Handwritten Math Equations to LaTeX

Spring 2018 Group 34 - Brandon Smith, Jeffrey Huang, Anthony Chou, Suva Shahria

{bcs109, jh1127, ac1321, ss2479}@scarletmail.rutgers.edu

Advisor : Anand Sarwate

Motivations and Objectives

Objectives

- To create a mobile application that takes a picture of handwritten math notes and converts it in a Latex pdf.

Motivation

- Latex is widely used in the academic field but is time consuming to learn and a vast majority of people do not know how to use it. Our app will allow everyone to use Latex. Our application will conserve time for the user and is more convenient than loading up Latex software on the computer.
- This app will allow all users to save their notes in a digital format without the need to retype or do any sort of formatting.

Definitions

- **Optical Character Recognition (OCR)** - conversion of images of handwritten or typed text to machine-encoded text.

Research Challenges

- ❑ We had trouble using opencv to detect more than 1 equation at the same time.
- ❑ We initial wanted to do everything on the app but we switched to server client model so that everyone could develop separately. We all agreed to use Python in order to make implementation and integration easier than if we all used different languages.
- ❑ Erasing noise in the photo, removing shadows, different lighting conditions.
- ❑ Extracting the equation from the detected page in the image.

Acknowledgement

We would like to thank Professor Anand Sarwate for his guidance and support in this project.

Methodology

Figure 1 : Page Detection

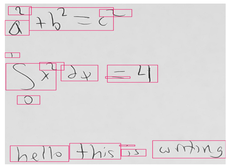

Figure 2: Character Detection

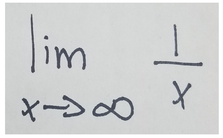

Figure 3: Image Cleanup

- ❑ **PICTURE -** Take a photo of the page using the app.
- ❑ **DETECTION & PROCESSING -** The image is then sent to the server where the page will be detected, the area of the equation will be determined and cropped into a new image which will then be processed for easier character detection to handle issues like lighting.
- ❑ **CONVERSION -** The cleaned image is converted into vector graphics through AutoTrace which is then converted from an .eps file into a .svg file through InkScape.
- ❑ **RECOGNITION -** The resulting file is then converted into a .scgink file and run through Seshat which returns the recognized equation using the stroke data.
- ❑ **OUTPUT -** Then the pylatex library is used to convert the string representing the equation into a pdf containing the LaTeX form of the processed equation which is returned to the app for the user to have.

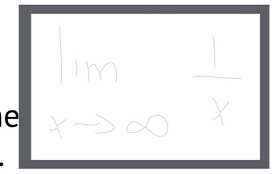

Figure 4: Vector Graphics Conversion

Current State and Results

We created the app using Android Studio.
After taking a picture , the user can select a name for the output and then click process where the picture is sent to the server and then our app outputs a Latex pdf with the equation inside. It also stores the tex file on the server side.

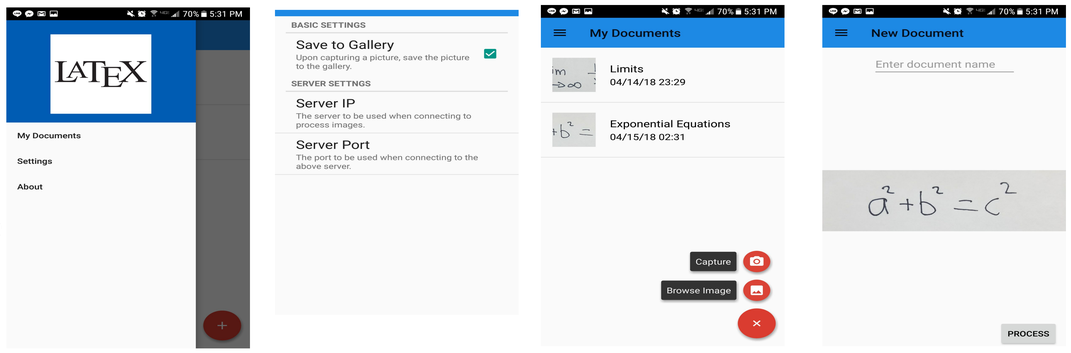

Figure 5a,b,c,d : The Application

Future Direction

Extensions:

- Multiple Line detection
- Further training on the model to incorporate more symbols
- Speed up processing time from image to pdf
- Recognizing typed text as well rather than just handwritten text
- Improved formatting for the digital output

References

[1] F. Álvaro, J. A. Sánchez, and J. M. Benedí, "An Integrated Grammar-Based Approach for Mathematical Expression Recognition," ***Pattern Recognition***, vol. 51, pp. 135-147, Mar. 2016.