Model Development Phase Template

Date	11-03-2025
Team ID	740047
Project Title	AI-POWERED VEHICLE DAMAGE ASSESSMENT FOR COST ESTIMATION AND INSURANCE CLAIMS.
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot.

Initial Model Training Code:

```
[ ] #create a model object
    model = Model(inputs=vgg.input, outputs=prediction)

    #view the structure of model
    model.summary()
```

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_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, 4)	100,356

Total params: 14,815,044 (56.51 MB)
Trainable params: 100,356 (392.02 KB)
Non-trainable params: 14,714,688 (56.13 MB)

Compiling the model

```
[ ]  #compiling the cnn model
     model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
```

```
[ ]  # Compile the model with sparse_categorical_crossentropy
     model.compile(
         optimizer='adam',
         loss='sparse_categorical_crossentropy',
         metrics=['accuracy']
     )
```

Fitting the model

```
import sys
# Fit the model
r = model.fit(
    training_set,
    validation_data=test_set,
    epochs=25,
    steps_per_epoch = 979//10,
    validation_steps=171//10
)
```