Digit Recognizer Using CNN

This project builds a digit recognition model using a Convolutional Neural Network (CNN) to classify handwritten digits from the MNIST dataset. It also includes a Streamlit web app for user interaction.

by Adil naeem

Dataset Loading and Preparation

Dataset Load

The MNIST dataset is loaded using the TensorFlow Keras API. It contains 60,000 training images and 10,000 test images of handwritten digits.

Data Split

The dataset is split into training and testing sets for model training and evaluation.

Data Normalization

Pixel values are normalized to the range 0-1 by dividing by 255. This improves model performance.

Model Architecture

1 Convolutional Layers

Conv2D layers extract features from the input images using convolutional filters.

2 Max Pooling Layers

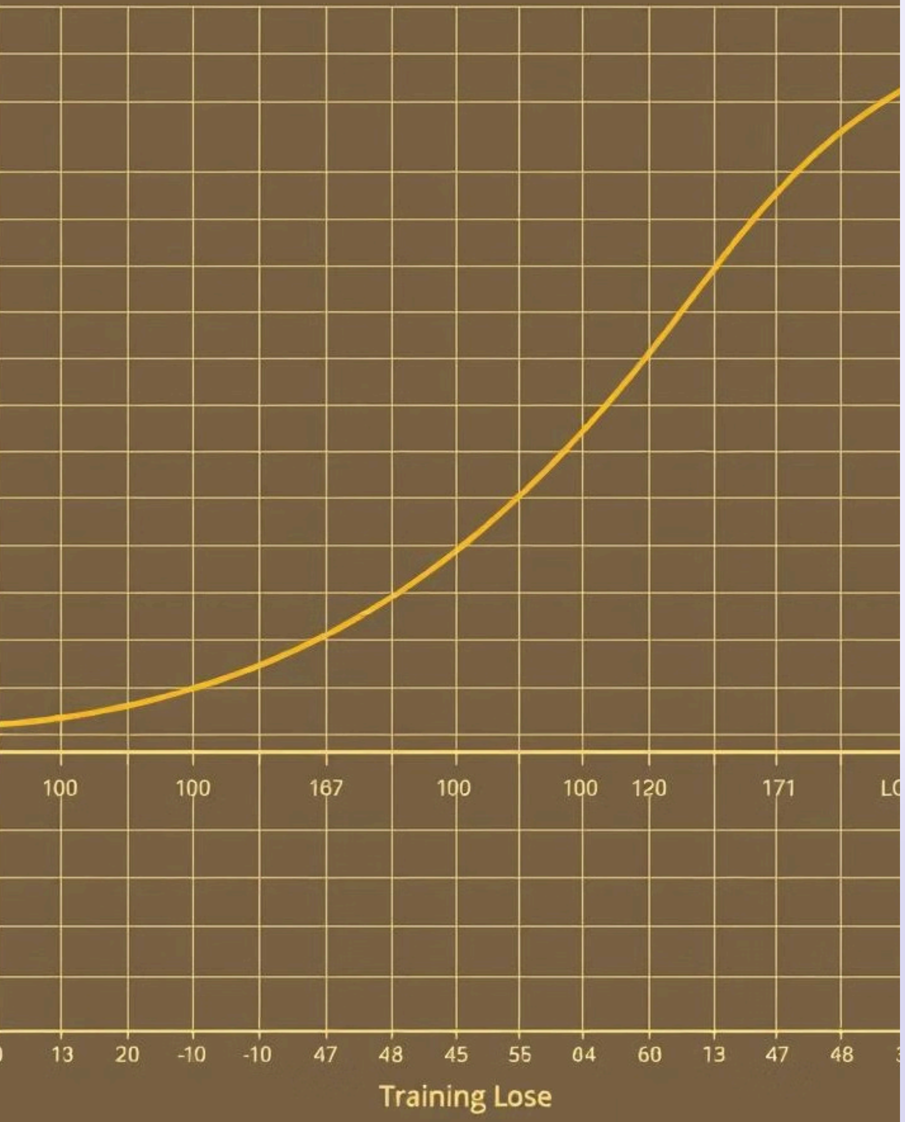
MaxPooling layers downsample the feature maps, reducing the spatial dimension and preserving the most important features.

3 Flatten Layer

The flattened layer converts the output of the convolutional layers into a one-dimensional vector, suitable for the fully connected layers.

4 Dense Layers

Dense layers perform fully connected operations, combining features from the flattened layer and applying activation functions.



Model Training and Evaluation

1

Model Compilation

The model is compiled with the Adam optimizer, sparse categorical cross-entropy loss, and accuracy as the evaluation metric.

2

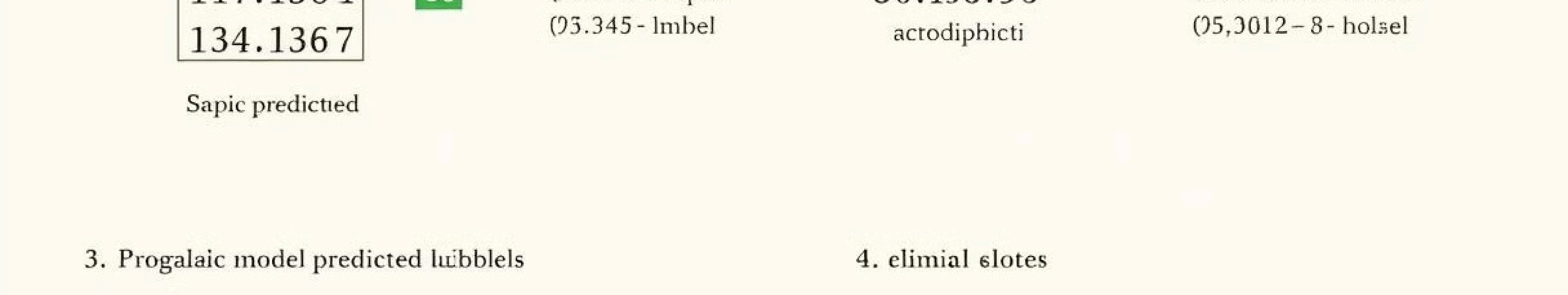
Model Training

The model is trained on the training data for a specified number of epochs, using the validation set to monitor performance.

3

Performance Evaluation

The model's performance is evaluated on the test set using metrics such as accuracy, precision, recall, and F1-score.



Model Prediction and Visualization

1

Predict Digit Labels

The trained model predicts digit labels for the test images.

2

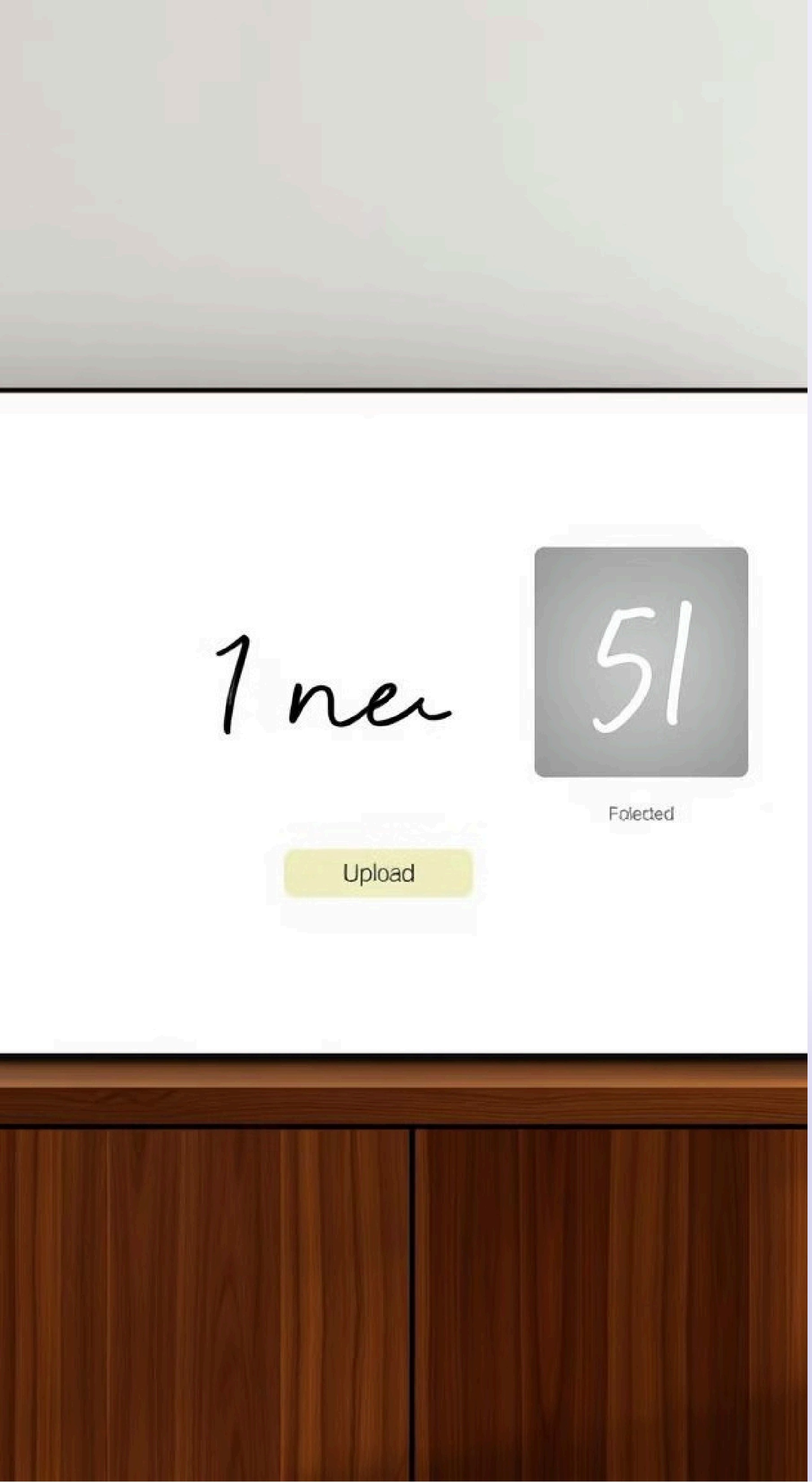
Compare Predictions

The predicted labels are compared with the actual labels, and the results are visualized to assess the model's performance.

3

Performance Analysis

The accuracy and other metrics are analyzed to evaluate the model's effectiveness in recognizing handwritten digits.



Model Saving and Deployment

Model Saving

The trained model is saved as an HDF5 file for future use, allowing for quick loading and prediction.

Streamlit App

A Streamlit web app is built to allow users to upload images of handwritten digits and receive predictions from the saved model.

User Interaction

Users can easily interact with the app by uploading images and viewing the predictions, making the digit recognition model readily accessible.

Future Enhancements

Improved Accuracy

Further training with larger datasets or using more sophisticated network architectures can potentially improve model accuracy.

Real-Time Prediction

The model can be integrated with real-time applications, such as mobile apps or embedded systems, for immediate digit recognition.

Enhanced User Interface

The web app interface can be enhanced with features like image drawing tools, history of past predictions, and user authentication.

Conclusion

This project successfully demonstrates a digit recognition model built using a Convolutional Neural Network. The model achieved satisfactory accuracy in classifying handwritten digits, and the Streamlit web app provides an intuitive interface for user interaction. The project showcases the potential of deep learning in practical applications, such as image recognition and data analysis.