Skill-12

2100030910

Sec-23

Main.py

```
import numpy as np
import pandas as pd
import os
import model7 as mc

train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train'
valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid'

cloud_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\data\cloudy'
cloud_train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train\cloudy'
cloud_valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid\cloudy'

water_directory = r'C:\Users\dell\PycharmProjex\deepLearn\Skill\data\water'
water_train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train\water'
water_valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid\water'

green_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\data\green_area'
green_train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train\green'
green_valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid\green'

desert_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\data\desert'
desert_train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train\desert'
desert_valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid\desert'

cloud_image_files = [f for f in os.listdir(cloud_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]


water_image_files = [f for f in os.listdir(water_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]
```

```
desert_image_files = [f for f in os.listdir(desert_directory) if
                      f.lower().endswith(('.jpg', '.jpeg'))]


green_image_files = [f for f in os.listdir(green_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]



os.makedirs(train_directory, exist_ok=True)
os.makedirs(valid_directory, exist_ok=True)


import os
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
BatchNormalization, Dropout

# Directories
train_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\train'
valid_directory =
r'C:\Users\dell\PycharmProjex\deepLearn\Skill\genData\valid'

# Image data preprocessing
train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1.0 / 255)

batch_size = 32

train_generator = train_datagen.flow_from_directory(
    train_directory,
    target_size=(224, 224),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

validation_generator = validation_datagen.flow_from_directory(
    valid_directory,
    target_size=(224, 224),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

# Sample CNN model
```

```
mc2=mc.Modelsc()
model2=mc2.Lstm()

# Display model summary
model2.summary()

# Training the model
epochs = 4

history = model2.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)

# Plotting accuracy and loss curves
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```

model.py

```
import tensorflow as tf
from keras import Sequential, Input, Model
from keras.src.applications import VGG16
from keras.src.layers import Flatten, Dense, Conv2D, MaxPool2D, Dropout,
SimpleRNN, Reshape, LSTM
from tensorflow import keras
from tensorflow.keras import layers
from keras.applications.vgg16 import VGG16, preprocess_input
class Modelsc:
    def Lstm(self):
        input_shape = (224, 224, 3)
        model=Sequential()

model.add(Reshape((input_shape[0]*input_shape[1],input_shape[2]),input_shap
e=input_shape))
        model.add(LSTM(16))
        model.add(Dense(256, activation='relu'))
        model.add(Dense(128, activation='relu'))
        model.add(Dense(4, activation='softmax'))
        model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
        return model
```

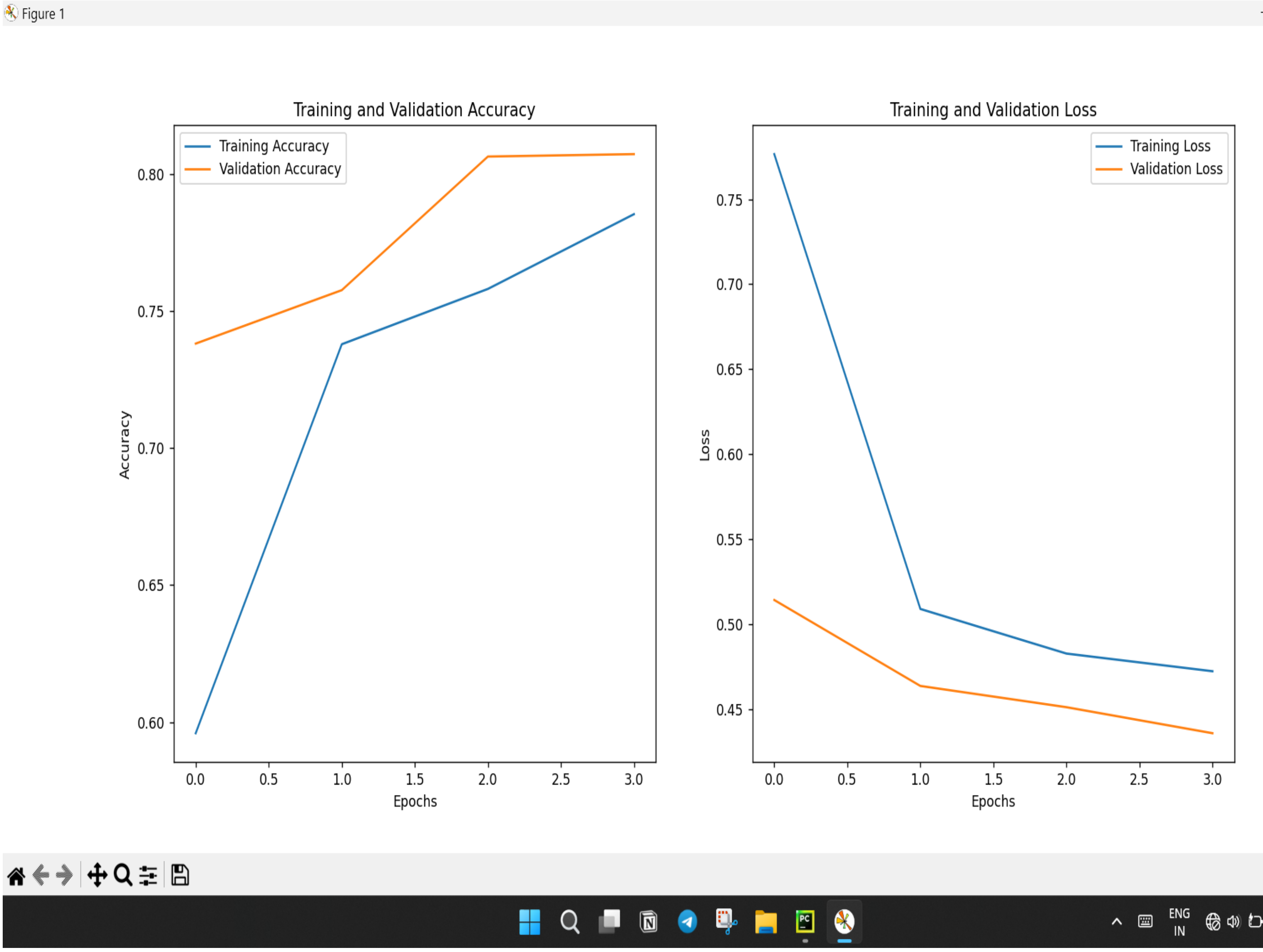
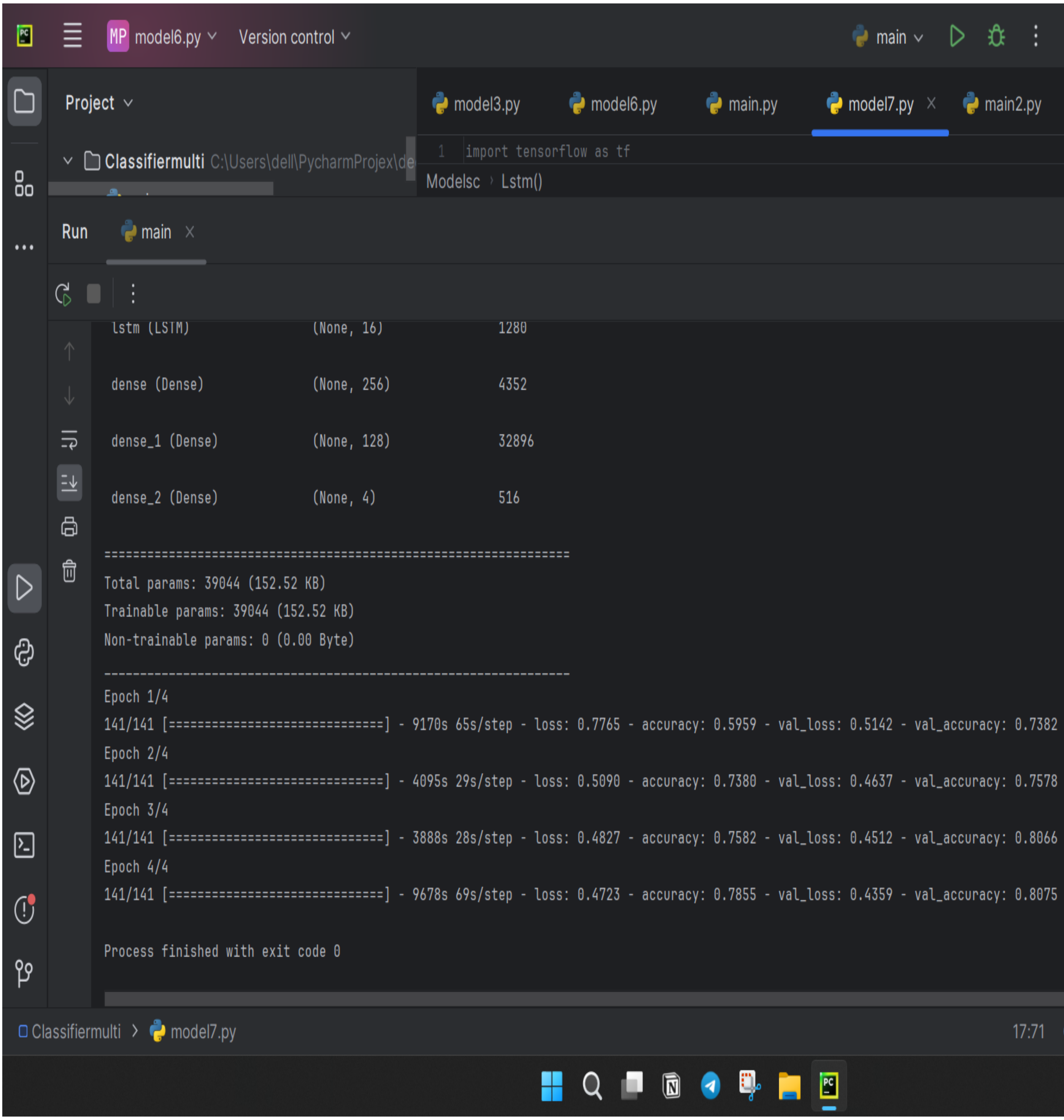


Figure 1
Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.80
0.75
0.70
0.65
0.60
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Epochs
Training and Validation Loss
Training Loss
Validation Loss
Loss
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Epochs
ENG
IN



```
import tensorflow as tf
```

```
 lstm (LSTM)                 (None, 16)                1280

 dense (Dense)               (None, 256)               4352

 dense_1 (Dense)             (None, 128)               32896

 dense_2 (Dense)             (None, 4)                 516

=================================================================
Total params: 39044 (152.52 KB)
Trainable params: 39044 (152.52 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/4
141/141 [==============================] - 9170s 65s/step - loss: 0.7765 - accuracy: 0.5959 - val_loss: 0.5142 - val_accuracy: 0.7382
Epoch 2/4
141/141 [==============================] - 4095s 29s/step - loss: 0.5090 - accuracy: 0.7380 - val_loss: 0.4637 - val_accuracy: 0.7578
Epoch 3/4
141/141 [==============================] - 3888s 28s/step - loss: 0.4827 - accuracy: 0.7582 - val_loss: 0.4512 - val_accuracy: 0.8066
Epoch 4/4
141/141 [==============================] - 9678s 69s/step - loss: 0.4723 - accuracy: 0.7855 - val_loss: 0.4359 - val_accuracy: 0.8075

Process finished with exit code 0
```