# Model Development Phase

| | |
|---|---|
| Date | 25 June 2025 |
| Team ID | SWTID1750058607 |
| Project Title | Early-Stage Disease Diagnosis System Using Human Nail Image |
| 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)

```
from tensorflow.keras.callbacks import EarlyStopping
# from sklearn.utils import class_weight

early_stop = EarlyStopping(
    monitor='val_accuracy',
    mode= 'max',
    patience=3,
    restore_best_weights=True
)

history =model.fit(train_set,validation_data=test_set, epochs=30, steps_per_epoch = len(train_set)//3, validation_steps = len(test_set)//3)
# Get final epoch's training and validation accuracy
train_acc = history.history['accuracy'][-1]
val_acc = history.history['val_accuracy'][-1]

# Print formatted values without scientific notation
print("Training Accuracy: {:.4f}".format(train_acc))
print("Validation Accuracy: {:.4f}".format(val_acc))
```

```
Epoch 1/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 484s 70s/step - accuracy: 0.4922 - loss: 1.6717 - val_accuracy: 0.4688 - val_loss: 1.8478
Epoch 2/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.4300 - loss: 1.9040 - val_accuracy: 0.4531 - val_loss: 1.7980
Epoch 3/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 480s 69s/step - accuracy: 0.4987 - loss: 1.7352 - val_accuracy: 0.6250 - val_loss: 1.2784
Epoch 4/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 537s 70s/step - accuracy: 0.5278 - loss: 1.5493 - val_accuracy: 0.5312 - val_loss: 1.6145
Epoch 5/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.6104 - loss: 1.5069 - val_accuracy: 0.5625 - val_loss: 1.2087
Epoch 6/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.4942 - loss: 1.5548 - val_accuracy: 0.5938 - val_loss: 1.3542
Epoch 7/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 480s 63s/step - accuracy: 0.6169 - loss: 1.4705 - val_accuracy: 0.5469 - val_loss: 1.4762
Epoch 8/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 473s 69s/step - accuracy: 0.5223 - loss: 1.5237 - val_accuracy: 0.5312 - val_loss: 1.4541
Epoch 9/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.5257 - loss: 1.5129 - val_accuracy: 0.6094 - val_loss: 1.2638
Epoch 10/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 438s 62s/step - accuracy: 0.6196 - loss: 1.3614 - val_accuracy: 0.6406 - val_loss: 1.1630
Epoch 11/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.6252 - loss: 1.2998 - val_accuracy: 0.6406 - val_loss: 1.1506
Epoch 12/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.5932 - loss: 1.3225 - val_accuracy: 0.6875 - val_loss: 1.0736
Epoch 13/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 75s/step - accuracy: 0.6233 - loss: 1.3270 - val_accuracy: 0.7344 - val_loss: 0.9664
Epoch 14/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 465s 67s/step - accuracy: 0.6618 - loss: 1.3277 - val_accuracy: 0.6562 - val_loss: 1.0499
Epoch 15/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 74s/step - accuracy: 0.7299 - loss: 1.0386 - val_accuracy: 0.7188 - val_loss: 0.8890
Epoch 16/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.6709 - loss: 1.2410 - val_accuracy: 0.7188 - val_loss: 1.0808
Epoch 17/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.6817 - loss: 1.1629 - val_accuracy: 0.6562 - val_loss: 1.0311
Epoch 18/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 431s 67s/step - accuracy: 0.6430 - loss: 1.0337 - val_accuracy: 0.7344 - val_loss: 0.8879
Epoch 19/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 490s 67s/step - accuracy: 0.7705 - loss: 0.8687 - val_accuracy: 0.7812 - val_loss: 0.8805
Epoch 20/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 462s 67s/step - accuracy: 0.6845 - loss: 1.0979 - val_accuracy: 0.6562 - val_loss: 1.0892
Training Accuracy: 0.7009
Validation Accuracy: 0.6562
```

# Model Validation and Evaluation Report (5 marks)

| Model | Summary |
| --- | --- |
| VGG16 | |

```
from tensorflow.keras.applications.vgg16 import VGG16,preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Flatten,Input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
# from tensorflow.keras.optimizers import Adam
# from tensorflow.keras import regularizers


# base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
vgg = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
for layer in vgg.layers:
    layer.trainable = False

x = Flatten()(vgg.output)
# x = Dense(256, activation='relu', kernel_regularizer=regularizers.l2(0.001))(x)
# x = Dropout(0.5)(x)
predictions = Dense(17, activation='softmax')(x)

model = Model(inputs=vgg.input, outputs=predictions)



model.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Model: "functional"
```

| Layer (type) | Output Shape | Param # |
| --- | --- | --- |
| input_layer (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 |
| flatten (Flatten) | (None, 25088) | 0 |
| dense (Dense) | (None, 17) | 426,513 |

```
Total params: 15,141,201 (57.76 MB)
Trainable params: 426,513 (1.63 MB)
Non-trainable params: 14,714,688 (56.13 MB)
```

# Training and Validation Performance Metrics

```python
from tensorflow.keras.callbacks import EarlyStopping
# from sklearn.utils import class_weight

early_stop = EarlyStopping(
    monitor='val_accuracy',
    mode= 'max',
    patience=3,
    restore_best_weights=True
)

history =model.fit(train_set,validation_data=test_set, epochs=30, steps_per_epoch = len(train_set)//3, validation_steps = len(test_set)//3)
# Get final epoch's training and validation accuracy
train_acc = history.history['accuracy'][-1]
val_acc = history.history['val_accuracy'][-1]

# Print formatted values without scientific notation
print("Training Accuracy: {:.4f}".format(train_acc))
print("Validation Accuracy: {:.4f}".format(val_acc))
```

```
Epoch 1/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 484s 70s/step - accuracy: 0.4922 - loss: 1.6717 - val_accuracy: 0.4688 - val_loss: 1.8478
Epoch 2/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.4300 - loss: 1.9040 - val_accuracy: 0.4531 - val_loss: 1.7980
Epoch 3/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 480s 69s/step - accuracy: 0.4987 - loss: 1.7352 - val_accuracy: 0.6250 - val_loss: 1.2784
Epoch 4/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 537s 70s/step - accuracy: 0.5278 - loss: 1.5493 - val_accuracy: 0.5312 - val_loss: 1.6145
Epoch 5/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.6104 - loss: 1.5069 - val_accuracy: 0.5625 - val_loss: 1.2087
Epoch 6/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.4942 - loss: 1.5548 - val_accuracy: 0.5938 - val_loss: 1.3542
Epoch 7/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 480s 63s/step - accuracy: 0.6169 - loss: 1.4705 - val_accuracy: 0.5469 - val_loss: 1.4762
Epoch 8/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 473s 69s/step - accuracy: 0.5223 - loss: 1.5237 - val_accuracy: 0.5312 - val_loss: 1.4541
Epoch 9/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.5257 - loss: 1.5129 - val_accuracy: 0.6094 - val_loss: 1.2638
Epoch 10/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 438s 62s/step - accuracy: 0.6196 - loss: 1.3614 - val_accuracy: 0.6406 - val_loss: 1.1630
Epoch 11/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.6252 - loss: 1.2998 - val_accuracy: 0.6406 - val_loss: 1.1506
Epoch 12/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.5932 - loss: 1.3225 - val_accuracy: 0.6875 - val_loss: 1.0736
Epoch 13/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 75s/step - accuracy: 0.6233 - loss: 1.3270 - val_accuracy: 0.7344 - val_loss: 0.9664
Epoch 14/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 465s 67s/step - accuracy: 0.6618 - loss: 1.3277 - val_accuracy: 0.6562 - val_loss: 1.0499
Epoch 15/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 74s/step - accuracy: 0.7299 - loss: 1.0386 - val_accuracy: 0.7188 - val_loss: 0.8890
Epoch 16/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 502s 73s/step - accuracy: 0.6709 - loss: 1.2410 - val_accuracy: 0.7188 - val_loss: 1.0808
Epoch 17/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 501s 73s/step - accuracy: 0.6817 - loss: 1.1629 - val_accuracy: 0.6562 - val_loss: 1.0311
Epoch 18/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 431s 67s/step - accuracy: 0.6430 - loss: 1.0337 - val_accuracy: 0.7344 - val_loss: 0.8879
Epoch 19/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 490s 67s/step - accuracy: 0.7705 - loss: 0.8687 - val_accuracy: 0.7812 - val_loss: 0.8805
Epoch 20/20
7/7 ━━━━━━━━━━━━━━━━━━━━ 462s 67s/step - accuracy: 0.6845 - loss: 1.0979 - val_accuracy: 0.6562 - val_loss: 1.0892
Training Accuracy: 0.7009
Validation Accuracy: 0.6562
```