Project Topic

Your Name

October 23, 2025

1 Introduction

Currently many different computer vision systems are used to interpret handwriting as digital text, but these systems are constrained usually to one style or lack the complexity to comprehend more challenging handwriting. By restricting themselves to focused or large diverse datasets in the development of the system, they create potential for inaccuracy. My goal is to create a series of computer vision systems that maintain accuracy while circumventing some of these constraints by having an AI redirect the given image of text to the appropriate computer vision system trained to accurately the style of text that the AI has perceived.

2 Related Works

Within the AI field of computer vision, one popular goal is the transcription of handwritten text. This is called handwriting recognition technology or Handwritten Text Recognition (HTR). According to Yuhe Tian and Ying Yu in their research paper, "Research Application of Computer Vision-Based Convolutional Neural Network in Handwriting Recognition Technology," a convolutional neural network system was utilized to build a computer vision system able to recognize digits from the MNIST and achieve accuracy of up to 100% on the number 1 in the test data. Essentially a convolutional neural network uses a combination of a single input layer then convolution layers that create feature maps, pooling layers that decrease data, and fully connected layers to create and process feature maps from images and fully connected layers that link the neurons to output results. These are very effective systems but according to the research article "A User Perspective on HTR Methods for the Automatic Transcription of Rare Scripts: The Case of *Codex Runicus*" by Mohamed Ali Souibgui and others, there are issues with methods like convolutional neural networks, and recurrent neural networks because they require deep learning that involves vast datasets inaccessible to some less common scripts. In their paper, they sought to transcribe the *Codex Runicus* which is in a very uncommon language. To do this, they used learning free methods such as unsupervised clustering which creates clusters which are subdivided, then the labels of each cluster are propagated through the other symbols, and the few-shot classification method which seeks to represent each individual symbol as a node in a graph and compare similarity between each pair of symbols. These methods may all be used to accomplish similar tasks but at different scales. At the moment, it is very possible to perform highly accurate machine learning for popular fonts and styles but the pursuit of handwritten text recognition for any obscure text may result in lower accuracy systems that avoid machine learning methods.

There are several different popular datasets for handwritten text recognition computer vision. As was seen in Yuhe Tian and Ying Yu's paper, they used the MNIST. This is a popular data set used for testing computer vision systems provided by the National Institute of Standards and Technology consisting of 70,000 grayscale images of 28 by 28 pixel digits. Another dataset the EMNIST is described in a research paper "EMNIST: an extension of MNIST to handwritten letters" by Gregory Cohen and others. This new data set includes images of characters from a-z in the same grayscale 28 by 28 pixel format used by the MNIST dataset. Each character in the dataset is labelled with its ASCII value. This makes many methods Yuhe Tian and Ying Yu described very easily applicable to the wider data set. As was noted by by Mohamed Ali Souibgui in their paper, it is difficult to find labelled data for some obscure texts.

The preparation of images for computer vision is a complex process involving simplifying images and often storing them in gray scale. As was seen with the EMNIST images, they were represented in gray scale with gray scale pixels. Additionally, a gaussian blur ROI Extraction Centered Frame, and resizing and resampling were applied to ensure that the images were the correct consistent size and had the proper shading. Another article, "Design and implementation of interactive image measurement software based on Tkinter and PIL library" by Xiao He and others show how tools may be used to perform these tasks. They explain that the Python PIL libraries may be used to perform image processing such as grayscale conversion, Gaussian blur, image sharpening, binarization and more. The PIL package in Python could effectively be used for the preparation of images for EMNIST computer vision systems.

To connect AI models with one another, there are several different popular methods. One recently released was the Model Context Protocol (MCP). According to its own website, this protocol is a standardized way to connect AI applications to other systems. This is primarily described in the context of agentic AI applications that utilize an AI to perform a task rather than just return an output.

For the implementation of this project, I will create a simple application allowing image and optional text input. The image input will take the image of the text while the optional text box would allow for a description of the writing style, skipping the optional computer vision to detect a specific writing style. From here, the file will be run through the required computer vision system or systems to convert it to plain text by MCP an MCP Client sending the image to the MCP server for the determined writing style. This will be returned to the user in the form of a text file that they can download.

The images chosen are diverse Latin alphabet handwriting styles including the generic modern handwriting found in the EMNIST data sets. I convert the images of this text using the PIL library to convert it to a similar size to that of the individual image files in the EMNIST database. From here, if they are a part of a larger or more popular writing style, the images will be fed into a convolutional neural network module which will perform the training and measure accuracy as it goes. If the accuracy is satisfactory, the resulting computer vision system will be made accessible to the AI implemented by MCP. Small fluctuations in accuracy do occur in each run of the program. If the text is from a smaller dataset then I perform the necessary labelling myself and attempt to process text with the unmonitored clustering method. This provides far less accuracy, but the system is still kept. This involves the same processing steps as convolutional neural networks and also may be used by the AI. The text is output on the simple user interface

3 Methodology

Dijkstra's algorithm was chosen for single-source shortest path computation on graphs with non-negative edge weights, following the implementation presented in CLRS [**?**].

4 Results

5 Conclusion

References