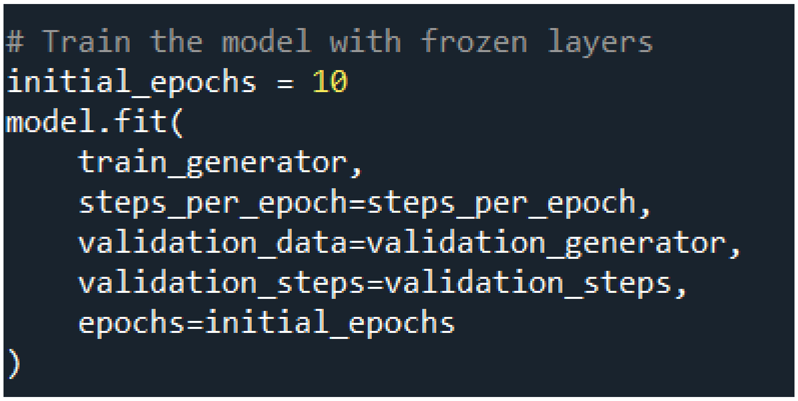
Model Development Phase Template

Date	1 December 2024
Team ID	739846
Project Title	Garbage Classification Using Deep 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):



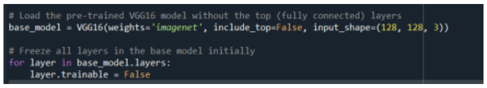
```
# Train the model with frozen layers
initial_epochs = 10
model.fit(
    train_generator,
    steps_per_epoch=steps_per_epoch,
    validation_data=validation_generator,
    validation_steps=validation_steps,
    epochs=initial_epochs
)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics

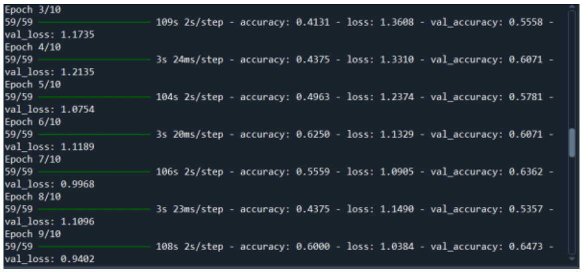
<table>
<tr>
<td>CNN model

(Base Model)</td>
<td>



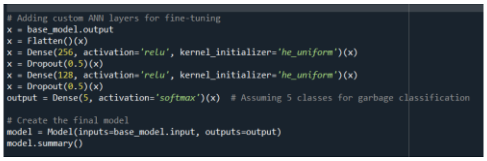
```
# Load the pre-trained VGG16 model without the top (fully connected) layers
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(128, 128, 3))

# Freeze all layers in the base model initially
for layer in base_model.layers:
    layer.trainable = False
```

</td>
<td>



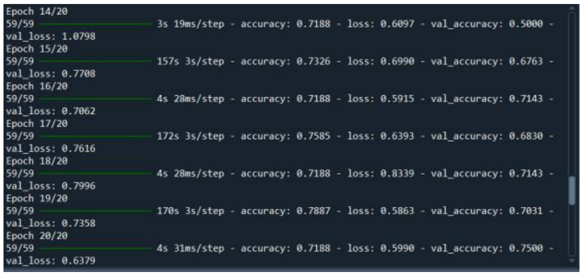
```
Epoch 3/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 109s 2s/step - accuracy: 0.4131 - loss: 1.3608 - val_accuracy: 0.5558 -
val_loss: 1.1735
Epoch 4/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 3s 24ms/step - accuracy: 0.4375 - loss: 1.3310 - val_accuracy: 0.6071 -
val_loss: 1.2135
Epoch 5/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 104s 2s/step - accuracy: 0.4963 - loss: 1.2374 - val_accuracy: 0.5781 -
val_loss: 1.0754
Epoch 6/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 3s 20ms/step - accuracy: 0.6250 - loss: 1.1329 - val_accuracy: 0.6071 -
val_loss: 1.1189
Epoch 7/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 106s 2s/step - accuracy: 0.5559 - loss: 1.0905 - val_accuracy: 0.6362 -
val_loss: 0.9968
Epoch 8/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 3s 23ms/step - accuracy: 0.4375 - loss: 1.1490 - val_accuracy: 0.5357 -
val_loss: 1.1096
Epoch 9/10
59/59 ━━━━━━━━━━━━━━━━━━━━ 108s 2s/step - accuracy: 0.6000 - loss: 1.0384 - val_accuracy: 0.6473 -
val_loss: 0.9402
```

</td>
</tr>
<tr>
<td>Fine Tuning</td>
<td>



```
# Adding custom ANN layers for fine-tuning
x = base_model.output
x = Flatten()(x)
x = Dense(256, activation='relu', kernel_initializer='he_uniform')(x)
x = Dropout(0.5)(x)
x = Dense(128, activation='relu', kernel_initializer='he_uniform')(x)
x = Dropout(0.5)(x)
output = Dense(5, activation='softmax')(x)  # Assuming 5 classes for garbage classification

# Create the final model
model = Model(inputs=base_model.input, outputs=output)
model.summary()
```

</td>
<td>



```
Epoch 14/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 3s 19ms/step - accuracy: 0.7188 - loss: 0.6097 - val_accuracy: 0.5000 -
val_loss: 1.0798
Epoch 15/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 157s 3s/step - accuracy: 0.7326 - loss: 0.6990 - val_accuracy: 0.6763 -
val_loss: 0.7708
Epoch 16/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 4s 28ms/step - accuracy: 0.7188 - loss: 0.5915 - val_accuracy: 0.7143 -
val_loss: 0.7062
Epoch 17/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 172s 3s/step - accuracy: 0.7585 - loss: 0.6393 - val_accuracy: 0.6830 -
val_loss: 0.7616
Epoch 18/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 4s 28ms/step - accuracy: 0.7188 - loss: 0.8339 - val_accuracy: 0.7143 -
val_loss: 0.7996
Epoch 19/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 170s 3s/step - accuracy: 0.7887 - loss: 0.5863 - val_accuracy: 0.7031 -
val_loss: 0.7358
Epoch 20/20
59/59 ━━━━━━━━━━━━━━━━━━━━ 4s 31ms/step - accuracy: 0.7188 - loss: 0.5990 - val_accuracy: 0.7500 -
val_loss: 0.6379
```

</td>
</tr>
</table>