

Model Development Phase Template

Date	16 June 2025
Team Lead Name	Jayanth Srinivas Bommisetty
Project Title	Sloan Digital Sky Survey (SDSS) galaxy classification using machine learning
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. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
CNN_one = keras.models.Sequential()

# first conv and max layer + dropout layer to prevent overfitting
CNN_one.add(Conv2D(filters=32, kernel_size=3, activation="relu", input_shape=[256,256,3]))
CNN_one.add(MaxPool2D(pool_size=2, strides=2))
CNN_one.add(BatchNormalization())
CNN_one.add(Dropout(0.5))

# second conv and max layer + dropout layer
CNN_one.add(Conv2D(filters=64, kernel_size=3, activation="relu", kernel_regularizer="l2"))
CNN_one.add(MaxPool2D(pool_size=2, strides=2))
CNN_one.add(BatchNormalization())
CNN_one.add(Dropout(0.2))

# third conv and max layer + dropout layer
CNN_one.add(Conv2D(filters=128, kernel_size=3, activation="relu", kernel_regularizer="l2"))
CNN_one.add(MaxPool2D(pool_size=2, strides=2))
CNN_one.add(BatchNormalization())
CNN_one.add(Dropout(0.2))

# forth conv and max layer + dropout layer
CNN_one.add(Conv2D(filters=256, kernel_size=3, activation="relu", kernel_regularizer="l2"))
CNN_one.add(MaxPool2D(pool_size=2, strides=2))
CNN_one.add(BatchNormalization())
CNN_one.add(Dropout(0.2))

# flattening and dense layers
CNN_one.add(Flatten())
CNN_one.add(Dense(units=512, activation="relu"))
CNN_one.add(Dense(units=5, activation="softmax"))

CNN_one.summary()
```

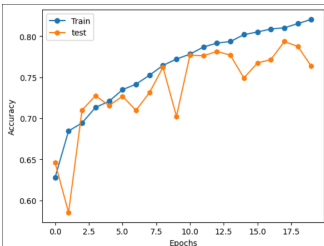
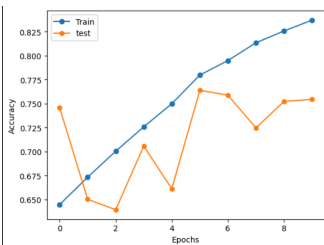
```
# Training the model
CNN_one.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
history1 = CNN_one.fit(x=training_data, validation_data = validation_data, epochs=20)
```

```
vgg16 = keras.applications.vgg16.VGG16(weights = "imagenet", input_shape=(256,256,3), include_top=False)
vgg16.trainable = False # Freezing convolutional layers
vgg16.summary()
```

```
vgg16_model2.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
```

```
history2 = vgg16_model2.fit(training_data,epochs=10, validation_data=validation_data)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix																																																												
CNN	<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d (conv2d)</td><td>(None, 10, 10, 32)</td><td>960</td></tr><tr><td>max_pooling2d (maxpooling2d)</td><td>(None, 5, 5, 32)</td><td>0</td></tr><tr><td>batch_normalization (batchnormalization)</td><td>(None, 5, 5, 32)</td><td>128</td></tr><tr><td>dropout (dropout)</td><td>(None, 5, 5, 32)</td><td>0</td></tr><tr><td>conv2d_1 (conv2d)</td><td>(None, 5, 5, 32)</td><td>960</td></tr><tr><td>max_pooling2d_1 (maxpooling2d)</td><td>(None, 2, 2, 32)</td><td>0</td></tr><tr><td>batch_normalization_1 (batchnormalization)</td><td>(None, 2, 2, 32)</td><td>128</td></tr><tr><td>dropout_1 (dropout)</td><td>(None, 2, 2, 32)</td><td>0</td></tr><tr><td>conv2d_2 (conv2d)</td><td>(None, 2, 2, 32)</td><td>960</td></tr><tr><td>max_pooling2d_2 (maxpooling2d)</td><td>(None, 1, 1, 32)</td><td>0</td></tr><tr><td>batch_normalization_2 (batchnormalization)</td><td>(None, 1, 1, 32)</td><td>128</td></tr><tr><td>dropout_2 (dropout)</td><td>(None, 1, 1, 32)</td><td>0</td></tr><tr><td>conv2d_3 (conv2d)</td><td>(None, 1, 1, 32)</td><td>960</td></tr><tr><td>max_pooling2d_3 (maxpooling2d)</td><td>(None, 1, 1, 32)</td><td>0</td></tr><tr><td>batch_normalization_3 (batchnormalization)</td><td>(None, 1, 1, 32)</td><td>128</td></tr><tr><td>dropout_3 (dropout)</td><td>(None, 1, 1, 32)</td><td>0</td></tr><tr><td>flatten (flatten)</td><td>(None, 1024)</td><td>0</td></tr><tr><td>dense (dense)</td><td>(None, 1)</td><td>10,480,032</td></tr><tr><td>dense_1 (dense)</td><td>(None, 1)</td><td>1,024</td></tr></table>	Layer (type)	Output Shape	Param #	conv2d (conv2d)	(None, 10, 10, 32)	960	max_pooling2d (maxpooling2d)	(None, 5, 5, 32)	0	batch_normalization (batchnormalization)	(None, 5, 5, 32)	128	dropout (dropout)	(None, 5, 5, 32)	0	conv2d_1 (conv2d)	(None, 5, 5, 32)	960	max_pooling2d_1 (maxpooling2d)	(None, 2, 2, 32)	0	batch_normalization_1 (batchnormalization)	(None, 2, 2, 32)	128	dropout_1 (dropout)	(None, 2, 2, 32)	0	conv2d_2 (conv2d)	(None, 2, 2, 32)	960	max_pooling2d_2 (maxpooling2d)	(None, 1, 1, 32)	0	batch_normalization_2 (batchnormalization)	(None, 1, 1, 32)	128	dropout_2 (dropout)	(None, 1, 1, 32)	0	conv2d_3 (conv2d)	(None, 1, 1, 32)	960	max_pooling2d_3 (maxpooling2d)	(None, 1, 1, 32)	0	batch_normalization_3 (batchnormalization)	(None, 1, 1, 32)	128	dropout_3 (dropout)	(None, 1, 1, 32)	0	flatten (flatten)	(None, 1024)	0	dense (dense)	(None, 1)	10,480,032	dense_1 (dense)	(None, 1)	1,024	0.7660804986953735	
Layer (type)	Output Shape	Param #																																																													
conv2d (conv2d)	(None, 10, 10, 32)	960																																																													
max_pooling2d (maxpooling2d)	(None, 5, 5, 32)	0																																																													
batch_normalization (batchnormalization)	(None, 5, 5, 32)	128																																																													
dropout (dropout)	(None, 5, 5, 32)	0																																																													
conv2d_1 (conv2d)	(None, 5, 5, 32)	960																																																													
max_pooling2d_1 (maxpooling2d)	(None, 2, 2, 32)	0																																																													
batch_normalization_1 (batchnormalization)	(None, 2, 2, 32)	128																																																													
dropout_1 (dropout)	(None, 2, 2, 32)	0																																																													
conv2d_2 (conv2d)	(None, 2, 2, 32)	960																																																													
max_pooling2d_2 (maxpooling2d)	(None, 1, 1, 32)	0																																																													
batch_normalization_2 (batchnormalization)	(None, 1, 1, 32)	128																																																													
dropout_2 (dropout)	(None, 1, 1, 32)	0																																																													
conv2d_3 (conv2d)	(None, 1, 1, 32)	960																																																													
max_pooling2d_3 (maxpooling2d)	(None, 1, 1, 32)	0																																																													
batch_normalization_3 (batchnormalization)	(None, 1, 1, 32)	128																																																													
dropout_3 (dropout)	(None, 1, 1, 32)	0																																																													
flatten (flatten)	(None, 1024)	0																																																													
dense (dense)	(None, 1)	10,480,032																																																													
dense_1 (dense)	(None, 1)	1,024																																																													
VGG16	<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>input_layer_1 (inputlayer)</td><td>(None, 256, 256, 3)</td><td>0</td></tr><tr><td>block1_conv1 (conv2d)</td><td>(None, 128, 128, 3)</td><td>1,152</td></tr><tr><td>block1_conv2 (conv2d)</td><td>(None, 128, 128, 3)</td><td>96,768</td></tr><tr><td>block1_pool (maxpooling2d)</td><td>(None, 64, 64, 3)</td><td>0</td></tr><tr><td>block2_conv1 (conv2d)</td><td>(None, 64, 64, 3)</td><td>71,680</td></tr><tr><td>block2_conv2 (conv2d)</td><td>(None, 64, 64, 3)</td><td>541,760</td></tr><tr><td>block2_pool (maxpooling2d)</td><td>(None, 32, 32, 3)</td><td>0</td></tr><tr><td>block3_conv1 (conv2d)</td><td>(None, 32, 32, 3)</td><td>605,184</td></tr><tr><td>block3_conv2 (conv2d)</td><td>(None, 32, 32, 3)</td><td>588,800</td></tr><tr><td>block3_conv3 (conv2d)</td><td>(None, 32, 32, 3)</td><td>588,800</td></tr><tr><td>block3_pool (maxpooling2d)</td><td>(None, 16, 16, 3)</td><td>0</td></tr><tr><td>block4_conv1 (conv2d)</td><td>(None, 16, 16, 3)</td><td>1,088,000</td></tr><tr><td>block4_conv2 (conv2d)</td><td>(None, 16, 16, 3)</td><td>1,088,000</td></tr><tr><td>block4_conv3 (conv2d)</td><td>(None, 16, 16, 3)</td><td>1,088,000</td></tr><tr><td>block4_pool (maxpooling2d)</td><td>(None, 8, 8, 3)</td><td>0</td></tr><tr><td>block5_conv1 (conv2d)</td><td>(None, 8, 8, 3)</td><td>1,536,000</td></tr><tr><td>block5_conv2 (conv2d)</td><td>(None, 8, 8, 3)</td><td>1,536,000</td></tr><tr><td>block5_conv3 (conv2d)</td><td>(None, 8, 8, 3)</td><td>1,536,000</td></tr><tr><td>block5_pool (maxpooling2d)</td><td>(None, 4, 4, 3)</td><td>0</td></tr></table>	Layer (type)	Output Shape	Param #	input_layer_1 (inputlayer)	(None, 256, 256, 3)	0	block1_conv1 (conv2d)	(None, 128, 128, 3)	1,152	block1_conv2 (conv2d)	(None, 128, 128, 3)	96,768	block1_pool (maxpooling2d)	(None, 64, 64, 3)	0	block2_conv1 (conv2d)	(None, 64, 64, 3)	71,680	block2_conv2 (conv2d)	(None, 64, 64, 3)	541,760	block2_pool (maxpooling2d)	(None, 32, 32, 3)	0	block3_conv1 (conv2d)	(None, 32, 32, 3)	605,184	block3_conv2 (conv2d)	(None, 32, 32, 3)	588,800	block3_conv3 (conv2d)	(None, 32, 32, 3)	588,800	block3_pool (maxpooling2d)	(None, 16, 16, 3)	0	block4_conv1 (conv2d)	(None, 16, 16, 3)	1,088,000	block4_conv2 (conv2d)	(None, 16, 16, 3)	1,088,000	block4_conv3 (conv2d)	(None, 16, 16, 3)	1,088,000	block4_pool (maxpooling2d)	(None, 8, 8, 3)	0	block5_conv1 (conv2d)	(None, 8, 8, 3)	1,536,000	block5_conv2 (conv2d)	(None, 8, 8, 3)	1,536,000	block5_conv3 (conv2d)	(None, 8, 8, 3)	1,536,000	block5_pool (maxpooling2d)	(None, 4, 4, 3)	0	0.294928752754983	
Layer (type)	Output Shape	Param #																																																													
input_layer_1 (inputlayer)	(None, 256, 256, 3)	0																																																													
block1_conv1 (conv2d)	(None, 128, 128, 3)	1,152																																																													
block1_conv2 (conv2d)	(None, 128, 128, 3)	96,768																																																													
block1_pool (maxpooling2d)	(None, 64, 64, 3)	0																																																													
block2_conv1 (conv2d)	(None, 64, 64, 3)	71,680																																																													
block2_conv2 (conv2d)	(None, 64, 64, 3)	541,760																																																													
block2_pool (maxpooling2d)	(None, 32, 32, 3)	0																																																													
block3_conv1 (conv2d)	(None, 32, 32, 3)	605,184																																																													
block3_conv2 (conv2d)	(None, 32, 32, 3)	588,800																																																													
block3_conv3 (conv2d)	(None, 32, 32, 3)	588,800																																																													
block3_pool (maxpooling2d)	(None, 16, 16, 3)	0																																																													
block4_conv1 (conv2d)	(None, 16, 16, 3)	1,088,000																																																													
block4_conv2 (conv2d)	(None, 16, 16, 3)	1,088,000																																																													
block4_conv3 (conv2d)	(None, 16, 16, 3)	1,088,000																																																													
block4_pool (maxpooling2d)	(None, 8, 8, 3)	0																																																													
block5_conv1 (conv2d)	(None, 8, 8, 3)	1,536,000																																																													
block5_conv2 (conv2d)	(None, 8, 8, 3)	1,536,000																																																													
block5_conv3 (conv2d)	(None, 8, 8, 3)	1,536,000																																																													
block5_pool (maxpooling2d)	(None, 4, 4, 3)	0																																																													