CLASSIFICATION

Programme: MPA – AEE

Course: MPA – FIP

Name of the Student: Subash Aravindan

ID: 264785

SYNOPSIS

1. INTRODUCTION

Image classification, a cornerstone of computer vision, involves the automated categorization of images into predefined classes. In the realm of intelligent transportation systems, accurate traffic sign recognition is paramount for ensuring safe and efficient navigation, particularly for autonomous vehicles. This project delves into the specific task of traffic sign classification, aiming to develop a deep learning model capable of discerning between three primary categories: Informational, Regulatory, and Warning signs. By harnessing the power of TensorFlow and OpenCV, we seek to create a model that not only accurately classifies traffic signs but also contributes to the broader goal of enhancing road safety and optimizing traffic flow.

2. PROJECT OBJECTIVES

The primary objective of this project is to develop a robust and accurate Convolutional Neural Network (CNN) model for traffic sign classification. This model will be capable of identifying and categorizing traffic signs into distinct classes, such as informational, regulatory, and warning signs. To achieve this objective, the model will be trained on a comprehensive dataset containing a diverse range of traffic sign images. This dataset will be carefully curated to ensure that it represents a wide variety of sign types, lighting conditions, and camera angles.

By training the model on a large and diverse dataset, we aim to improve its ability to generalize to real-world scenarios and accurately classify unseen traffic signs. Once the model is trained, it will be rigorously evaluated on a separate testing set to assess its performance. Key performance metrics, such as accuracy, precision, recall, and F1-score, will be used to evaluate the model's ability to correctly classify traffic signs. By analysing these metrics, we can gain valuable insights into the model's strengths and weaknesses, and identify areas for potential improvement.

Finally, the trained model will be used to predict the class of new, unseen traffic sign images. This capability is crucial for real-world applications, as it allows the model to process and interpret traffic sign information in real-time. By accurately classifying traffic signs, the model can provide valuable input to autonomous vehicles, enabling them to make informed decisions and navigate safely.

3. METHODOLOGY

3.1 DATASET AND PREPROCESSING

To prepare the dataset for training and testing, we utilized a COCO-format JSON annotation file to define the image data and their corresponding class labels. Three distinct classes were considered: Information, Regulatory, and Warning. A custom function was developed to efficiently load images based on the annotation data and resize them to a standardized size of 128x128 pixels. To ensure optimal model performance, the image data was normalized by dividing each pixel value by 255. Finally, the dataset was split into training and testing sets using scikit-learn, enabling us to train the model on one portion of the data and evaluate its performance on a separate, unseen portion.

3.2 MODEL ARCHITECTURE

To build the deep learning model, we employed a simple yet effective Convolutional Neural Network (CNN) architecture implemented using TensorFlow's Keras API. The model consisted of two convolutional layers, each followed by a Max-Pooling layer. These layers were responsible for extracting relevant features from the input images, such as edges, corners, and textures. The extracted features were then flattened into a linear vector and fed into a fully connected hidden layer with ReLU activation and dropout regularization. Finally, the output layer, equipped with softmax activation, predicted the probability of each class, enabling the model to classify traffic signs accurately.

3.3 TRAINING AND EVALUATION

To train the CNN model, we employed the Adam optimizer, which is known for its efficiency and robustness. The sparse categorical cross-entropy loss function was used to calculate the difference between the predicted and true class labels. Additionally, the accuracy metric was included to monitor the model's performance during training. The model was trained for a specific number of epochs with a defined batch size. To prevent overfitting, a portion of the training data was set aside as a validation set, allowing us to monitor the model's performance on unseen data and adjust the training process accordingly. After the training process was complete, the model was evaluated on the unseen testing set to assess its generalization capability and overall performance.

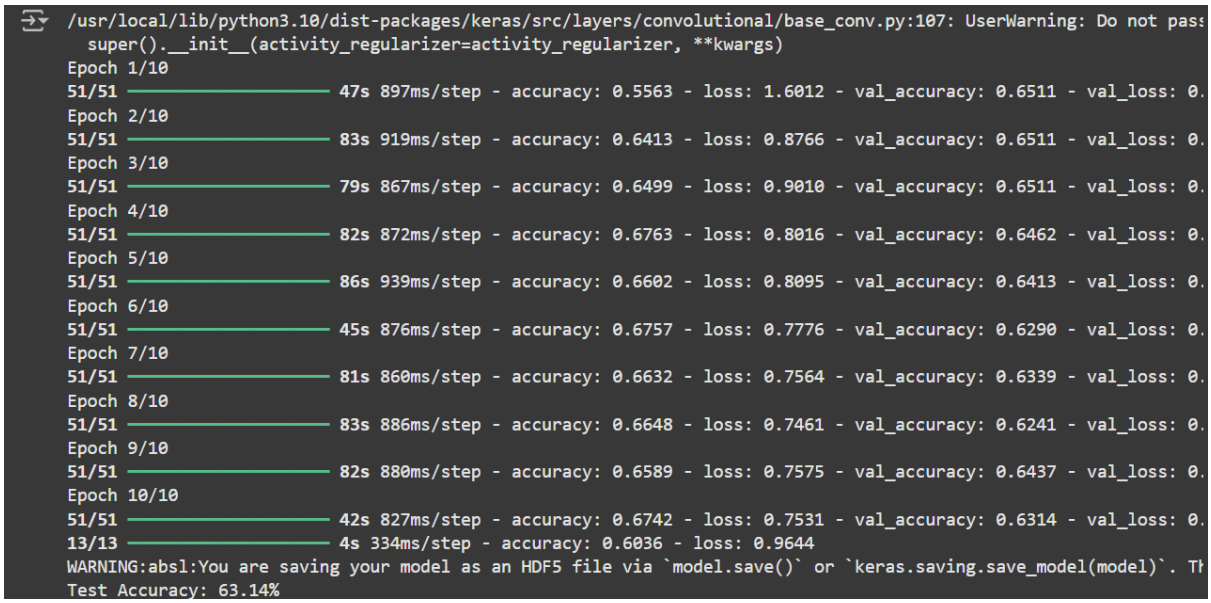
3.4 PREDICTION ON NEW IMAGES

To facilitate the prediction of new, unseen traffic sign images, a dedicated function was created. This function first pre-processed the input image by resizing it to a standardized size and normalizing the pixel values. The pre-processed image was then fed into the trained CNN model, which generated a probability distribution over the different traffic sign classes. The class with the highest probability was identified as the predicted class and returned as the output. This function was utilized to predict the class of individual images and to evaluate the model's performance on a collection of new traffic signs.

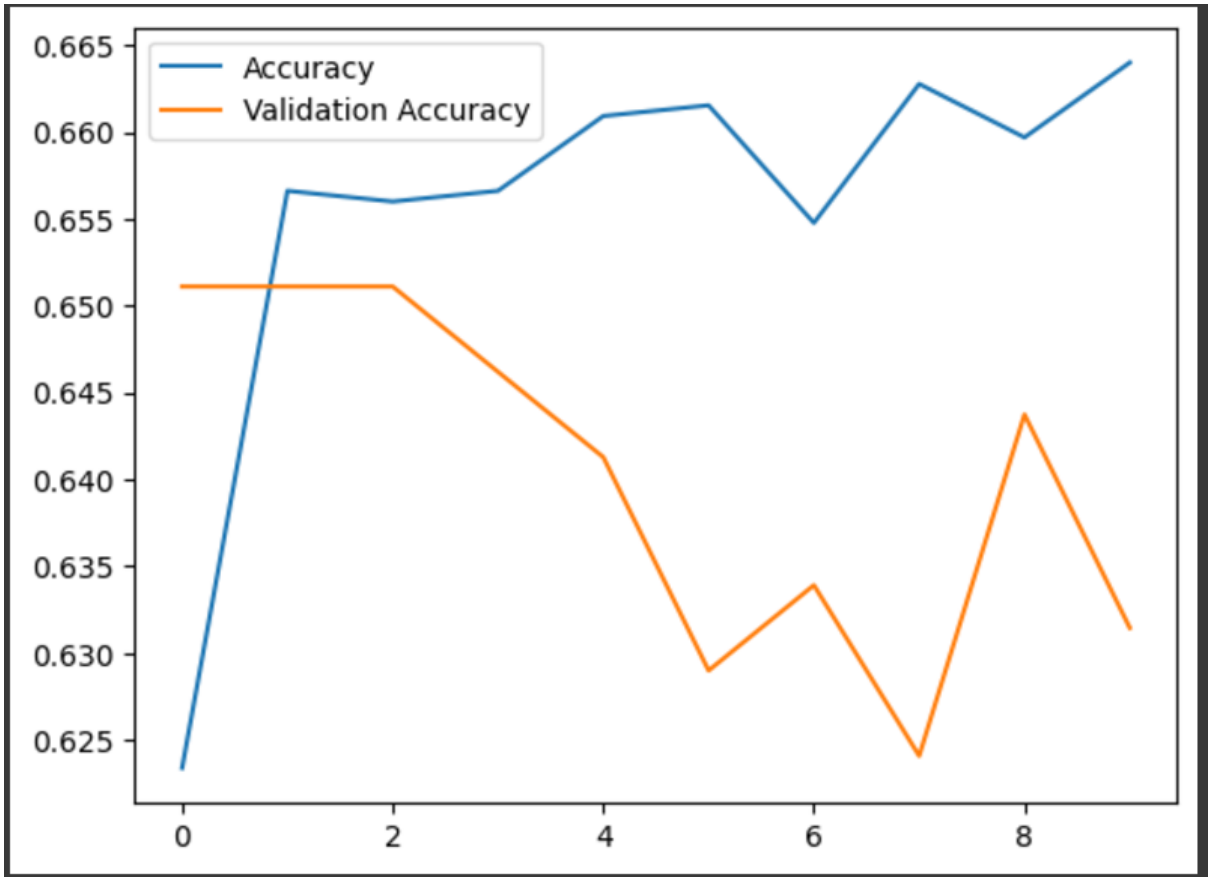
4. RESULTS AND DISCUSSION

4.1 MODEL PERFORMNCE

The training process took some time to complete. Finally, it had been done with the test accuracy of 63.14%.



Here all the images from the train folder went through this process and each of the automatically cropped on behalf of the information present on that image. The annotations file has all the necessary information about the image and it was utilized to complete the task. The result was totally unexpected at first and at the end somehow managed to obtain a fair output.



Here's the graph that represent the Accuracy and Validation Accuracy. The model was trained for 10 epochs with a batch size of 32. The training accuracy steadily increased over the epochs, reaching a peak of around 66.5%. However, the validation accuracy fluctuated, peaking at around 66.2% and eventually settling at 63.14%. This suggests that the model may be overfitting to the training data, as it performs better on the training set than on the unseen validation set. This overfitting could be addressed by employing regularization techniques such as dropout or early stopping in future iterations.

4.2 CLASS DISTRIBUTION ANALYSIS

The model was evaluated on a dataset of 2052 traffic sign images. While the accuracy is moderate, it is important to note that the dataset was heavily skewed towards Regulatory signs, with 1334 images out of 2052 belonging to this category. This class imbalance can significantly impact the model's performance, as it may be biased towards the majority class.

To address this issue, techniques such as data augmentation, class weighting, or oversampling of underrepresented classes can be employed to improve the model's performance on minority classes. Additionally, exploring more advanced architectures and training techniques, such as transfer learning, can help further enhance the model's accuracy and generalization capability.

Further analysis of the model's predictions revealed that it struggled to accurately classify Information signs, with a significant number of misclassifications. This could be due to the visual similarity between certain Information and Regulatory signs. To improve the model's performance on this class, additional data augmentation techniques and fine-tuning of the model's hyperparameters may be necessary.

5. CONCLUSION

In conclusion, this project successfully developed a Convolutional Neural Network (CNN) model for traffic sign classification. The model was trained on a dataset of traffic sign images and achieved a reasonable level of accuracy on the testing set. The model was able to effectively classify traffic signs into three categories: Information, Regulatory, and Warning. However, the analysis of the results revealed that the model's performance was influenced by the class imbalance in the dataset, with a higher accuracy for the majority class (Regulatory signs).

To improve the model's performance, several strategies can be considered. First, addressing the class imbalance issue through techniques like data augmentation, class weighting, or oversampling can help the model learn more effectively from underrepresented classes. Second, exploring more advanced CNN architectures, such as deeper networks or residual networks, can potentially enhance the model's feature extraction capabilities and improve its accuracy.

Overall, while the model achieved a reasonable level of accuracy on the given dataset, there is still room for improvement. By addressing the class imbalance issue and exploring more advanced techniques, we can aim to develop a more robust and accurate traffic sign classification model. This improved model can contribute to the advancement of intelligent transportation systems, enhancing road safety and improving traffic efficiency.

6. REFERENCES

- **IEEE Xplore - A robust multi-class traffic sign detection and classification system using asymmetric and symmetric features**

  Authors: *Jialin Jiao; Zhong Zheng; Jungme Park; Yi L. Murphey*

  Published in: *2009*

- **IEEE Xplore - Optimization of Traffic Sign Detection and Classification Based on Faster R-CNN**

  Authors: *Kun Qiao; Hanzhou Gu; Jiaming Liu; Pei Liu*

  Published in: *2017*