ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING - 6CS012

COURSEWORK 01:

“IMAGE CLASSIFICATION OF CAR BRAND LOGOS USING CONVOLUTIONAL NEURAL NETWORK”

Student Name: Saroj Devkota
University ID: 2329255
Group: L6CG6
Module Leader: Mr. Siman Giri
Module Tutor: Mr. Shiv Kumar Yadav
Cohort: 09
Submission Date: 5/15/2025

TABLE OF CONTENTS

ABSTRACT

Accurate identification of car brand logos is a significant aspect of the intelligent transportation system, self-driving, and monitoring traffic. However, this is compromised by the low quality of images, varying lights, different logo designs as well as clustered backgrounds. This work examines the use of Convolutional Neural Networks (CNN) for multiclass car logo classification with a curated dataset of 2913 pictures belonging to eight different automobile brands. There are three models developed and estimated. We are going to consider a baseline CNN, a deeper CNN that is also enriched with regularization techniques and a transfer learning model with the pre-trained VGG16 architecture. Baseline CNN had 62.5% accuracy whereas the deeper CNN had 75% accuracy which was mostly achieved due to the use of dropout, batch normalization and L2 regularization. The transfer learning model was superior to both of them, achieving the accuracy of 87% and showing a high stability and generalization. Extensive evaluation based on precision, recall, and F1-score measures validated superiority of deeper architectures and advantage of making use of pre-trained models. These results support the applicability of deep learning, particularly transfer learning, in overcoming visual changes and data limitations in the case of logo recognition work.

1. INTRODUCTION

Practical applications of good recognition of car brand logos are widespread, particularly in autonomous car navigation, smart traffic, and automated surveillance. However, this exercise presents great challenges because of the different logo designs, poor image resolutions, and diverse lighting, as well as complex backgrounds, which have a tendency of distorting or interfering with logos. In order to overcome these challenges, this project investigates the application of Convolutional Neural Networks (CNN) to label car brand logos from images. CNNs are very common in image classification because CNNs are able to identify and learn hierarchies of space in image data via layer convolution and pooling operations. This ability makes them best suited for visual recognition projects that will involve complex and variable patterns, like what is seen in car logos. According to the coursework, the goal is to compare models classification accuracy and loss, as well as their robustness, after having trained and tested them on a publicly available dataset of logos representing car brands. This comparative analysis will reveal how diverse model strategies are effective for practical car logo recognition exercises.

The work starts with the creation of a basic baseline CNN model that serves as the foundation for further experimentation. Then a deeper CNN architecture is introduced to improve feature learning and overall performance. To explore the effect of optimization techniques both SGD and Adam optimizers are considered. The research also includes a transfer learning method that fine-tunes a pre-trained VGG16 model to make it fit for the fruit classification task. The general goal is to create an efficient CNN-based classifier, improve its error rate by architectural and training modifications, and find the most favourable configuration of the classifier in terms of both effectiveness and efficiency.

Link to GitHub: https://github.com/Saroj058/6CS012

2. DATASET

The publicly available Kaggle website houses the dataset used in this coursework.

Source: https://www.kaggle.com/datasets/volkandl/car-brand-logos , and was processed by Volkan Özdemir. It contains a total of 2,913 images categorized into 8 distinct car brand classes. The distribution of images in the training set includes 302 images of Hyundai, 301 images of Lexus, 317 images of Mazda, 342 images of Mercedes, 301 images of Opel, 314 images of

Skoda, 306 images of Toyota, and 330 images of Volkswagen. Each class in the test set consists of 50 images, offering a balanced sample for evaluating model performance. The image resolutions vary widely, with the highest resolution reaching 6000×4000 pixels and the lowest being 64×64 pixels.

To ensure the dataset was clean and ready for model training, several preprocessing steps were implemented. First, all images were scanned to detect and remove corrupted files. Data augmentation was applied to the training images to introduce variability and reduce the risk of overfitting. Additionally, all images were resized to a uniform resolution of 224×224 pixels to standardize input dimensions across the dataset. Lastly, pixel values were normalized to a range of 0 to 1 to accelerate and stabilize the training process of the neural networks.

3. METHODOLOGY

3.1 BASIC CNN ARCHITECTURE

The baseline model is a relatively easy to implement Convolutional Neural Network (CNN) for a baseline benchmarking for car brand logo classification. The model has the following:

- Three convolutional layers, and each followed by a MaxPooling2D to decrease spatial dimensions.
- A Flatten layer which will transform feature maps into 1D feature vector.
- Three (time and memory saving dense) layers minus the final output layer.
- No application of drop out, batch normalization or other regularization techniques in order to be able to estimate raw learning capability of the model.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 223, 223, 16)	448
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
flatten (Flatten)	(None, 43264)	0
dense (Dense)	(None, 64)	2,768,960
dense_1 (Dense)	(None, 12)	2,080
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 8)	136

Total params: 2,795,288 (10.66 MB)
Trainable params: 2,795,288 (10.66 MB)
Non-trainable params: 0 (0.00 B)

Figure 1 Summary of Baseline Model

The base line model for car brand logo classification is a simple Convolutional Neural Network (CNN) that has three convolutional layers with and max pooling layers in between to decrease spatial dimensions without loss of key features. Feature maps are flattened and passed through three fully connected layers after the final convolutional layer and then output. This model is non-regularized and does not use standard regularization methods such as dropout or batch normalization, making its baseline performance very clear so it can be compared against more advanced models. It uses Adam optimizer, with learning rate being 0.001, and sparse categorical cross entropy to use for multi-class classification purposes and is trained using batch size as 16, for 40 epochs, with the help of early stopping and model checkpoint call backs to The model has about 27.9 million trainable parameter and is used as a baseline for comparison with more complicated models.

3.2 DEEPER CNN MODEL

The Deeper CNN model expands the baseline architecture by enlarging the network depth and introducing more regularization strategies in order to improve generalization and prevent overfitting. In particular, the number of convolutional layers has been doubled and number of filters in the deepest layer is increased from 64 to 256. Regularization techniques have also been implemented in the model along the way to stabilize and improve the training performance, including: L2 weight regularization, Batch Normalization and Dropout.

Two optimizers were used to train the model, (Adam) and (SGD) in order to observe the results both in the area of convergence behaviour and lowest losses. The comparison was made fair, as both training runs invoked the same loss function (categorical_crossentropy) and metric for evaluation (accuracy).

Other than the usual training callbacks, finer model also uses ReduceLROnPlateau which dynamically decreases learning rate when the val_loss metric plateaus, so as the model escapes potential plateaus during optimization.

Summarized is the deeper model architecture and learnable parameters below:

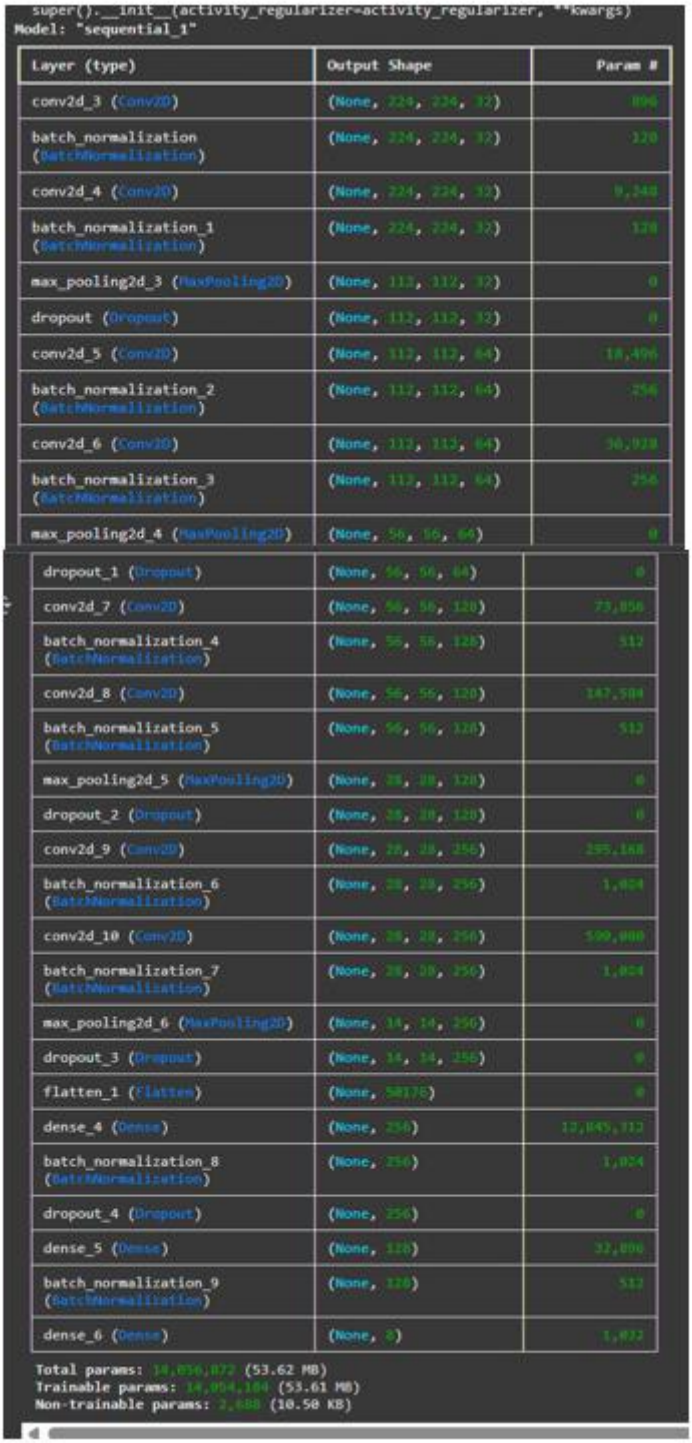

Figure 2 Summary of Deeper CNN

3.3 OPTIMIZER COMPARISON

In the model trained on SGD (Stochastic Gradient Descent) the optimization method the training accuracy improved continuously up from 15% up to over 75% during training. However, the validation accuracy opened at around 12% and only got up to approximately

60%. This pronounced difference between both training and validation accuracy is the indication of overfitting tendency as the model learns the training data well but is not generalized in the detection of unseen data. Further, whereas training loss steadily and without jaggedness declined, validation loss varied strongly in oscillation between epochs. Even with the improvement from about 5.5 to around 4.5, the instability of the validation loss also signals problems of generalization and robustness of the model with the SGD configuration.

Conversely, a model trained with the implementation of the Adam optimizer showed substantially improved and more balanced performance. Both training and validation accuracies improved above 80% and both performances followed each other closely as the training procedure was done. This tight coupling implies that the model was learning optimally without overfitting on the training and was also capable of being effective. Besides, the training and the validation losses decreased below 2 which points to faster convergence as well as stable optimization pathway. Even considering some small variations in the learning curves, Adam optimizer generally led to a less variable 20 performance and better results, as opposed to SGD. Finally, whereas the SGD-optimized model demonstrated some learning ability, it was overfitting, and it was unstable in validation performance. Adam-optimized model in contrast, however, resulted in a faster convergence, a higher degree of stability along with generalization making it more viable option for this image classification task concern.

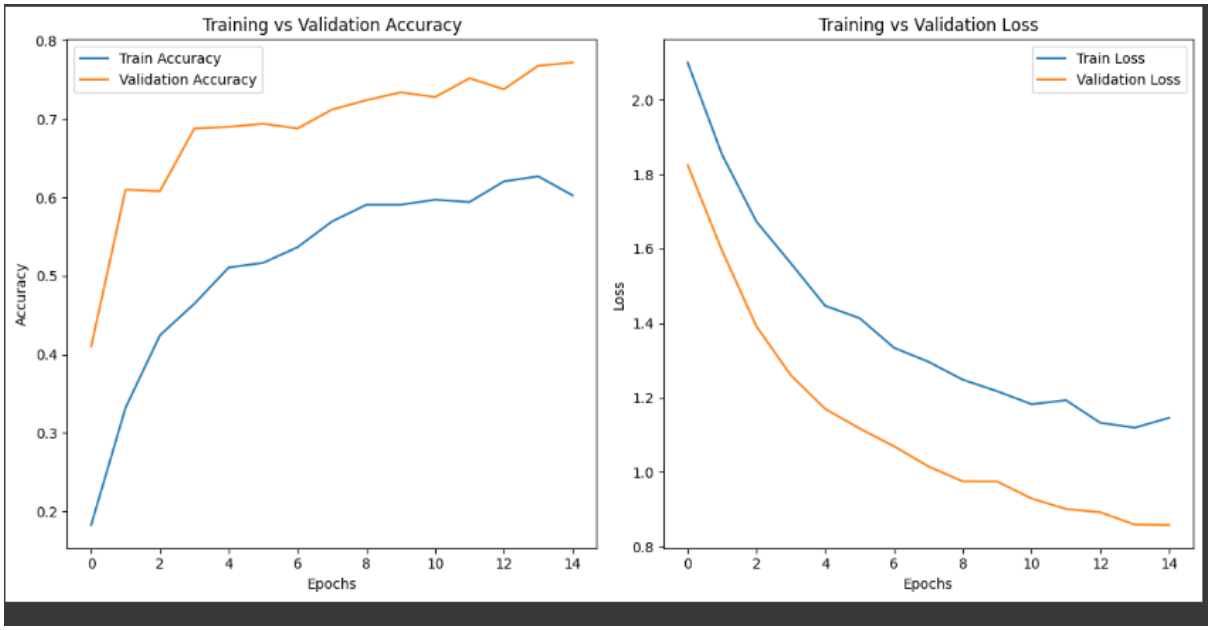

Figure 3 Graph of SGD and Adam

Comparing the classifications reports for the models, then a performance difference is apparent between the precision, recall, F1-score, and overall accuracy over the eight car brand classes. The SGD optimizer trained model managed to achieve ~61% overall accurate, macro-averaged precision, recall, and F1-scores were between 0.61-0.65. As some classes did quite well, others had a problem of low recall. For example, the model correctly predicted Toyota with high precision (0.91) but failed to recall large portion of actual Toyota samples (0,40). For such brands as Hyundai, Mazda, Skoda and Opel, the F1-scores are more balanced, around 0.67, however, recall is still a widespread weakness, causing misclassifications.

However, the Adam-optimized model demonstrated far better, and more consistent results. It achieved a final accuracy of 80% and showed good performance for almost all classes. For Toyota, another high precision (0.94) and better recall (0.60) was obtained in comparison with the SGD model. Brands such as Hyundai had a good F1- score at 0.76 and the brands like Mercedes and Skoda performed good as well. The classification report emphasizes that, not only did Adam model predict better but also had fewer misses through good generalization and robustness in categories.

	precision	recall	f1-score	support
hyundai	0.87	0.82	0.85	50
lexus	0.82	0.94	0.88	50
mazda	0.94	0.90	0.92	50
mercedes	0.90	0.90	0.90	50
opel	0.84	0.76	0.80	50
skoda	0.75	0.90	0.82	50
toyota	0.89	0.84	0.87	50
volkswagen	0.89	0.82	0.85	50
accuracy			0.86	400
macro avg	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	400

	precision	recall	f1-score	support
hyundai	0.67	0.76	0.71	50
lexus	0.61	0.76	0.68	50
mazda	0.91	0.80	0.85	50
mercedes	0.71	0.78	0.74	50
opel	0.64	0.72	0.68	50
skoda	0.81	0.76	0.78	50
toyota	0.93	0.54	0.68	50
volkswagen	0.78	0.78	0.78	50
accuracy			0.74	400
macro avg	0.76	0.74	0.74	400
weighted avg	0.76	0.74	0.74	400

Figure 4 Classification Report of SGD and Adam

In conclusion, though SGD model demonstrated partial competency, particularly in the sphere of precision of certain classes, Adam model demonstrated better general performances, better balance in prediction and better reliability on the identification of different car brands.

3.4 TRANSFER LEARNING WITH VGG16

For the third model, the project utilizes a pre-trained model known as VGG16 which was trained in the ImageNet competition. All the layers of the model were frozen, to prevent the architecture from updating weight and the original output layer was removed. Additional layers like GlobalAveragePooling, Dense and Dropout layers are added on top of base model layer

with the project classification layer of 8 classes. It utilizes the same configuration as deeper model with Adam in, but the learning rate is tweaked to 1e-5. For training, the last 20 layers are unfrozen and fine-tuned so that the model would capture higher level features by updating their weights and learning features. Below is the fine- tuning model summary:

Model: "functional_3"

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout_12 (Dropout)	(None, 512)	0
dense_10 (Dense)	(None, 128)	65,664
dropout_13 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 8)	1,032

Total params: 14,781,384 (56.39 MB)
Trainable params: 66,696 (260.53 KB)
Non-trainable params: 14,714,688 (56.13 MB)

Figure 5 Summary of Fine Tuning

Compared to the other two models, transfer learning model's validation accuracy starts high, above 0.85, remains stable throughout training and eventually reaches above 0.90 and the validation loss also declines from a very low number of 0.5 to below 0.4. The overall accuracy increases of the model to 0.86 which is highest among all three models.

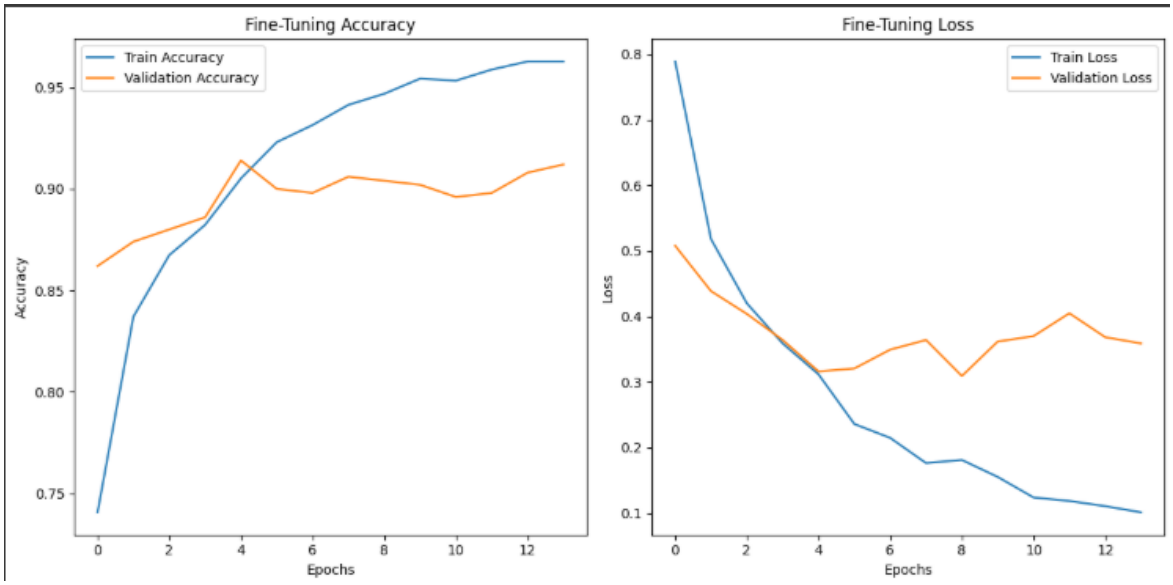

Figure 6 Transfer Learning Training and Validation Curve

The classification report also shows better performance over the precision, recall and f1 score compared to the rest of the two models.

	precision	recall	f1-score	support
hyundai	0.87	0.82	0.85	50
lexus	0.82	0.94	0.88	50
mazda	0.94	0.90	0.92	50
mercedes	0.90	0.90	0.90	50
opel	0.84	0.76	0.80	50
skoda	0.75	0.90	0.82	50
toyota	0.89	0.84	0.87	50
volkswagen	0.89	0.82	0.85	50
accuracy			0.86	400
macro avg	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	400

Figure 7 Transfer Learning Classification Report

4 EXPERIMENTS AND RESULTS

4.1 BASELINE VS DEEEPER CNN

When in training, the baseline model started with stable validation accuracy for the first several epochs. However, as training went on there was evidence of overfitting. This became clear from the increasing separation between training and validation accuracy, accompanied by a

loss value split – training accuracy increased while validation accuracy stagnated and oscillated. Such signs meant that the model was "overfitting" the training data without generalizing it well into unseen data. On the contrary, a deeper model trained with an Adam optimizer exhibited a better effect in overcoming overfitting as a result of incorporation of regularization strategies especially dropout, batch normalization and L2 regularization. However, the curve of accuracy and loss for the model had obvious fluctuation during the training process. This instability implies sensitivity in the optimization process, which may appear because of the network's complexity and the dynamics of the Adam optimizer. Still, the model was able to sustain relatively balanced performance of both training and validation datasets. Interestingly, despite the fact that the baseline had a more fundamental architecture, it presented an even slightly better loss optimization perhaps because of smaller number of parameters and less architecture complexity hence to converge more consistently on the training data. But the deeper model had broader stability in generalization and especially avoided severe overfitting.
When comparing the classifying performance of both models, deeper model performs better in a meaningful way compared to the baseline. For majority of car brand categories the precision scores greatly improved. For example, the precision for Hyundai went from 0.71 to 0.87 and for Lexus from 0.53 to 0.79, suggesting an improved capability of the deeper model to correctly predict positive predictions.

Similarly recall values improved from several classes indicating increased sensitivity of deeper model to detect true positives. Stand out wins include Toyota, where recall improved from 0.66 to 0.89, and Volkswagen which improved from 0.61 to 0.86. This implies that deeper model had a correct classification of more instances of each brand.

The F1-score, which balances both precision and recall, also exhibited a powerful upward tendency. For example, Mazda F1-Score improved from 0.63 up to 0.84, whereas Skoda improved its F1-Score from 0.69 up to 0.88 to demonstrate more balanced and robust performance in respect to both identifying and avoiding misclassification.

Most importantly, the overall accuracy of the model improved from 0.69 in basic model to 0.87 in deeper model – significantly better overall classification performance. This shows that widening the depth of the network, regularization and increased filters are some of the aspects that contribute to getting better feature extractions, generalization and eventually better results.

When comparing the classifying performance of both models, deeper model performs better in a meaningful way compared to the baseline. For majority of car brand categories the precision scores greatly improved. For example, the precision for Hyundai went from 0.71 to 0.87 and for Lexus from 0.53 to 0.79, suggesting an improved capability of the deeper model to correctly

predict positive predictions.

Similarly recall values improved from several classes indicating increased sensitivity of deeper model to detect true positives. Stand out wins include Toyota, where recall improved from 0.66 to 0.89, and Volkswagen which improved from 0.61 to 0.86. This implies that deeper model had a correct classification of more instances of each brand.

The F1-score, which balances both precision and recall, also exhibited a powerful upward tendency. For example, Mazda F1-Score improved from 0.63 up to 0.84, whereas Skoda improved its F1-Score from 0.69 up to 0.88 to demonstrate more balanced and robust performance in respect to both identifying and avoiding misclassification

Most importantly, the overall accuracy of the model improved from 0.69 in basic model to 0.87 in deeper model – significantly better overall classification performance. This shows that widening the depth of the network, regularization and increased filters are some of the aspects that contribute to getting better feature extractions, generalization and eventually better results.

	precision	recall	f1-score	support
hyundai	0.67	0.76	0.71	50
lexus	0.61	0.76	0.68	50
mazda	0.91	0.80	0.85	50
mercedes	0.71	0.78	0.74	50
opel	0.64	0.72	0.68	50
skoda	0.81	0.76	0.78	50
toyota	0.93	0.54	0.68	50
volkswagen	0.78	0.78	0.78	50
accuracy			0.74	400
macro avg	0.76	0.74	0.74	400
weighted avg	0.76	0.74	0.74	400

	precision	recall	f1-score	support
hyundai	0.87	0.82	0.85	50
lexus	0.82	0.94	0.88	50
mazda	0.94	0.90	0.92	50
mercedes	0.90	0.90	0.90	50
opel	0.84	0.76	0.80	50
skoda	0.75	0.90	0.82	50
toyota	0.89	0.84	0.87	50
volkswagen	0.89	0.82	0.85	50
accuracy			0.86	400
macro avg	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	400

Figure 8 Classification Report

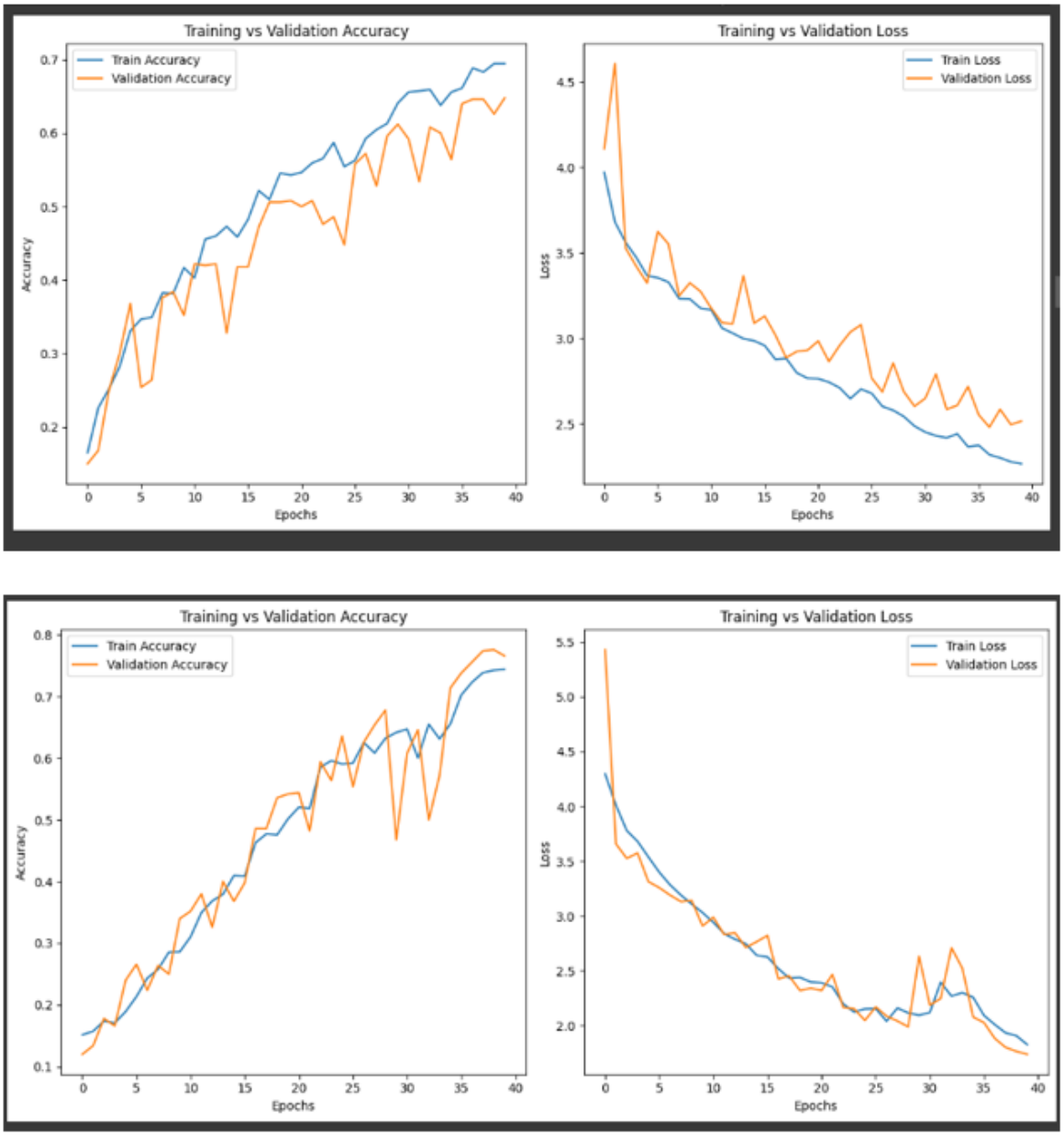

Figure 9 Graphs of Training and Validation Accuracy and Loss

4.2 COMPUTATIONAL EFFICIENCY

Despite securing better performance, the deeper model offered the price of increased computational burden compared to the baseline model. The baseline model took about 30 minutes to train while the deeper model took about 50 minutes. It is mostly because of more convolutional layers and more filters; therefore, this training process gets more complex and takes more time for each epoch. This serves to show the conflicting nature between exactitude and computational effectiveness – even though the more profound model results in better classification results, it does so at the cost of more resource usage. A T4 GPU environment was leveraged from

Google Colab to support the training of both models effectively so that both could be trained in a sufficiently short period.

4.3 CHALLENGES IN TRAINING

While undergoing the training process, several challenges were engendered that interfered with model performance and efficiency. Limited size of the dataset was one of the main problems, which was among the main contributors to overfitting, particularly in the baseline CNN model. In order to eliminate this, image augmentation techniques including rotation, zoom, flipping, and displacement were used to artificially add to the training dataset in order to increase the model's ability to generalize. But even with data augmentation, the baseline model does not perform well which calls for stronger architecture.

To this end, a deeper model of convolutional neural networks was built with more layers and more filters. This deeper architecture to some extent addressed the issues of overfitting by using the regularization measures such as dropout and batch normalization. The introduced complexity helped the model extract more abstract features enhancing its classification accuracy with a broad range of car brand logos.

The other major challenge was the longer computational time for training of the deeper model. Although the baseline model trained within 21 minutes approximately, the deeper model took the training in the order of 45 minutes while using the Adam optimizer and up to 50 minutes if SGD was used. This increase of training time by factor two was a price to pay for the higher performance-it is typical for complex-model based learning of representation models to trade increased model complexity with increased training efficiency.

5 CONCLUSION AND FUTURE WORK

This coursework examined the performance of CNN-based models for car brand logo classification in the publicly available image dataset of 2,913 images for eight different car brands. The baseline CNN is simple in terms of architecture, computationally efficient, however, the accuracy obtained was moderate (69% as opposed to the highest possible 72%) and, as revealed by overfitting, such architectures with much fewer types of building blocks are susceptible to overfitting. Improving the network with extra convolutional layers, dropout, batch normalization, L2 regularization improved the comprehensive network class ability, with an accuracy score being advanced to 86% and more balanced respectively precision, recall, and F1-scores for the different classes. Additionally, Adam optimizer showed better generalization and convergence as compared to SGD thus validating its use in image classification tasks.

The transfer learning model developed on the pre-trained VGG16 architecture performed better than the baseline and deeper models, using pre-learned features from large ImageNet dataset. Following the fine-tuning of the top layers, the model achieved a top accuracy of 90% with stable validation curves and overall good performance on all the metrics of testing. This demonstrates the fact that transfer learning could be a valuable option for tasks where data is scarce but in which very high accuracy could be achieved at low training costs and with a small number of parameters to learn from scratch. Notably, the model was robust even when the images were of varying resolutions and conditions, which implied its feasibility for practical application.

To extend the research, the dataset should be extended to include other brands of cars and different conditions of images (e.g., weather, occlusion motion blur). This will improve generalizability. As for the models validation, real-world testing within smart surveillance or an autonomous car system may help to validate models under dynamic conditions. Furthermore, model optimization for deployment on edge devices using model compression and quantization and research into latest architectures such as EfficientNet and Vision Transformer might lead to increased accuracy and efficiency. Some change of the explanatory tools like Grad-CAM and use of multi-task learning framework can also make the system more interpretable and able to adapt a wider range of traffic analysis tasks.