

Skill8

2100030910

Sec-23

Main.py

```
import numpy as np
import pandas as pd
import os
import model5 as mc

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

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

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

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

desert_directory = r'C:\Users\dell\PycharmProjex\dlSkill\Skill\data\desert'
desert_train_directory =
r'C:\Users\dell\PycharmProjex\dlSkill\Skill\genData\train\desert'
desert_valid_directory =
r'C:\Users\dell\PycharmProjex\dlSkill\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\dlSkill\Skill\genData\train'
valid_directory =
r'C:\Users\dell\PycharmProjex\dlSkill\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.adam_with_regularization()

```

```

# 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
from tensorflow import keras
from tensorflow.keras import layers
from keras.applications.vgg16 import VGG16, preprocess_input
class Modelsc:

    def adam_with_regularization(self):
        model = tf.keras.Sequential()

        model.add(layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(224, 224, 3)))
        model.add(layers.MaxPooling2D((2, 2)))
        model.add(layers.BatchNormalization())

        model.add(layers.Conv2D(64, (3, 3), activation='relu'))
        model.add(layers.MaxPooling2D((2, 2)))
        model.add(layers.BatchNormalization())

        model.add(layers.Conv2D(128, (3, 3), activation='relu'))
        model.add(layers.MaxPooling2D((2, 2)))

```

```

model.add(layers.BatchNormalization())

model.add(layers.Flatten())

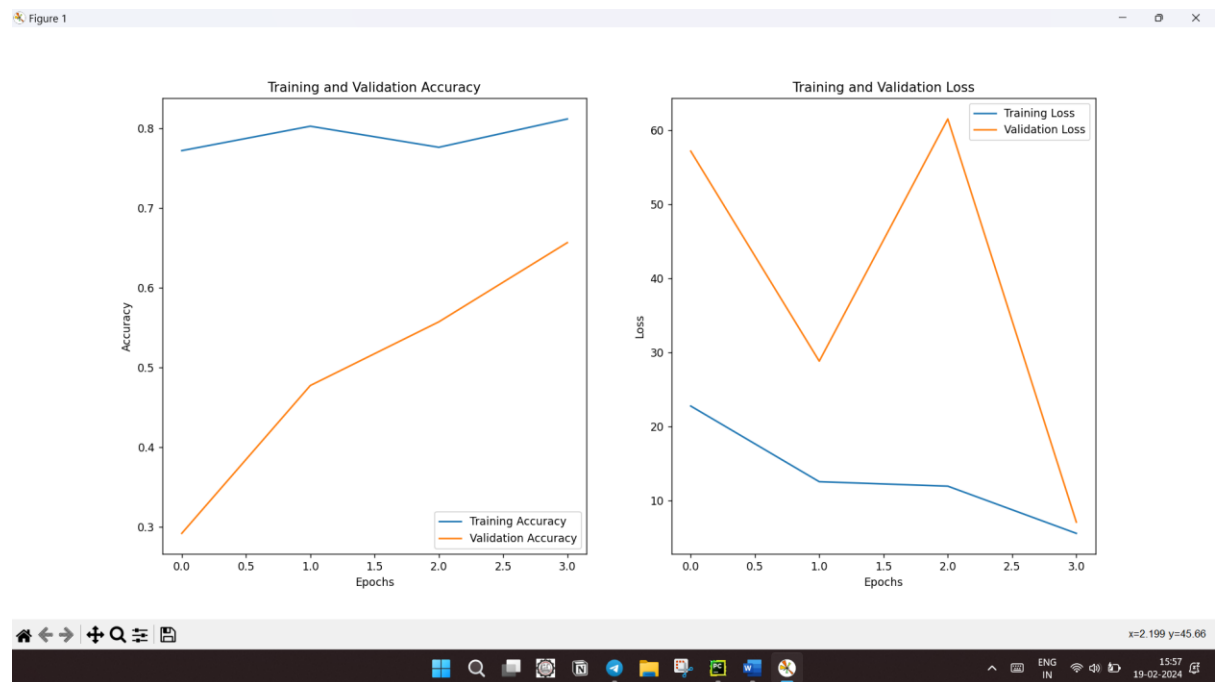
model.add(layers.Dense(256, activation='relu',
kernel_regularizer=keras.regularizers.l2(
    0.001))) # Adding L2 regularization with 0.001 strength
model.add(layers.Dropout(0.25)) # Adding dropout with 50% rate

model.add(layers.Dense(4, activation='softmax')) # Assuming 4
classes: cloudy, water, green, desert

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

return model

```



PC

☰

Classifiermulti

Version control

main

▶

⚙️

⋮

Project

Classifiermulti

main.py

main2.py

model.py

model2.py

model3.py

model4.py

model5.py

model.py

model4.py

model5.py

model3.py

model2.py

1

import numpy as np

2

import pandas as pd

3

import os

4

import model5 as mc

5

6

train_directory = r'C:\Users\de11\PycharmProjex\d1Skill\Skill\genData\t

7

valid_directory = r'C:\Users\de11\PycharmProjex\d1Skill\Skill\genData\v

8

9

cloud_directory = r'C:\Users\de11\PycharmProjex\d1Skill\Skill\data\clou

10

cloud_train_directory = r'C:\Users\de11\PycharmProjex\d1Skill\Skill\gen

Run

main

2024-02-19 15:36:19.338233: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 201867264 exceeds 10% of free system memory

2024-02-19 15:36:22.082273: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 201867264 exceeds 10% of free system memory

1/141 [.....] - ETA: 11:12 - loss: 2.6092 - accuracy: 0.1562

2024-02-19 15:36:22.530541: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 201867264 exceeds 10% of free system memory

2024-02-19 15:36:24.276098: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 201867264 exceeds 10% of free system memory

141/141 [=====] - 327s 2s/step - loss: 22.7333 - accuracy: 0.7720 - val_loss: 57.1494 - val_accuracy: 0.2

Epoch 2/4

141/141 [=====] - 299s 2s/step - loss: 12.5177 - accuracy: 0.8026 - val_loss: 28.7943 - val_accuracy: 0.4

Epoch 3/4

141/141 [=====] - 305s 2s/step - loss: 11.9165 - accuracy: 0.7762 - val_loss: 61.4924 - val_accuracy: 0.5

Epoch 4/4

141/141 [=====] - 317s 2s/step - loss: 5.5433 - accuracy: 0.8117 - val_loss: 7.0574 - val_accuracy: 0.656

Process finished with exit code 0

Classifiermulti

>

main.py

3:1

Windows Taskbar