0.Preparation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Double-click (or enter) to edit

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Activation,Dropout,Conv2D, MaxPooling2D,BatchNormalization
from tensorflow.keras.optimizers import Adam, Adamax
from tensorflow.keras.metrics import categorical_crossentropy
```

1.Read the dataset

Create ImageData Generator

```
size=(224,224)
lables=['Coast','Desert','Forest','Glacier','Mountain']

def data_generator(location):
    data_gen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1/255.0,horizontal_flip=True)
    return data_gen.flow_from_directory(location,target_size=size,classes=lables,class_mode='sparse',seed=42)
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

Import training dataset

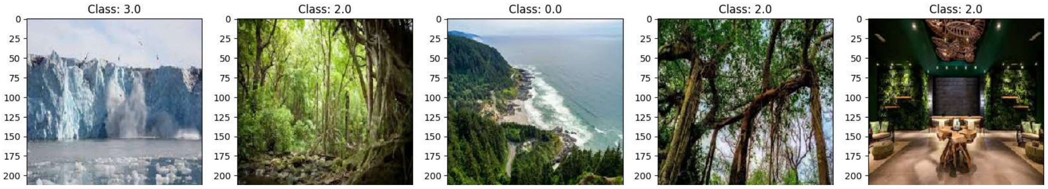
```
train_dir='/content/gdrive/My Drive/MLProject/Training Data'

train = data_generator(train_dir)
```

```
Found 10000 images belonging to 5 classes.
```

```
train.class_indices
```

```
{'Coast': 0, 'Desert': 1, 'Forest': 2, 'Glacier': 3, 'Mountain': 4}
```

```
fig, ax = plt.subplots(1, 5, figsize=(20, 20))
for images, labels in train:
    for i in range(5):
        ax[i].imshow(images[i])
        ax[i].set_title('Class: ' + str(labels[i]))
    break
```



2.Build a CNN Classcifier

```
tf.random.set_seed(42)

classifier = tf.keras.Sequential()

classifier.add(tf.keras.layers.Conv2D(10, 3, activation="relu",input_shape=size + (3,)))

 classifier.add(tf.keras.layers.Conv2D(10, 3, activation="relu"))

classifier.add( tf.keras.layers.MaxPool2D(2))

 classifier.add(tf.keras.layers.Conv2D(10, 3, activation="relu"))

 classifier.add(tf.keras.layers.Conv2D(10, 3, activation="relu"))

classifier.add( tf.keras.layers.MaxPool2D(2))

classifier.add( tf.keras.layers.Flatten(),)

classifier.add( tf.keras.layers.Dense(5, activation="softmax"))

classifier.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(),optimizer=tf.keras.optimizers.Adam(),metrics=["accuracy"])
classifier.summary()
```

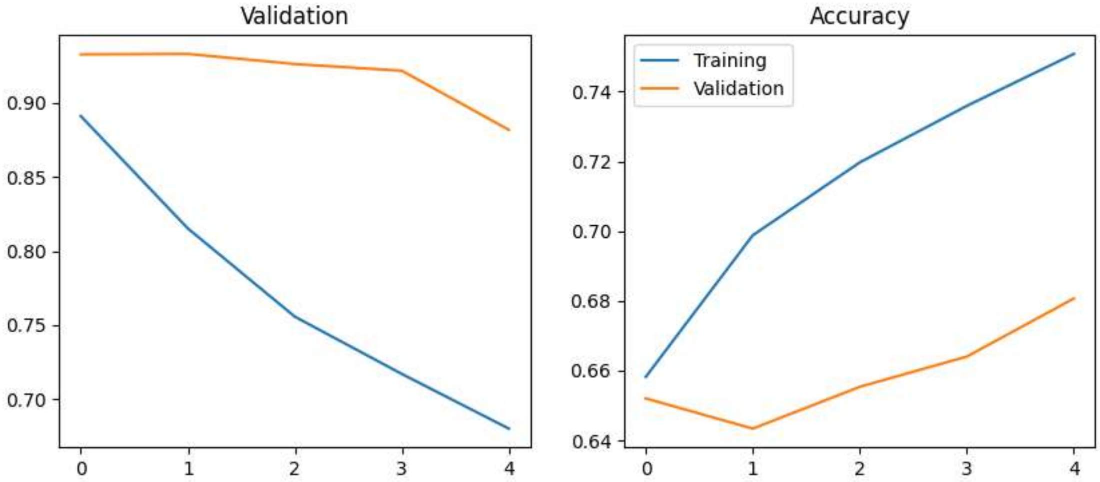
```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_32 (Conv2D)          (None, 222, 222, 10)      280

 conv2d_33 (Conv2D)          (None, 220, 220, 10)      910

 max_pooling2d_18 (MaxPoolin  (None, 110, 110, 10)     0
 g2D)

 conv2d_34 (Conv2D)          (None, 108, 108, 10)      910

 conv2d_35 (Conv2D)          (None, 106, 106, 10)      910

 max_pooling2d_19 (MaxPoolin  (None, 53, 53, 10)       0
 g2D)

 flatten_10 (Flatten)        (None, 28090)             0

 dense_9 (Dense)             (None, 5)                 140455

=================================================================
Total params: 143,465
Trainable params: 143,465
Non-trainable params: 0
_________________________________________________________________
```

3.Fit Model

```
valid_dir='/content/gdrive/My Drive/MLProject/Validation Data'
validation=data_generator(valid_dir)
model_history = classifier.fit(train, epochs=5,steps_per_epoch=len(train),validation_data=validation, validation_steps=len(validation))
```

```
Found 1500 images belonging to 5 classes.
Epoch 1/5
313/313 [==============================] - 1287s 4s/step - loss: 0.8910 - accuracy: 0.6582 - val_loss: 0.9327 - val_accuracy: 0.6520
Epoch 2/5
313/313 [==============================] - 607s 2s/step - loss: 0.8152 - accuracy: 0.6988 - val_loss: 0.9330 - val_accuracy: 0.6433
Epoch 3/5
313/313 [==============================] - 599s 2s/step - loss: 0.7556 - accuracy: 0.7197 - val_loss: 0.9262 - val_accuracy: 0.6553
Epoch 4/5
313/313 [==============================] - 605s 2s/step - loss: 0.7170 - accuracy: 0.7359 - val_loss: 0.9216 - val_accuracy: 0.6640
Epoch 5/5
313/313 [==============================] - 607s 2s/step - loss: 0.6801 - accuracy: 0.7508 - val_loss: 0.8817 - val_accuracy: 0.6807
```

```
def plot_history(history):
    plt.figure(figsize=(10, 4))
    plt.subplot(121)
    plt.plot(history.history['loss'], label="Training")
    plt.plot(history.history['val_loss'], label="Validation")
    plt.title("Validation")

    plt.subplot(122)
    plt.plot(history.history['accuracy'], label="Training")
    plt.plot(history.history['val_accuracy'], label="Validation")
    plt.title("Accuracy")

    plt.legend()
    plt.show()

plot_history(model_history)
```



```
classifier.save('the_model.h5')
print("Saved model.")
```

```
Saved model.
```

4.Predict on the test set

```
model = keras.models.load_model('the_model.h5')
print("Model is loaded.")
```

```
Model is loaded.
```

```
test_dir='/content/gdrive/My Drive/MLProject/Testing Data'

test = data_generator(test_dir)
actual_result=test.labels
```

```
Found 500 images belonging to 5 classes.
```

```
def evaluate_model(model):
    metrics = model.evaluate(test)
    print(f"Accuracy: {metrics[1] * 100:.2f}%")

evaluate_model(classifier)
evaluate_model(classifier)
```

```
16/16 [==============================] - 147s 10s/step - loss: 0.7201 - accuracy: 0.7300
Accuracy: 73.00%
```

```
import os, glob
from tensorflow.keras.preprocessing import image
img_dir="/content/gdrive/My Drive/MLProject/Testing Data"
data_path = os.path.join(img_dir, '*g')
files = glob.glob(data_path)
```

```
files
```

```
[]
```

```
data = []
test_result = []
for f1 in files:
    img = image.load_img(f1, target_size = (224, 224))
    img = image.img_to_array(img)
    img = np.expand_dims(img, axis = 0)
    data.append(img)
    result = model.predict(img)
    r = np.argmax(result, axis=1)
    test_result.append(r)
```

```
test_result
```

```
[]
```

✓ 3m 22s completed at 8:37 PM