1. Report

The goal of this project was to develop a system for detecting handwritten digits in images. The system involved generating training images, training a YOLO-based model, and then using the trained model to detect digits in test images. The detected digits were stored in a CSV file, with each row containing the image name and its corresponding detected digits.

**2. Approach**

The project was divided into three main phases:

1. **Data Generation:** We generated synthetic training images of handwritten digits to train the model upto 10^8 images with took long hrs on gpu also.
2. **Model Training:** We trained a YOLO (You Only Look Once) model to detect the digits in the images and the training is done on just numbers detection and it took around 8-10 hours for 1000 epochs for better results otherwise result might hinder.
3. **Digit Detection & CSV Saving:** Once the model was trained, we used it to detect digits in test images and saved the results (image name and detected digits) to a CSV file.

**2.1 Data Generation**

* **Objective:** The first step in the project was to create a dataset of handwritten digits for model training.
* **Method:** We used tools like Python libraries and synthetic data generation methods to generate labeled images of handwritten digits. The images were created with varying backgrounds and noise to simulate real-world scenarios. This variety helped improve the model's robustness.

**2.2 Model Training**

* **Objective:** Train a model to recognize handwritten digits in images.
* **Model Chosen:** YOLO (You Only Look Once), an object detection model known for its speed and accuracy, was selected due to its ability to detect objects (digits, in this case) in images.
* **Training Time:** The model took approximately 8-10 hours to train on a GPU.
* **Training Process:**
  + **Data Preprocessing:** The synthetic images were preprocessed to fit the input size required by the YOLO model.
  + **Model Architecture:** The YOLOv5 architecture was used, which involves convolutional neural networks (CNNs) for detecting and classifying objects in images.
  + **Hyperparameters:** Various hyperparameters were tuned, including learning rate, batch size, and the number of epochs.
  + **Tools Used:** The model was trained using the Ultralytics YOLOv5 library, which provides a simple interface for training and deploying YOLO models.

**2.3 Digit Detection & Saving to CSV**

* **Objective:** Use the trained model to detect digits in test images and store the results.
* **Process:**
  + The YOLO model was applied to test images. For each detected digit, the model returned its class (digit) and bounding box coordinates.
  + The detected digits were extracted, and the image name along with the list of detected digits were saved in a CSV file for future analysis.
  + **CSV File Structure:** The CSV file contained two columns:
    1. **Image Name:** The path of the test image.
    2. **Detected Digits:** A comma-separated string of digits detected in the image.

**3. Tools Used**

* **Programming Language:** Python
* **Libraries/Frameworks:**
  + **Ultralytics YOLOv8:** For object detection using YOLO.
  + **OpenCV:** For image manipulation and bounding box drawing.
  + **CSV:** For storing the results.
  + **NumPy and Pandas:** For handling data and processing.

**4. Results**

* After training, the model was able to detect handwritten digits with high accuracy. The detection results were saved in the CSV file, with each image name and its corresponding detected digits.