Car multi-class image classification

Table of contents

Abstract

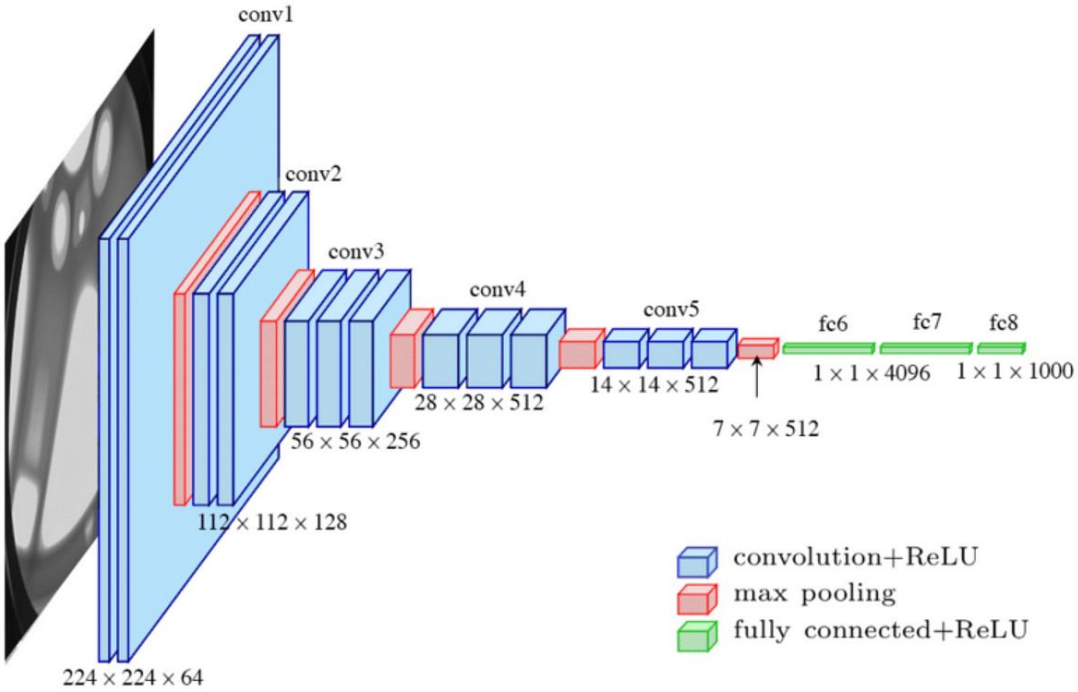
This research develops a convolutional neural network (CNN) to classify images from the Vehicle Type Image Dataset version 2 (Olarik, 2023), containing 4,793 vehicle photos across five classes. The model achieved a validation accuracy of 92.72%, surpassing the initial target of 70%. Explainable AI (XAI) methods, such as LIME, were employed to interpret the model's classification decisions, highlighting the features used for image recognition. The final model builds upon a basic CNN architecture, incorporating advanced techniques like L2 regularization, dynamic learning rates, and dropout to improve generalization and performance.

CNN Algorithm analysis

A convolutional neural network (CNN) is a type of artificial neural network that is inspired by the way the human brain processes visual information, CNN is primarily used for image recognition and processing. They achieve this with a series of specialised layers, each of which is designed to learn and extract distinct features of the target class, such as edges, textures and shapes. To build CNN, it will be comprehensive to understand the well-known CNN architectures and to develop on.

1. VGG-16

VGG-16 architecture was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It contains 16 layers, including 13 convolutional layers with 3 fully connected layers. It is known for its effectiveness as well as its simplicity, VGG-16 was designed to handle datasets as large as 14 million images across up to 1000 classes with 138 million parameters, it often leads to overfitting and significant computational demands, especially when used for a small dataset.



2. ResNet

   ResNet, or Residual Network, introduced the concept of residual learning, which skips connections to bypass certain layers in the network. These skip connections allow the model to retain important information and address the vanishing gradient problem, this innovation enables ResNet to scale up to 152 layers with suitable datasets without suffering from overfitting or performance degradation.

3. AlexNet

   AlexNet was known as a milestone in machine learning, it has changed the landscape with the use of ReLU activation to address the vanishing gradient problem and overlapping pooling for better feature learning. Even more, it is one of the first models that uses GPU to train, enabling efficient training on large datasets, it contains 8 layers, with 5 convolutional layers and 3 fully connected layers.

4. MobileNet

   MobileNet is designed for mobile devices, it aims to provide a high-performance, low-latency image classification on light devices, it uses depthwise separable convolutions and pointwise convolutions.

Initialise Model

AlexNet has inspired the initial baseline model. Because of its simplicity and effectiveness. with room for experimentation with different methods making it an ideal choice for a baseline while still maintaining reliable performance from the start. As seen in Figure One, the architecture of the model is similar to AlexNet, meanwhile, to improve its performance, several improvements have been made to the AlexNet-inspired model, the depth has been increased to 13 layers with batch normalization to improve stability and reduces its sensitivity to initialization and learning rates, addressing a key limitation of the original AlexNet.

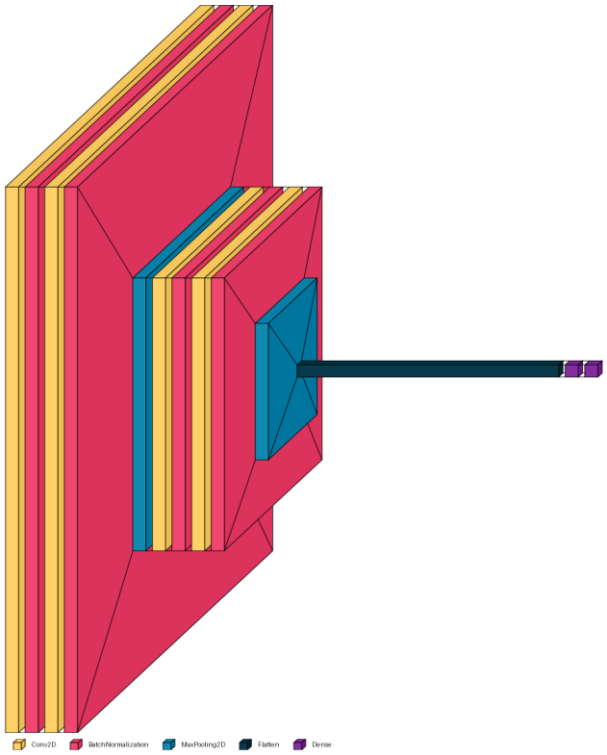


Figure 1: Base Model

As seen in Figure One, the architecture of the model is similar to AlexNet, meanwhile, to improve its performance, several improvements have been made to the AlexNet-inspired model, the depth has been increased to 13 layers with batch normalization to improve stability and reduces its sensitivity to initialization and learning rates, addressing a key limitation of the original AlexNet.

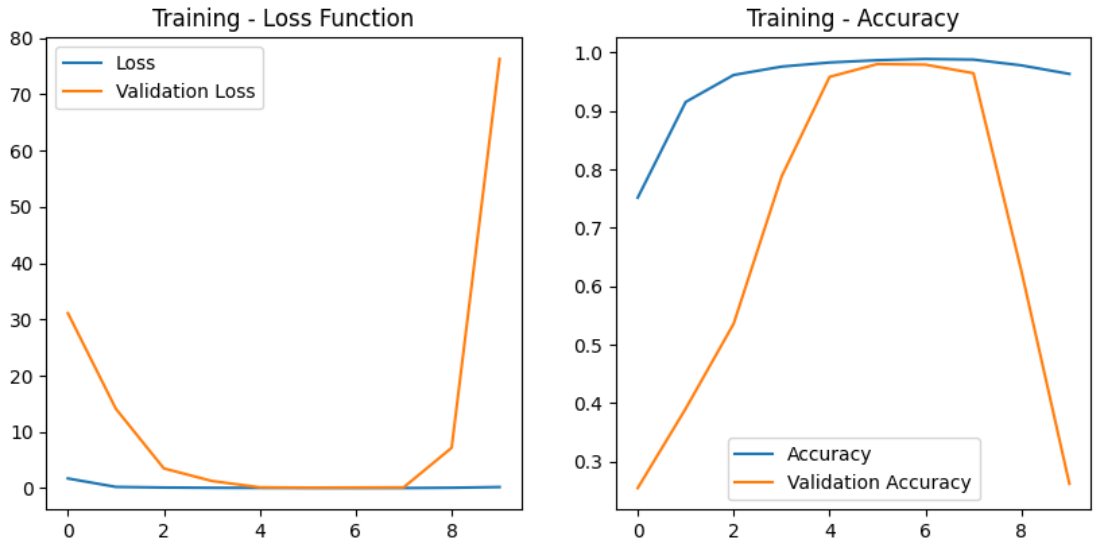


Figure 2: Base Model Results

Model	Epochs	Training Accuracy	Validation Accuracy
Base Model	1	75.16%	25.50%
	5	98.28%	95.82%
	10	96.32%	26.23%

Table 1: Base Model Results

Both Figure 2 and Table 1 show the results of the initialised base model. It has been able to achieve the highest training accuracy, 98.28%. However, as Figure 2 suggests, the model is unstable on unprocessed data and has the potential for data leakage, as it is “too good to be true” for a first model therefore, data cleaning will proceed.

In the selected dataset, duplicates, miss-classing and data imbalance have been found across 5 classes and are required to be cleaned for reliable results for later experiments. The dataset contained 4,793 photos, and after cleaning, 1,764 photos remained.

Data Processing

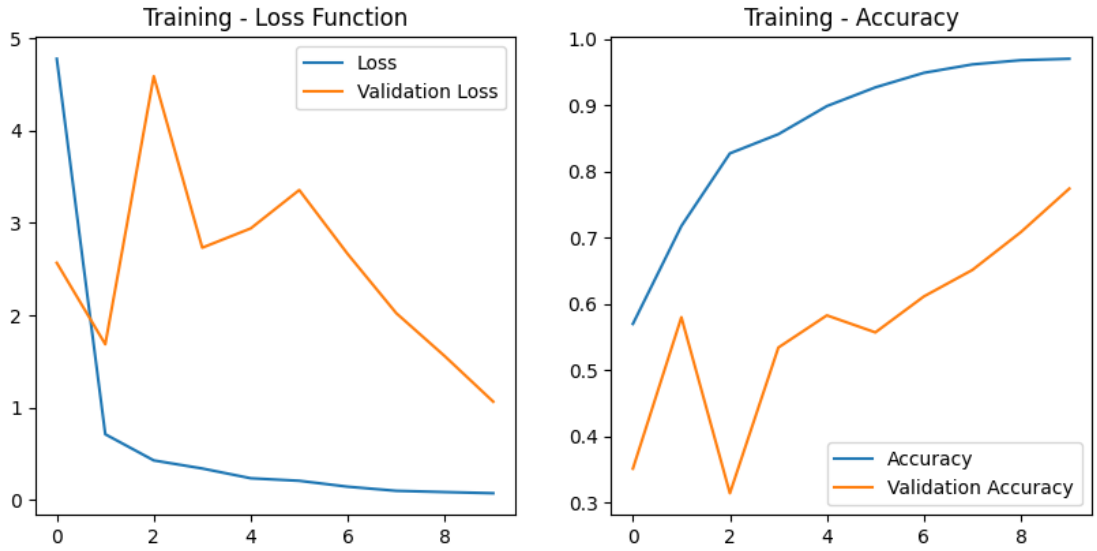


Figure 3: Base Model (Data Cleaned)

Model	Epochs	Training Accuracy	Validation Accuracy
Base Model (cleaned)	1	57.16%	35.14%
	5	89.89%	58.29%
	10	97.03%	77.42%

Table 2: Base Model (Data Cleaned)

The results after data cleaning have shown improvement in both stability and practical results, for that reason, the cleaned data will be used for further experiments. As discussed, data imbalances remain so data balance and image augmentation will be tested.

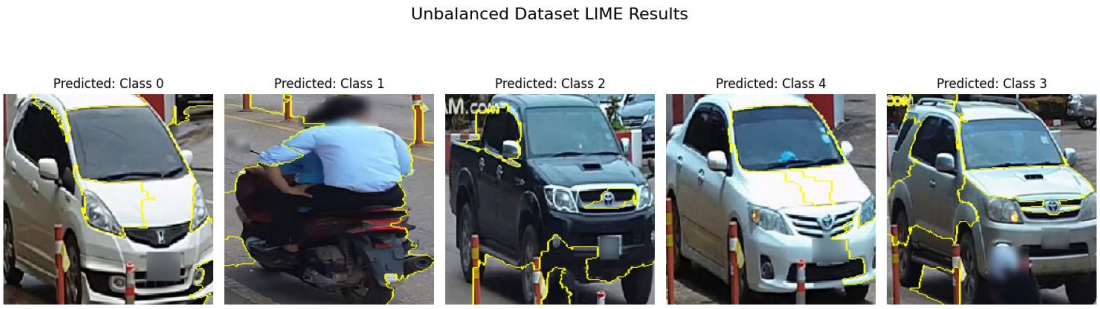


Figure 4: Base Model (Data Cleaned) LIME Results

Figure 4 suggests the model can key features as a basic model and cleaned data, this helps build the foundation for further improvement.

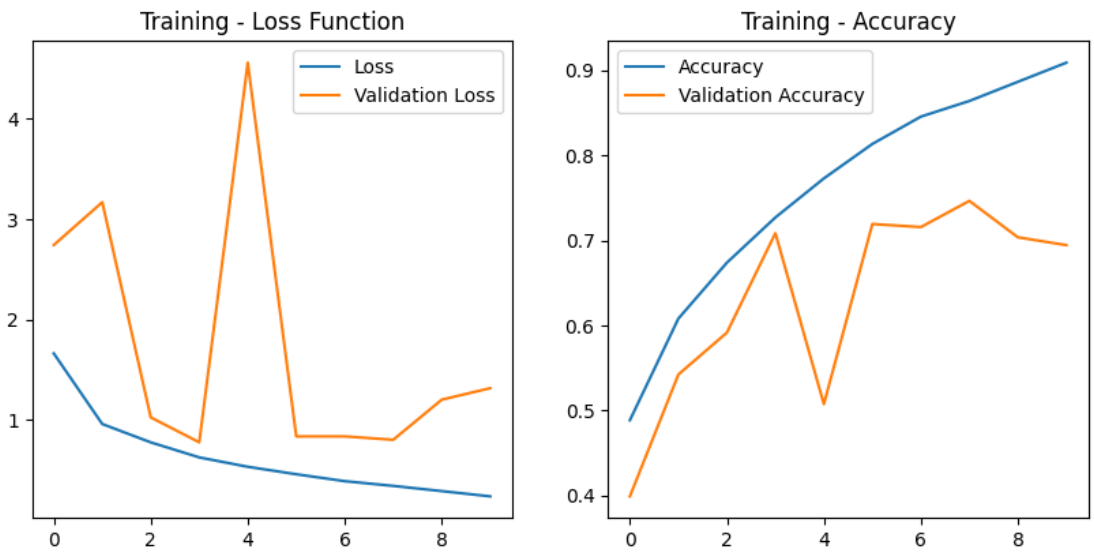


Figure 5: 2500 Photos Per Class

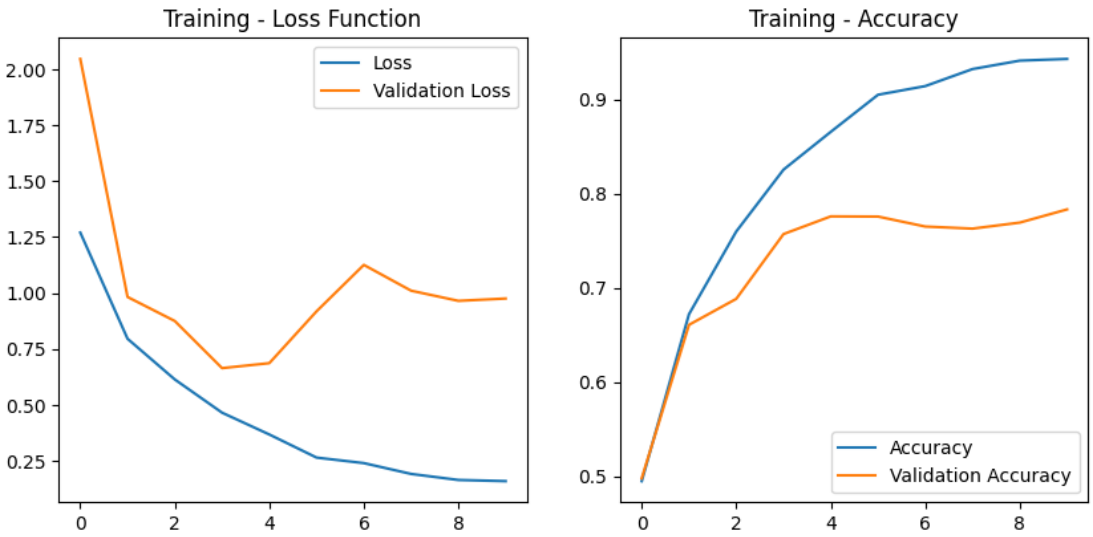


Figure 6: 5000 Photos Per Class

Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current Result	**10**	**97.03%**	**77.42%**
2500 Photo (Per Class)	1	48.87%	39.92%
	5	77.28%	50.76%
	10	**90.88%**	**69.44%**
5000 Photo (Per Class)	1	49.47%	49.78%
	5	86.58%	77.60%
	10	**94.33%**	**78.34%**

Table 2: Base Model (Data Balanced)

The dataset has been scaled up to 2,500 and 5,000 photos per class, a total of 12,500 and 25,000 to afford duplicates that create data leakage. Image augmentation has been implemented into both datasets, and 5000 photos per class have shown more consistent results compared to others, 5000 photos per will be used.

Drop Out Testing

Drop-out methodology is a common approach to reduce overfitting in CNN, by selecting a % of neurons to force the model to be more generalizable.

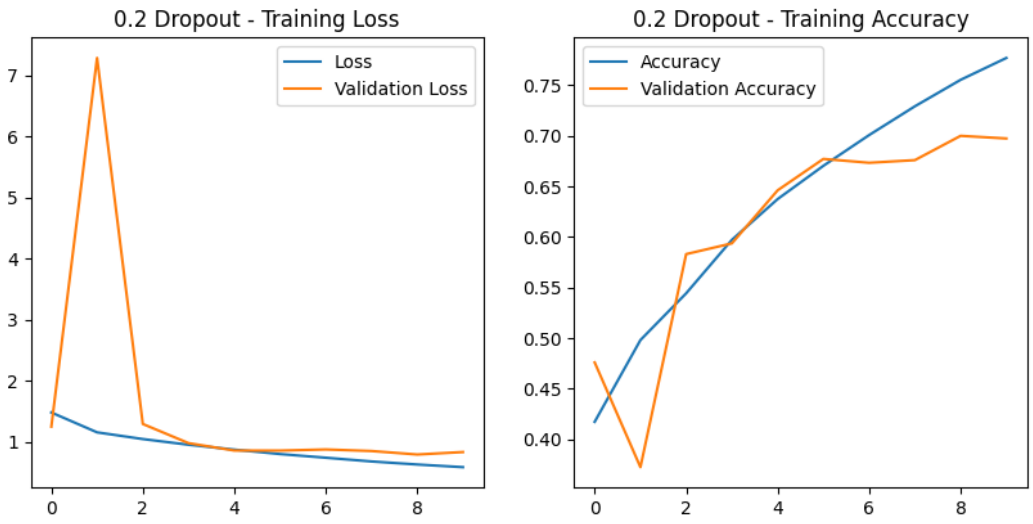


Figure 7: 20% Dropout Rate

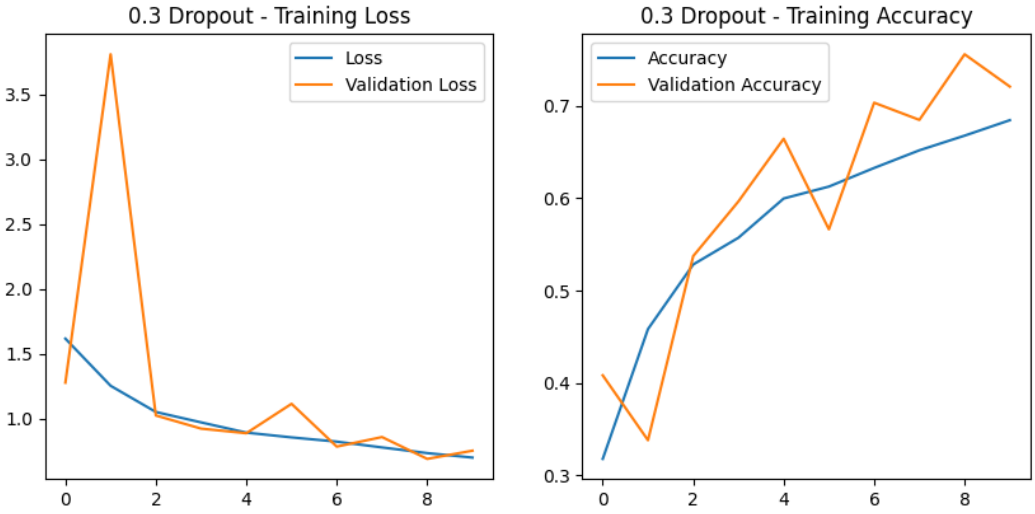


Figure 8: 30% Dropout Rate

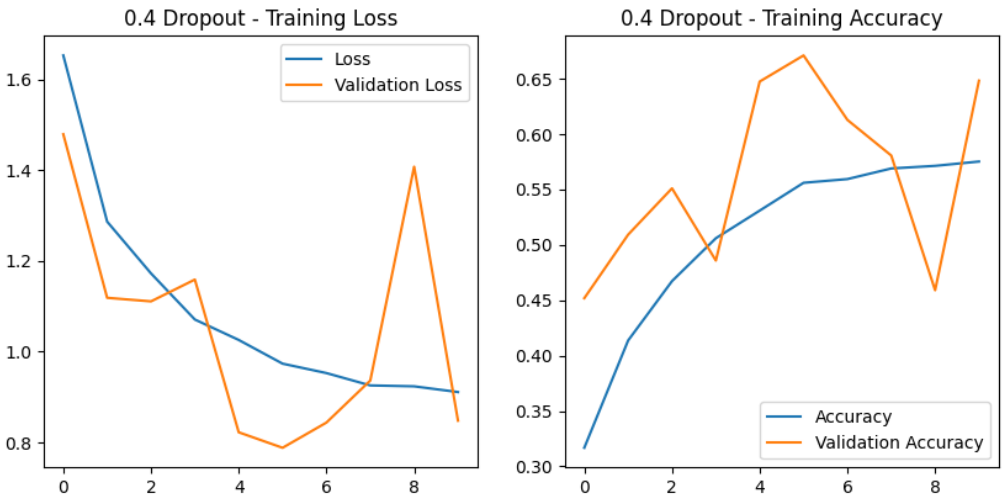


Figure 9: 40% Dropout Rate

Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current Result	**10**	**94.33%**	**78.34%**
20% Dropout Rate	1	41.74%	47.60%
	5	63.74%	64.60%
	10	**77.67%**	**69.72%**
30% Dropout Rate	1	31.80%	40.84%
	5	59.97%	66.44%
	10	**68.45%**	**72.08%**
40% Dropout Rate	1	31.67%	45.18%
	5	53.09%	64.74%
	10	**57.52%**	**64.82%**

Table 3: Dropout Rate Model Accuracy

The implementation of dropout has shown a positive result in terms of stability, with the cost of lower accuracy, as shown in Figure 7 of a 20% dropout rate. As expected, the higher the dropout rates, the more accuracy drops and the model loses the ability to learn; therefore, a 20% dropout rate will be used.

Extra layer

As the execution of the dropout rate has decreased the model accuracy, implementing an extra layer such as the convolutional and fully connected layer will be a good practice to strengthen the model's accuracy.

Figure 10: 3 Convolutional Block Layers

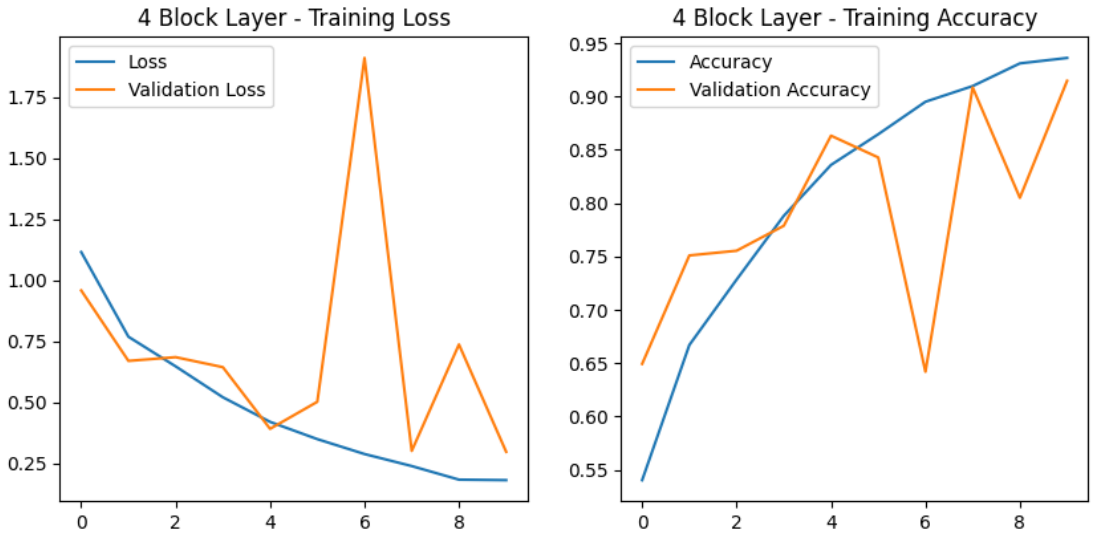


Figure 11: 4 Convolutional Block Layers

Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**77.67%**	**69.72%**
3 Block Model	1	49.52%	47.10%
	5	75.34%	78.68%
	10	**87.84%**	**86.74%**
4 Block Model	1	54.04%	64.94%
	5	83 58%	86.32%
	10	**93.60%**	**91.46%**

Table 4: Extra Convolutional Layer Model Accuracy

Accuracy has been largely improved by the extra convolutional layers, however, while both models show varying levels of instability in their training and validation processes, the 4-block model achieves consistently higher accuracy and maintains a close gap between training and validation accuracy. Therefore, 4 block model will be used and add fully connected layers to stabilize it.

Extra Dense Layer

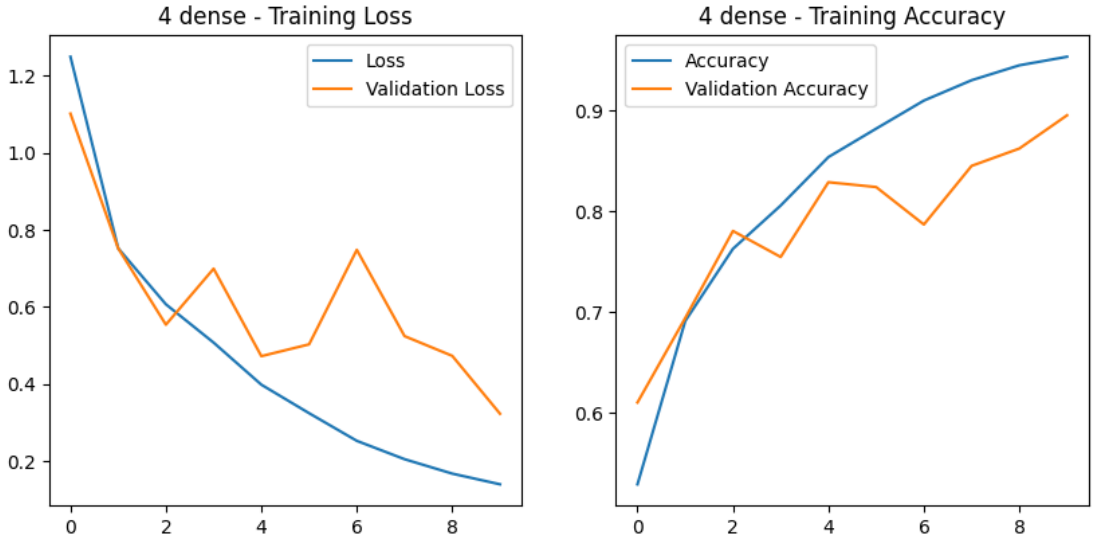


Figure 12: 4 Dense Layers

Figure 13: 5 Dense Layers

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**93.60%**	**91.46%**
4 Block Model	1	52.88%	61.00%
	5	85.36%	82.86%
	10	**95.31%**	**89.50%**
5 Block Model	1	46.00%	64.94%
	5	80 98%	81.60%
	10	**94.20%**	**82.60%**

Table 5: Extra Dense Layer Model Accuracy

The new dense layer model has been able to smooth out the result while achieving a higher accuracy showing it is stable again with a huge improvement with 5 dense layers, it will be used for further development.

For further development, Local Interpretable Model-agnostic Explanations (LIME) will be implemented, because the model has been able to achieve impressive accuracy based on it, LIME can help development by better understanding of its model decision-making.

Batch Normalization

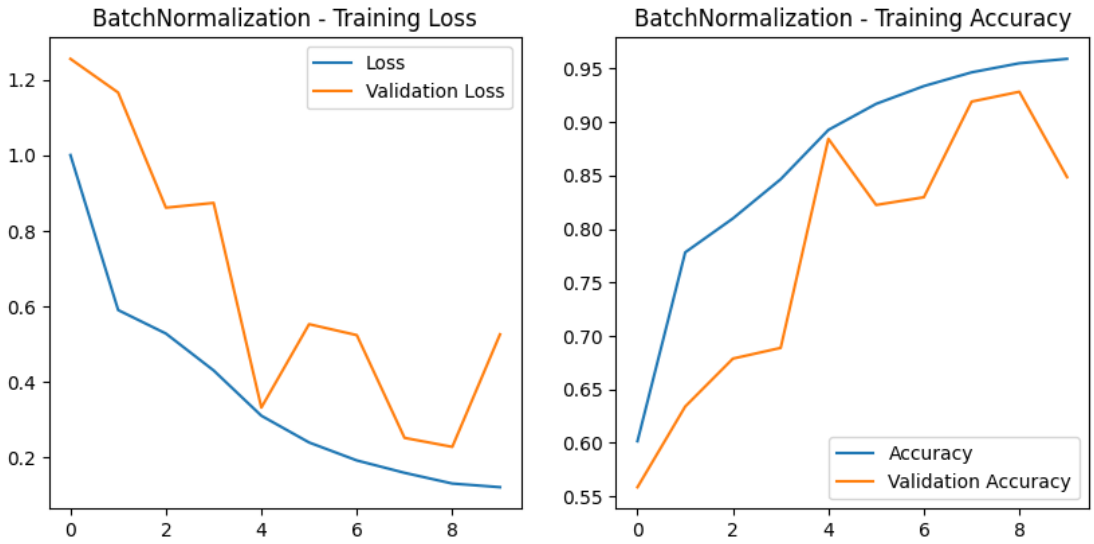


Figure 14: Batch Normalization

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**94.20%**	**82.60%**
Batch Normalization	1	60.16%	55.86%
	5	89.25%	88.42%
	10	**95.89%**	**84.84%**

Table 5: Extra Batch Normalization Between Dense Layer Model Accuracy

Batch Normalization has been added between each fully connected layer the performance has shown a slight improvement.

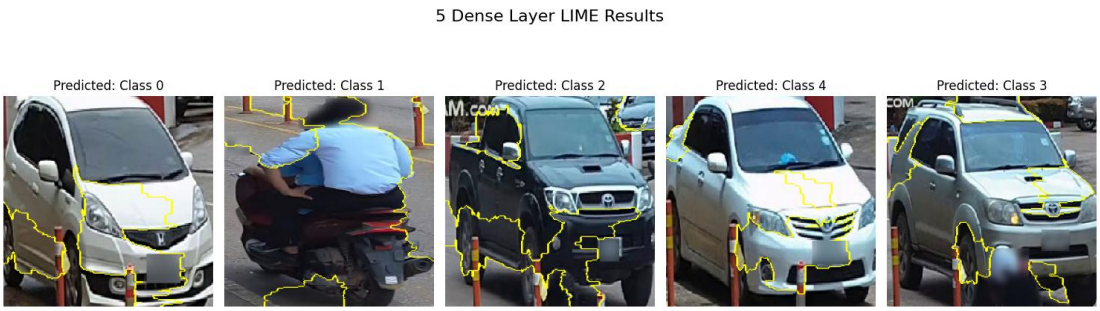


Figure 15: 5 Dense Layers LIME Results

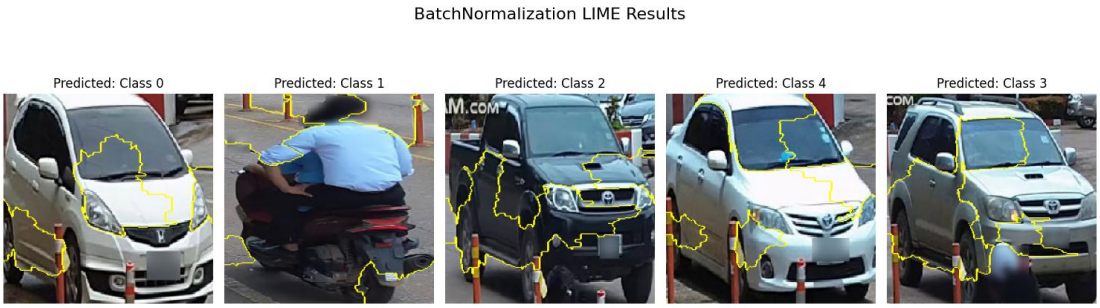


Figure 16: Batch Normalization Lime Results

As Figure 16 shows compares to Figure 15, Batch Normalization can help the model learn more key features of different classes, therefore It will be used for further development.

Regularization

Regularization also is a common technique used to improve the generalization of models by reducing overfitting and further stabilizing it.

Figure 17: 0.01 L2 Rate

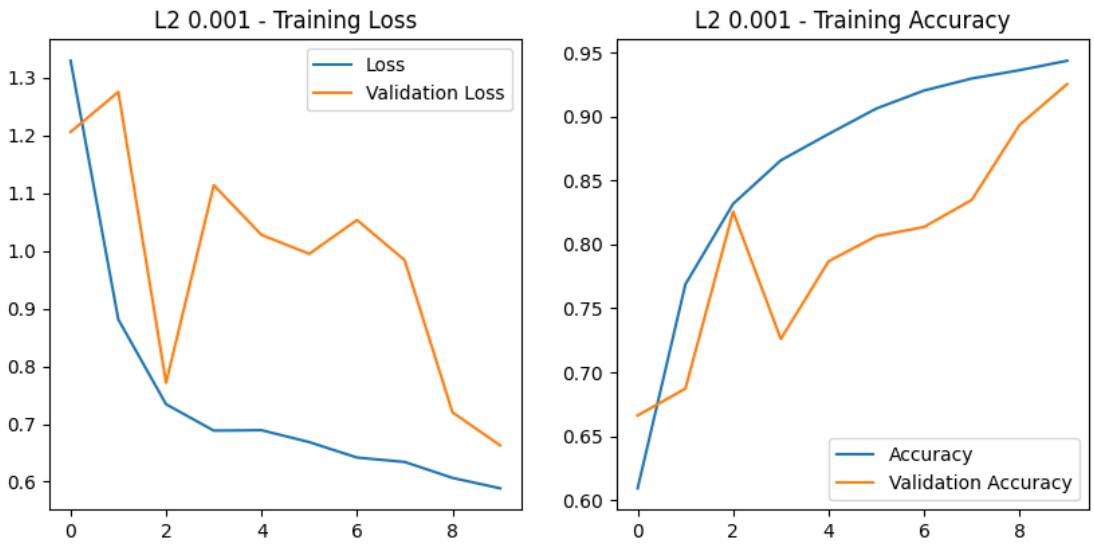


Figure 18: 0.001 L2 Rate

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**95.89%**	**84.84%**
0.01 L2 Rate	1	45.95%	60.94%
	5	75.60%	79.32%
	10	**91.91%**	**86.14%**
0.001 L2 Rate	1	60.92%	66.62%
	5	88 63%	78.68%
	10	**94.37%**	**92.54%**

Table 5: L2 Model Accuracy

As expected, the L2 model has been able to further stabilize the model without overly suppressing the accuracy of the model, therefore 0.01 with be used for further development.

Learning Rate

The learning rate will determine the step size at which the model updates its weight during training.

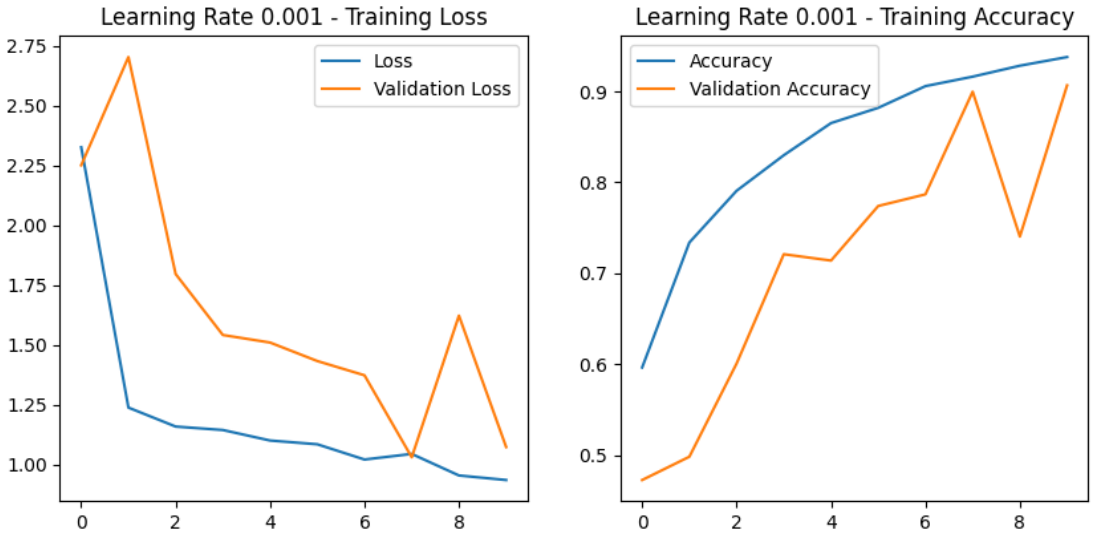


Figure 19: Learning Rate 0.001

Figure 20: Learning Rate 0.0001

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**91.91%**	**86.14%**
0.001 Learning Rate	1	59.65%	47.28%
	5	86.51%	71.40%
	10	**93.77%**	**90.66%**
0.0001 Learning Rate	1	56.16%	62.94%
	5	88.49%	86.84%
	10	**95.16%**	**84.26%**

Table 6: Learning Rate Model Accuracy

0.0001 Learning Rate has shown a smoother curve and better training accuracy of 95,16%, therefore for further development 0.0001 Learning Rate will be used for further development

Dynamic Learning Rate

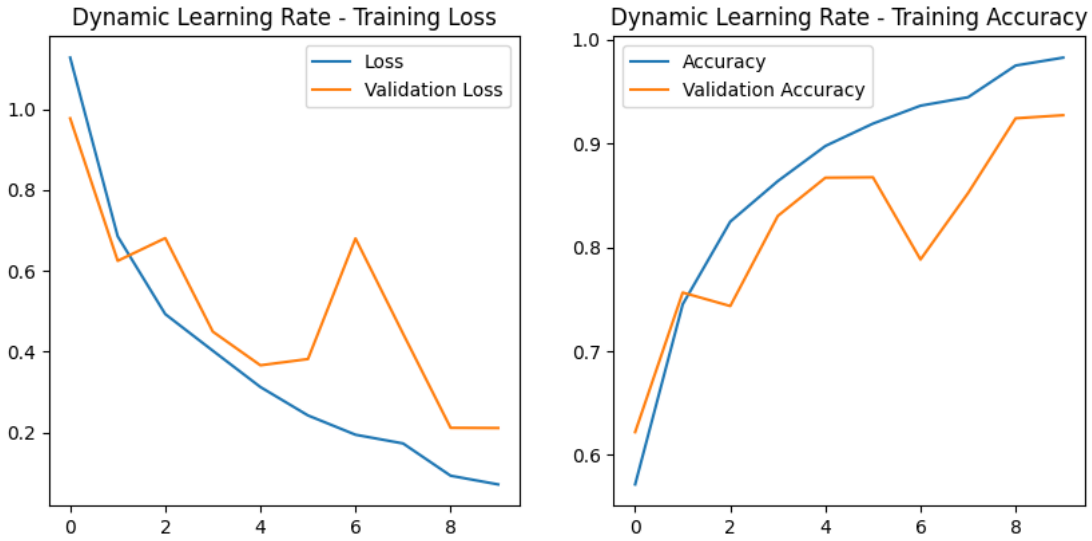


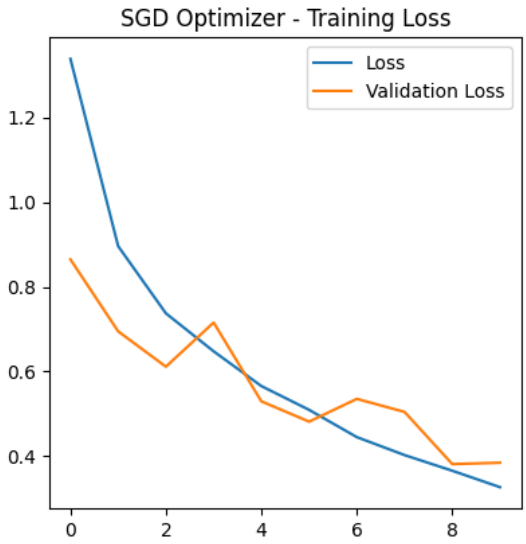
Figure 21: Dynamic Learning Rate 0.0001

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**95.16%**	**84.26%**
0.0001 Dynamic Learning Rate	1	57.16%	62.20%
	5	89.75%	86.70%
	10	**98.26%**	**92.72%**

Table 6: Dynamic Learning Rate Model Accuracy

A dynamic learning rate decreases over time, enabling faster initial convergence and finer weight updates as training progresses. It achieved higher validation accuracy 92.72% by preventing overfitting, improving generalization, and ensuring stable convergence, making it ideal for further development.

Optimizer



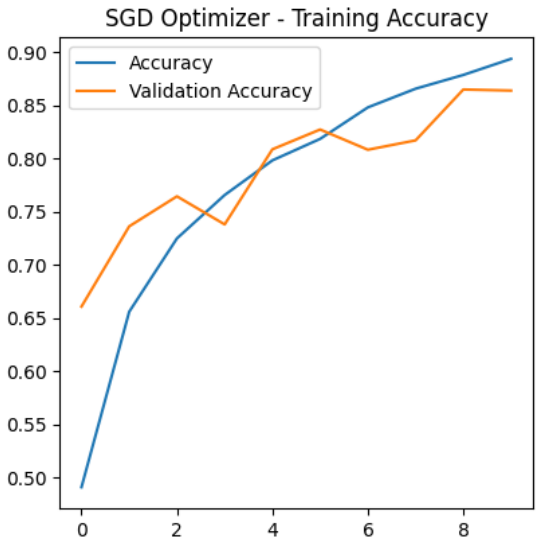
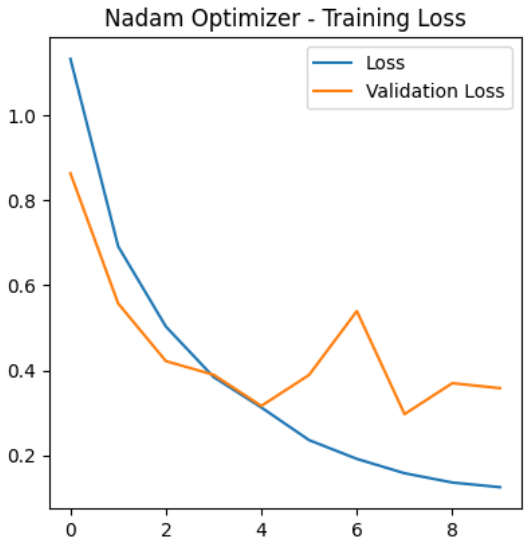


Figure 22: SGD Optimizer



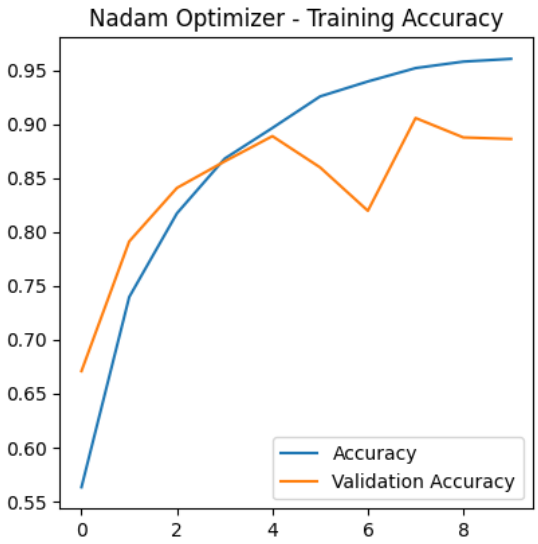


Figure 23: Nadam Optimizer

Figure 24: AdamW Optimizer

5,000 Balanced Data	Epochs	Training Accuracy	Validation Accuracy
Current result	**10**	**98.26%**	**92.72%**
SGD	1	49.13%	66.10%
	5	79.84%	80.88%
	10	**89.38%**	**86.40%**
Nadm	1	56.34%	67.10%
	5	89.64%	88.88%
	10	**96.04%**	**88.62%**
AdamW	1	57.14%	54.53%
	5	90.74%	87.12%
	10	**96.09%**	**86.88%**

Table 7: Different Optimizer Model Accuracy

Different optimizer method will use different method to learn, however for our dataset none of the result can match with the current result therefore, it will not be used in the final model.

Final Model

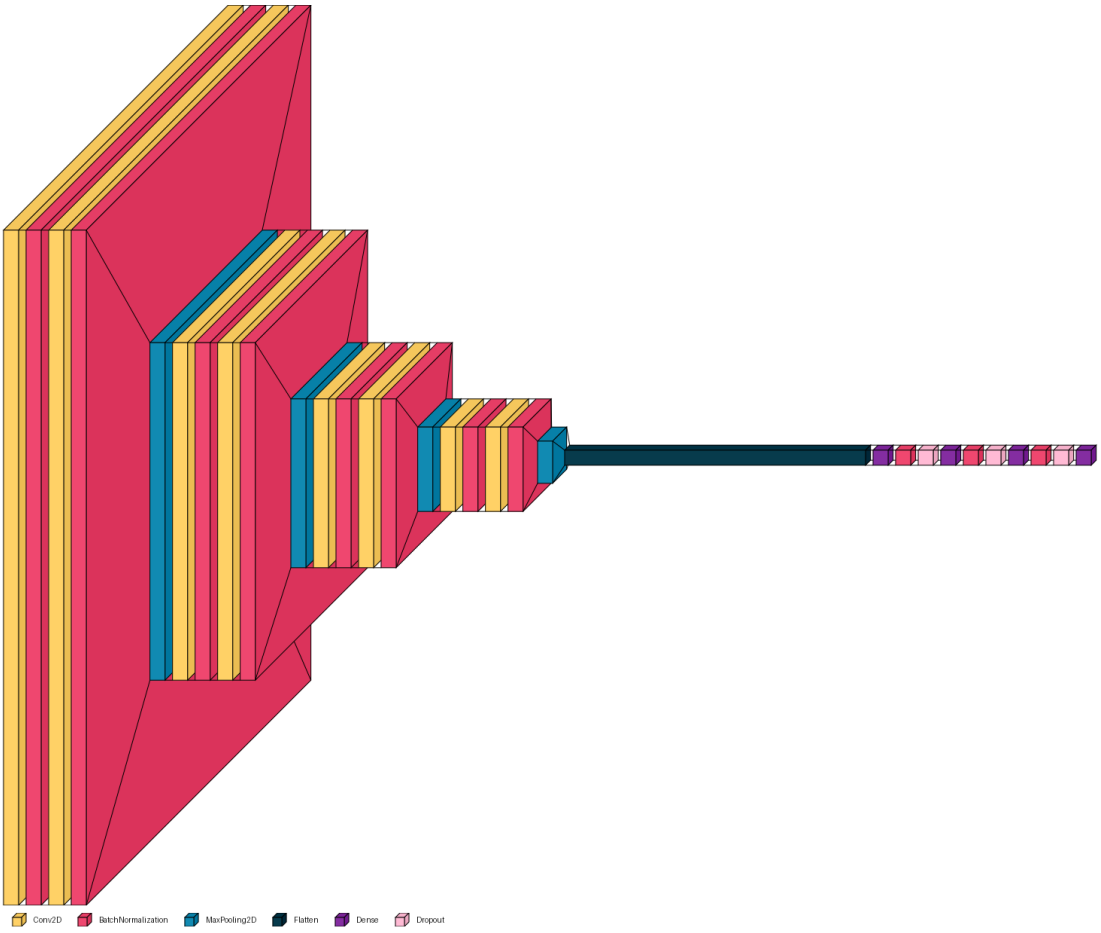


Figure 25: Final Model

The final model contains four convolutional blocks and 5 fully connected layers, with a 20% dropout rate, L2 0.001, and dynamic learning rate with Adam optimizer.

Conclusion and future work

The final model successfully achieves 98.26% Training Accuracy and 92.72% Validation Accuracy, although it has successfully achieved the target of 70%, the result can be unreliable due to the dataset having up to 90% of the data created by the dataset itself, to develop a more realistic and reliable model a higher quality of dataset are needed and use different XAI method to better help to understand the behaver of each model development.

Reference

Anon., 2025a. *Image segmentation* [online]. TensorFlow. Available from: https://www.tensorflow.org/tutorials/images/segmentation [Accessed 17 Jan 2025].

Anon., 2025b. *Semantic segmentation with KerasHub* [online]. Keras.io. Available from: https://keras.io/keras_hub/guides/semantic_segmentation_deeplab_v3/ [Accessed 17 Jan 2025].

Anon., 2025c. *Image classification with KerasHub* [online]. Keras.io. Available from: https://keras.io/keras_hub/guides/classification_with_keras_hub/ [Accessed 17 Jan 2025].

Anon., 2025d. *MobileNet - Hugging Face Community Computer Vision Course* [online]. Huggingface.co. Available from: https://huggingface.co/learn/computer-vision-course/unit2/cnns/mobilenet [Accessed 17 Jan 2025].

Bangar, S., 2022. *AlexNet architecture explained - siddhesh bangar* [online]. Medium. Available from: https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5 [Accessed 17 Jan 2025].

Boesch, G., 2023. *Deep Residual networks (ResNet, ResNet-50) - 2024 guide* [online]. viso.ai. Available from: https://viso.ai/deep-learning/resnet-residual-neural-network/ [Accessed 17 Jan 2025].

Great Learning, 2021. *Everything you need to know about VGG16 - Great Learning* [online]. Medium. Available from: https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918 [Accessed 17 Jan 2025].

Rahman, M. A., Tanim, A. S., Sanjid, I., Fahim, P., G., M. S. and Shawon, M. T. R., 2023. Evaluating the reliability of CNN models on classifying traffic and road signs using LIME. *arXiv [cs.CV]* [online]. Available from: http://arxiv.org/abs/2309.05747.

Bloice, M. D., 2017. Augmentor: Image augmentation library in Python for machine learning. [online]. Available from: https://github.com/mdbloice/Augmentor [Accessed 17 Jan 2025].