

Model Development Phase Template

Date	1 JUNE 2025
Team ID	xxxxxxx
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

```
[ ] 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
```

```
▶ 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"])
```

```
▶  
# 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	<pre>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(0, activation="softmax")]) model.summary() return model model = create_model() model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])</pre> <div>Download data from https://storage.googleapis.com/tensorflow/tf-datasets/dataset_info/tf_datasets_model_01 Model: 'sequential'</div> <table><thead><tr><th>Layer (type)</th><th>Input Shape</th><th>Param #</th></tr></thead><tbody><tr><td>Rescaling (Rescaling)</td><td>(None, 28, 28, 3)</td><td>0</td></tr><tr><td>conv2d1 (Conv2D)</td><td>(None, 2, 2, 32x32)</td><td>7,392,384</td></tr><tr><td>global_average_pooling2d (GlobalAveragePooling2D)</td><td>(None, 32x32)</td><td>0</td></tr><tr><td>Dense (Dense)</td><td>(None, 256)</td><td>702,400</td></tr><tr><td>Dropout (Dropout)</td><td>(None, 256)</td><td>0</td></tr><tr><td>Dense_1 (Dense)</td><td>(None, 512)</td><td>131,008</td></tr><tr><td>Dropout_1 (Dropout)</td><td>(None, 512)</td><td>0</td></tr><tr><td>Dense_2 (Dense)</td><td>(None, 1024)</td><td>1,048,320</td></tr><tr><td>Dense_3 (Dense)</td><td>(None, 10)</td><td>10,100</td></tr></tbody></table> <div>Total params: 9,151,104 (35.45 MB) Trainable params: 8,151,104 (31.45 MB) Non-trainable params: 1,000,000 (3.99 MB)</div>	Layer (type)	Input Shape	Param #	Rescaling (Rescaling)	(None, 28, 28, 3)	0	conv2d1 (Conv2D)	(None, 2, 2, 32x32)	7,392,384	global_average_pooling2d (GlobalAveragePooling2D)	(None, 32x32)	0	Dense (Dense)	(None, 256)	702,400	Dropout (Dropout)	(None, 256)	0	Dense_1 (Dense)	(None, 512)	131,008	Dropout_1 (Dropout)	(None, 512)	0	Dense_2 (Dense)	(None, 1024)	1,048,320	Dense_3 (Dense)	(None, 10)	10,100	<div># Train the model History = model.fit(x=train_set, validation_data=val_set, epochs=EPOCHS)</div> <div>Epoch 1/5 7915/7915 ----- 1381s 165ms/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: 0.0829 - 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: 0.0268 Epoch 5/5 7915/7915 ----- 1182s 139ms/step - accuracy: 0.9862 - loss: 0.0524 - val_accuracy: 0.9938 - val_loss: 0.0234</div> <div># Save model model.save('crime.h5')</div>
Layer (type)	Input Shape	Param #																														
Rescaling (Rescaling)	(None, 28, 28, 3)	0																														
conv2d1 (Conv2D)	(None, 2, 2, 32x32)	7,392,384																														
global_average_pooling2d (GlobalAveragePooling2D)	(None, 32x32)	0																														
Dense (Dense)	(None, 256)	702,400																														
Dropout (Dropout)	(None, 256)	0																														
Dense_1 (Dense)	(None, 512)	131,008																														
Dropout_1 (Dropout)	(None, 512)	0																														
Dense_2 (Dense)	(None, 1024)	1,048,320																														
Dense_3 (Dense)	(None, 10)	10,100																														