Project Report: CNN Model for MNIST Digit Classification

1. Introduction

This project focuses on developing a Convolutional Neural Network (CNN) model to classify handwritten digits using the MNIST dataset. The dataset consists of 60,000 training images and 10,000 test images of digits ranging from 0 to 9. The images are grayscale, each with a size of 28x28 pixels.

2. Data Preprocessing

- **Loading and Normalizing Data:** The MNIST dataset was loaded using TensorFlow. The images were normalized by scaling pixel values to a range of 0 to 1.
- **Reshaping Data:** The images were reshaped from a 28x28 matrix to a 28x28x1 array, which represents the width, height, and the single color channel (grayscale).

3. Data Augmentation

To enhance the model's robustness and prevent overfitting, a data augmentation layer was added. This layer applies random rotations, flips, zooms, and translations to the input images during training.

4. Model Architecture

The CNN model was built using the following layers:

- **Input Layer:** Receives images with a shape of 28x28x1.
- **Convolutional Layers:** Multiple convolutional layers with ReLU activation and batch normalization were used to extract features. Padding was applied to maintain the spatial dimensions.
- **Pooling Layers:** Max pooling layers were employed to reduce the spatial dimensions and computational load.
- **Spatial Dropout:** Dropout was used after pooling layers to prevent overfitting.
- **Residual Connections:** A residual block was implemented to help the model learn more efficiently by allowing gradients to flow through the network directly.
- **Global Average Pooling:** This layer replaced the fully connected layers, reducing the number of parameters and preventing overfitting.

- **Output Layer:** A dense layer with 10 units and softmax activation was used to output the probabilities for each of the 10 digit classes.

5. Model Training

- **Optimizer:** The model was compiled using the Adam optimizer with a learning rate of 0.001.
- **Loss Function:** Sparse categorical cross-entropy was used as the loss function.
- **Early Stopping:** The training process was monitored with early stopping to restore the best weights if the validation accuracy did not improve for 10 consecutive epochs.

6. Results

The model was trained over 35 epochs, and the best validation accuracy achieved was 94%. The final evaluation on the test set resulted in the following metrics:

- **Accuracy:** 94%
- **Precision, Recall, F1-Score:** These metrics were computed for each digit class, with an overall weighted average of 94%.

7. Visualization

The training and validation accuracy and loss were plotted to analyze the model's performance over epochs. Additionally, a confusion matrix was generated to visualize the model's performance on each digit class.

8. Prediction and Evaluation

A function was implemented to preprocess and predict the class of a new image. The predicted class and its confidence level were displayed. The model was saved for future use, though an error occurred when attempting to pickle the model using joblib.

9. Conclusion

The CNN model demonstrated strong performance on the MNIST dataset, achieving high accuracy and generalization. However, further improvements could be made by addressing the model saving issue and exploring other model architectures or hyperparameter tuning for potentially better results.