**MSA 8600 Deep learning Homework 3 – Group 5**

**Project Report, April 4th, 2023**

**Car Brand Logo Image Classification using CNN**



**Team Members:**

Shreyashi Mukhopadhyay

Swathy Raja

Jibi Jacob

Anesia Mccarthy

**Business Problem:** Car brand logo image classification

**Introduction:**

Car brand logo image classification is an emerging field that involves the use of deep learning algorithms to analyze and identify the logos of different car brands from images. This technology has a wide range of use cases across various industries, including marketing, research, automotive aftermarket, vehicle maintenance and repair, insurance claims. By accurately identifying car brand logos, businesses can gain valuable insights into consumer preferences, make informed decisions about product placement and understand customer experience better.

Furthermore, logo classification can be used to protect intellectual property rights, track brand perception and loyalty over time, and compare brand recognition with that of competitors. With the growing demand for automation and data-driven decision-making, car brand logo image classification is becoming an increasingly important tool for businesses in the automotive industry and beyond.

The use cases for car brand logo image classification are numerous, and some of them include:

**Intellectual Property Protection:** Logo classification can help companies protect their intellectual property rights by identifying instances of unauthorized use or infringement of their logos.

**Marketing and Advertising:** Researchers in the automotive industry can use logo classification to gain insights into consumer preferences and how a particular brand logo influences buying behavior. They can also use it to track brand perception and loyalty over time.

**Competitive Analysis:** Companies can use logo classification to compare their brand recognition with that of their competitors and gain a better understanding of the market landscape.

**Product Placement and Sponsorship:** Logo classification can be used to study the effect of the design of a particular logo within different market segments based on age, gender and income. This information can help companies make informed decisions about product placement and sponsorship opportunities.

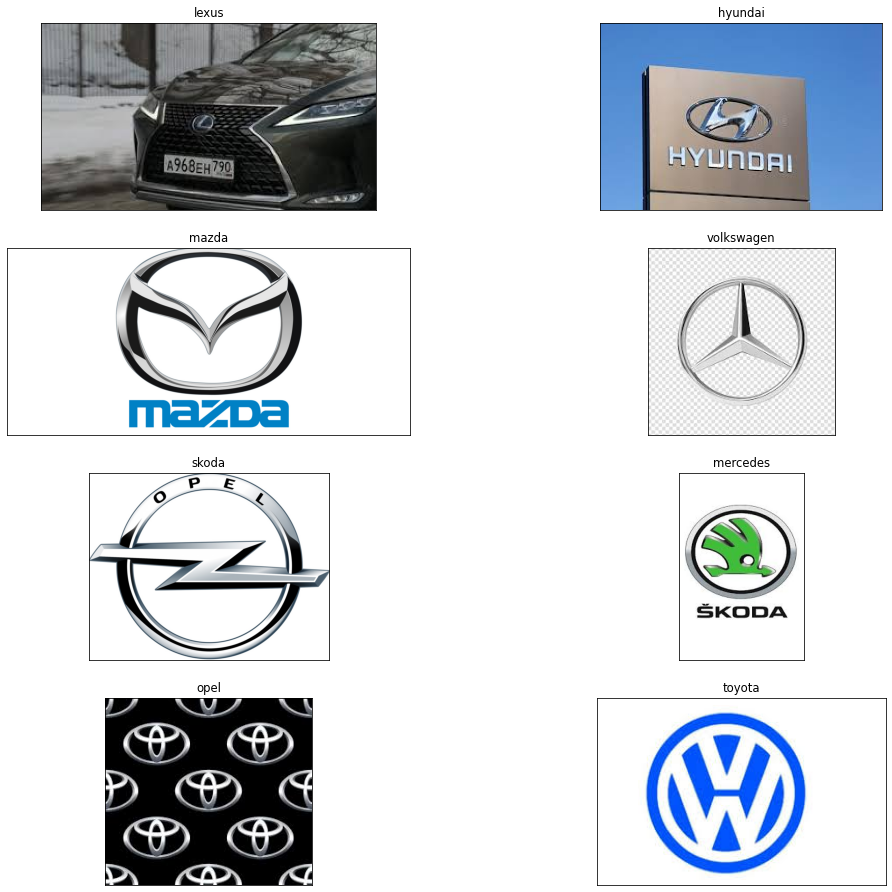
**Vehicle Maintenance and Repair:** Logo classification can be used by mechanics and technicians to quickly and accurately identify the brands of parts and components that need to be repaired or replaced which may be imprinted on a particular spare part.

**Insurance Claims:** Logo classification can be used by insurance companies to identify the make and model of vehicles involved in accidents, which can help speed up the claims process.

**Data Source:**

This data was sourced from Kaggle.com at [Car Brand Logos | Kaggle](https://www.kaggle.com/datasets/volkandl/car-brand-logos). It is compiled of logos from 8 different car companies including: hyundai, toyota, opel, skoda, volkswagen, mercedes, lexus, and mazda. The dataset is split into two folders: training and test; the training data contained over 300 images for each logo while the test folder contained 50 images for each logo.

**Data Visualization: Visualizing an image from each class.**

****

**Data Pre-processing:**

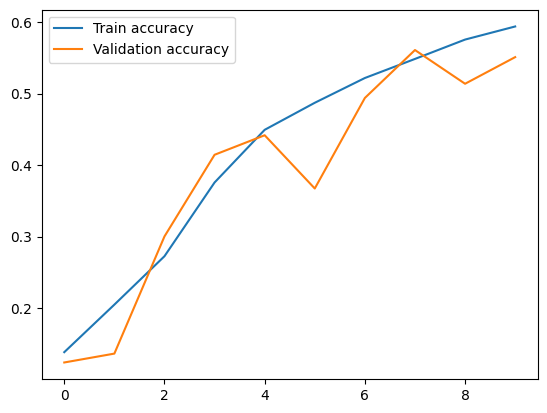
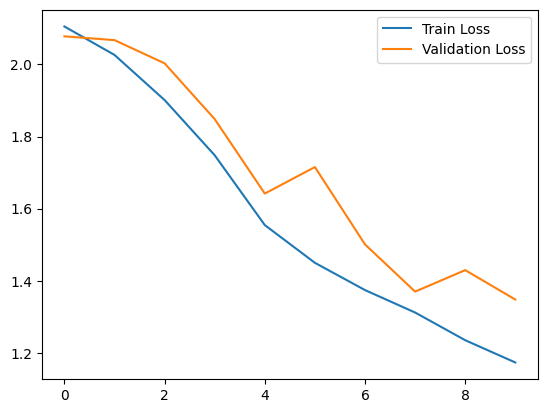
As an initial step the average height and width of all the images is calculated for the test and train data folders. The average width to height ratio is obtained as 0.8068. Based on the ratio, the input shape of the image is given as (193,240).

This is followed by data rescaling and augmentation. The ImageDataGenerator class from Keras is used to define the transformation that was applied to the images during training. This involves rescaling the pixel values to be between 0 and 1, random rotations up to 10 degrees, and horizontal flipping. Finally we have 2513 images belonging to 8 classes in the training set and 403 images belonging to 8 classes in the test set.

**CNN: Model 1**

The first model is a Convolutional Neural Network (CNN) that takes input images of size 193x240 with 3 color channels (RGB). It consists of 4 pairs of convolutional and MaxPooling layers that progressively reduce the spatial size of the output, followed by a Dropout layer for regularization. The convolutional layers have 32 filters of size 3x3 each and the MaxPooling layers have a pool size of 2x2 with a stride of 2x2. BatchNormalization is applied to the output of the first convolutional layer. After the convolutional and pooling layers, the output is flattened and fed into 2 fully connected layers with 96 and 32 units respectively, both with a ReLU activation function. Finally, the output is passed through a dense layer with 8 units and a softmax activation function to predict the class probabilities for the input image. The model has a total of 585,192 trainable parameters.

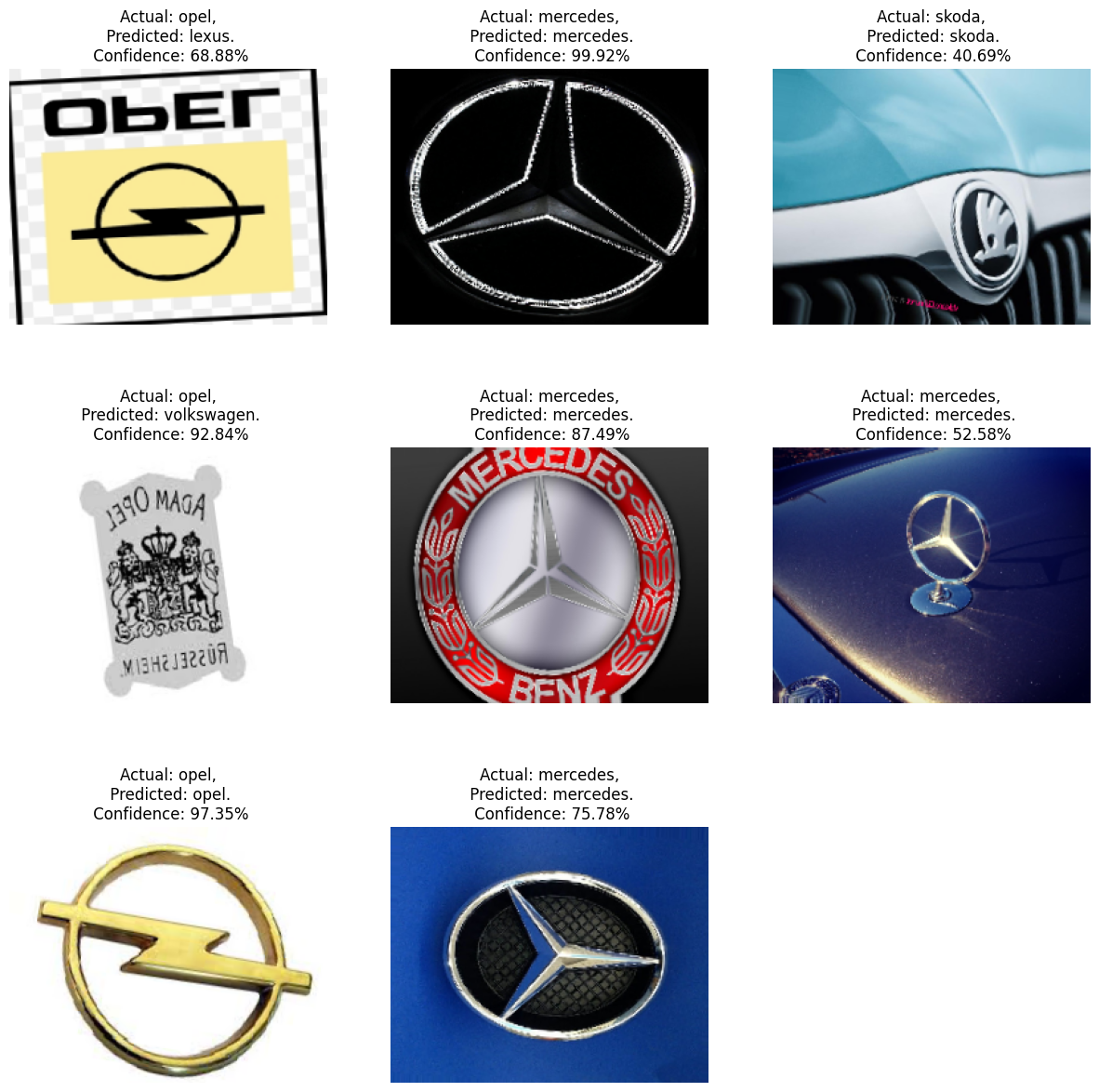
The model was trained using 10 epochs and a batch size of 32. From the results, it can be seen that the loss decreased and the accuracy improved as the number of epochs increased. This indicates that the model is learning from the data and improving its performance. However, it can also be seen that the validation accuracy is consistently lower than the training accuracy which may indicate overfitting. The model's accuracy on the validation set increases at first, reaches one of its highest points at epoch 3 with an accuracy of 30%, then somewhat decreases, but it still maintains a reasonable performance throughout the remaining epochs ending at 55% accuracy with the last epoch.



Test loss: 1.393

Test accuracy: 0.6079

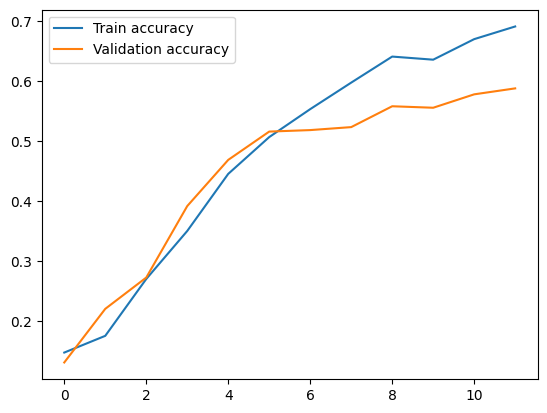
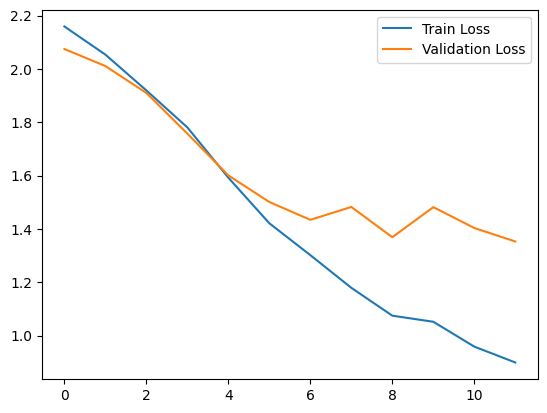
*Predictions:*

****

**CNN: Model 2**

Model 2 has 5 layers - 2 convolutional layers with max-pooling layers, 1 flatten layer, and 2 fully connected layers. Similarly to model 1, the first convolutional layer has 32 filters however, the size is 5x5 with a ReLU activation function. The second layer has 32 filters of size 3x3 with a ReLU activation function, and is followed by a max-pooling layer with a pool size of 2x2 and a stride of 2x2. After the max-pooling layer, the output feature maps are flattened into a one-dimensional array of size 92160. Next, there are two fully connected layers with 96 and 32 units. The first dense layer has a ReLU activation function and a dropout of 0.4 to prevent overfitting. The second dense layer also has a ReLU activation function. Lastly, there is an output layer with 8 units and a softmax activation function. The total number of trainable parameters in this model is 8,862,504.

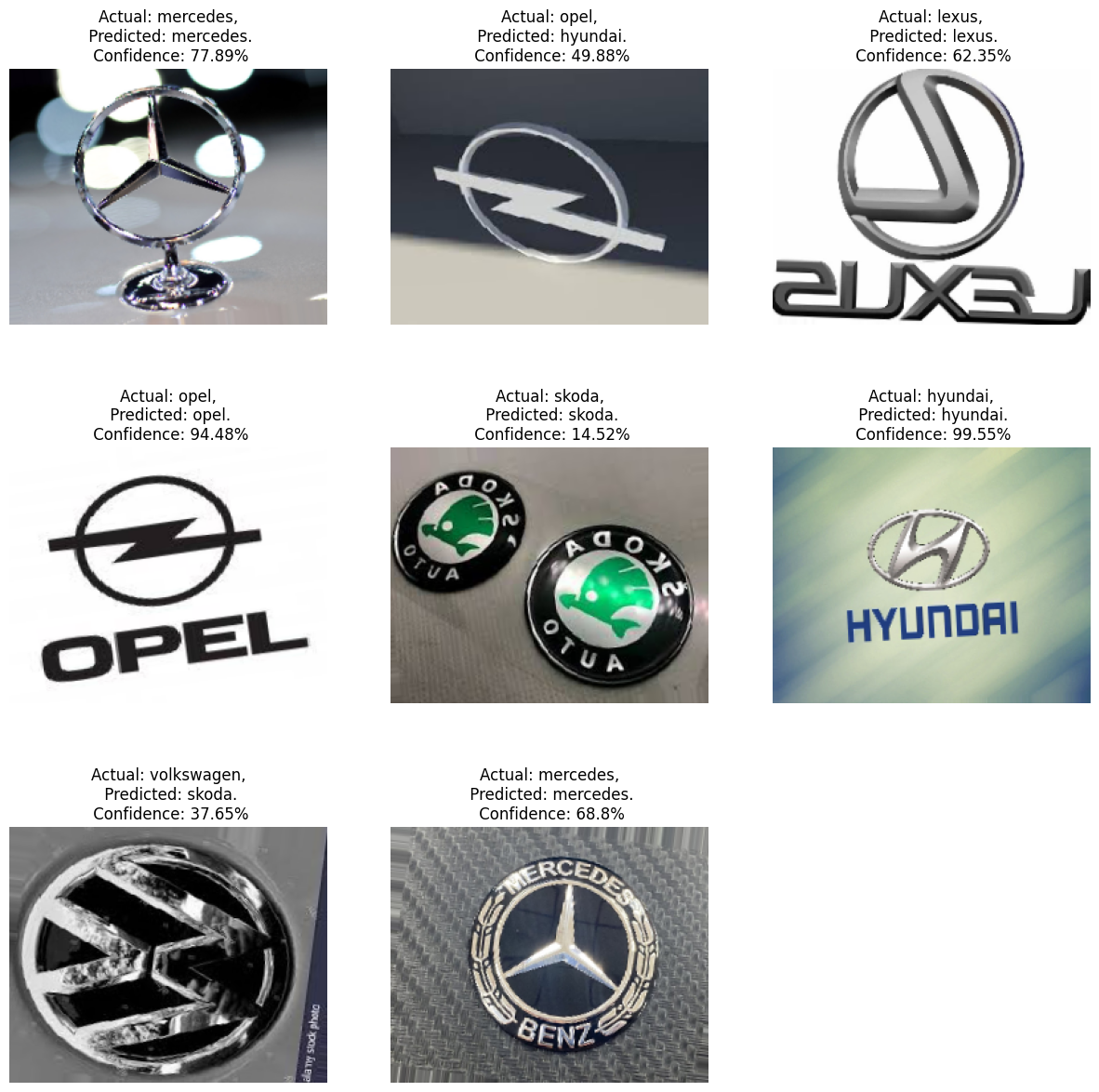
During training, the results show that the training and validation accuracy gradually increase over time, while the training and validation loss decrease. The training accuracy starts at 14.8% and reaches 69.1% at the end of 12 epochs. The validation accuracy also improves from 13.1% to 58.8%. The training loss decreases from 2.16 to 0.89, and the validation loss decreases from 2.07 to 1.35. The training and validation curves seem to diverge after the 7th epoch, indicating that the model may also be overfitting to the training data.



Test loss: 1.3663

Test accuracy: 0.5732

*Predictions:*



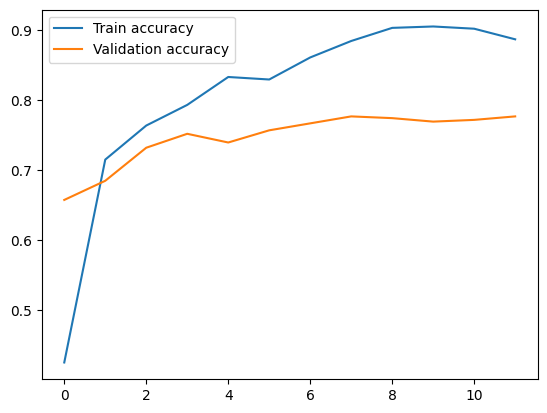
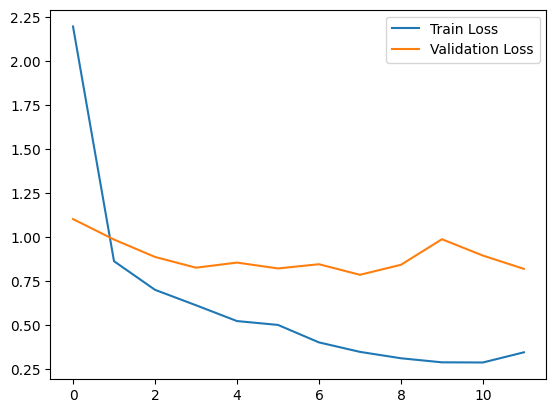
**VGG19**

VGG19 is a deep neural network with 19 layers. It has been pre-trained on the large ImageNet dataset, which contains millions of images and thousands of different classes. As a result, the model has learned to extract meaningful features from images, which can be used for a variety of computer vision tasks such as image classification, object detection, and image segmentation.

The base model, VGG19, has 20,024,384 non-trainable parameters, which are the pre-trained weights and biases. The remaining 22,550,024 trainable parameters are from the layers that were added on top of the base model.

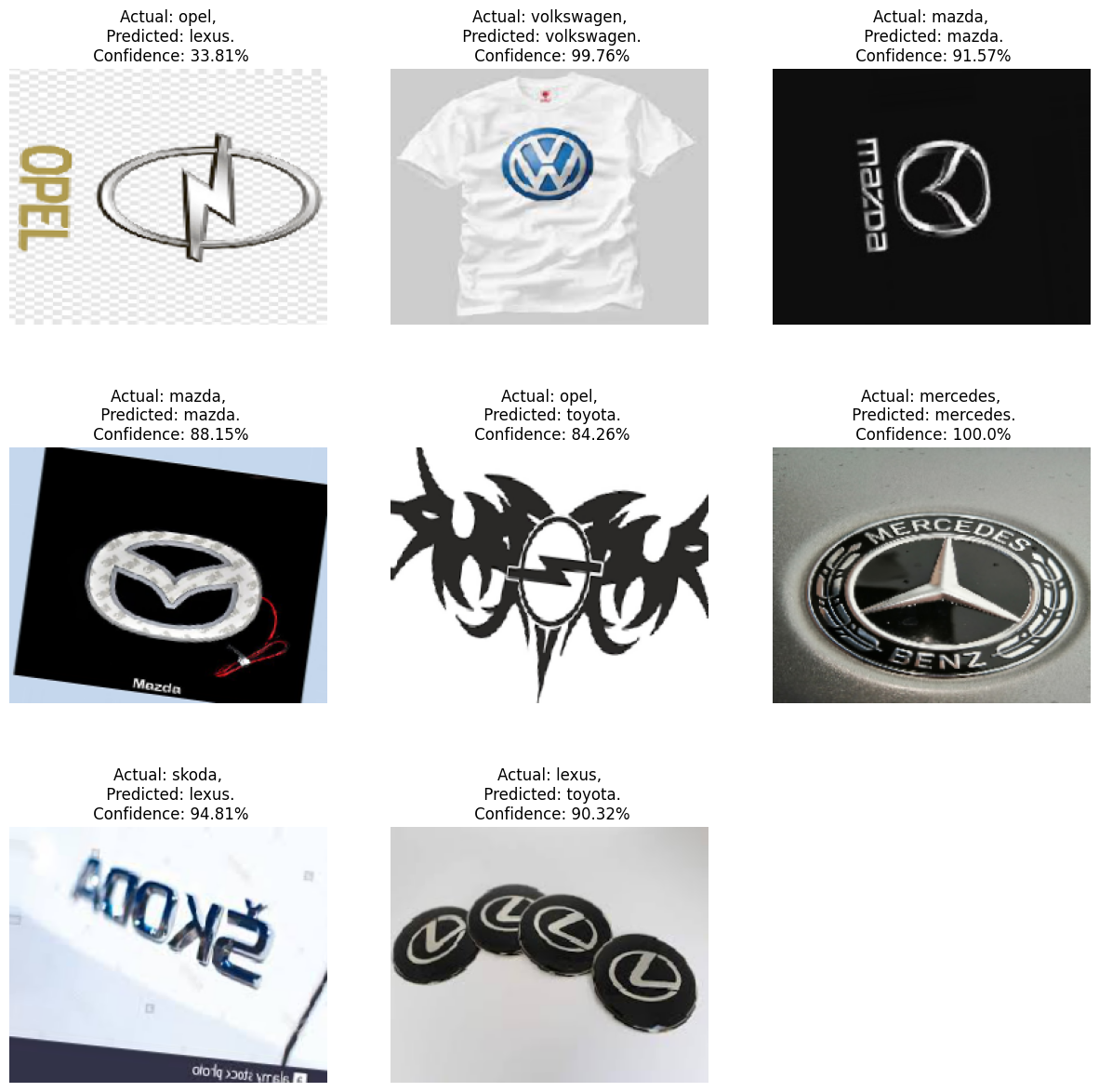
The added layers are: Flatten: this layer flattens the output of the convolutional layers into a 1D array to be fed to the fully connected layers; Dense with 1024 units: this layer has 22,021,120 trainable parameters and is the first fully connected layer. It uses the ReLU activation function; Dropout with 0.2 rate: this randomly sets 20% of the input units to 0 at each update during training time, to help prevent overfitting; Dense layer with 512 units: this layer has 524,800 trainable parameters and is the second fully connected layer. It uses ReLU activation function; dropout layer with 0.2 rate; and a Dense layer with 8 units: this layer has 4,104 trainable parameters and is the output layer. It uses softmax activation over the 8 classes.

The results show the performance of the model for 12 epochs. The model achieved an accuracy of 42.58% on the training set and 65.76% on the validation set in the first epoch. As the number of epochs increased, the accuracy on the training set also increased, reaching a maximum of 88.66% in the 12th epoch. The accuracy on the validation set did not improve significantly after the 5th epoch and hovered around 76-77%. The loss function decreased steadily during the training process, indicating that the model was learning to make more accurate predictions. These results suggest that the model may have started to overfit to the training data, as the accuracy on the validation set did not improve much after the 5th epoch.



Test accuracy for this model was 0.5756 while test loss was 1.32.

*Predictions:*

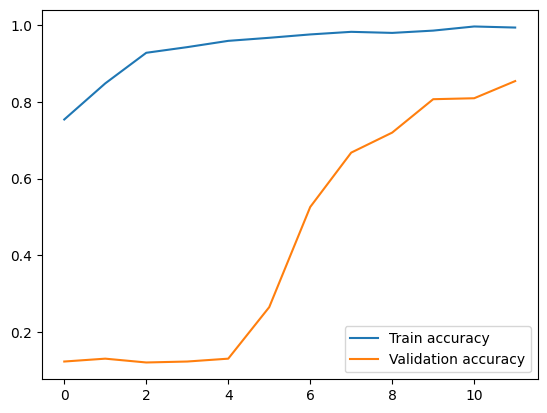
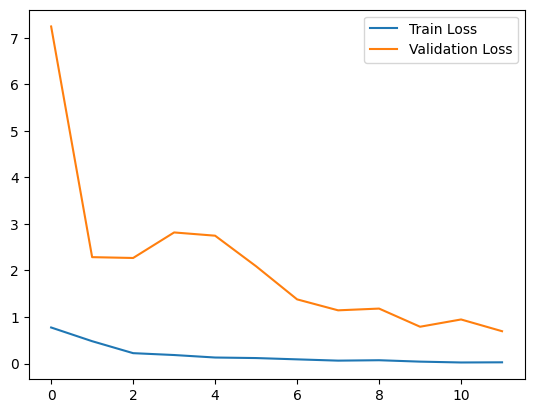


**ResNet 50**

ResNet-50 is composed of 50 layers and is based on the residual learning framework, which allows for the training of very deep neural networks. The key innovation of ResNet-50 is the use of residual blocks, which enable the network to learn residual mappings instead of trying to learn the underlying mapping directly. This helps to alleviate the problem of vanishing gradients and enables the network to be much deeper than previous architectures while maintaining high accuracy.

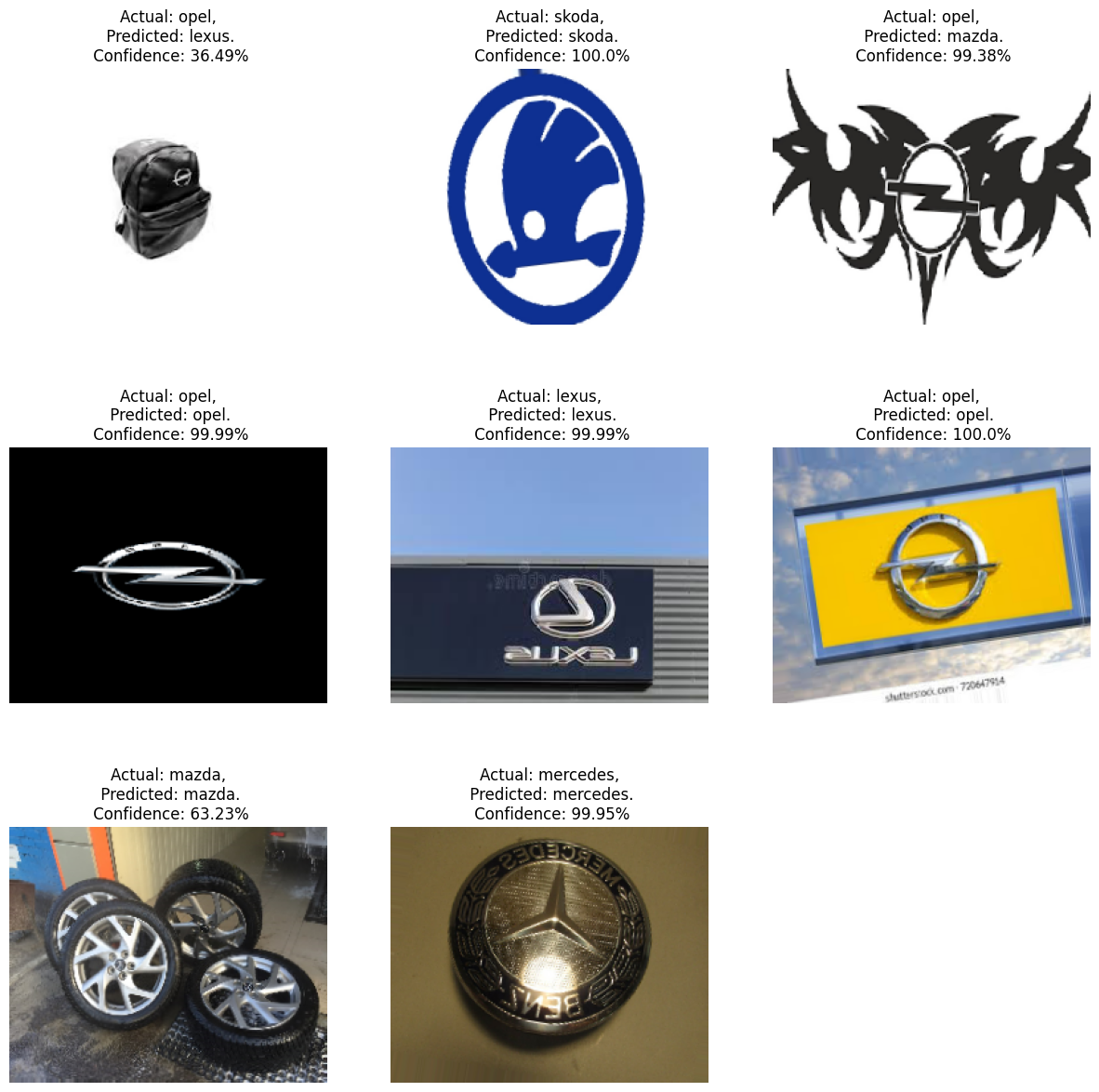
The ResNet50 model was defined using the SGD optimizer with specific learning rate, decay, momentum, and nesterov momentum, and uses sparse categorical cross-entropy loss function and accuracy metric for training and evaluation.

The model started with a training accuracy of 75.37% and a validation accuracy of 12.41% in the first epoch. As training progressed, the model's performance improved, and the accuracy increased to 99.28% on the training set and 85.36% on the validation set in the last epoch. Similarly, the loss decreased from 0.7741 to 0.0259 on the training set and from 7.2416 to 0.6942 on the validation set, which indicates that the model was able to make more accurate predictions as training progressed. The results also show that the model started to possibly overfit after the sixth epoch. The model's performance on the training set continued to improve, but this improvement did not generalize to the validation set. This could indicate that the model was memorizing the training set instead of learning general features.



ResNet50 has an accuracy of 0.8337 and a loss of 0.7527.

*Predictions:*

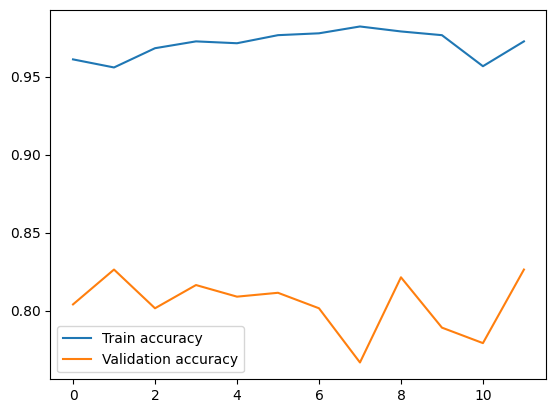
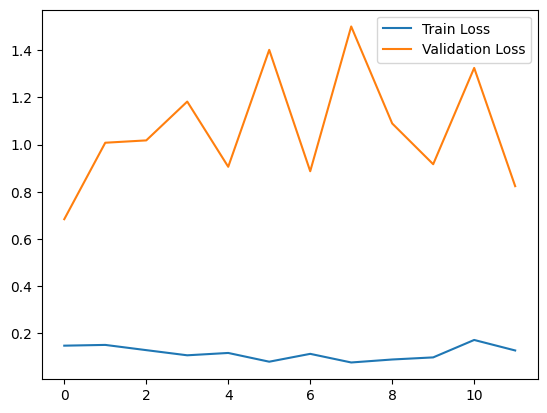


**MobileNet**

MobileNet is a type of neural network architecture designed for mobile and embedded vision applications. MobileNet uses a depthwise separable convolution operation, which separates the spatial and depthwise convolution operations. This allows it to reduce the number of parameters and operations required while still maintaining high accuracy.

The pre-trained MobileNet model is used as a base model and its output tensor is passed as input to the new classification layers. The new classification layers consist of a GlobalAveragePooling2D layer followed by three fully connected Dense layers and a softmax activation layer with 8 outputs. The model then sets the first 20 layers of the MobileNet model to be non-trainable while allowing the remaining layers to be trained during the model training process.

The model achieved high accuracy on the training set with values ranging from 95.58% to 98.21%. However, the accuracy on the validation set is lower, ranging from 76.67% to 82.63%. This suggests that the model could be overfitting to the training data and is not generalizing well to new, unseen data. Additionally, the loss values for the validation set are generally higher than those for the training set, which also suggests that the model is overfitting.



The MobileNet model has a test loss of 0.823 and a test accuracy of 0.826 which means that it correctly classified 82.6% of the test data. This suggests that the model is performing reasonably well in identifying car logos.

*Predictions:*

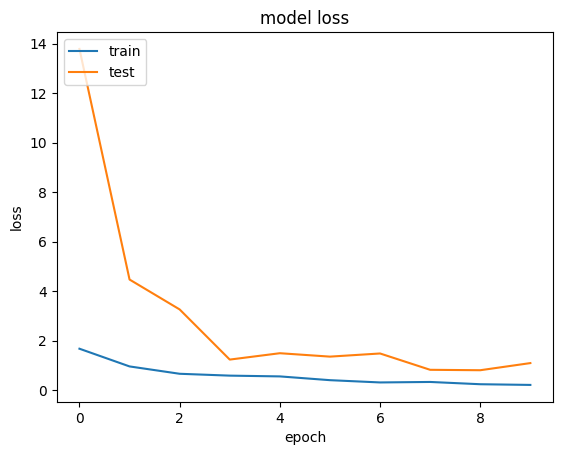
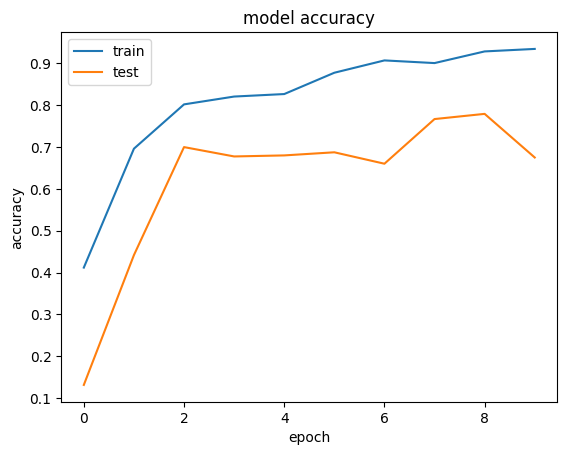
**

**InceptionV3**

Inception v3 is the third iteration in the Inception series of CNN architectures that were designed to be very computationally efficient while still achieving high performance on image classification tasks. It is composed of many convolutional layers with varying layers and multiple pooling layers.

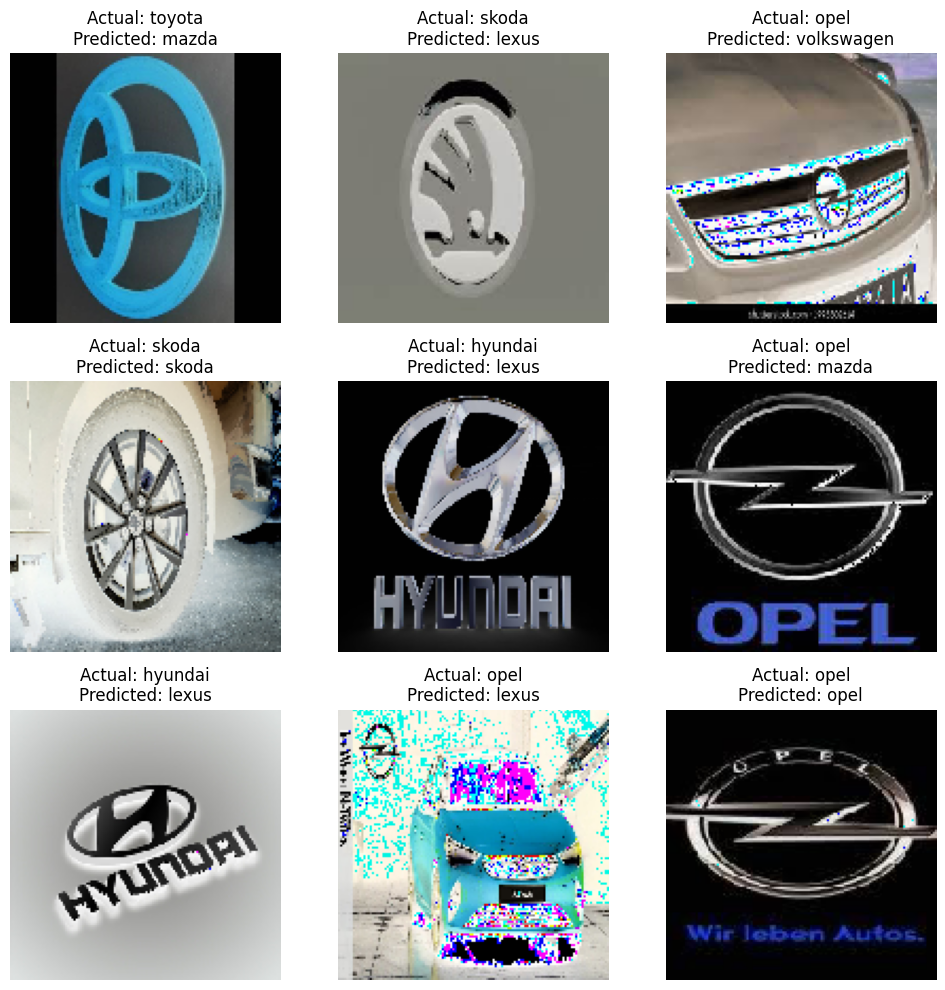
For the purpose of this project, a new model was built using transfer learning with the pre-trained InceptionV3 model. A global average pooling layer was added to reduce the spatial dimensions of the output followed by a fully-connected layer with 1024 units and a ReLU activation function. A softmax layer was added with 8 units for multi-class classification.

During the training process, the model achieved an accuracy of 41.19% on the training set and 13.15% on the validation set in the first epoch, showing poor performance. As training continued, the model's accuracy increased significantly, achieving an accuracy of 93.43% on the training set and 67.49% on the validation set in the final epoch. The loss values decreased notably throughout the training process indicating that the model was learning to classify the images correctly. The validation loss was initially high, but decreased after a few epochs, indicating that the model was generalizing well to new, unseen data. Overall, the results show that the InceptionV3 model was able to learn from the dataset and classify the images.



The test accuracy was 0.5459 and test loss was 1.3213.

*Prediction:*



**Conclusion**

Based on the results, the ResNet50 model performed the best with a test loss of 0.7527 and test accuracy of 0.8337. The second best model was MobileNet which had a slightly higher test loss and lower test accuracy, indicating it did not perform as well as ResNet50. The remaining models had significantly higher test losses and lower test accuracies, indicating they were not effective in predicting the test data. Therefore, ResNet50 is the best performing model among the models tested.

In conclusion, car brand logo image classification has various applications across different industries, such as automotive, marketing, and insurance. With the rise of automation and data-driven decision-making, accurately identifying car brand logos has become increasingly important. The convolutional neural network models discussed in this analysis showed promising results in classifying car brand logos with high accuracy. Although some performed better than others, all the models have room for improvement, such as reducing overfitting and increasing accuracy on the validation set. Overall, car brand logo image classification has the potential to provide valuable insights to businesses and organizations and help them make informed decisions in their respective industries.