Model Optimization and Tuning Phase

Date	7 Feb 2026
Student Name	Aarya Sunil Dalal
Project Title	Uncovering the Hidden Treasures of the Mushroom Kingdom A Classification Analysis
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves improving our neural network models to get the best results. This means adjusting the model's settings, comparing how well different settings work, and explaining why we chose our final model.

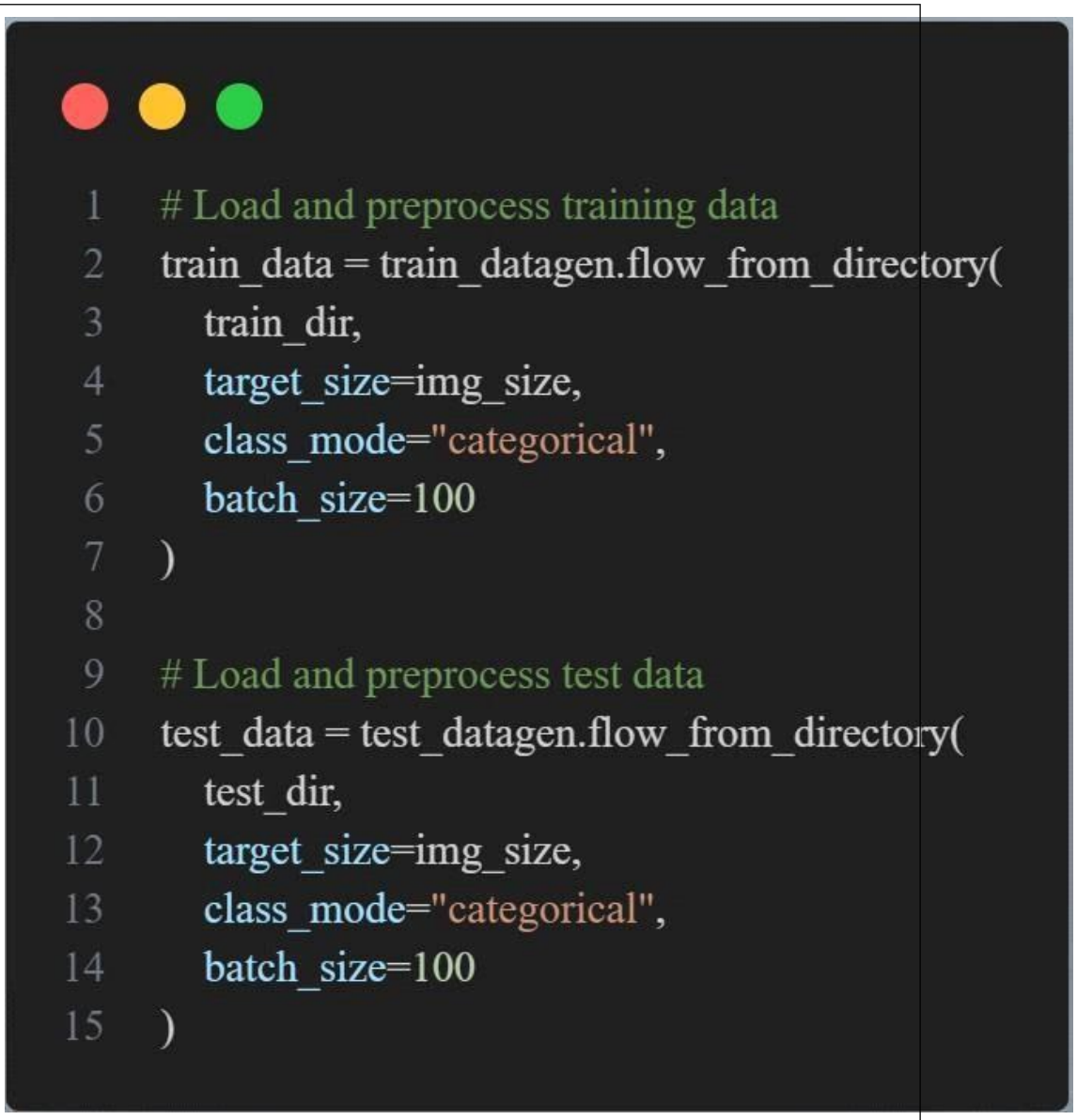
The neural network models were trained to classify mushroom images into the following three classes: Boletus, Lactarius, and Russula. The training dataset consisted of 911 labeled mushroom images across the three target classes. A separate dataset of 292 images was used for validation and final evaluation of the models.

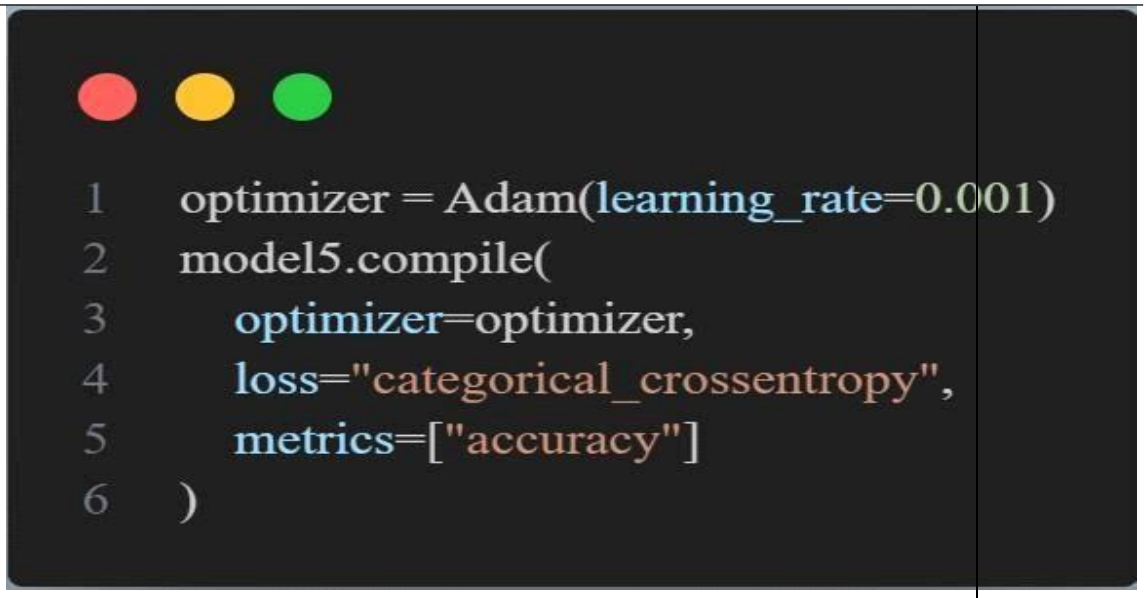
Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
Model 1: Inception V3 (Baseline)	Learning Rate: We adjusted the learning rate, which controls how much the model learns from its mistakes. We tried different learning rates to find one that helps the model learn effectively without becoming unstable.

```
optimizer = Adam(learning_rate=0.001)
model5.compile(
    optimizer=optimizer,
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
```

	Batch Size: We changed the batch size, which is the number of images the model processes at once before updating its knowledge. We tested different batch sizes to balance speed and memory usage.

	
	Learning Rate: We made finer adjustments to the learning rate, building on what we learned from Model 1, to see if we could improve performance further.

Model 2: Inception V3 (Optimized)	 Batch Size: We used the best batch size from Model 1.

```
optimizer = Adam(learning_rate=0.001)
model5.compile(
    optimizer=optimizer,
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
```

```
# Load and preprocess training data
train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)

# Load and preprocess test data
test_data = test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)
```

Accuracy:

```
...  Found 911 images belonging to 3 classes.
     Found 292 images belonging to 3 classes.
     Epoch 1/50
     10/10 [==============================] - 409s 44s/step - loss: 1.4139 - accuracy: 0.4577 - val_loss: 1.2089 - val_accuracy: 0.2808
     Epoch 2/50
     10/10 [==============================] - 24s 2s/step - loss: 0.9628 - accuracy: 0.6081 - val_loss: 1.3015 - val_accuracy: 0.3219
     Epoch 3/50
     10/10 [==============================] - 25s 3s/step - loss: 0.8397 - accuracy: 0.6773 - val_loss: 1.2984 - val_accuracy: 0.3938
     Epoch 4/50
     10/10 [==============================] - 23s 2s/step - loss: 0.7214 - accuracy: 0.7080 - val_loss: 0.8995 - val_accuracy: 0.5445
     Epoch 5/50
     10/10 [==============================] - 24s 2s/step - loss: 0.6653 - accuracy: 0.7333 - val_loss: 0.7398 - val_accuracy: 0.6096
     Epoch 6/50
     10/10 [==============================] - 24s 2s/step - loss: 0.6067 - accuracy: 0.7673 - val_loss: 0.6160 - val_accuracy: 0.7055
     Epoch 7/50
     10/10 [==============================] - 24s 2s/step - loss: 0.6118 - accuracy: 0.7453 - val_loss: 0.5257 - val_accuracy: 0.7500
     Epoch 8/50
     10/10 [==============================] - 24s 2s/step - loss: 0.5409 - accuracy: 0.7794 - val_loss: 0.4759 - val_accuracy: 0.7979
     Epoch 9/50
     10/10 [==============================] - 24s 2s/step - loss: 0.5510 - accuracy: 0.7629 - val_loss: 0.4671 - val_accuracy: 0.7877
     Epoch 10/50
     10/10 [==============================] - 23s 2s/step - loss: 0.5515 - accuracy: 0.7783 - val_loss: 0.3937 - val_accuracy: 0.8356
     Epoch 11/50
     10/10 [==============================] - 25s 3s/step - loss: 0.5000 - accuracy: 0.7859 - val_loss: 0.3540 - val_accuracy: 0.8699
     Epoch 12/50
     ...
     10/10 [==============================] - 24s 2s/step - loss: 0.3308 - accuracy: 0.8573 - val_loss: 0.2649 - val_accuracy: 0.8836
     3/3 [==============================] - 3s 1s/step - loss: 0.2649 - accuracy: 0.8836
     Test loss: 0.26492103934288025
     Test accuracy: 88.35616707801819
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning