Skill2

2100030910

Sec-23

preprocessLoad.py

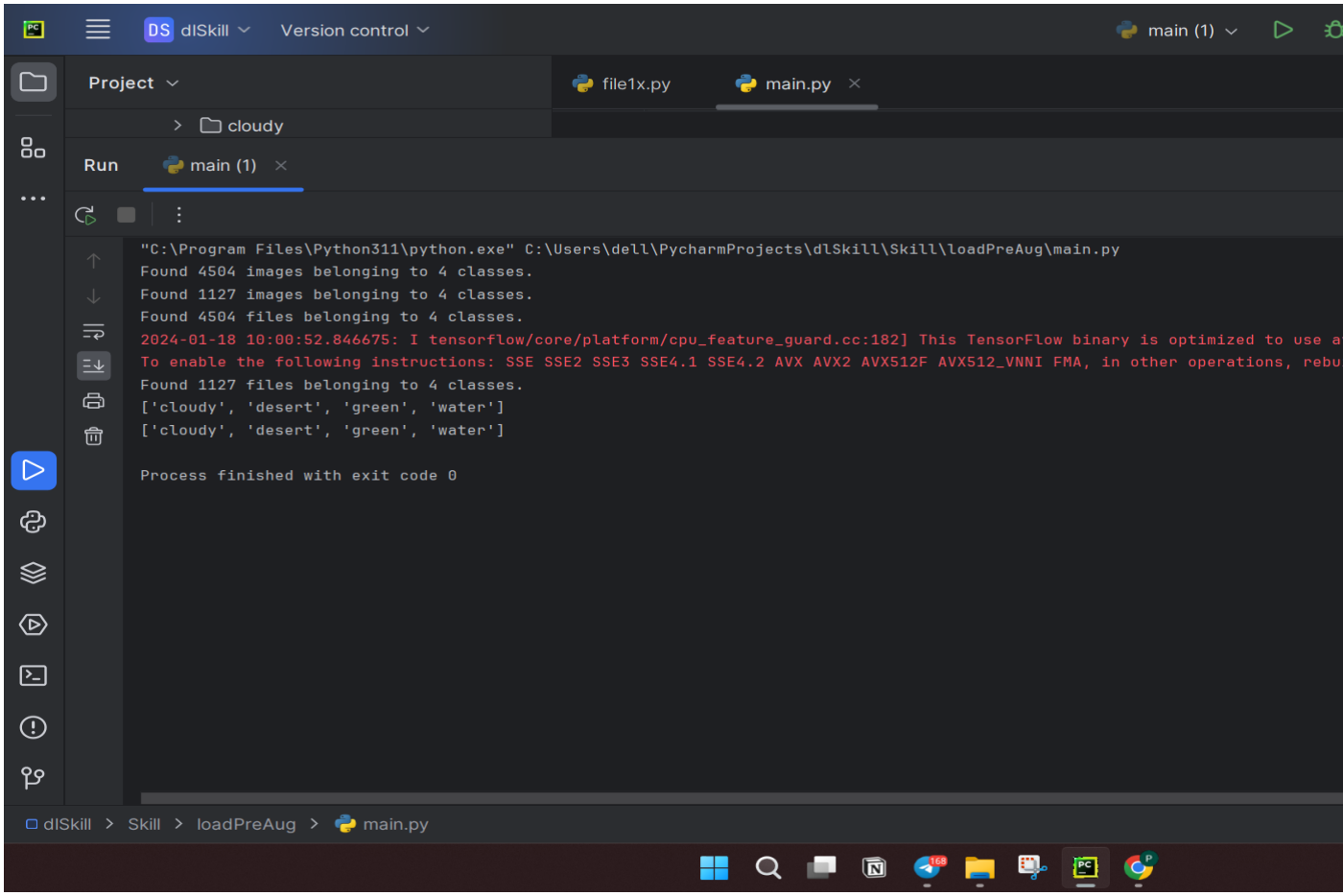
```
import numpy as np
import os

import matplotlib.pyplot as plt

import shutil
from sklearn.model_selection import train_test_split

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

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

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

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

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

cloud_image_files = [f for f in os.listdir(cloud_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]
# Specify a new name format
# You can customize the format according to your needs
# In this example, we're using a base name and appending an incremental
number
new_name_base = 'cloudy'
```

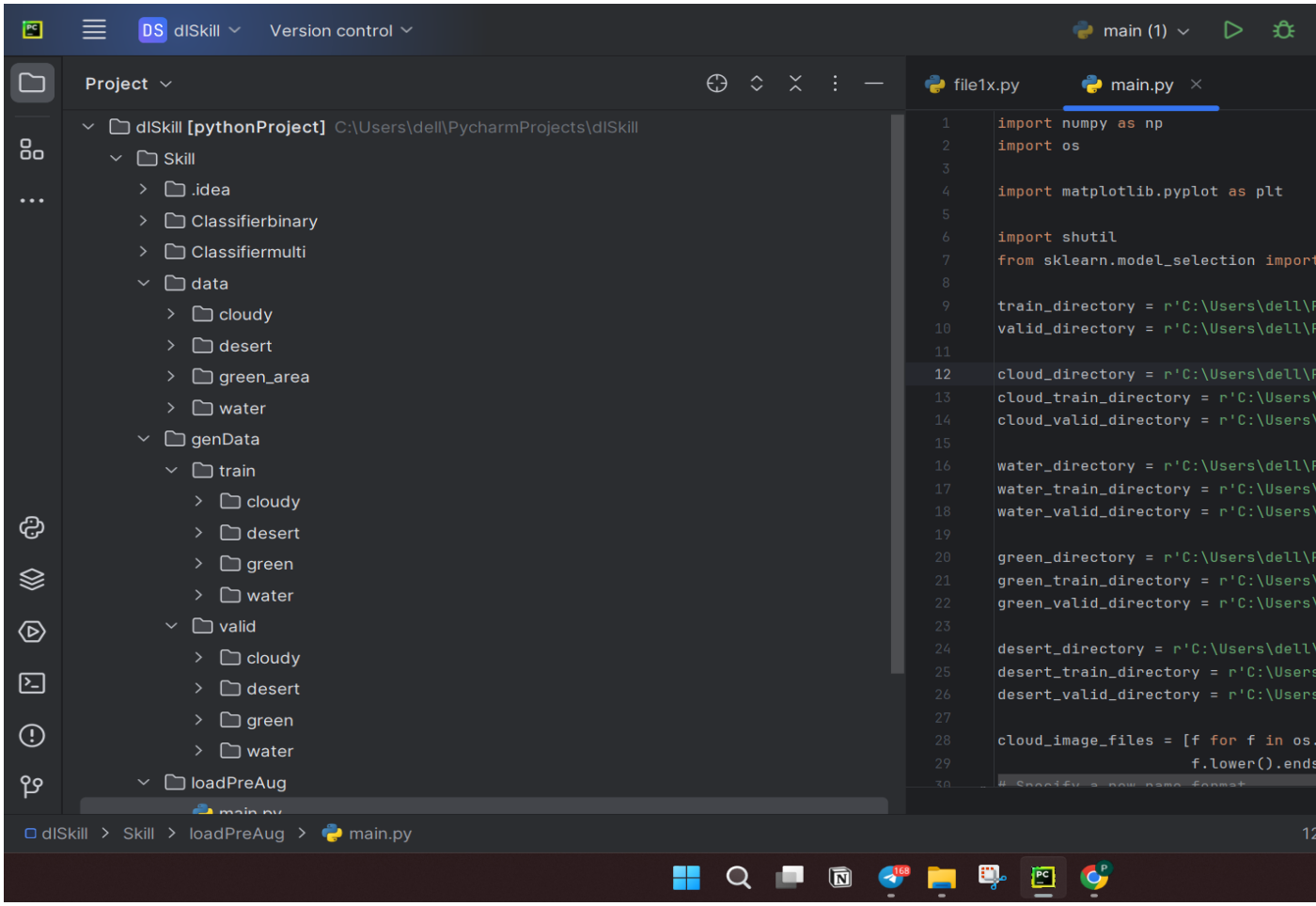
```
start_index = 1

# Loop through the image files and rename them
for old_name in cloud_image_files:
    # Determine the file extension
    file_ext = os.path.splitext(old_name)[-1]

    # Create the new file name with the desired format
    new_name = f'{new_name_base}_{start_index:04d}{file_ext}'

    # Construct the old and new file paths
    old_path = os.path.join(cloud_directory, old_name)
    new_path = os.path.join(cloud_directory, new_name)

    # Rename the file
    os.rename(old_path, new_path)

    # Increment the start index
    start_index += 1

water_image_files = [f for f in os.listdir(water_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]
# Specify a new name format
# You can customize the format according to your needs
# In this example, we're using a base name and appending an incremental
number
new_name_base = 'water'
start_index = 1

# Loop through the image files and rename them
for old_name in water_image_files:
    # Determine the file extension
    file_ext = os.path.splitext(old_name)[-1]

    # Create the new file name with the desired format
    new_name = f'{new_name_base}_{start_index:04d}{file_ext}'

    # Construct the old and new file paths
    old_path = os.path.join(water_directory, old_name)
    new_path = os.path.join(water_directory, new_name)

    # Rename the file
    os.rename(old_path, new_path)

    # Increment the start index
    start_index += 1

desert_image_files = [f for f in os.listdir(desert_directory) if
                      f.lower().endswith(('.jpg', '.jpeg'))]
# Specify a new name format
# You can customize the format according to your needs
# In this example, we're using a base name and appending an incremental
number
new_name_base = 'desert'
start_index = 1

# Loop through the image files and rename them
for old_name in desert_image_files:
    # Determine the file extension
    file_ext = os.path.splitext(old_name)[-1]
```

```
    # Create the new file name with the desired format
    new_name = f'{new_name_base}_{start_index:04d}{file_ext}'

    # Construct the old and new file paths
    old_path = os.path.join(desert_directory, old_name)
    new_path = os.path.join(desert_directory, new_name)

    # Rename the file
    os.rename(old_path, new_path)

    # Increment the start index
    start_index += 1

green_image_files = [f for f in os.listdir(green_directory) if
                     f.lower().endswith(('.jpg', '.jpeg'))]
# Specify a new name format
# You can customize the format according to your needs
# In this example, we're using a base name and appending an incremental
number
new_name_base = 'green_area'
start_index = 1

# Loop through the image files and rename them
for old_name in green_image_files:
    # Determine the file extension
    file_ext = os.path.splitext(old_name)[-1]

    # Create the new file name with the desired format
    new_name = f'{new_name_base}_{start_index:04d}{file_ext}'

    # Construct the old and new file paths
    old_path = os.path.join(green_directory, old_name)
    new_path = os.path.join(green_directory, new_name)

    # Rename the file
    os.rename(old_path, new_path)

    # Increment the start index
    start_index += 1


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

cloud_train_files, cloud_valid_files = train_test_split(cloud_image_files,
test_size=0.2, random_state=42)
for file in cloud_train_files:
    source_file_path = os.path.join(cloud_directory, file)
    destination_file_path = os.path.join(cloud_train_directory, file)
    shutil.copy(source_file_path, destination_file_path)

for file in cloud_valid_files:
    source_file_path = os.path.join(cloud_directory, file)
    destination_file_path = os.path.join(cloud_valid_directory, file)
    shutil.copy(source_file_path, destination_file_path)

water_train_files, water_valid_files = train_test_split(water_image_files,
test_size=0.2, random_state=42)
for file in water_train_files:
    source_file_path = os.path.join(water_directory, file)
    destination_file_path = os.path.join(water_train_directory, file)
```

```
    shutil.copy(source_file_path, destination_file_path)

for file in water_valid_files:
    source_file_path = os.path.join(water_directory, file)
    destination_file_path = os.path.join(water_valid_directory, file)
    shutil.copy(source_file_path, destination_file_path)

desert_train_files, desert_valid_files =
train_test_split(desert_image_files, test_size=0.2, random_state=42)
for file in desert_train_files:
    source_file_path = os.path.join(desert_directory, file)
    destination_file_path = os.path.join(desert_train_directory, file)
    shutil.copy(source_file_path, destination_file_path)

for file in desert_valid_files:
    source_file_path = os.path.join(desert_directory, file)
    destination_file_path = os.path.join(desert_valid_directory, file)
    shutil.copy(source_file_path, destination_file_path)

green_train_files, green_valid_files = train_test_split(green_image_files,
test_size=0.2, random_state=42)
for file in green_train_files:
    source_file_path = os.path.join(green_directory, file)
    destination_file_path = os.path.join(green_train_directory, file)
    shutil.copy(source_file_path, destination_file_path)

for file in green_valid_files:
    source_file_path = os.path.join(green_directory, file)
    destination_file_path = os.path.join(green_valid_directory, file)
    shutil.copy(source_file_path, destination_file_path)

# Creating training image data generator
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_imagenerator = 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'

)

train_generator = train_imagenerator.flow_from_directory(
    train_directory,
    target_size=(224,224),
    batch_size=20,
    class_mode='categorical',
#     classes=class_labels,
    shuffle=True
)

val_imagenerator = ImageDataGenerator(rescale=1.0/255)

validation_generator = val_imagenerator.flow_from_directory(
    valid_directory,
    target_size=(224,224),
```

```
    batch_size=20,
    class_mode='categorical',
    shuffle=True
)

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
batch_size = 28
image_size = (224, 224)

# Create ImageDataGenerator
train_imagenerator = ImageDataGenerator(
    rescale=1.0/255,   # Normalize pixel values to [0, 1]
    # You can add other data augmentation options here
)

# Create a tf.data.Dataset using image_dataset_from_directory
train_data = tf.keras.preprocessing.image_dataset_from_directory(
    train_directory,
    seed=45,
    shuffle=True,
    image_size=image_size,
    batch_size=batch_size
)

val_data = tf.keras.preprocessing.image_dataset_from_directory(
    valid_directory,
    seed=45,
    shuffle=False,
    image_size=image_size,
    batch_size=batch_size
)

class_names = train_data.class_names
print(class_names)

class_names = train_data.class_names

plt.figure(figsize=(12, 8))
for images, labels in train_data.take(1):
    for i in range(4):
        ax = plt.subplot(2, 2, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")

augmented_images, labels = train_generator.next()

plt.figure(figsize=(12, 8))
for i in range(min(6, augmented_images.shape[0])):
    ax = plt.subplot(2, 3, i + 1)
    plt.imshow(augmented_images[i])
    plt.title(int(np.argmax(labels[i])))  # Convert one-hot encoded label
to integer category
    plt.axis("off")

class_names = train_data.class_names
print(class_names)
```


dlSkill
Version control
main (1)
Project
file1x.py
main.py
cloudy
Run
main (1)
"C:\Program Files\Python311\python.exe" C:\Users\dell\PycharmProjects\dlSkill\Skill\loadPreAug\main.py
Found 4504 images belonging to 4 classes.
Found 1127 images belonging to 4 classes.
Found 4504 files belonging to 4 classes.
2024-01-18 10:00:52.846675: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use a
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebu
Found 1127 files belonging to 4 classes.
['cloudy', 'desert', 'green', 'water']
['cloudy', 'desert', 'green', 'water']
Process finished with exit code 0
dlSkill > Skill > loadPreAug > main.py


DS dlSkill
Version control
main (1)
Project
dlSkill [pythonProject] C:\Users\dell\PycharmProjects\dlSkill
Skill
.idea
Classifierbinary
Classifiermulti
data
cloudy
desert
green_area
water
genData
train
cloudy
desert
green
water
valid
cloudy
desert
green
water
loadPreAug
file1x.py
main.py
import numpy as np
import os
import matplotlib.pyplot as plt
import shutil
from sklearn.model_selection import
train_directory = r'C:\Users\dell\
valid_directory = r'C:\Users\dell\
cloud_directory = r'C:\Users\dell\
cloud_train_directory = r'C:\Users
cloud_valid_directory = r'C:\Users
water_directory = r'C:\Users\dell\
water_train_directory = r'C:\Users
water_valid_directory = r'C:\Users
green_directory = r'C:\Users\dell\
green_train_directory = r'C:\Users
green_valid_directory = r'C:\Users
desert_directory = r'C:\Users\dell
desert_train_directory = r'C:\Users
desert_valid_directory = r'C:\Users
cloud_image_files = [f for f in os.
f.lower().ends
dlSkill > Skill > loadPreAug > main.py