**Transfer Learning Assignment**

Team member:

1. Cedric Oliveira da Silva Costa: 11369574
2. Abhishek Sapkal: 10831908

Unit: Artificial Intelligence

# **Introduction:**

In this assignment, we are performing and building an image classification (flower classifier) using the application of machine learning. Pre-trained models such as convolutional neural networks containing learned features such as corners, edges, shape, colour, etc. are implemented.

**Task 2 - Setting up Our Image Data:**

A total of 1000 images segregated into 5 categories were given to us and to train and test the datasets, we performed an 80-20 split, i.e., 800 are used for training and the remaining 200 images are used for validations.

# **Import the required libraries:**

A picture containing text

Description automatically generated

*Fig1: importing libraires*

# **Task 3 - Import the Model:**

We have created a base model from the MobileNetV2 model. The pre-trained MobileNetV2 is downloaded using tf.keras.applications. The include\_top parameter is set to False, which will load a network that doesn’t include the classification layers (output layer) at the top.

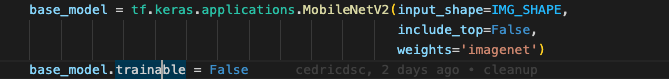
Text

Description automatically generated

When is model is loaded, it contain around 3 millions of trainable params, but after replacing the last layer with the Dense layer of our image classifications, the params got reduced to 6,405.

Graphical user interface, text

Description automatically generated



*Fig2: Downloading the MobileNetV2 model*

# **Task 4 - Data preparation for non-accelerated version:**

Figure 3 refers to an 80,20 split, also out of 32 batch sizes, 25 is for training, 6 for validation, and 1 for testing. For data augmentation, since we are using the MobileNetV2 model which requires pixel values in [-1, 1], we have rescaled the data accordingly.

A picture containing graphical user interface

Description automatically generated

*Fig3: Data pre-processing*

# **Task 5 - Compile the model:**

Once the preparation of training, validation and test dataset is done, the model is then compiled with SGD optimizer. With the parameters learning\_rate=0.01, momentum=0.0, nesterov=False. To get accuracy, the test dataset is evaluated with the compiled model.



*Fig:4: Accuracy value with SGD optimizer*

Since the accuracy is low, the prediction is also very low. In the below image, one can see the out of 24 test images, only 4 have been predicted correctly.

Diagram

Description automatically generated

*Fig5: Model predictions for non-accelerated version*

# **Task 6 – Plotting the graphs:**

# **Chart, line chart Description automatically generatedChart, line chart Description automatically generated**

*Fig6: Training and validation loss v/s Time(epochs)*

The above two graphs provide the data for e training and validation errors vs time as well as the training and validation accuracies. Starting with the left graph, the training accuracy is represented by blue graph, and we could see that after epoch 7 contains the maximum value of 0.218. However, the validation accuracy measured on validation dataset and is represented by red graph is very low and which states that the compiled model was unable to classify the images. Moreover, one the right graph, the validation loss is much higher.

# **Task 7 – Experimenting with 3 new learning rates:**

In this task, we have used 3 new learning rates. i.e. 0.00005, 0.00001and 0.1. The model is then trained with this new value and the one which provides the highest accuracy is picked. The bar plot shows the accuracy/loss value for each learning rates values. The image depicts that the learning rate of 0.1 gives the highest accuracy.



*Fig7: Learning rate with highest accuracy*

Chart, bar chart

Description automatically generated

*Fig8: Plotting the accuracy and loss value for different learning rates*

# **Task 8 - Experimenting with 3 new momentum rates with best learning rate with best accuracy:**

After getting the best learning rate, the value is passed to model wo compile with 3 more new momentum rate to the maximum accuracy from the model. Simliar to task above the accuracy values with different momentum rate is compared and one with highest accuracy is picked. Using these techniques, we have managed to increase the accuracy level from 0.21 to 0.25

\ *Fig9: Momentum rate with highest accuracy*

Chart, bar chart

Description automatically generated

*Fig10: Plotting the accuracy and loss value for different momentum rates*

Furthermore, using these parameters, the prediction is visualized. The top two plots anticipated the higher number of correct predictions.

Timeline

Description automatically generated with low confidenceDiagram

Description automatically generatedTimeline

Description automatically generatedDiagram

Description automatically generated

*Fig11: Model predictions from top left to right for 0.0, 0.001, 0.01 and 0.1 respectively*

# **Task 9 – Prepare the datasets with accelerated version:**

In our experiment, the model has under-performed i.e., the accuracy value is very much on the lower side. In this step, the data is augmented. The data augmentation is process in which the existing images are changed in minor scale. These changes include transformation such as rotation, flipping, rescaling, and resizing.

# **Task 10 – Experimenting for the accelerated version with new momentum rates:**

The same process of compiling the existing model with 3 different learning rates and with new augmented dataset is done. The accuracy values are compared and the learning rate with highest accuracy is picked and model is compiled again with 3 new momentum values.

Chart, bar chart

Description automatically generated

*Fig12: Plotting the accuracy and loss value for different learning rates for accelerated version*

The using the best learning rate and momentum rate, the highest accuracy is executed. However, the accuracy is again low, and it explains the less prediction as in the figure 14.

Chart, bar chart

Description automatically generated

*Fig13: Plotting the accuracy and loss value for different momentum rates for accelerated version*

Chart

Description automatically generated with medium confidenceDiagram

Description automatically generated with medium confidenceDiagram

Description automatically generatedA picture containing diagram

Description automatically generated *Fig14: Model predictions from top left to right for 0.0, 0.001, 0.01 and 0.1 respectively*