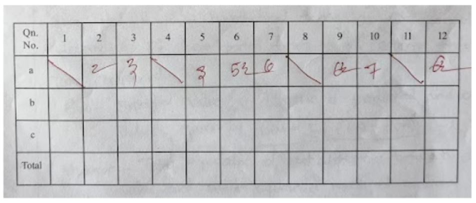
# Chapter 1

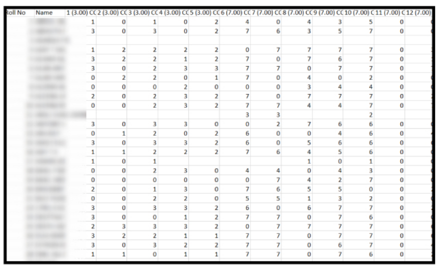
# Introduction

The rapid advancement of Artificial Intelligence (AI) has revolutionized various aspects of human workflow, particularly in the field of data processing. One prominent application is the automation of handwritten digit recognition, which holds immense potential for numerous purposes. While recognizing digits from a small set can be relatively straightforward, dealing with vast quantities of handwritten digits poses significant challenges, as it becomes a time-consuming task prone to errors.
To address this critical issue, the project aims to develop a robust and reliable system capable of instantaneously converting handwritten digits into structured Comma Separated Values(CSV) files. By doing so, it alleviates the substantial burden placed on educators during the manual data entry process. This innovative system will empower teachers and researchers to allocate their valuable time more efficiently towards other essential responsibilities, ultimately enhancing the efficiency and accuracy of digit data processing in various domains.



| Qn. No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | \ | 2 | 3 | \ | 3 | 5½ | 6 | \ | 6 | 7 | \ | 6 |
| b | | | | | | | | | | | | |
| c | | | | | | | | | | | | |
| Total | | | | | | | | | | | | |

$\rightarrow$



| Roll No | Name | 1 (3.00) CC | 2 (3.00) CC | 3 (3.00) CC | 4 (3.00) CC | 5 (3.00) CC | 6 (7.00) CC | 7 (7.00) CC | 8 (7.00) CC | 9 (7.00) CC | 10 (7.00) C | 11 (7.00) C | 12 (7.00) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [redacted] | [redacted] | 1 | 0 | 1 | 0 | 2 | 4 | 0 | 4 | 3 | 5 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 3 | 0 | 2 | 7 | 6 | 3 | 5 | 7 | 0 | |
| [redacted] | [redacted] | | | | | | | | | | | | |
| [redacted] | [redacted] | 1 | 2 | 2 | 2 | 2 | 0 | 7 | 7 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 3 | 2 | 2 | 1 | 2 | 7 | 0 | 7 | 6 | 7 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 2 | 3 | 3 | 7 | 7 | 0 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 0 | 2 | 2 | 0 | 1 | 7 | 0 | 4 | 0 | 2 | 0 | |
| [redacted] | [redacted] | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 3 | 4 | 4 | 0 | |
| [redacted] | [redacted] | 2 | 0 | 2 | 3 | 2 | 7 | 0 | 7 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 0 | 0 | 2 | 3 | 2 | 7 | 7 | 4 | 4 | 7 | 0 | |
| [redacted] | [redacted] | | | | | | 3 | 3 | | | 2 | | |
| [redacted] | [redacted] | 3 | 0 | 3 | 3 | 0 | 0 | 2 | 7 | 6 | 6 | 0 | |
| [redacted] | [redacted] | 0 | 1 | 2 | 0 | 2 | 6 | 0 | 0 | 4 | 6 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 3 | 3 | 2 | 6 | 0 | 5 | 6 | 6 | 0 | |
| [redacted] | [redacted] | 1 | 1 | 2 | 2 | 2 | 7 | 6 | 4 | 5 | 6 | 0 | |
| [redacted] | [redacted] | 1 | 0 | 1 | | | | | 1 | 0 | 1 | 0 | |
| [redacted] | [redacted] | 0 | 0 | 2 | 3 | 0 | 4 | 4 | 0 | 4 | 3 | 0 | |
| [redacted] | [redacted] | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 4 | 2 | 7 | 0 | |
| [redacted] | [redacted] | 2 | 0 | 1 | 3 | 0 | 7 | 6 | 5 | 5 | 0 | 0 | |
| [redacted] | [redacted] | 0 | 0 | 2 | 2 | 0 | 5 | 5 | 1 | 3 | 2 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 3 | 3 | 2 | 6 | 0 | 6 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 0 | 1 | 2 | 7 | 7 | 0 | 7 | 6 | 0 | |
| [redacted] | [redacted] | 2 | 3 | 3 | 3 | 2 | 0 | 7 | 7 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 3 | 2 | 2 | 1 | 1 | 7 | 7 | 0 | 7 | 7 | 0 | |
| [redacted] | [redacted] | 3 | 0 | 3 | 2 | 2 | 7 | 7 | 0 | 6 | 7 | 0 | |
| [redacted] | [redacted] | 1 | 1 | 0 | 1 | 1 | 7 | 7 | 0 | 7 | 6 | 0 | |

Figure 1.1: Handwritten Text to Digital Text

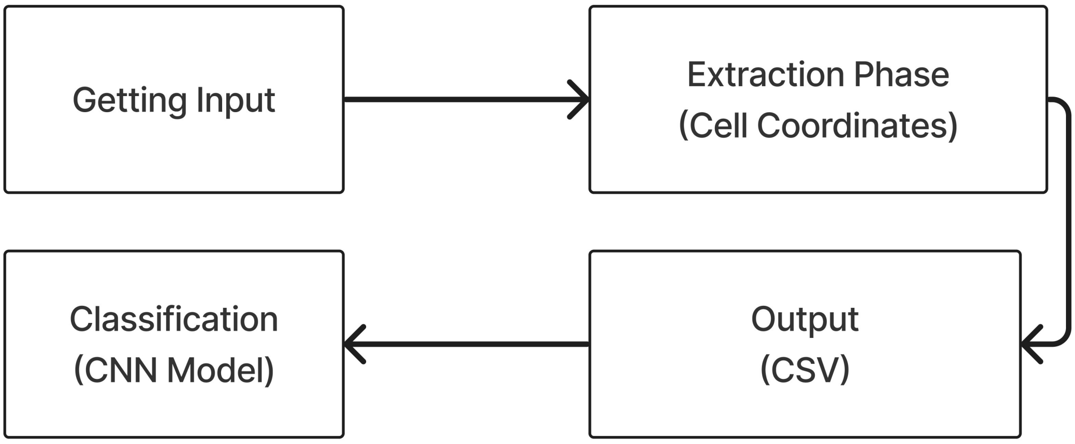
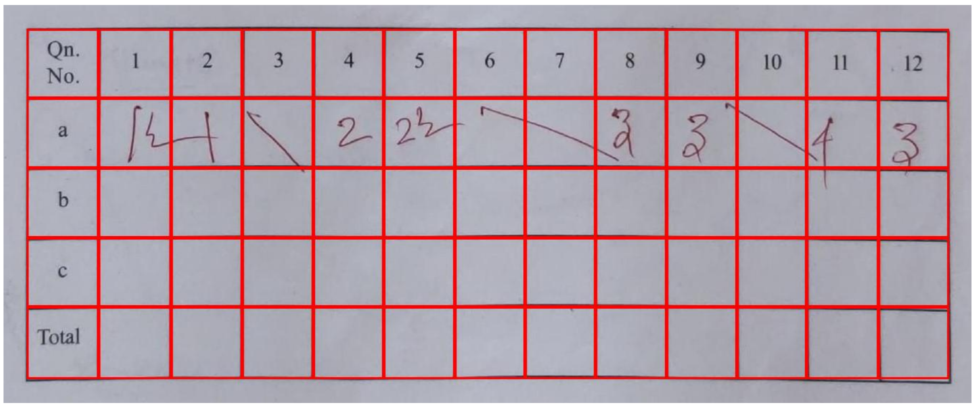

Getting Input
Extraction Phase (Cell Coordinates)
Output (CSV)
Classification (CNN Model)

Figure 3.2: Overview of proposed system

## 3.2 Detailed Description Of The System

The application features a professional and efficient user interface, developed using HTML and Flask. Designed to enhance interactivity and performance, this interface serves as the entry point for teachers to interact seamlessly. A camera icon in the center enables users to quickly open the camera and capture images of answer scripts, streamlining the process. This intuitive design fosters a professional environment, empowering teachers with a streamlined approach to their tasks.
The input images are processed by the img2table library for table recognition. Figure 3.3 shows the recognized table structure.



| Qn. No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 1½ | 4 |  | 2 | 2½ |  |  | 3 | 3 |  | 4 | 3 |
| b |  |  |  |  |  |  |  |  |  |  |  |  |
| c |  |  |  |  |  |  |  |  |  |  |  |  |
| Total |  |  |  |  |  |  |  |  |  |  |  |  |

Figure 3.3: Sample Output Of Table Detection Process

### 3.3.2 Data Collection

Initially, 614 images of data were collected privately. Understanding the fact that this data was insufficient to train the model properly, a collection of an additional 21,600 images from a public Kaggle dataset was made. As per the project requirements, the removal of image classes - numbers 8 and 9 from the public dataset reduced the public dataset image count from 21,600 images to 17,454 images. So the final dataset has a sum of 18,068 images.

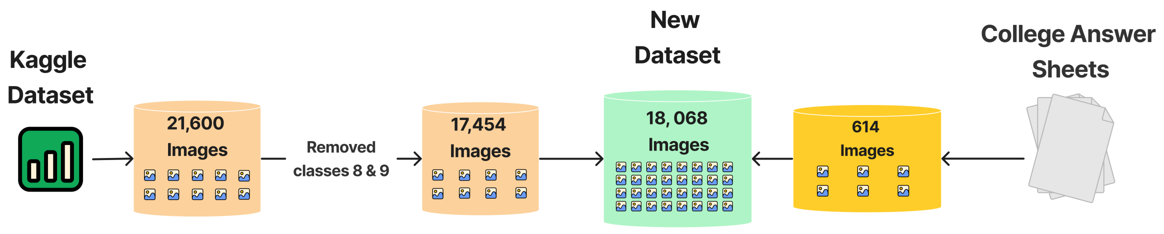


Figure 3.5: Dataset Collection

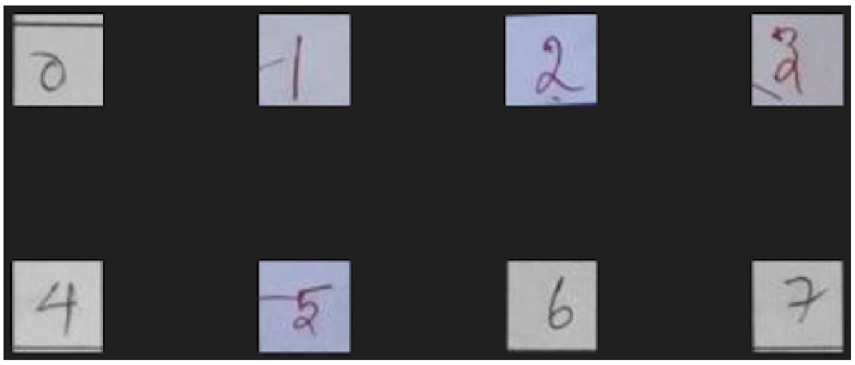


Figure 3.6: Mark cells of private dataset

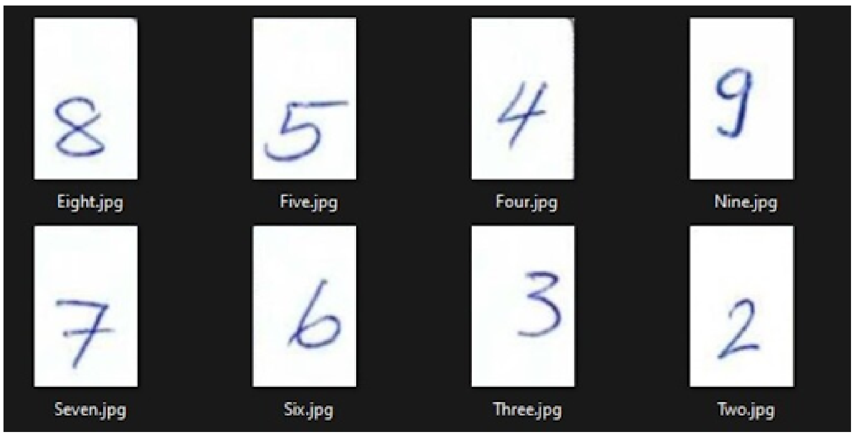


Figure 3.7: Mark cells of the public dataset from Kaggle

## 4.1 Performance Evaluation

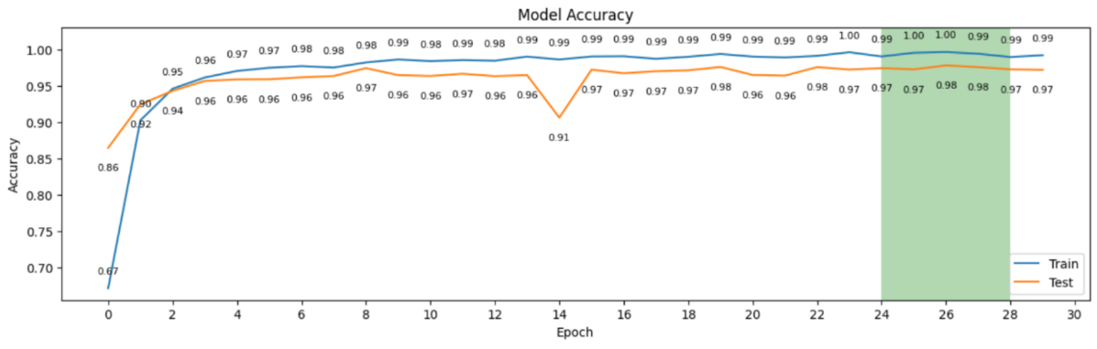


Figure 4.1: Training accuracy per epoch of CNN Model

CNN_Model_0 is improved upon by the addition of a new convolutional layer, which leads to the creation of CNN_Model_1. The smallest of changes has an impact on various performance measures related to the CNN model. Though CNN_Model_0 outperforms CNN_Model_1 in training accuracies, CNN_Model_1 gets the upper hand when it comes to validation and testing accuracies. Upon analyzing Figure 4.1, CNN_Model_1 has a dip in accuracy during the 14th epoch of training and it can be attributed to a phenomenon called "overfitting".

*Overfitting occurs when the model becomes too specialized in learning the training data, losing its generalization ability to new, unseen data.*

At the 14th epoch, the model might have started to memorize specific patterns in the training set, leading to a decrease in accuracy on the validation data. This memorization can cause the model to perform poorly on examples it has not encountered before, resulting in a temporary drop in accuracy.

As a remedy to this issue, several approaches such as early stopping and regularization can be implemented. But the overall graph shows only a negligible sign of overfitting, indicating effective training and resource utilization. The green shade, representing the best epoch-accuracy range, is prominently located towards the end. This signifies optimal performance without wasting resources.

As per the performance, the system is capable of recognizing most of the number written on a paper correctly (Figure 4.2), but still, there are limitations for the model, like the decimal marks (Figure 4.3) or unrecognizable writing styles (Figure 4.4) or due to cut and corrections in the image (Figure 4.5).

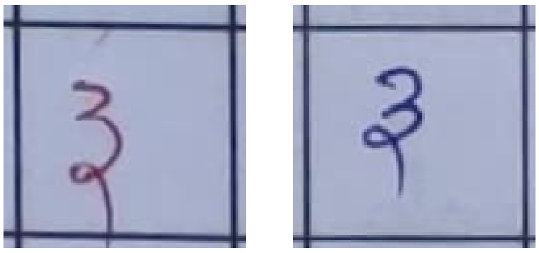


Figure 4.2: Cells with mark written correctly

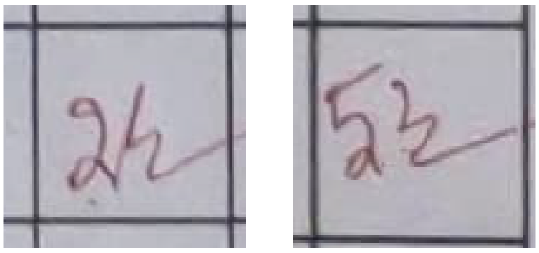


Figure 4.3: Cells with half marks (unable to detect)

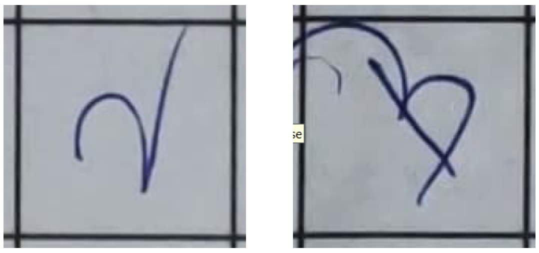


Figure 4.4: Cells with hard-to-recognize marks (may give false result)

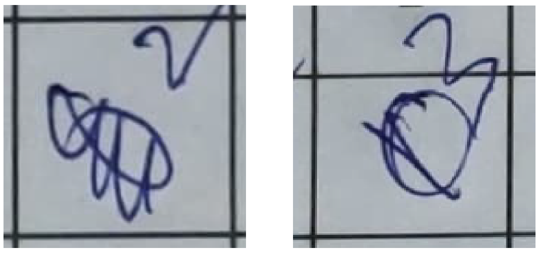


Figure 4.5: Cells with cuts and corrections (unable to detect)

### 4.2.2 Comparision of CNN_Model_0 and CNN_Model_1

The models have achieved a testing accuracy of 99% and 99.2% for CNN_Model_0 and CNN_Model_1 respectively. For an in-depth comparison, Table 4.3 is useful as it can be seen that class 3 of CNN_Model_0 has a little dip while in the figure of CNN_Model_1, the model has learned all classes equally.

Comparing the confusion matrices (given in Table 4.3), CNN_Model_0 showed a slight dip in accuracy for classes 3 and 5, indicating room for improvement. In contrast, CNN_Model_1 demonstrated balanced learning across all classes, indicating its ability to classify instances accurately. These insights gives guidance in refining the models for enhanced performance and accuracy.

The data, given in Table 4.5, compare the performance metrics for the two versions of CNN models. It can be seen that the precision values have not changed with the change in versions. But the recall values have decreased by 0.001 to reach 0.986 in CNN_ Model_1. Also, the F1-scores have decreased by 0.011 to reach 0.988 in CNN_ Model_1.

Table 4.3: Accuracy By Class Comparison CNN_Model_0 and CNN_Model_1

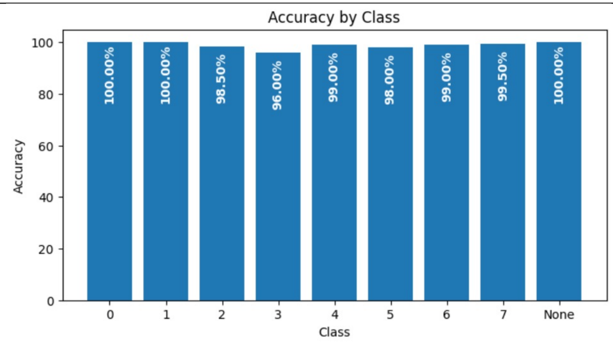
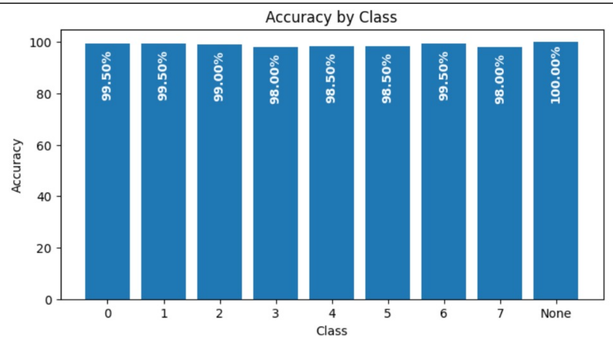
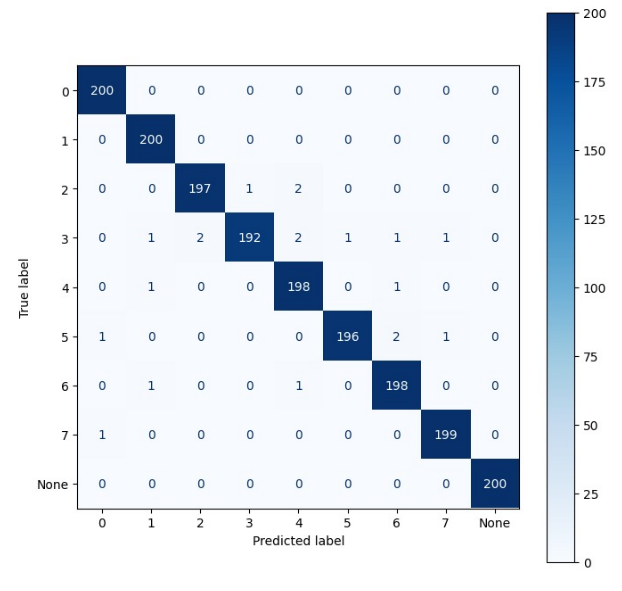
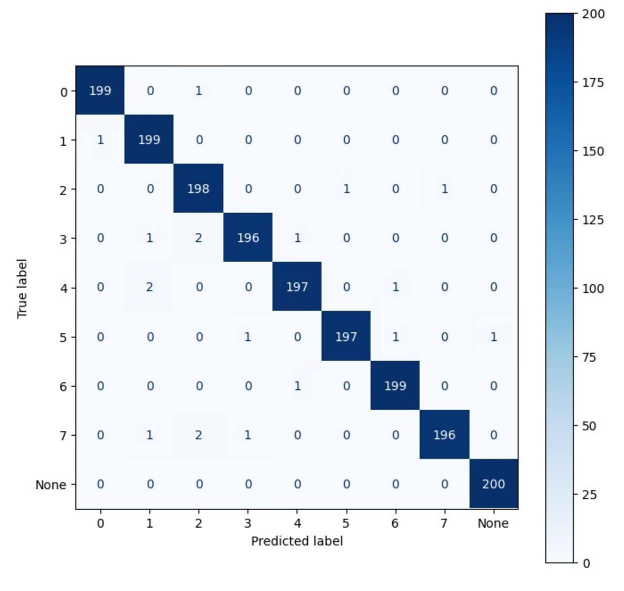
| **CNN_Model_0** | **CNN_Model_1** |
|---|---|
|  |  |

Table 4.4: Confusion Matrix Comparison CNN_Model_0 and CNN_Model_1

| **CNN_Model_0** | **CNN_Model_1** |
|---|---|
|  |  |

| True label \ Predicted label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | None |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 197 | 1 | 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 2 | 192 | 2 | 1 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 198 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 | 196 | 2 | 1 | 0 |
| 6 | 0 | 1 | 0 | 0 | 1 | 0 | 198 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 199 | 0 |
| None | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 |

| True label \ Predicted label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | None |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 199 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 199 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 198 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 2 | 196 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 2 | 0 | 0 | 197 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 197 | 1 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 199 | 0 | 0 |
| 7 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 196 | 0 |
| None | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 |

Table 4.5: Performance metrics comparison for two versions of CNN Models

|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | None | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| CNN_Model_0 |  |  |  |  |  |  |  |  |  |  |
| Precision | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 | 0.98 | 0.99 | 1 | 0.988 |
| Recall | 1 | 1 | 0.98 | 0.96 | 0.99 | 0.98 | 0.99 | 0.99 | 1 | 0.987 |
| F1-Score | 1 | 0.99 | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 1 | 0.999 |
| CNN_Model_1 |  |  |  |  |  |  |  |  |  |  |
| Precision | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 1 | 0.988 |
| Recall | 0.99 | 0.99 | 0.99 | 0.98 | 0.98 | 0.98 | 0.99 | 0.98 | 1 | 0.986 |
| F1-Score | 0.99 | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 1 | 0.988 |

Given below is the graphical representation of Table 4.6

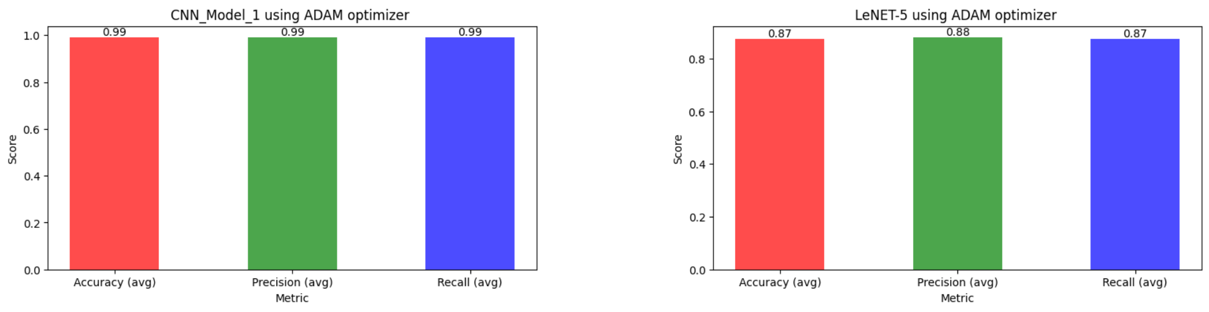


Figure 4.6: Model Comparison: CNN Model 1 vs LeNet5 with ADAM Optimizer

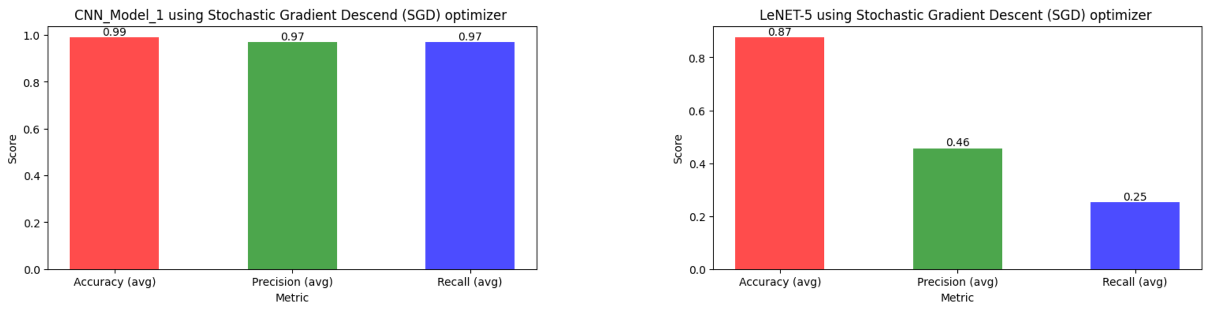


Figure 4.7: Model Comparison: CNN Model 1 vs LeNet5 with SGD Optimizer

## 4.4 Discussion

In the evaluation of two models, namely CNN-Model-1 and LeNet5, their performance was analyzed using two different optimizers: Adam and SGD. The objective was to assess the impact of optimizer choice on the accuracy of the models.

When testing CNN-Model-1 with both Adam and SGD optimizers, it was observed that there was no significant change in accuracy. Regardless of the optimizer used, CNN-Model-1 consistently performed well, indicating its robustness and stability. This suggests that the choice of optimizer had minimal influence on the overall accuracy of the model. Thus, CNN-Model-1 demonstrated its superiority by consistently maintaining a high level of accuracy in both optimizer scenarios.