

Model Development Phase Template

Date	12 July 2024
Team ID	SWTID1720067113
Project Title	Dog Breed Identification using Transfer Learning
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):

```
[ ] train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

[ ] selected_classes = []
for breed in labels.breed.unique():
    selected_classes.append(breed)

▶ datagen = ImageDataGenerator()

generator = datagen.flow_from_directory(
    'subset/train',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False,
    classes=selected_classes
)

Found 10222 images belonging to 120 classes.

[ ] test_datagen = ImageDataGenerator(rescale=1./255)
```

```
[ ] Image_size=[224,224]
sol=MobileNetV2(input_shape=Image_size + [3], weights='imagenet', include_top = False)
for i in sol.layers:
    i.trainable = False
y=Flatten()(sol.output)
```

↔ Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/9406464/9406464 [=====] - 1s 0us/step

```
[ ] final = Dense(120, activation='softmax')(y)
```

▶ model = Model(inputs=sol.input, outputs=final)

```
[ ] model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['Accuracy'])
```

```
[ ] for data, labels in generator:
    print('Data shape:', data.shape)
    print('Labels shape:', labels.shape)
    break # Only need to check one batch
```

↔ Data shape: (32, 224, 224, 3)
Labels shape: (32, 120)

```
[ ] # Fit the model
model.fit(generator, epochs=15)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1 MobileNet V2	<pre>[] final = Dense(120, activation='softmax')(y)</pre>	<pre># Fit the model model.fit(generator, epochs=15) Epoch 1/15 320/320 [=====] - 31s 83ms/step - loss: 86.9138 - Accuracy: 0.0079 Epoch 2/15 320/320 [=====] - 27s 83ms/step - loss: 76.3017 - Accuracy: 0.0332 Epoch 3/15 320/320 [=====] - 25s 78ms/step - loss: 66.1721 - Accuracy: 0.0764 Epoch 4/15 320/320 [=====] - 26s 80ms/step - loss: 58.8636 - Accuracy: 0.1043 Epoch 5/15 320/320 [=====] - 27s 84ms/step - loss: 51.4098 - Accuracy: 0.1517 Epoch 6/15 320/320 [=====] - 25s 78ms/step - loss: 48.1306 - Accuracy: 0.1793 Epoch 7/15 320/320 [=====] - 26s 81ms/step - loss: 36.5324 - Accuracy: 0.2756 Epoch 8/15 320/320 [=====] - 26s 81ms/step - loss: 28.9178 - Accuracy: 0.3685 Epoch 9/15 320/320 [=====] - 26s 81ms/step - loss: 26.5497 - Accuracy: 0.4278 Epoch 10/15 320/320 [=====] - 26s 82ms/step - loss: 20.8736 - Accuracy: 0.4984 Epoch 11/15 320/320 [=====] - 26s 81ms/step - loss: 13.9174 - Accuracy: 0.6419 Epoch 12/15 320/320 [=====] - 27s 84ms/step - loss: 17.4532 - Accuracy: 0.5563 Epoch 13/15 320/320 [=====] - 27s 83ms/step - loss: 14.1683 - Accuracy: 0.6471 Epoch 14/15 320/320 [=====] - 26s 82ms/step - loss: 10.1021 - Accuracy: 0.7393 Epoch 15/15 320/320 [=====] - 25s 79ms/step - loss: 8.7054 - Accuracy: 0.7707 <keras.src.callbacks.History at 0x7b2454b89fc0></pre>

<p>Model 2</p> <p>VGG19</p>	<pre>[] final = Dense(120, activation='softmax')(y)</pre>	<pre># Fit the model model.fit(generator, epochs=15) Epoch 1/15 320/320 [=====] - 65s 176ms/step - loss: 294.5328 - Accuracy: 0.0252 Epoch 2/15 320/320 [=====] - 56s 174ms/step - loss: 124.2027 - Accuracy: 0.3150 Epoch 3/15 320/320 [=====] - 57s 178ms/step - loss: 26.9172 - Accuracy: 0.7512 Epoch 4/15 320/320 [=====] - 55s 172ms/step - loss: 9.4837 - Accuracy: 0.8945 Epoch 5/15 320/320 [=====] - 56s 173ms/step - loss: 7.1624 - Accuracy: 0.9158 Epoch 6/15 320/320 [=====] - 57s 178ms/step - loss: 3.5580 - Accuracy: 0.9541 Epoch 7/15 320/320 [=====] - 57s 177ms/step - loss: 21.3413 - Accuracy: 0.8532 Epoch 8/15 320/320 [=====] - 55s 172ms/step - loss: 18.8485 - Accuracy: 0.8492 Epoch 9/15 320/320 [=====] - 57s 178ms/step - loss: 37.6908 - Accuracy: 0.8043 Epoch 10/15 320/320 [=====] - 55s 172ms/step - loss: 20.9604 - Accuracy: 0.8862 Epoch 11/15 320/320 [=====] - 55s 173ms/step - loss: 4.1757 - Accuracy: 0.9661 Epoch 12/15 320/320 [=====] - 56s 173ms/step - loss: 5.9315 - Accuracy: 0.9617 Epoch 13/15 320/320 [=====] - 56s 173ms/step - loss: 13.8605 - Accuracy: 0.9338 Epoch 14/15 320/320 [=====] - 55s 173ms/step - loss: 12.6411 - Accuracy: 0.9422 Epoch 15/15 320/320 [=====] - 55s 173ms/step - loss: 5.7751 - Accuracy: 0.9589 <keras.src.callbacks.History at 0x792ffc1866e0></pre>
<p>Model 3</p> <p>EfficientNet B7</p>	<pre>[] final = Dense(120, activation='softmax')(y)</pre>	<pre># Fit the model model.fit(generator, epochs=15) Epoch 1/15 320/320 [=====] - 114s 355ms/step - loss: 22.3111 - Accuracy: 0.6911 Epoch 2/15 320/320 [=====] - 114s 357ms/step - loss: 14.3997 - Accuracy: 0.8023 Epoch 3/15 320/320 [=====] - 114s 356ms/step - loss: 9.5306 - Accuracy: 0.8554 Epoch 4/15 320/320 [=====] - 114s 356ms/step - loss: 12.6607 - Accuracy: 0.8418 Epoch 5/15 320/320 [=====] - 114s 355ms/step - loss: 6.0558 - Accuracy: 0.9063 Epoch 6/15 320/320 [=====] - 114s 356ms/step - loss: 9.5931 - Accuracy: 0.8874 Epoch 7/15 320/320 [=====] - 114s 356ms/step - loss: 7.7925 - Accuracy: 0.8862 Epoch 8/15 320/320 [=====] - 114s 355ms/step - loss: 7.0943 - Accuracy: 0.9031 Epoch 9/15 320/320 [=====] - 114s 356ms/step - loss: 4.9911 - Accuracy: 0.9409 Epoch 10/15 320/320 [=====] - 114s 356ms/step - loss: 5.9198 - Accuracy: 0.9237 Epoch 11/15 320/320 [=====] - 114s 356ms/step - loss: 9.4890 - Accuracy: 0.9129 Epoch 12/15 320/320 [=====] - 114s 356ms/step - loss: 5.3840 - Accuracy: 0.9354 Epoch 13/15 320/320 [=====] - 114s 355ms/step - loss: 4.4254 - Accuracy: 0.9587 Epoch 14/15 320/320 [=====] - 114s 356ms/step - loss: 12.4857 - Accuracy: 0.8890 Epoch 15/15 320/320 [=====] - 114s 356ms/step - loss: 6.6507 - Accuracy: 0.9415 <keras.src.callbacks.History at 0x7ac7a0541660></pre>