, accurate and flexible while providing good user interaction experience. With good scalability, the system can be further extended to and find more applications in situations such as multi-language translation and larger database, or the information about the dish can be obtained via online searching. People may further extend the application to showing the information on a wearable VR device.

The presented pipeline brought up several novel ideas, including the featureless rotation and passing in segmented ROI for each dish name to the OCR engine. As noted in previous section, the results of these two techniques heavily

depend on the structural element sizes used in the dilation process. As such, we think one possible way to overcome such scaling problem is to create a pyramid of dilated results based on different structural element sizes. With this pyramid of dilated images, we can pick the scale that yields the maximum bounding box aspect ratio for the rotation correction and the highest OCR accuracy rate for the dish name segmentation. This will definitely make our system more robust but also adds significant amount of runtime cost.

Another possible improvement of the OCR correction is that we can first group and categorize the dish names in the database every time we import more dish data. This kind of preprocessing on the database is the only way to make the computation complexity not linearly grow with the size of the database, and be able to further incorporate faster data searching techniques such as hash table or binary search.

VIII. ACKNOWLEDGEMENT

We would like to thank Professor Gordon Wetzstein, Jean-Baptiste Boin, Matt Yu, and course assistant Kushagr Gupta for continuous guidance and care throughout the quarter and throughout the final project.

REFERENCES

[1] A. Heng, "L'Addition: Splitting the Check, Made Easy".

[2] C. N. Nshuti, "Mobile Scanner and OCR (A first step towards receipt to spreadsheet)".

[3] M. Jin, L. X. Wang, B. Y. Zhang, "Real Time Word to Picture Translation for Chinese Restaurant Menus," EE268 Project Report, Spring 2014.

[4] R. Smith, "An Overview of the Tesseract OCR Engine," Tesseract OSCON, Google Inc.