Prof Helper

Anshit Sobti
Computer Engineering Department
San Jose State University
San Jose, USA
anshit.sobti@gmail.com

Shripal Modani
Computer Engineering Department
San Jose State University
San Jose, USA
shripal555@gmail.com

Tianxiang Liu
Computer Engineering Department
San Jose State University
San Jose, USA
liutianxiang344@gmail.com

Pavan Kothawade
Computer Engineering Department
San Jose State University
San Jose, USA
pavan.kothawade@gmail.com

Abstract—Prof helper, as the name suggests, aims to help professors in an institution by decoding the handwriting of every student into standardized text to understand the exact answers of the students and avoid any mistakes while grading their papers.

I. Problem Statement

Many students lose important marks in examinations even after writing the correct answers due to ambiguous/bad handwriting. Prof helper converts the hand-written text into standard text on the computer screen which helps the professor to read the exact answers of the students and grade them accordingly.

II. Introduction

Prof Helper, is a web application that allows the professor to upload photos of the writing piece of every student and the converted computerized text can be seen on the same page. After judging the text, the professor is allowed to grade the student's work, comment on it and email the grades and comments to the student.
The teacher can also save the marks, comments and name/email of the student in a spreadsheet which can be used later.

III. Implementation

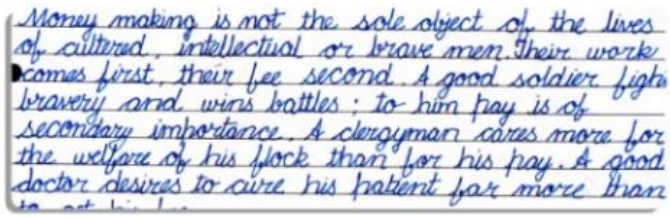
The application allows a professor to access it after signing up by entering a few details including the name of the institution he/she is a part of. This gives them access to the application even in the future by using the same credentials.
The major implementation of this project includes the use of Microsoft Azure API known as Computer Vision API which has the option of giving the URL of an image from the internet. The image can be analyzed to:

i. Tag images based on content
ii. Categorize images
iii. Identify type and quality of images
iv. Detect human faces and return coordinates
v. Recognize handwritten text

The professor can upload the images from local system using an uploader which is developed in Node.js and the computerized equivalent of that image is available on the same page. The professor can grade the student based on the submission and also comment on the work which will then be saved in a spreadsheet along with the name of the same student.

A spreadsheet is maintained where the marks, comments and name of the student is stored to maintain the record of every student, so that the professor can later have a look at it.
A sample of input and output image is shown below:

Input image:



Output on the computer:

```
"Money making is not the sole object of the livesof cultured intellectual or
bravecomes first , their fee second A good soldier fightbravery and wins
bottles : to him pay issecondary importance A clergyman cares ma""the welfare
of his flock than for his paydoctor desires to cure his patent hak more than"
```

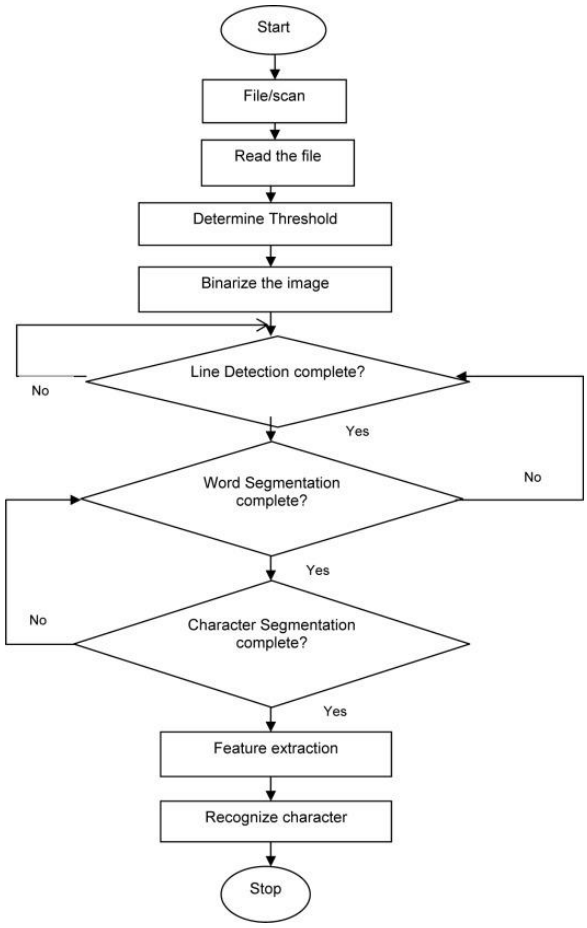
On implementation, the accuracy was seen to be more than 90% making it one of the most accurate text detection models.

IV. Technology

The model for identifying handwritten text through prediction of a handwriting sequence by using OCR (Optical Character Recognition) spots text in an image and abstracts the recognized characters into a machine-usable character stream. This approach is based on conclusions in the field of brain sciences. While identifying handwritten text, humans are said

to inadvertently trace the order of handwritten text in their brains. Similarly, we went through a model that predicts a handwriting sequence from a static image of handwritten text by using OCR. On success, the OCR results will be returned while on failure, the error code together with failure message will be returned. The error code can be Invalid Image URL, Invalid Image Format, Invalid Image Size, Non-Supported Image, Non-Supported Language, or Internal Server Error. The predicted handwriting sequence would be used to recognize the text.

First, the handwriting image sequences for training are self-organized into image features using a self-organizing map. The self-organized image features are used to train the neuro-dynamics learning model. For recognition, both trained and untrained image sequences are used to evaluate the competence of the model to familiarize to strange data. Written text recognition is an essential facility for self-sufficient systems expected to play a role in human society.



Flow chart of OCR system

Current approaches and their limitations:

Presently, two main approaches exist for realizing text recognition systems: (i) Pattern recognition-based text recognition. (ii) Online recognition-based writing recognition. Although these methods have shown high efficiency for identifying letters, they own a problem in that they require a large dataset to train the model to apply it to differences in writing outlines. The second approach is mainly directed by dynamical systems to distinguish the written text by outlining the changeover of the pen.

Though these methods were extremely efficient for influential handwritten letters, offline recognition (without writing transition) was out of the scope. We focus on a new approach for text recognition based on the human cognition of text. Humans are said to apply their handwriting experiences to trace the written text inadvertently when identifying the text. Experimentations with people, training them with writing and typing practices, have shown that writing experiences enhance the ability to recognize characters. Experimentations with English characters have also made known that handwriting experiences have a great impact on their recognition capabilities. These works infer that recognition, not only by static evidence, but also by estimate of handwriting, progresses the text recognition ability for people. The goal line of our effort is to present the estimate of a writing sequence from pictorial text for recognition of written text. The method focuses on demonstrating human recognition rather than developing a high-precision recognition system. We apply a dynamical system for creating the estimate system. There are two benefits to our approach: (i) The model can simplify from few handwriting practices. (ii) The model can familiarize to differences in writing patterns.

Image specifications:

i. The image formats supported include JPEG, PNG, GIF and BNP.
ii. Image file size less than 4 MB.
iii. Image dimension greater than 50 x 50 pixels.

V. Other Alternatives to solve the same problem

Before using the Computer Vision API from Microsoft, we were implementing the same using Google Cloud Vision API but the accuracy of the text detection was close to 70% while the current accuracy is close to 90%.

We did a lot of research with both Google Cloud Vision API and Azure Computer Vision API by testing all kinds of images and came up with various comparisons to choose the better option.

Moreover, if we want to expand the project further like getting the coordinates of the image or objects, type of image etcetera our current implementation allows us to do so while the Google API was limited to only text detection. Hence, using Computer Vision API seemed to be a logical choice.

VI. Future Scope and Refinements

There are many refinements possible in this application such as an instructor can upload multiple pages at a time and see all the converted texts at once.

This application supports only English language at the moment but similarly it can be implemented in other languages making it useful for a wider range of professors.
One of the flaws that can be resolved in the near future is that if the words

VII. Conclusion

To conclude, after trying out a few APIs available for hand writing detection we got a lot more accuracy (approximately 90%) than what we were expecting initially. We successfully managed to solve the problem of helping the professors or for that matter anyone to understand any handwritten text by converting it into standardized computer text.

VIII. Deployment

Our application is deployed on an instance of Amazon AWS EC2 which makes it accessible. The link is given in the GitHub repository of our project.

IX. Acknowledgement

We would like to thank Professor Rakesh Ranjan for giving us an opportunity to explore wide range of topics without any restriction and asking us to do the one we are most most passionate about. This made us explore many more possible possibilities and helped us to work constructively towards the project.

X. References

[1]http://www.tandfonline.com/doi/abs/10.1163/016918611X595026
[2] www.google.com
[3]https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home#Tagging
[4] http://ieeexplore.ieee.org/document/7193210/
[5] http://eprints.sztaki.hu/7890/1/Kornai_1762363_ny.pdf
[6]https://pdfs.semanticscholar.org/7ca5/584576423b366ad7bdf03d5fc136a2e958e6.pdf