Project Report

The code for this project can be found on GitHub at the following GitHub Repository.

1. Model Implementation:

To implement the model, I initiated the process with image preprocessing. This involved applying morphing and blur techniques to enhance object outlines and reduce the impact of black dots in the images. Dilation was then applied to expand the shapes within the images. These preprocessing steps were performed on both the training and test datasets. For the model architecture, I adopted a design proposed in a YouTube video, here. The Convolutional Neural Network (CNN) consists of three ReLU activated convolutional layers followed by max-pooling to reduce parameter count. A 25% dropout layer was introduced for regularization, followed by flattening to convert the results into a vector. A dense layer creates a fully connected graph, followed by another dropout layer. The model concludes with a dense layer using the SoftMax activation function, producing a vector of length 10, representing the ten data classes. The model was then trained and tested, with the output vector analyzed to determine the class with the highest probability, which was reported as the predicted result.

2. Results:

Initially, I used the original model from the YouTube video, achieving a training accuracy of approximately 85% at epoch 12. After predicting the test data, the model achieved an accuracy of 86.6%. To further experiment, I added a third convolutional layer to the model, which led to more complexity and increased training time. However, the results improved, with a training accuracy of 89% and a test set accuracy of 87.98%, the highest accuracy achieved. Despite the model's increased complexity, it was retained due to its stability and superior performance compared to the original model.

3. Challenges:

The primary challenges I encountered were related to understanding the models and dealing with errors generated by Keras. It took time to grasp concepts like dense and flatten layers, but with increased understanding, these issues became more manageable. Many errors were vague and provided no clear indication of their source. To address these, I sought solutions on platforms like Stack Overflow, learning to discern issues related to input dimensions and layer dimensions, enabling me to tackle similar problems effectively.

4. Conclusion:

This project has been a valuable learning experience, enhancing my understanding of Convolutional Neural Networks (CNNs) and their practical applications. While CNN concepts were well-understood in theory, this project provided hands-on experience, helping me comprehend model architecture, layer application, and how to adapt them to achieve specific output dimensions. It also shed light on issues of overfitting and model performance, which vary with data complexity, model depth, and the number of training epochs. I gained insights into different activation functions, such as ReLU and SoftMax, and improved my understanding of image preprocessing techniques to enhance model results.