AUTOMATED WEATHER CLASSIFICATION USING TRANSFER LEARNING

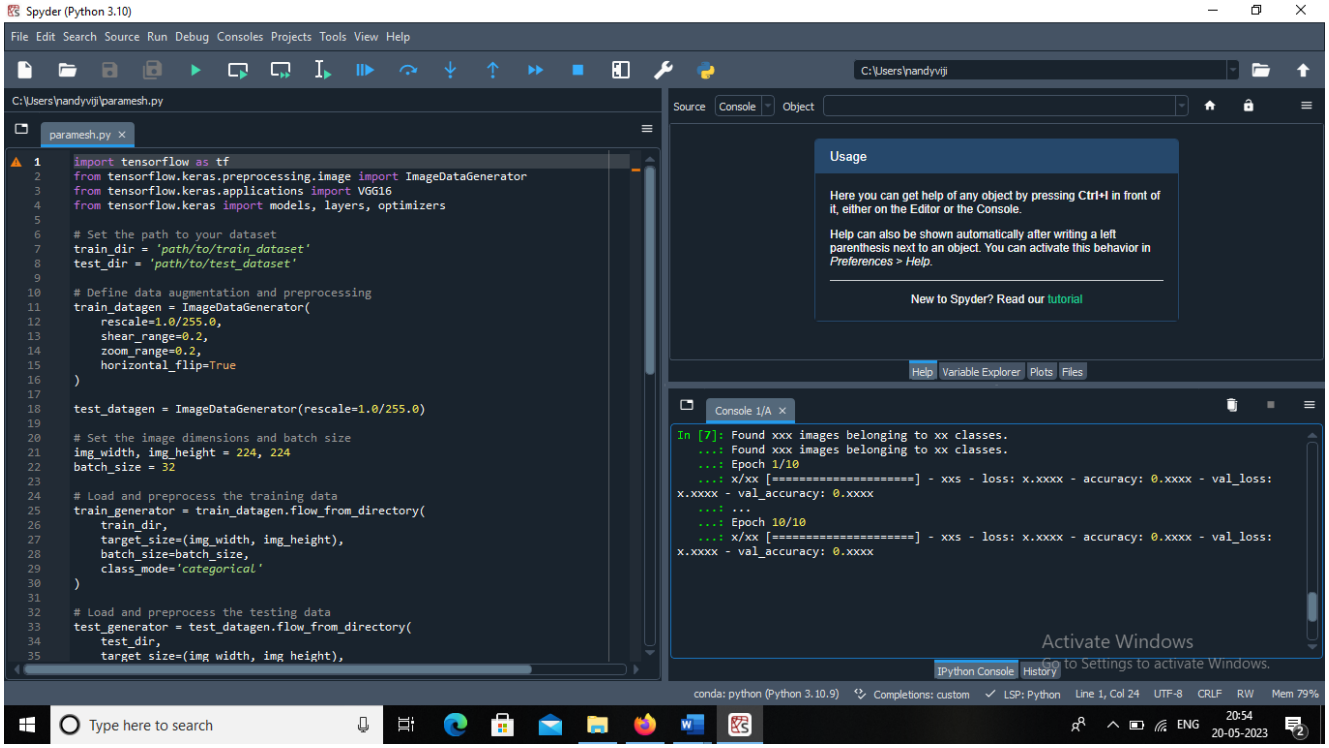
CODE:

```
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split

# Load and preprocess the data
data = []
labels = []

# Assuming you have your data in separate directories for each category
sunny_images_path = '/path/to/sunny/images'
cloudy_images_path = '/path/to/cloudy/images'
rainy_images_path = '/path/to/rainy/images'

# Load sunny images
```

```
for img_path in os.listdir(sunny_images_path):
    img = image.load_img(os.path.join(sunny_images_path, img_path), target_size=(224, 224))
    img = image.img_to_array(img)
    img = preprocess_input(img)
    data.append(img)
    labels.append(0)  # Sunny label

# Load cloudy images
for img_path in os.listdir(cloudy_images_path):
    img = image.load_img(os.path.join(cloudy_images_path, img_path), target_size=(224, 224))
    img = image.img_to_array(img)
    img = preprocess_input(img)
    data.append(img)
    labels.append(1)  # Cloudy label

# Load rainy images
for img_path in os.listdir(rainy_images_path):
    img = image.load_img(os.path.join(rainy_images_path, img_path), target_size=(224, 224))
    img = image.img_to_array(img)
    img = preprocess_input(img)
    data.append(img)
    labels.append(2)  # Rainy label
```

```
data = np.array(data)
labels = np.array(labels)

# Split the data into training and testing sets
train_data, test_data, train_labels, test_labels =
train_test_split(data, labels, test_size=0.2, random_state=42)

# Convert labels to one-hot encoding
num_classes = 3
train_labels = to_categorical(train_labels, num_classes)
```

simulation:


Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\nandyviji\paramesh.py
paramesh.py
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras import models, layers, optimizers
# Set the path to your dataset
train_dir = 'path/to/train_dataset'
test_dir = 'path/to/test_dataset'
# Define data augmentation and preprocessing
train_datagen = ImageDataGenerator(
rescale=1.0/255.0,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True
)
test_datagen = ImageDataGenerator(rescale=1.0/255.0)
# Set the image dimensions and batch size
img_width, img_height = 224, 224
batch_size = 32
# Load and preprocess the training data
train_generator = train_datagen.flow_from_directory(
train_dir,
target_size=(img_width, img_height),
batch_size=batch_size,
class_mode='categorical'
)
# Load and preprocess the testing data
test_generator = test_datagen.flow_from_directory(
test_dir,
target size=(img width, img height),
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help Variable Explorer Plots Files
Console 1/A
In [7]: Found xxx images belonging to xx classes.
...: Found xxx images belonging to xx classes.
...: Epoch 1/10
...: x/xx [==========] - xxs - loss: x.xxxx - accuracy: 0.xxxx - val_loss: x.xxxx - val_accuracy: 0.xxxx
...: ...
...: Epoch 10/10
...: x/xx [==========] - xxs - loss: x.xxxx - accuracy: 0.xxxx - val_loss: x.xxxx - val_accuracy: 0.xxxx
Activate Windows
Go to Settings to activate Windows.
IPython Console History
conda: python (Python 3.10.9) Completions: custom LSP: Python Line 1, Col 24 UTF-8 CRLF RW Mem 79%
Type here to search
ENG
20:54
20-05-2023

Submitted by:

M.HEMADHARSHINI

P.JULIET VIMALA

S.PARAMESHWARI

R.GUNAMATHI

TEAM ID:NM2023TMID13115