Model Development Phase Template

Date	14 December 2024
Team ID	739696
Project Title	Plant seedling Classification using Deep 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):

```python
# Set the number of steps per epoch and validation steps
steps_per_epoch = len(training_set) // 32  # 32 is batch_size
validation_steps = len(test_dataset) // 32  # 32 is batch_size

# Start model training
r = model.fit(
    training_set,
    validation_data=test_dataset,
    epochs=10,
    steps_per_epoch=steps_per_epoch,
    validation_steps=validation_steps
)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics

| CNN model (Base Model) | ```python
# Add custom layers on top of the pre-trained model
X = Flatten()(xception.output)  # Flatten the feature maps
prediction = Dense(12, activation='softmax')(X)  # Final dense layer for classification

# Create the complete model
model = Model(inputs=xception.input, outputs=prediction)

# Print the model summary
model.summary()

# Compile the model
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
``` | ```
Epoch 1/10
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its
  self._warn_if_super_not_called()
4/4 ━━━━━━━━ 948s 248s/step - accuracy: 0.1511 - loss: 9.4482 - val_accuracy: 0.0869 - val_loss: 19.6136
Epoch 2/10
4/4 ━━━━━━━━ 53s 18s/step - accuracy: 0.3125 - loss: 11.3158 - val_accuracy: 0.0819 - val_loss: 17.8900
Epoch 3/10
4/4 ━━━━━━━━ 91s 25s/step - accuracy: 0.3187 - loss: 9.1201 - val_accuracy: 0.0781 - val_loss: 18.3931
Epoch 4/10
4/4 ━━━━━━━━ 91s 16s/step - accuracy: 0.4281 - loss: 7.3380 - val_accuracy: 0.0844 - val_loss: 24.8223
Epoch 5/10
4/4 ━━━━━━━━ 87s 24s/step - accuracy: 0.4096 - loss: 7.5730 - val_accuracy: 0.0819 - val_loss: 25.4974
Epoch 6/10
4/4 ━━━━━━━━ 76s 22s/step - accuracy: 0.3542 - loss: 8.7956 - val_accuracy: 0.0819 - val_loss: 23.7906
Epoch 7/10
4/4 ━━━━━━━━ 88s 24s/step - accuracy: 0.4698 - loss: 6.4703 - val_accuracy: 0.0919 - val_loss: 25.7789
Epoch 8/10
4/4 ━━━━━━━━ 88s 23s/step - accuracy: 0.5580 - loss: 6.1439 - val_accuracy: 0.0630 - val_loss: 29.2279
Epoch 9/10
4/4 ━━━━━━━━ 82s 23s/step - accuracy: 0.5719 - loss: 5.4253 - val_accuracy: 0.0705 - val_loss: 29.1835
Epoch 10/10
4/4 ━━━━━━━━ 68s 26s/step - accuracy: 0.5656 - loss: 6.1409 - val_accuracy: 0.0768 - val_loss: 29.5912
``` |
| Fine Tuning | ```python
# Unfreeze some layers in the Xception model for fine-tuning
for layer in xception.layers[-20:]:  # Example: Unfreeze the last 20 layers
    layer.trainable = True

# Recompile the model with a lower learning rate
model.compile(
    loss='categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),  # Reduced learning rate
    metrics=['accuracy']
)

# Fine-tune the model
r_fine_tune = model.fit(
    training_set,
    validation_data=test_dataset,
    epochs=5,  # Fine-tune for fewer epochs
    steps_per_epoch=steps_per_epoch,
    validation_steps=validation_steps
)
``` | ```
Epoch 1/5
4/4 ━━━━━━━━ 219s 19s/step - accuracy: 0.3021 - loss: 11.8456 - val_accuracy: 0.0781 - val_loss: 29.8364
Epoch 2/5
4/4 ━━━━━━━━ 78s 23s/step - accuracy: 0.2896 - loss: 11.5190 - val_accuracy: 0.0768 - val_loss: 30.0466
Epoch 3/5
4/4 ━━━━━━━━ 65s 20s/step - accuracy: 0.4115 - loss: 7.9222 - val_accuracy: 0.0756 - val_loss: 30.3446
Epoch 4/5
4/4 ━━━━━━━━ 59s 17s/step - accuracy: 0.3219 - loss: 8.5186 - val_accuracy: 0.0756 - val_loss: 30.2222
Epoch 5/5
4/4 ━━━━━━━━ 65s 20s/step - accuracy: 0.4698 - loss: 7.6808 - val_accuracy: 0.0730 - val_loss: 30.5238
```
+ Code   + Text |