Model Development Phase Template

Date	1 JUNE 2025
Team ID	SWTID1749902640
Project Title	CRIME VISION: ADVANCED CRIME CLASSIFICATION 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.

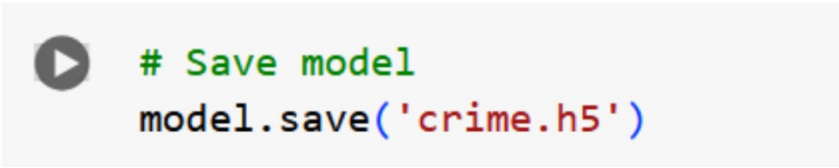**Initial Model Training Code (5 marks):**

Paste the screenshot of the model training code

```python
def transfer_learning():
    base_model = DenseNet121(include_top=False, input_shape=(*IMG_SHAPE, 3), weights="imagenet")
    base_model.trainable = False  # Freeze all layers
    return base_model
```

```python
def create_model():
    model = Sequential([
        Rescaling(1./255, input_shape=(*IMG_SHAPE, 3)),
        transfer_learning(),
        GlobalAveragePooling2D(),
        Dense(256, activation="relu"),
        Dropout(0.2),
        Dense(512, activation="relu"),
        Dropout(0.2),
        Dense(1024, activation="relu"),
        Dense(n, activation="softmax")
    ])
    model.summary()
    return model
```

```python
model = create_model()
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

```python
# Train the model
history = model.fit(x = train_set, validation_data = val_set,epochs = EPOCHS)
```

```
# Save model
model.save('crime.h5')
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1	```def create_model():
    model = Sequential([
        Rescaling(1./255, input_shape=(*IMG_SHAPE, 3)),
        transfer_learning(),
        GlobalAveragePooling2D(),
        Dense(256, activation="relu"),
        Dropout(0.2),
        Dense(512, activation="relu"),
        Dropout(0.2),
        Dense(1024, activation="relu"),
        Dense(n, activation="softmax")
    ])
    model.summary()
    return model

model = create_model()
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/densenet/densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29084464/29084464 ━━━ 0s 0us/step
Model: "sequential"

Layer (type) / Output Shape / Param #
rescaling (Rescaling) / (None, 60, 60, 3) / 0
densenet121 (Functional) / (None, 2, 2, 1024) / 7,037,504
global_average_pooling2d (GlobalAveragePooling2D) / (None, 1024) / 0
dense (Dense) / (None, 256) / 262,400
dropout (Dropout) / (None, 256) / 0
dense_1 (Dense) / (None, 512) / 131,584
dropout_1 (Dropout) / (None, 512) / 0
dense_2 (Dense) / (None, 1024) / 525,312
dense_3 (Dense) / (None, 50) / 51,350

Total params: 7,971,150 (30.41 MB)
Trainable params: 933,646 (3.56 MB)
Non-trainable params: 7,037,504 (26.85 MB) | ```# Train the model
history = model.fit(x = train_set, validation_data = val_set, epochs = EPOCHS)
```

Epoch 1/5
7915/7915 ━━━ 1301s 161ms/step - accuracy: 0.9211 - loss: 0.2912 - val_accuracy: 0.9858 - val_loss: 0.0537
Epoch 2/5
7915/7915 ━━━ 1155s 139ms/step - accuracy: 0.9772 - loss: 0.0819 - val_accuracy: 0.9918 - val_loss: 0.0329
Epoch 3/5
7915/7915 ━━━ 1128s 143ms/step - accuracy: 0.9823 - loss: 0.0641 - val_accuracy: 0.9927 - val_loss: 0.0272
Epoch 4/5
7915/7915 ━━━ 1128s 138ms/step - accuracy: 0.9845 - loss: 0.0570 - val_accuracy: 0.9933 - val_loss: 0.0260
Epoch 5/5
7915/7915 ━━━ 1102s 139ms/step - accuracy: 0.9862 - loss: 0.0524 - val_accuracy: 0.9938 - val_loss: 0.0234

+ Code + Text

```# Save model
model.save('crime.h5')
``` |