**Project Topic:** Creating a neural network by machine learning that can read handwriting

**Members and Roles:**

**Thomas McCullough:** Python Expert and accessibility consultant and accessibility consultant.

**Farahnaz Hosseini:** Python expert and data visualization.

**Project Abstract:**

We seek to provide a robust solution to handwriting recognition across a variety of applications. Whether our clients are teachers seeking to automate the grading process, author's attempting to compile notes, or blind programmers attempting to read a letter, our program will provide a reliable way to bring handwritten documents into a digital format. We have taken a realistic approach, but our stretch goals include multilingual handwriting recognition, with support for punctuation and mathematical symbols. The end user will be able to choose whether they want to use our model in a windows application, on a website, or simply import the model as a python package. We know that this will require a high degree of accuracy and thus have chosen to use a wide variety of datasets, and a model known for its utility in image processing: the convoluted neural network.

**Dataset:**

https://www.kaggle.com/datasets/landlord/handwriting-recognition

https://www.kaggle.com/datasets/dhruvildave/english-handwritten-characters-dataset

https://fki.tic.heia-fr.ch/databases/iam-handwriting-database

**Project Design and Milestones:**

We will be creating a Convolutional Neural Network to extract features and label textual characters with an alphanumeric label. Our language of choice is Python, and our approach will involve a combination of python modules and Google Collab Notebook files. Our implementation will make use of TensorFlow for the CNN, pandas for accessing the dataset, and both NumPy and Sklearn to compile our resulting statistics. Matplotlib will be used for graphical representations of data.

A picture containing shape

Description automatically generated

Fig 1- Handwriting system using Neural Network

Our initial focus will be proof of concept. We will create a function that takes an image file as input, and output a textual representation of the image through commands in the python interpreter or Google Collab cell. Optional stretch goals include a GUI for desktop using WXPython and A web interface for accessing our model. Our scope will be similarly structured. Our initial focus will be alphabetical characters, followed by numbers, and optionally mathematical special characters punctuation, and other languages.

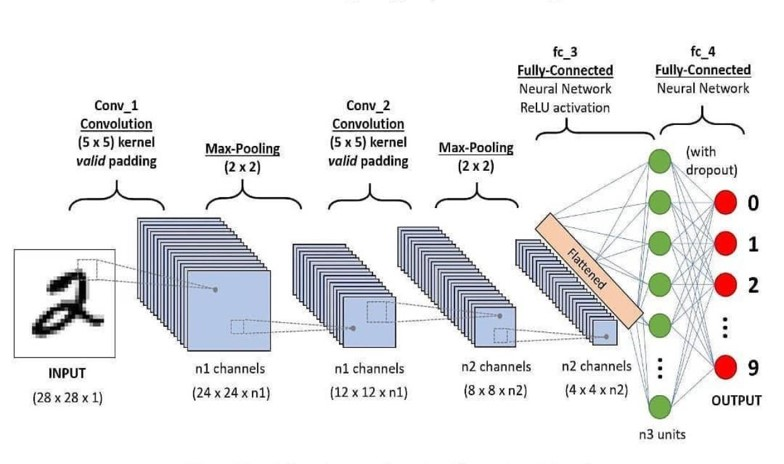


Fig 2- Handwriting recognition with convolutional neural network[2]

**Our milestones include the following:**

1: Create a GitHub repository for version control of code snippets.

2: Successfully import and clean our datasets.

3: Familiarize with TensorFlow and extract features from a small number of images as a proof of concept.

4: Make classifications based on initial features, making a proof of concept for our pipeline.

5: Scale up processing, applying our proof of concept to the entire dataset.

6: Tune hyperparameters to optimize feature selection and classification

7: (Optional) Use a cross-validation technique to pick optimal values for hyperparameters

8: Export the tuned model to the appropriate format, likely a Google Collab notebook that is appropriately descriptive.

9: (Optional) Create a WXPython GUI application for running the model on example images

10: (Optional) Create a GitHub-based website where users can run the finished model on example images.

11: Regardless of optional goals, spend last week finalizing extensive documentation and polishing front-end results.

**Related Projects:**

Some projects related to reading handwriting using machine learning are:

1. Ancestry’s AI Handwriting Recognition technology can take historical documents scanned in and convert the written text to Unicode characters.
2. The United State Postal Service’s Edge Computing Infrastructure Program (ECIP) can speed up the processing of mail in the United States by reading the address on the front of an envelope so that it can be sorted more efficiently.
3. CheckXpert.AI is an AI program that reads the written information on a cheque so that the bank can process the payment or transfer of money and provide proof of deposit and other remittance applications.
4. Tesseract-OCR is an open-source handwriting recognition software.

With commercial and even freely available applications, a lot of information is obfuscated or unavailable on how the creators created, trained, and deployed their software to read handwriting. The applications of this technology are very different through in both the way they need to read the handwriting as well as auxiliary requirements. In banking, the software not only needs to be able to read the handwriting of the cheque but do so with significant accuracy as well as checking to verify if the cheque is invalid or fraudulent. It then must also provide proof that it verified this information to both the customer and the institution itself. Any misstep in the process has the potential to be financially catastrophic. Whereas a free-to-use application such as genealogy is quite a bit more forgiving. The software can have a higher error rate since historical records regarding people’s lives are unlikely to create significant issues. Furthermore, they can supplement these errors by employing humans to verify and correct historical documents as Ancestry does. Often commercial applications versus free applications of machine learning to reach handwriting see the commercial application requires a much lower error rate with more precision and accuracy than a non-commercial application.

However, an example of a non-commercial application that still needs a very low error is the United States Postal Service Edge Computing Infrastructure Program. This AI reads the handwriting on envelopes and sorts them to be sent to the written destination. While they do have a facility in Utah that is dedicated to manually labeling mail that the AI is incapable of reading if the AI reads a piece of mail incorrectly and labels it with a valid address then it will be sent to that destination instead of being caught and sent to be labeled by a human.

Open-source software such as Tesseract-OCR gives the user more freedom in understanding what they are using and even making modifications should they desire to. This freedom of information regarding the handwriting recognition software allows users to decide if it suits their needs for their application and can be integrated or built upon unlike closed-source software such as many commercial and even freely available software.

**Reference:**

[1] https://www.myscript.com/ai/

[2] http://ijetemr.org/2020/06/19/handwritten-character-recognition-using-deep-learning/