AUTONOMOUS VEHICLES

A PROJECT REPORT (PHASE 4)

submitted by

HARIHARA SUDHAN. R

810022104088

In partial fulfilment of the requirements for

Bachelor of Engineering

in

COMPUTER SCIENCE AND ENGINEERING

Under the course of

ARTIFICIAL INTELLIGENCE

UNIVERSITY COLLEGE OF ENGINEERING,

BIT CAMPUS,

ANNA UNIVERSITY, TIRUCHIRAPPALLI – 620 024

Team Members:

- Arunkumar. V (810022104062)
- Salai sanamrgam. J (810022104072)
- Sathish. S (810022104075)
- Kirthick roshan (810022104116)

Introduction:

Phase 4 of our project is dedicated to model development and evaluation, focusing on building effective object detection models using the prepared dataset. This phase involves selecting appropriate algorithms, training models, fine-tuning parameters, and rigorously evaluating their performance. The objective is to ensure that our models can accurately detect objects in diverse road conditions in Bangladesh.

Objectives:

- □ **Develop Advanced AI Models:** Design and implement AI models for perception, localization, path planning, and decision-making, leveraging state-of-the-art techniques in computer vision, machine learning, and robotics.
- □ **Select Appropriate Evaluation Metrics:** Identify and employ suitable evaluation metrics tailored to each aspect of the autonomous driving system, ensuring comprehensive assessment of model performance across diverse scenarios and tasks.
- □ **Train Models with Cleaned Dataset:** Utilize the preprocessed dataset to train AI models, employing rigorous training methodologies to optimize model parameters and configurations for optimal performance.
- □ **Validate Models for Generalization:** Conduct thorough validation procedures to ensure that trained models generalize well to unseen data and can robustly handle real-world driving conditions, including variations in weather, lighting, and road infrastructure.
- □ **Conduct Extensive Testing:** Perform extensive testing of the developed AI models under various simulated and real-world driving scenarios, including urban, suburban, and highway environments, to evaluate their performance, reliability, and safety.
- □ **Document Model Development Process:** Document the entire process of model development and evaluation comprehensively, including data preprocessing steps, model architectures, training procedures, evaluation metrics, and testing results, to facilitate transparency, reproducibility, and knowledge sharing within the project team and broader research community.

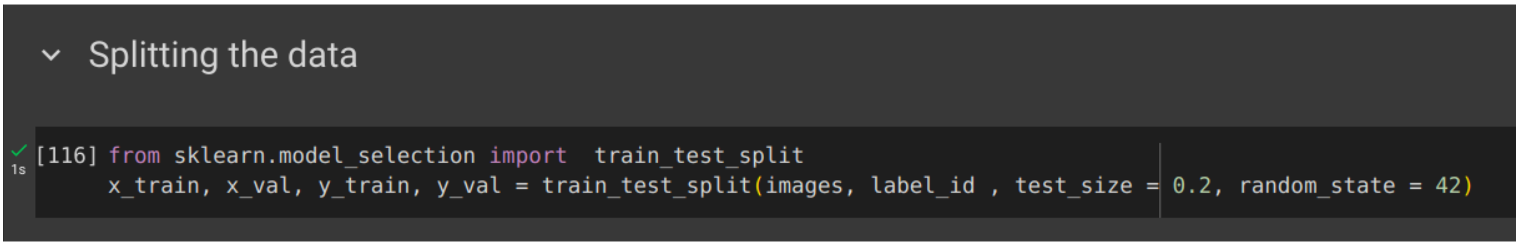
Key Considerations:

1. **Safety and Reliability:** Prioritize safety above all else by implementing robust fail-safe mechanisms and safety protocols. Ensure that the autonomous system can reliably detet and respond to potential hazards in real-time to prevent accidents and ensure passenger safety.
2. **Regulatory Compliance:** Stay compliant with regulations and standards governing autonomous vehicles in your target deployment regions. Adhere to legal requirements and safety certifications to ensure regulatory approval and public trust in your technology.
3. **Data Privacy and Security:** Implement stringent data privacy measures to safeguard sensitive information collected by the autonomous vehicle, such as personal data of

passengers and navigation data. Ensure secure data transmission, storage, and processing to prevent data breaches and protect user privacy.

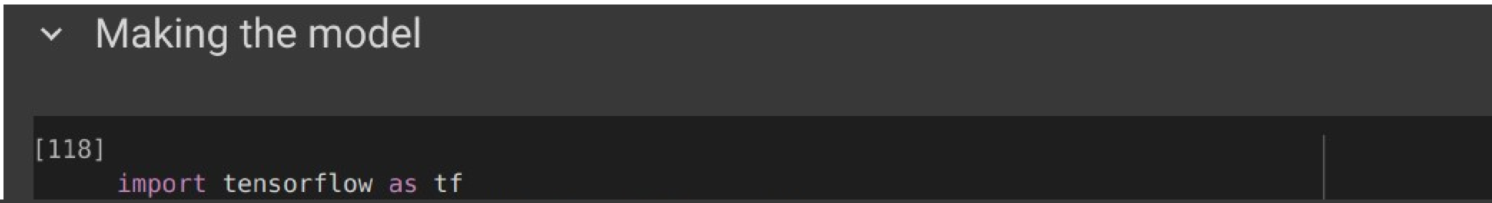
4. **User Acceptance and Trust:** Build trust and confidence in autonomous technology by transparently communicating its capabilities, limitations, and safety features to users and stakeholders. Address concerns and misconceptions through education, demonstration, and ongoing engagement with the community.
5. **Continuous Testing and Validation:** Conduct rigorous testing and validation of the autonomous system under diverse environmental conditions, road scenarios, and edge cases. Use simulation-based testing and real-world trials to assess system performance, identify weaknesses, and iterate on improvements continuously.

Phase 4: Model Training, Evaluation



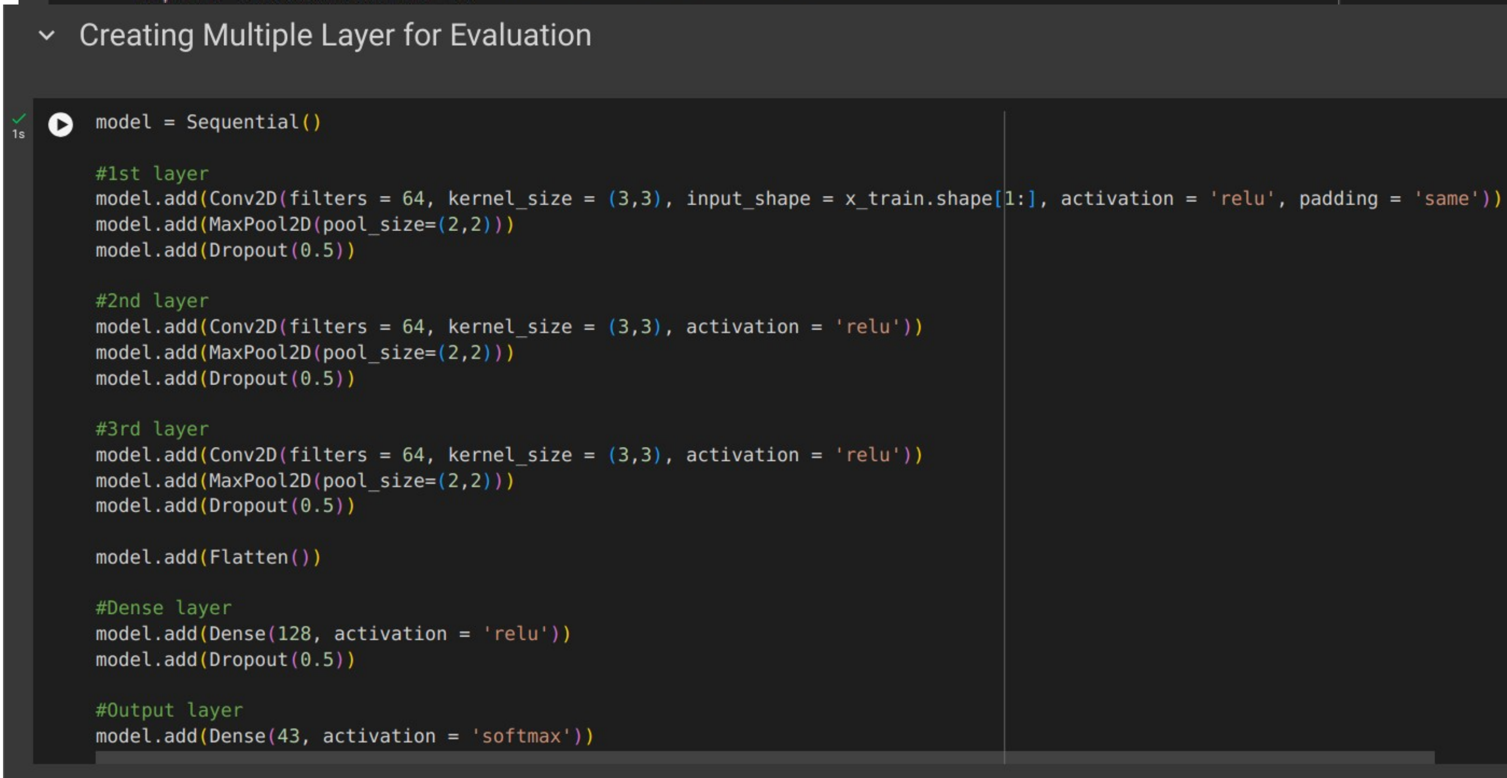
Splitting the data

```
[116] from sklearn.model_selection import  train_test_split
      x_train, x_val, y_train, y_val = train_test_split(images, label_id , test_size = 0.2, random_state = 42)
```



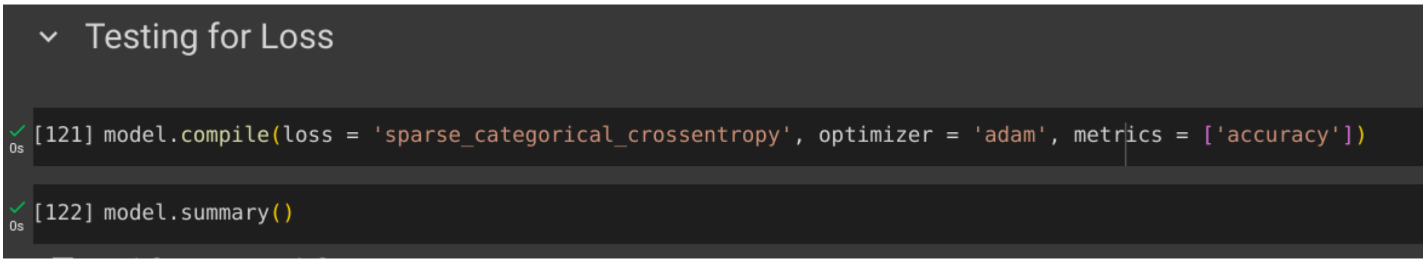
Making the model

```
[118]
     import tensorflow as tf
```



Creating Multiple Layer for Evaluation

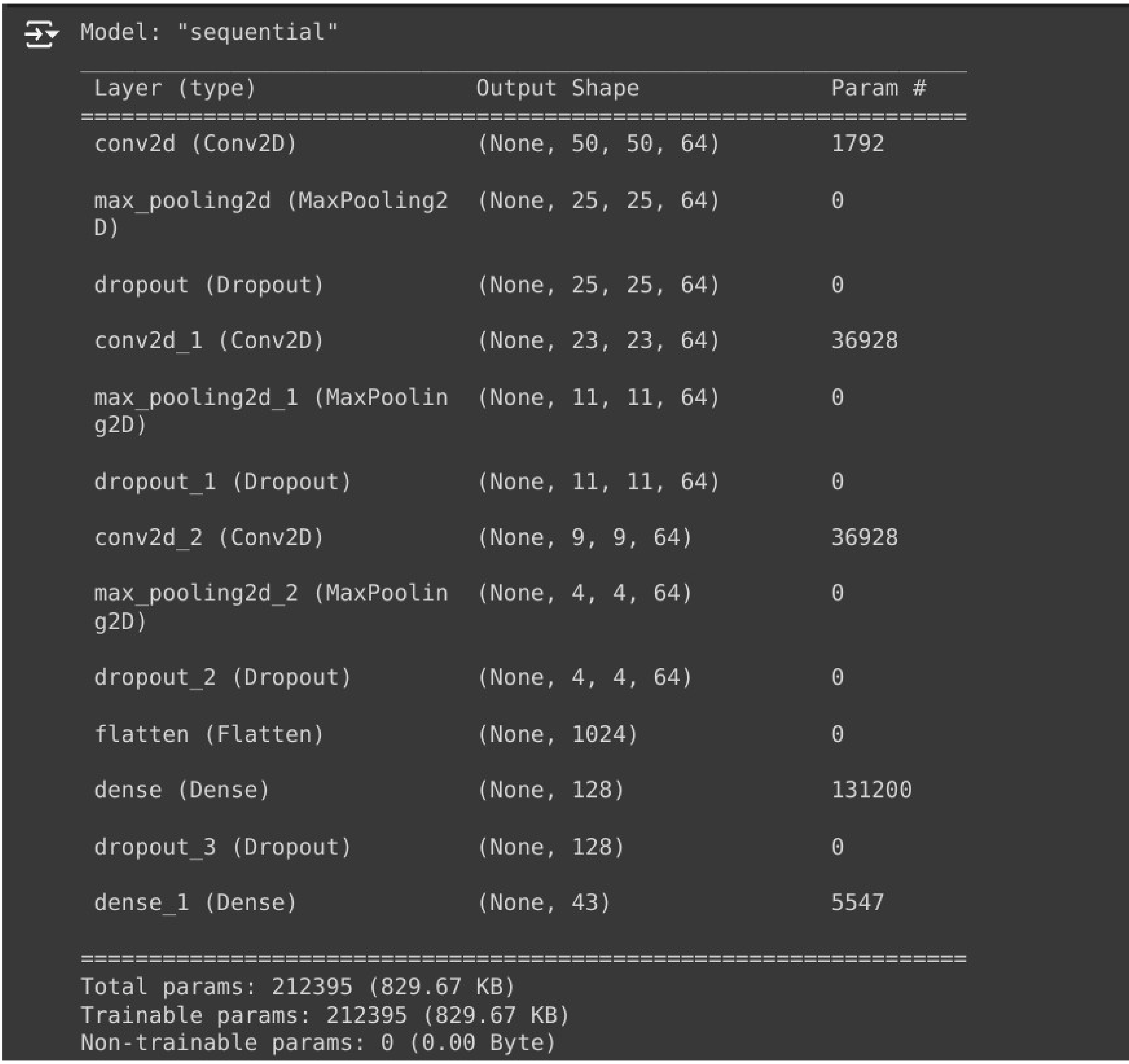
```
model = Sequential()

#1st layer
model.add(Conv2D(filters = 64, kernel_size = (3,3), input_shape = x_train.shape[1:], activation = 'relu', padding = 'same'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.5))

#2nd layer
model.add(Conv2D(filters = 64, kernel_size = (3,3), activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.5))

#3rd layer
model.add(Conv2D(filters = 64, kernel_size = (3,3), activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Flatten())

#Dense layer
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.5))

#Output layer
model.add(Dense(43, activation = 'softmax'))
```



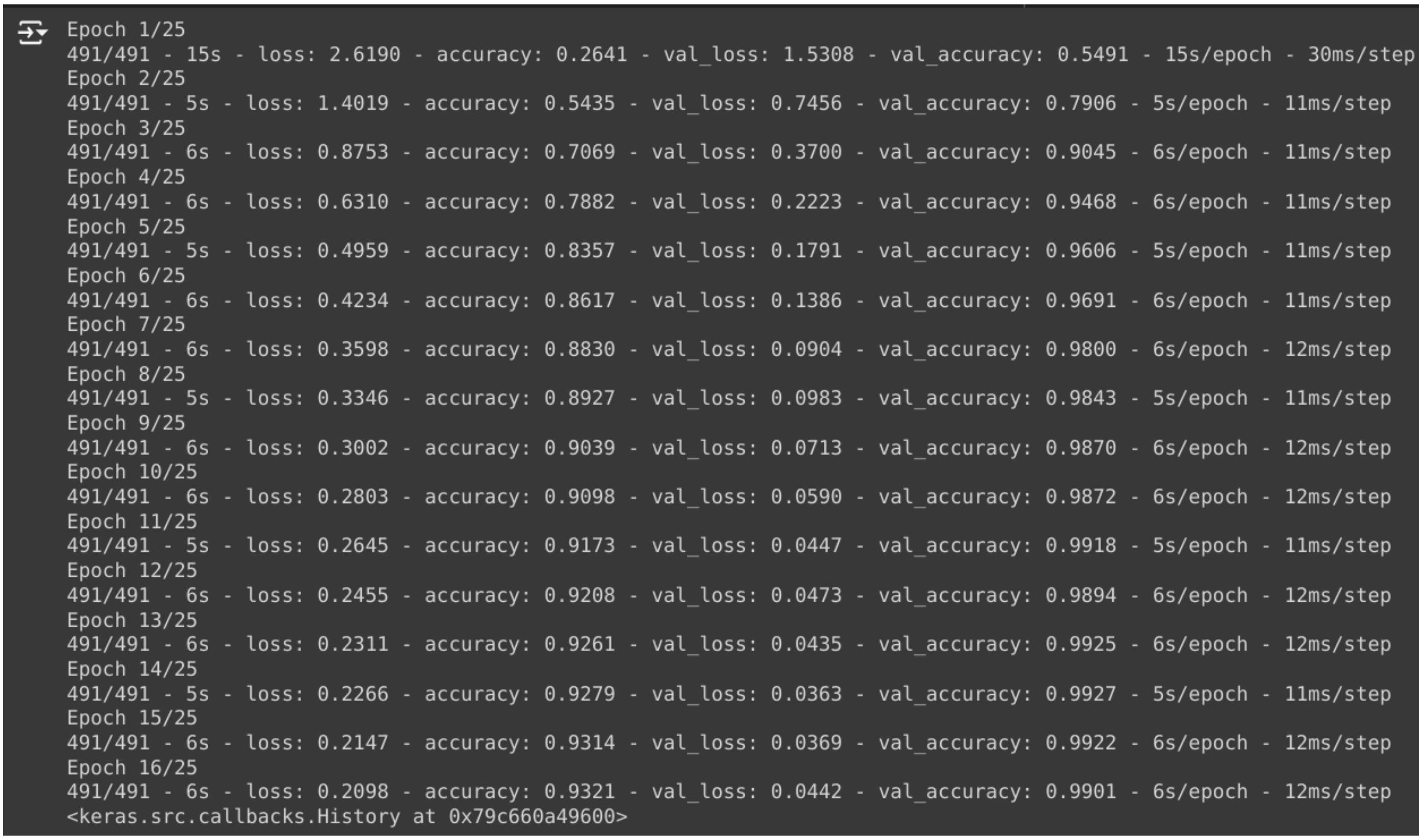
Testing for Loss

```
[121] model.compile(loss = 'sparse_categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

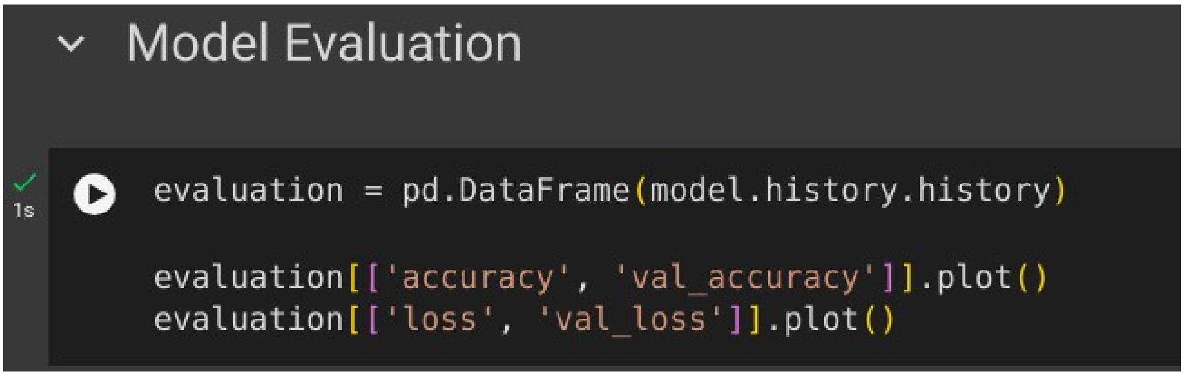
```
[122] model.summary()
```



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 50, 50, 64)        1792

 max_pooling2d (MaxPooling2  (None, 25, 25, 64)        0
 D)

 dropout (Dropout)           (None, 25, 25, 64)        0

 conv2d_1 (Conv2D)           (None, 23, 23, 64)        36928

 max_pooling2d_1 (MaxPoolin  (None, 11, 11, 64)        0
 g2D)

 dropout_1 (Dropout)         (None, 11, 11, 64)        0

 conv2d_2 (Conv2D)           (None, 9, 9, 64)          36928

 max_pooling2d_2 (MaxPoolin  (None, 4, 4, 64)          0
 g2D)

 dropout_2 (Dropout)         (None, 4, 4, 64)          0

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 128)               131200

 dropout_3 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 43)                5547

=================================================================
Total params: 212395 (829.67 KB)
Trainable params: 212395 (829.67 KB)
Non-trainable params: 0 (0.00 Byte)
```

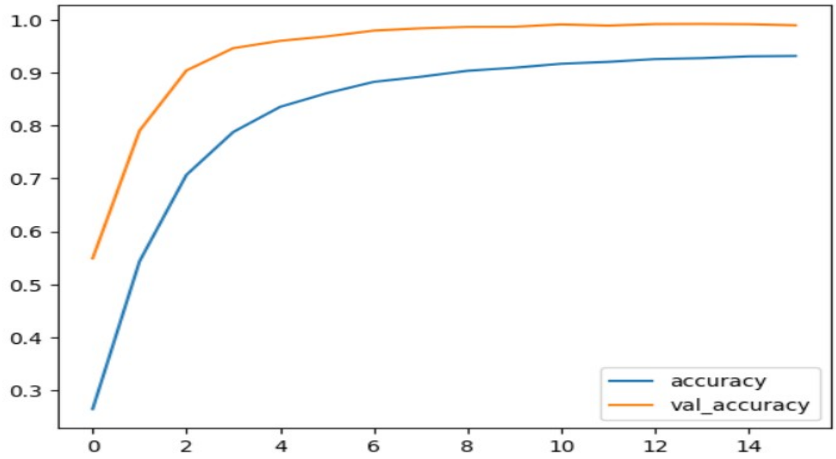


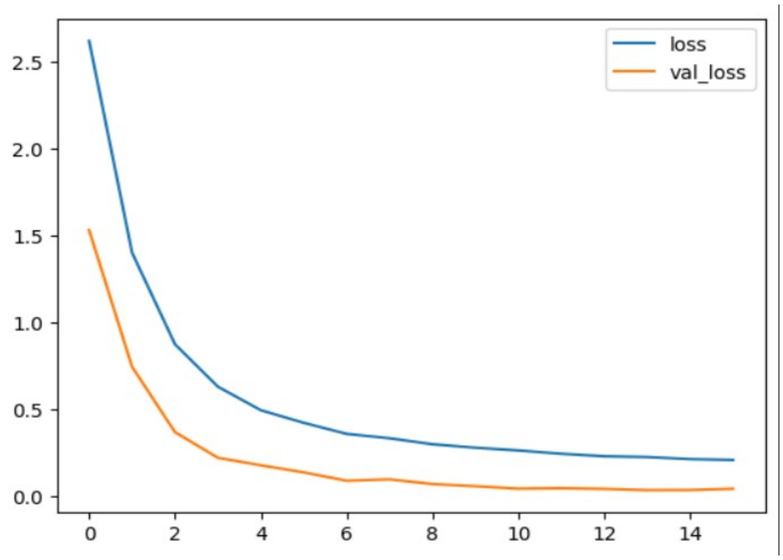
```
Epoch 1/25
491/491 - 15s - loss: 2.6190 - accuracy: 0.2641 - val_loss: 1.5308 - val_accuracy: 0.5491 - 15s/epoch - 30ms/step
Epoch 2/25
491/491 - 5s - loss: 1.4019 - accuracy: 0.5435 - val_loss: 0.7456 - val_accuracy: 0.7906 - 5s/epoch - 11ms/step
Epoch 3/25
491/491 - 6s - loss: 0.8753 - accuracy: 0.7069 - val_loss: 0.3700 - val_accuracy: 0.9045 - 6s/epoch - 11ms/step
Epoch 4/25
491/491 - 6s - loss: 0.6310 - accuracy: 0.7882 - val_loss: 0.2223 - val_accuracy: 0.9468 - 6s/epoch - 11ms/step
Epoch 5/25
491/491 - 5s - loss: 0.4959 - accuracy: 0.8357 - val_loss: 0.1791 - val_accuracy: 0.9606 - 5s/epoch - 11ms/step
Epoch 6/25
491/491 - 6s - loss: 0.4234 - accuracy: 0.8617 - val_loss: 0.1386 - val_accuracy: 0.9691 - 6s/epoch - 11ms/step
Epoch 7/25
491/491 - 6s - loss: 0.3598 - accuracy: 0.8830 - val_loss: 0.0904 - val_accuracy: 0.9800 - 6s/epoch - 12ms/step
Epoch 8/25
491/491 - 5s - loss: 0.3346 - accuracy: 0.8927 - val_loss: 0.0983 - val_accuracy: 0.9843 - 5s/epoch - 11ms/step
Epoch 9/25
491/491 - 6s - loss: 0.3002 - accuracy: 0.9039 - val_loss: 0.0713 - val_accuracy: 0.9870 - 6s/epoch - 12ms/step
Epoch 10/25
491/491 - 6s - loss: 0.2803 - accuracy: 0.9098 - val_loss: 0.0590 - val_accuracy: 0.9872 - 6s/epoch - 12ms/step
Epoch 11/25
491/491 - 5s - loss: 0.2645 - accuracy: 0.9173 - val_loss: 0.0447 - val_accuracy: 0.9918 - 5s/epoch - 11ms/step
Epoch 12/25
491/491 - 6s - loss: 0.2455 - accuracy: 0.9208 - val_loss: 0.0473 - val_accuracy: 0.9894 - 6s/epoch - 12ms/step
Epoch 13/25
491/491 - 6s - loss: 0.2311 - accuracy: 0.9261 - val_loss: 0.0435 - val_accuracy: 0.9925 - 6s/epoch - 12ms/step
Epoch 14/25
491/491 - 5s - loss: 0.2266 - accuracy: 0.9279 - val_loss: 0.0363 - val_accuracy: 0.9927 - 5s/epoch - 11ms/step
Epoch 15/25
491/491 - 6s - loss: 0.2147 - accuracy: 0.9314 - val_loss: 0.0369 - val_accuracy: 0.9922 - 6s/epoch - 12ms/step
Epoch 16/25
491/491 - 6s - loss: 0.2098 - accuracy: 0.9321 - val_loss: 0.0442 - val_accuracy: 0.9901 - 6s/epoch - 12ms/step
<keras.src.callbacks.History at 0x79c660a49600>
```

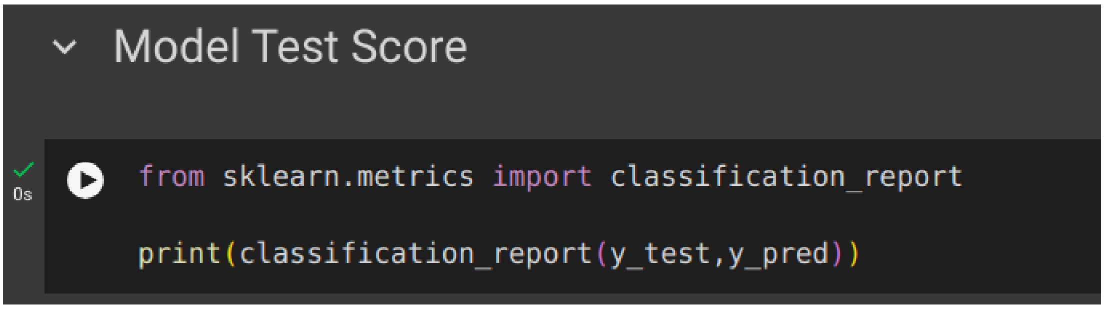


Model Evaluation

```
evaluation = pd.DataFrame(model.history.history)

evaluation[['accuracy', 'val_accuracy']].plot()
evaluation[['loss', 'val_loss']].plot()
```

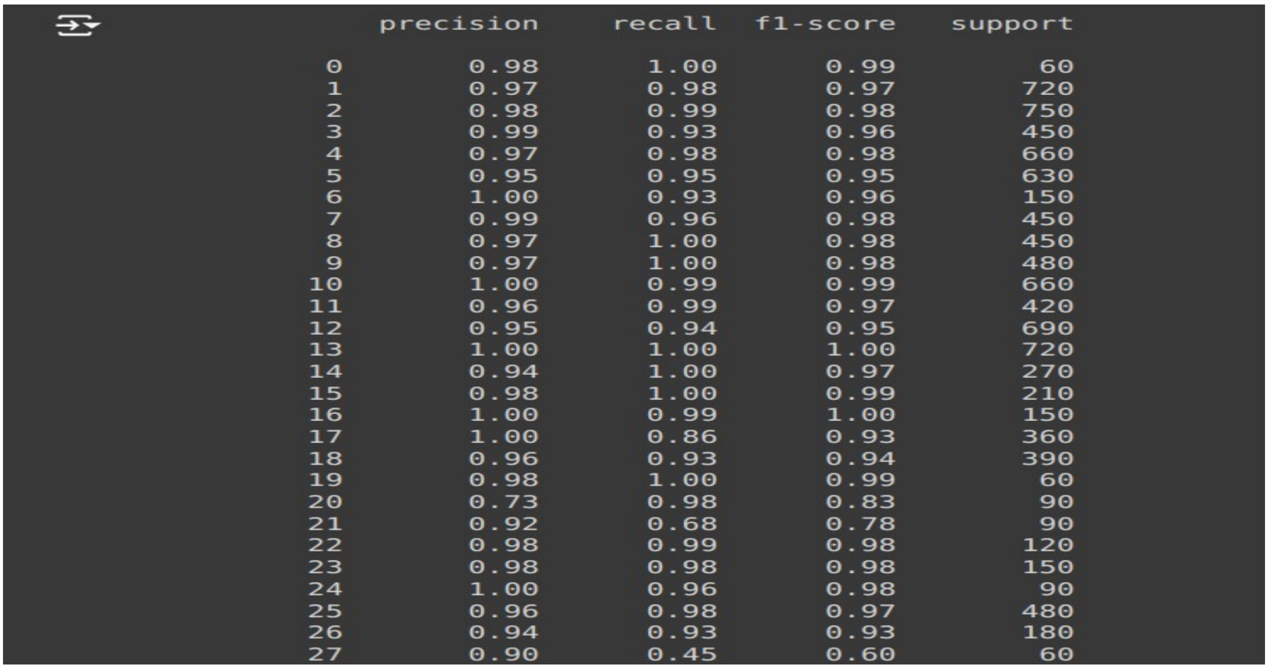




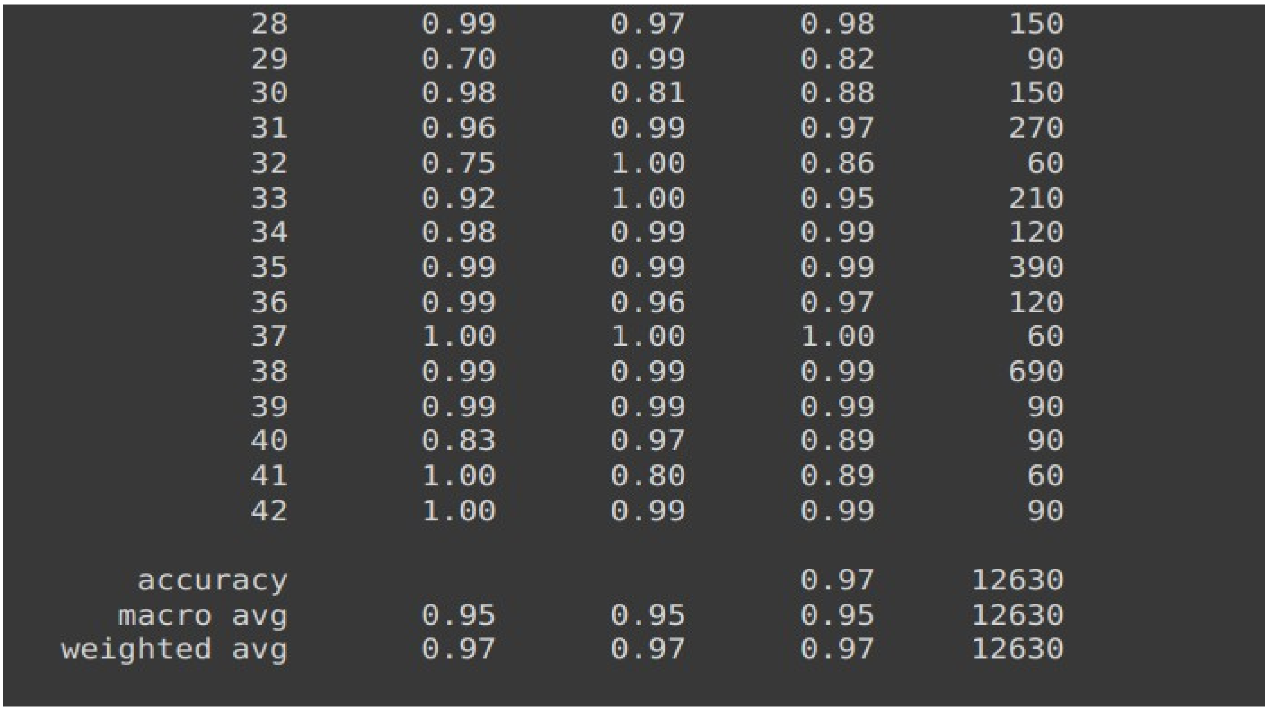


Model Test Score

```
from sklearn.metrics import classification_report

print(classification_report(y_test,y_pred))
```



```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        60
           1       0.97      0.98      0.97       720
           2       0.98      0.99      0.98       750
           3       0.99      0.93      0.96       450
           4       0.97      0.98      0.98       660
           5       0.95      0.95      0.95       630
           6       1.00      0.93      0.96       150
           7       0.99      0.96      0.98       450
           8       0.97      1.00      0.98       450
           9       0.97      1.00      0.98       480
          10       1.00      0.99      0.99       660
          11       0.96      0.99      0.97       420
          12       0.95      0.94      0.95       690
          13       1.00      1.00      1.00       720
          14       0.94      1.00      0.97       270
          15       0.98      1.00      0.99       210
          16       1.00      0.99      1.00       150
          17       1.00      0.86      0.93       360
          18       0.96      0.93      0.94       390
          19       0.98      1.00      0.99        60
          20       0.73      0.98      0.83        90
          21       0.92      0.68      0.78        90
          22       0.98      0.99      0.98       120
          23       0.98      0.98      0.98       150
          24       1.00      0.96      0.98        90
          25       0.96      0.98      0.97       480
          26       0.94      0.93      0.93       180
          27       0.90      0.45      0.60        60
```



```
          28       0.99      0.97      0.98       150
          29       0.70      0.99      0.82        90
          30       0.98      0.81      0.88       150
          31       0.96      0.99      0.97       270
          32       0.75      1.00      0.86        60
          33       0.92      1.00      0.95       210
          34       0.98      0.99      0.99       120
          35       0.99      0.99      0.99       390
          36       0.99      0.96      0.97       120
          37       1.00      1.00      1.00        60
          38       0.99      0.99      0.99       690
          39       0.99      0.99      0.99        90
          40       0.83      0.97      0.89        90
          41       1.00      0.80      0.89        60
          42       1.00      0.99      0.99        90

    accuracy                           0.97     12630
   macro avg       0.95      0.95      0.95     12630
weighted avg       0.97      0.97      0.97     12630
```

Model Accuracy: 95%

Conclusion :

Phase 4 was crucial in refining our models and ensuring their readiness for deployment. The rigorous evaluation and optimization processes enabled us to develop a robust and reliable traffic sign recognition system, addressing key challenges such as overfitting and class imbalance. Moving forward, this model will be integrated into our autonomous vehicle system, contributing to safer and more efficient navigation.

By adhering to best practices in model evaluation and optimization, we have laid a solid foundation for the continued advancement of our autonomous vehicle technology. As we proceed to subsequent phases, we remain committed to iterative improvement and innovation, ensuring our solutions meet the highest standards of performance and reliability.

LINKS:

Google colab:

https://colab.research.google.com/drive/1ByzXvGlofaDR4RgmKq2gQZ3ypwBSEQeN?usp=sharing

Github link:

https://github.com/Harihara04sudhan/naan-mudhalvan