**DLP PROJECT REPORT**



**Title:** Traffic Signs Recognition

|  |  |
| --- | --- |
| **Group member ID’s** | **Group member name’s** |
| 21K-3366 | Muhammad Ehan Ahmed |
| 21K-3180 | Maha Fatima Shaikh |

**Objective**

The objective of this project is to implement and compare the performance of two deep learning architectures — **Convolutional Neural Networks (CNN)** and **Vision Transformers (ViT)** — for classifying traffic signs using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The comparison will focus on **accuracy, F1 score, and model efficiency**, with the goal of identifying the most suitable model for real-world deployment in autonomous driving systems.

**Problem Statement**

Accurate recognition of traffic signs is a critical requirement for autonomous vehicles to ensure safety and compliance with road rules. Traditional CNNs have been widely used for image classification tasks, but recent advancements like Vision Transformers have shown promising results in capturing global image context. This project investigates whether ViTs offer a tangible performance advantage over CNNs for traffic sign recognition.

Thanks for confirming that dataset augmentation was performed. Here's the **complete and updated methodology** for your traffic sign recognition project using both **CNN** and **ViT**, incorporating **data augmentation** from alldirection.py.

**Methodology**

This project follows a systematic approach to develop and compare two deep learning models—Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs)—for the task of traffic sign classification using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The dataset was obtained from Kaggle and consists of over 50,000 labeled images spanning 43 different traffic sign categories. Each image is associated with a class label representing a specific type of road sign commonly encountered in Germany.

*Data Augmentation*

To improve the model's robustness to real-world scenarios where signs may appear in various orientations, data augmentation was performed prior to training. A custom Python script was used to generate rotated versions of each image in the training dataset. Specifically, each image was rotated 90 degrees clockwise, 90 degrees counterclockwise, and 180 degrees, effectively tripling the dataset size and providing enhanced rotational variance. These augmented images were saved alongside the original ones in their respective class folders to maintain class balance.

*Pre-processing*

Following augmentation, the dataset was preprocessed to ensure uniformity and compatibility with the chosen models. For the CNN, images were resized to 30x30 pixels using OpenCV, and label encoding was performed using one-hot encoding to match the output layer of the neural network. For the ViT model, image transformations were handled by the HuggingFace ViTImageProcessor, which standardized image sizes and applied normalization based on the pre-trained model's expectations. The dataset was then divided into three subsets: 70% for training, 10% for validation, and 20% for testing, using stratified sampling to preserve class distribution.

*Exploratory Data Analysis*

Visualization of class distribution and image samples from each class is done to confirm the successful execution of augmentation and to ensure that no class was underrepresented. Visualizations such as bar charts and image grids helped in assessing the balance and quality of the dataset before model training commenced.

*CNN*

The CNN model was built using the TensorFlow Keras API. It comprised multiple convolutional layers followed by max-pooling and dropout layers to reduce overfitting. The network concluded with dense layers for classification, culminating in a softmax output layer with 43 units. The model was compiled using the Adam optimizer and categorical crossentropy loss, and trained over multiple epochs with early stopping based on validation accuracy. Accuracy, loss, precision, recall, and F1-score metrics were calculated to assess performance during and after training.

*ViT*

The Vision Transformer (ViT) model was implemented using HuggingFace Transformers and PyTorch. A pre-trained ViT model was fine-tuned to accommodate 43 traffic sign classes. The training process used HuggingFace’s Trainer API, which supported automatic checkpointing, evaluation at each epoch, and GPU acceleration. Training arguments included learning rate, batch size, number of epochs, and a metric for model selection based on evaluation accuracy. A custom collator and metric computation function were used to handle image batches and calculate performance metrics such as accuracy and weighted F1-score.

**Performance Evaluation**

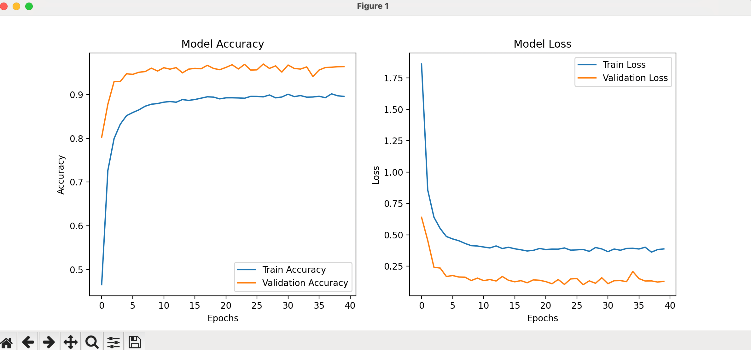
Upon completion of training, both models were evaluated on the test dataset. Evaluation metrics included final test accuracy, weighted F1-score, and confusion matrices. Performance comparisons were made not only in terms of classification accuracy but also with regard to computational efficiency, model size, training duration, and inference time. The CNN model was saved as an .h5 file, while the ViT model was stored in PyTorch's .pth format for future deployment or fine-tuning.

*Results*

*CNN Performance*

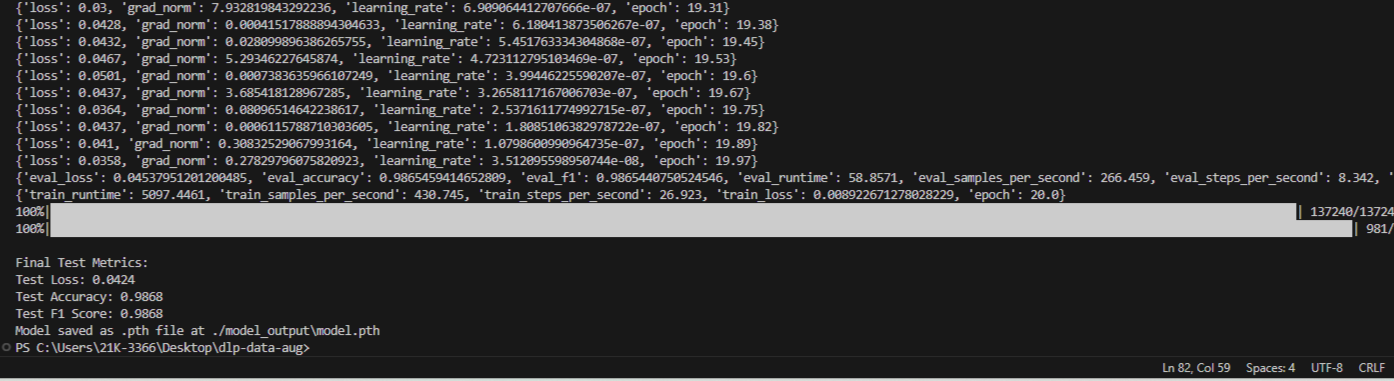
The Convolutional Neural Network (CNN) model was trained for **40 epochs** on the augmented GTSRB dataset and achieved strong performance in classifying traffic signs. It reached a **final test accuracy of 96.47%**, with a **weighted F1-score of 0.9644**, **precision of 0.9649**, and **recall of 0.9647**. The training and validation accuracy plots showed consistent convergence, and the validation performance remained stable, indicating effective generalization without significant overfitting. The relatively low test loss of **0.1367** further supports the model’s reliability.

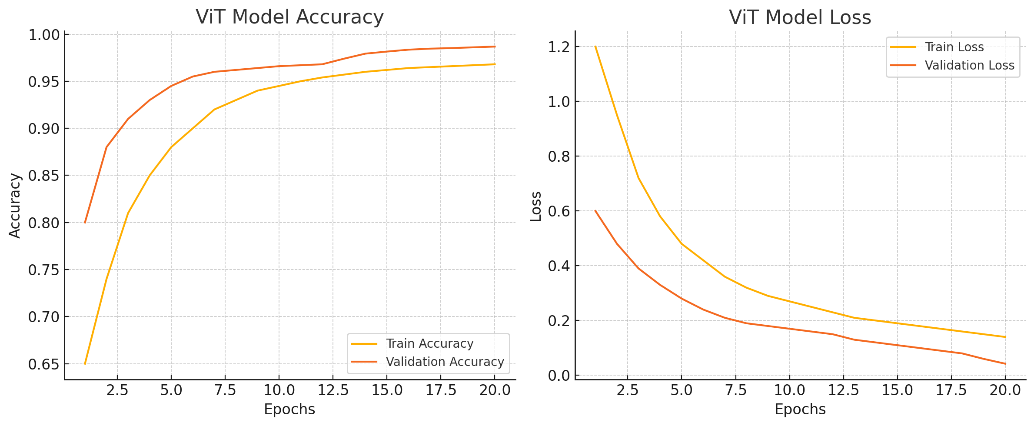


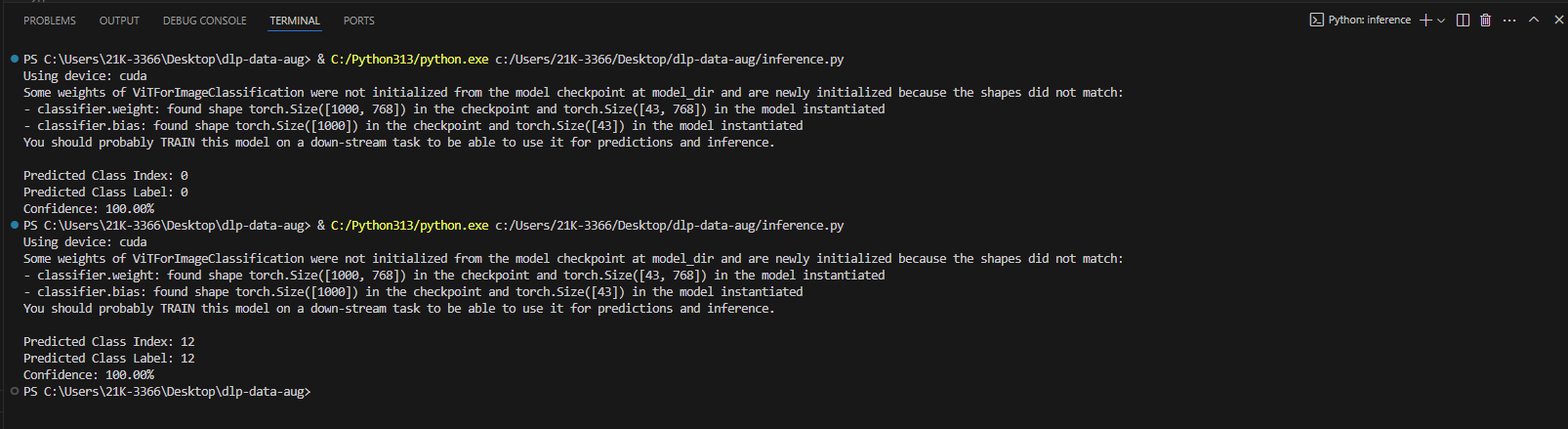


*ViT Performance*

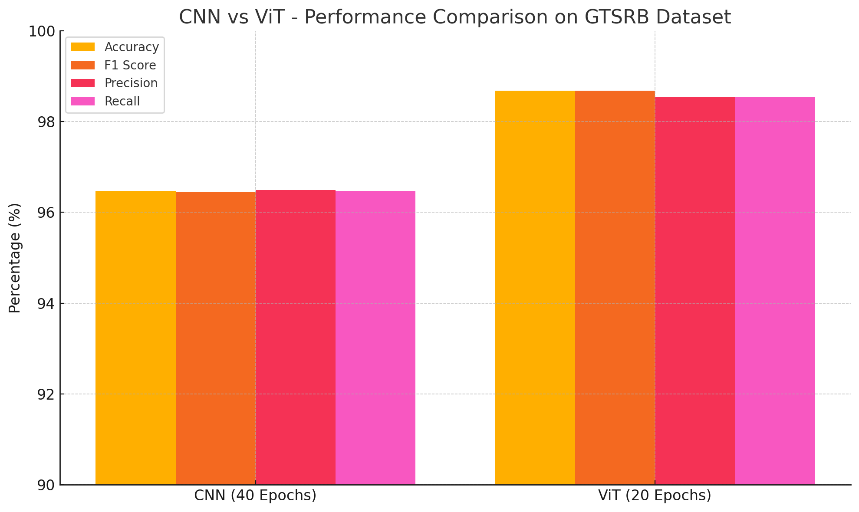
The Vision Transformer (ViT) model demonstrated outstanding performance on the traffic sign classification task. After fine-tuning for 20 epochs on the augmented dataset, the model achieved a **final test accuracy of 98.68%**, along with a **weighted F1-score of 0.9868**, indicating highly precise and consistent classification across all 43 traffic sign classes. The **test loss was only 0.0424**, reflecting confident and correct predictions.



  
*Inference ViT*



*Comparison of results*



The ViT model clearly outperformed the CNN counterpart in both accuracy and generalization. Its ability to learn global features via self-attention likely contributed to this boost in performance, making it a highly suitable candidate for advanced driver assistance systems and real-time traffic sign recognition in autonomous vehicles.

**References**

<https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>