Table of Contents

1 Introduction

In the realm of machine learning, the MNIST Handwritten Digit Classification to number prediction project stands as a pursuit aimed at crafting a proficient model capable of accurately identifying and classifying handwritten digits as well as numbers. At its core, this project aims at harnessing the potential of the MNIST dataset – a well-regarded repository of 28x28 pixel grayscale images featuring handwritten digits ranging from 0 to 9 where the training and testing size is of 60, 000 and 10, 000 respectively. The central objective is to implement a robust classification algorithm, one that possesses the ability to automatically discern and categorize distinct digits based on the nuanced pixel values within each image. That means, this project is all about teaching a computer to be smart with handwritten numbers. Our job is to train the computer to look at the pictures and say, "Oh, that's a 3!" or "Hey, that's a 7!" We're basically making the computer learn how to recognize and tell apart different numbers by looking at tiny dots that make up the pictures. It's like teaching the computer its own special way of reading our handwritten numbers.

1.1 Motivation

The motivation behind the MNIST Handwritten Digit Classification to number prediction project stems from its potential impact on daily life. By creating a smart system that can recognize handwritten numbers, we aim to improve postal services and preserve historical documents. This project also serves as a fun way for learners to explore machine learning using the MNIST dataset, enhancing both technical skills and understanding. It's about making technology smarter and more useful in our everyday experiences. Therefore, in fact the major motivations behind our project as follows-

- The realization that this project's practical impact extends to automating tasks like recognizing postal codes and digitizing historical documents, simplifying processes in daily life.
- The realization that it will be a helpful playground for trying out different algorithms and gaining hands-on experience.
- It can be used to enable automatic postal code recognition and digitization of historical documents as well.

1.2 Applications and Impacts

The developed model will find application in digit recognition systems, particularly in scenarios where manual digit entry is required. This includes applications in finance, postal services, and any domain where accurate and efficient digit recognition is essential. Therefore, the major applications of this model can be as follows-

- In Optical Character Recognition (OCR), handwritten digit classification is the basic and former things.
- A robust digit classification model can enhance the accuracy of such systems, reducing errors in financial transactions.
- Implementation of the digit classification model in postal services can automate the sorting process, improving efficiency.
- Recognizing handwritten digits can be applied in educational technology. For instance, grading handwritten assignments, evaluating mathematical expressions, or providing feedback on handwritten solutions as well.

2 Project Overview

The proposed project about handwritten digit classification to number detection contains several features as follows-

- **MNIST Dataset Utilization:** Leveraging the MNIST dataset, a repository of testing images featuring handwritten digits (0 to 9).
- **Preprocessing Techniques:** Employing preprocessing methods, including normalization and reshaping of images. Enhancing the model's efficiency by preparing the data for effective learning.
- **Traditional Machine Learning Models:** Implementing classical machine learning algorithms to classify handwritten digits. Exploring methods such as support vector machines or k-nearest neighbors for baseline comparisons though this is not used in this project.
- **Deep Learning Models:** Incorporating deep learning techniques for more complex pattern recognition. Utilizing neural network architectures, like convolutional neural networks (CNNs), to capture intricate features.

Following figures represents the deployment of the project which can predict numbers from handwritten images.

Figure 01: Interface of Home Page

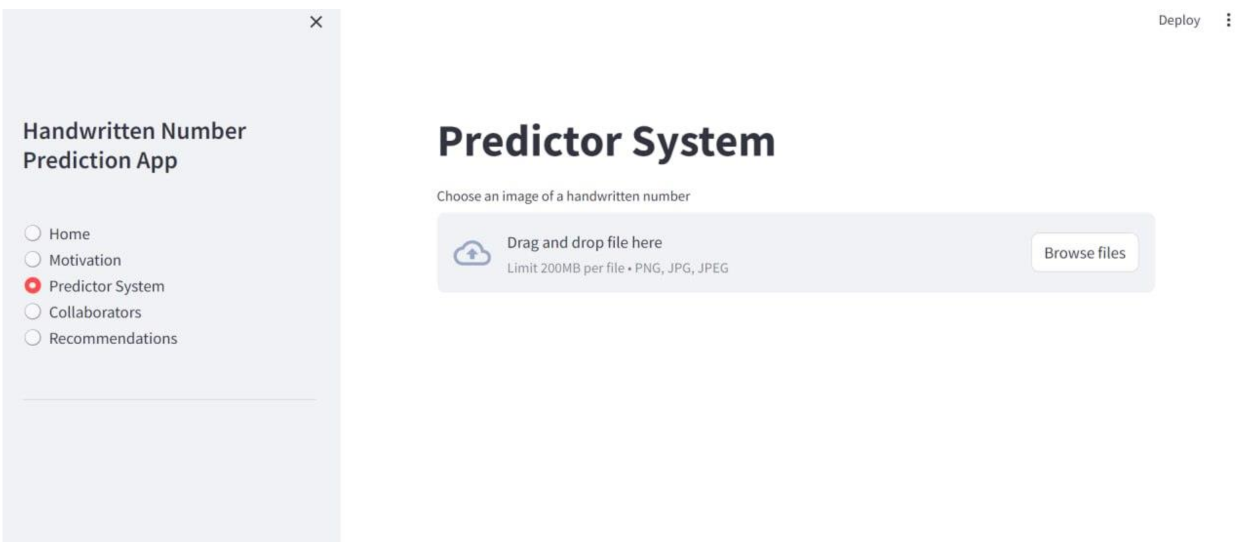

Figure 02: Interface of Predictor Screen

The following figure represents the core of the project which is to predict the handwritten number based on given input image by using trained model on MNIST dataset. It also includes the voice feature which can tell what is the predicted number by using text-to-voice function from python library.

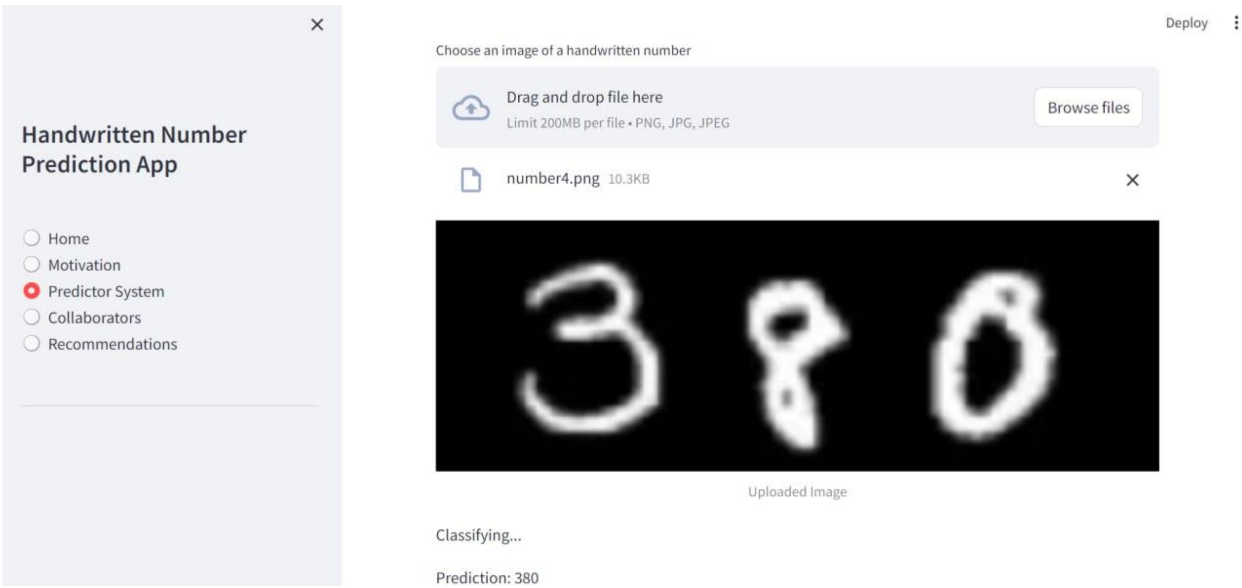

Figure 03: Interface of Predictor Screen (Including Prediction)

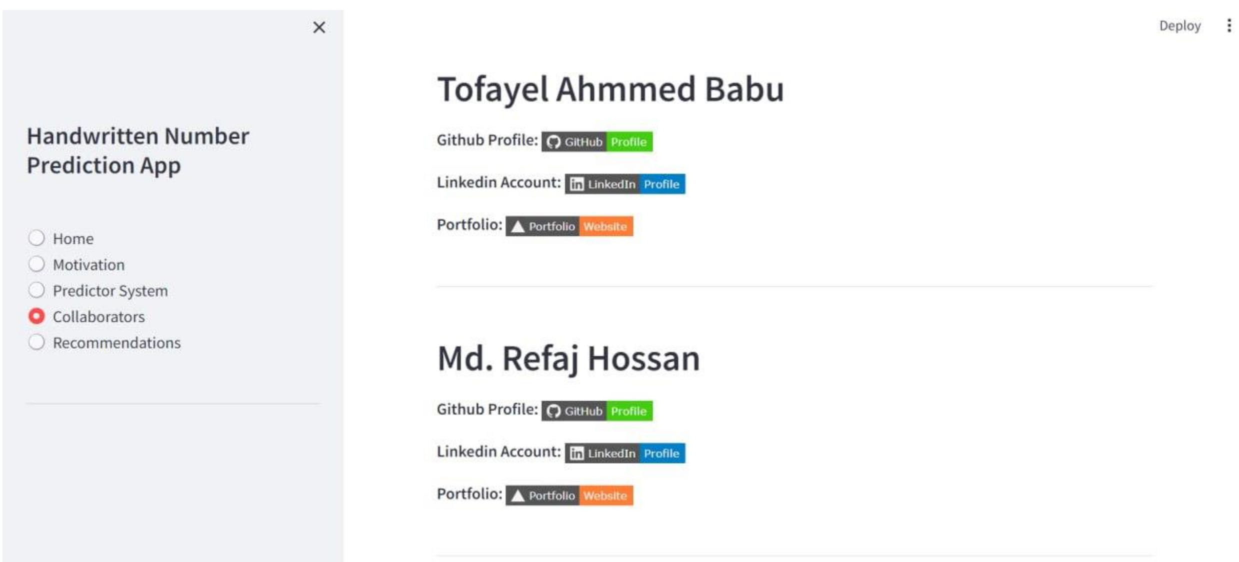

Figure 04: Interface of Collaborator Screen

3 Required Technologies

- **Development Platforms** : Google Collaboratory, Anaconda, VS Code
- **Language** : Python3
- **Frameworks** : Numpy, Matplotlib, Keras, Tensorflow, Seaborn, CV2 etc.
- **Frontend Development** : Streamlit
- **Model Deployment** : Streamlit

4 System Implementation

The project explores deep learning methods, particularly Convolutional Neural Networks (CNNs). By the word "System Implementation", we refer to the phase of System Development Life Cycle including coding, testing, software and hardware configuration. The proposed methodology involves data preprocessing, model training, and evaluation to identify the most suitable algorithm for accurate digit classification. Therefore, it includes the following steps for a machine learning based project-

1. Data collection
2. Data preprocessing
    a. Normalize pixel values to the range [0, 1].
    b. Flatten 28x28 images into a 1D array.
    c. Split the dataset into training and testing sets.
3. Implementation of CNN model for image classification.
4. Model training and model evaluation.
5. Result analysis and deployment.

Following figures represents some of the core things during train the model by using necessary frameworks as well as functions.

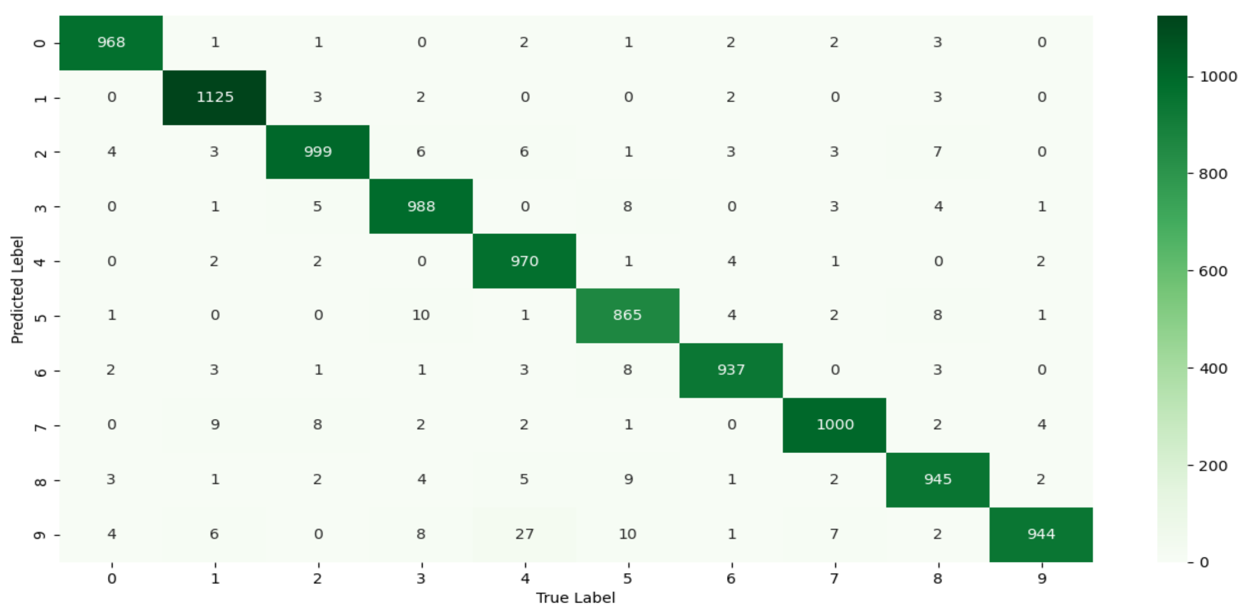

Figure 05: Heatmap of Predicted Value Versus True Value

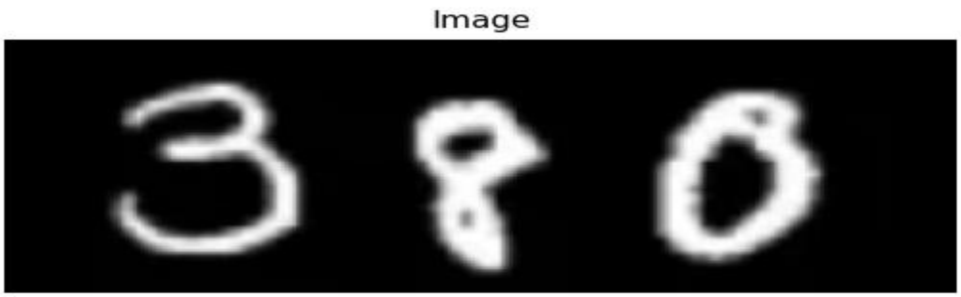

Figure 06: Sample Image for Predicting a Number

```
def find_digits(img):
    # Get image width and height
    height, width = img.shape

    # Store the starting and ending columns for each digit
    digit_columns = []

    is_digit = False
    start_col = 0

    # Traverse columns
    for col in range(width):
        column_data = img[:, col]

        # Check if the column has any white pixels (digits)
        if np.any(column_data == 255):
            if not is_digit:
                start_col = col
                is_digit = True
        else:
            if is_digit:
                digit_columns.append((start_col-10, col+10))
                is_digit = False

    # Check if the last digit extends to the end
    if is_digit:
        digit_columns.append((start_col, width))

    # Filter out small columns
    min_column_width = 5  # Adjust this threshold based on your images
    digit_columns = [(start, end) for start, end in digit_columns if (end - 
start) > min_column_width]

    # Extract digits based on identified columns
    digits = [resize_digit(img[:, start:end]) for start, end in digit_columns]

    return digits
```

By using the above function, the number is classified by using MNIST dataset which only includes single digit. The function acts like a black-box during training the model.

5 Conclusion

In drawing the curtains on this project, the attained success in training the model to achieve an impressive accuracy of approximately 97% signifies a significant milestone. The system, adept at predicting handwritten numbers with a quite high degree of precision, not only showcases the efficacy of machine learning but also underscores the potential for real-world applications. The integration of a voice feature adds a layer of accessibility, making the technology not just accurate but also user-friendly. While the project has proven its mettle in the realm of digit classification, it's essential to acknowledge the ongoing journey of technological evolution. There's room for further refinement, exploration, and adaptation to tackle the ever-expanding landscape of challenges. The practical application of Python programming skills, exploration of diverse algorithms, and the utilization of the MNIST dataset collectively enrich our understanding of the dynamic intersection between technology and handwritten digit recognition. In essence, this project is not just a conclusion but a stepping stone for future endeavors.

6 Future Recommendations

Though this project marks a visionary journey in digital number prediction systems, but it has some limitations as well which can be outcome in future. There are some recommendations that can be done in future for better services as follows-

- Exploration of advanced deep learning architectures and techniques to further refine the model. Experimentation with more layers, different activation functions, or optimization algorithms to potentially boost accuracy.
- Implementation of data augmentation techniques to diversify the training dataset.
- Enhance the user interface of the project, making it more intuitive and user-friendly. Implement features like interactive visualizations.

7 References

1. The MNIST Database of Handwritten Digit Images for Machine Learning Research by Li Deng in IEEE Signal Processing Journal, Page(s):141-142, DOI: 10.1109/MSP.2012.2211477.