Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	SWTID1749906821
Project Title	Neural Networks Ahoy: Cutting-Edge Ship Classification
Maximum Marks	6 Marks

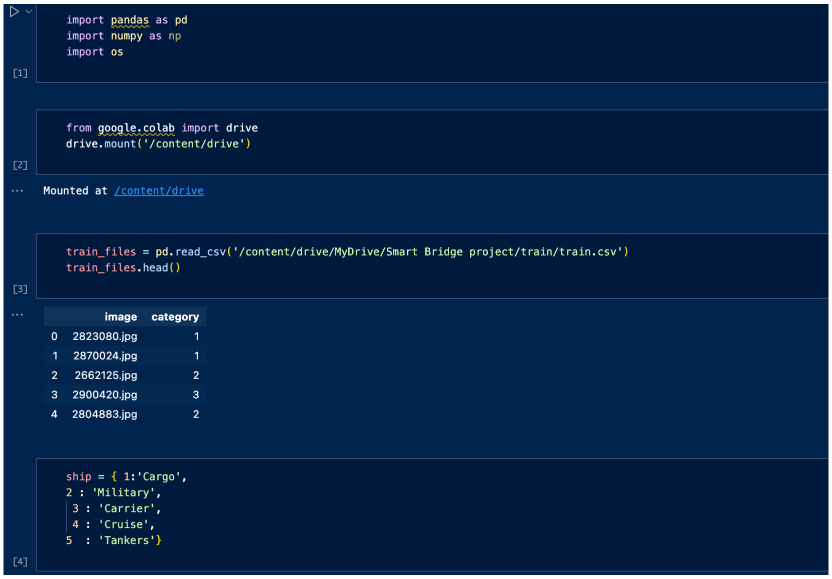
Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Loading Data	Give an overview of the data, which you're going to use in your project.
Categorize Train images	The dataset has ship images and a train.csv file mapping image IDs to encoded categories. A dictionary is used to decode category labels and add a readable column. Images are then moved into subfolders named after their categories for better organization.
Import ImageDataGenerator Library And Configure It	ImageDataGenerator class is used to augment the images with different modifications like considering the rotation, flipping the image etc. A function known as the preprocess_input from VGG16 library will perform the necessary preprocessing operations on the images in order to make them suitable for getting trained.

Apply ImageDataGenerator Functionality To Train And Test Set	Specify the path of both the folders in flow_from_directory method. We are importing the images in 224*224 pixels.

Data Preprocessing Code Screenshots

Loading Data	

```
import pandas as pd
import numpy as np
import os
```
[1]

```
from google.colab import drive
drive.mount('/content/drive')
```
[2]

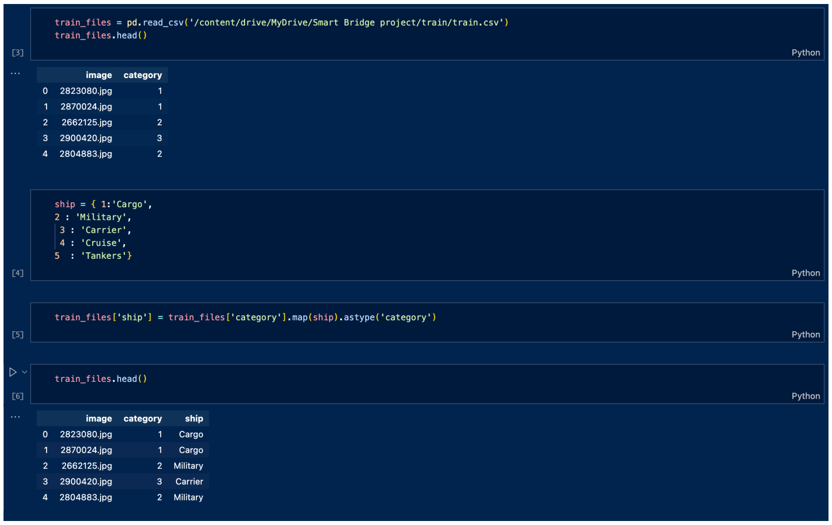
```
Mounted at /content/drive
```

```
train_files = pd.read_csv('/content/drive/MyDrive/Smart Bridge project/train/train.csv')
train_files.head()
```
[3]

```
         image  category
0  2823080.jpg         1
1  2870024.jpg         1
2  2662125.jpg         2
3  2900420.jpg         3
4  2804883.jpg         2
```

```
ship = { 1:'Cargo',
2 : 'Military',
 3 : 'Carrier',
 4 : 'Cruise',
5  : 'Tankers'}
```
[4]

Categorize Train images



```
train_files = pd.read_csv('/content/drive/MyDrive/Smart Bridge project/train/train.csv')
train_files.head()
```

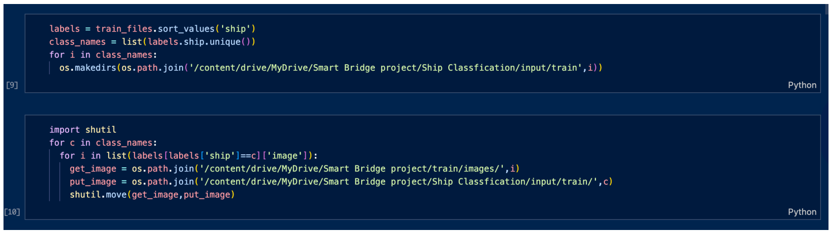
```
       image  category
0  2823080.jpg        1
1  2870024.jpg        1
2  2662125.jpg        2
3  2900420.jpg        3
4  2804883.jpg        2
```

```
ship = { 1:'Cargo',
2 : 'Military',
 3 : 'Carrier',
 4 : 'Cruise',
5  : 'Tankers'}
```

```
train_files['ship'] = train_files['category'].map(ship).astype('category')
```

```
train_files.head()
```

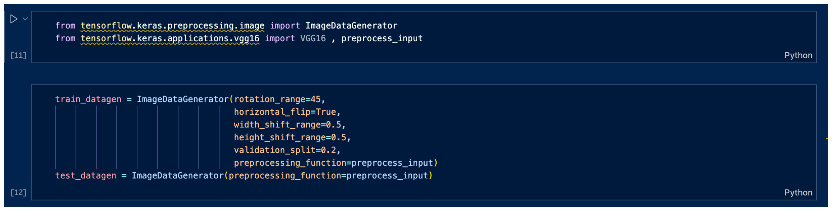
```
       image  category      ship
0  2823080.jpg        1     Cargo
1  2870024.jpg        1     Cargo
2  2662125.jpg        2  Military
3  2900420.jpg        3   Carrier
4  2804883.jpg        2  Military
```



```
labels = train_files.sort_values('ship')
class_names = list(labels.ship.unique())
for i in class_names:
  os.makedirs(os.path.join('/content/drive/MyDrive/Smart Bridge project/Ship Classfication/input/train',i))
```

```
import shutil
for c in class_names:
  for i in list(labels[labels['ship']==c]['image']):
    get_image = os.path.join('/content/drive/MyDrive/Smart Bridge project/train/images/',i)
    put_image = os.path.join('/content/drive/MyDrive/Smart Bridge project/Ship Classfication/input/train/',c)
    shutil.move(get_image,put_image)
```

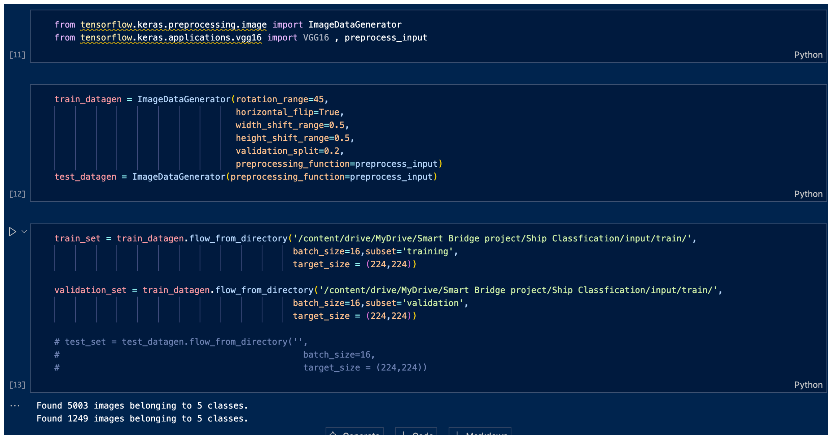
Import ImageDataGenerator Library And Configure It



```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.vgg16 import VGG16 , preprocess_input
```

```
train_datagen = ImageDataGenerator(rotation_range=45,
                                   horizontal_flip=True,
                                   width_shift_range=0.5,
                                   height_shift_range=0.5,
                                   validation_split=0.2,
                                   preprocessing_function=preprocess_input)
test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```

Apply ImageDataGenerator Functionality To Train And Test Set



```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.vgg16 import VGG16 , preprocess_input
```

```
train_datagen = ImageDataGenerator(rotation_range=45,
                                   horizontal_flip=True,
                                   width_shift_range=0.5,
                                   height_shift_range=0.5,
                                   validation_split=0.2,
                                   preprocessing_function=preprocess_input)
test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```

```
train_set = train_datagen.flow_from_directory('/content/drive/MyDrive/Smart Bridge project/Ship Classfication/input/train/',
                                              batch_size=16,subset='training',
                                              target_size = (224,224))

validation_set = train_datagen.flow_from_directory('/content/drive/MyDrive/Smart Bridge project/Ship Classfication/input/train/',
                                              batch_size=16,subset='validation',
                                              target_size = (224,224))

# test_set = test_datagen.flow_from_directory('',
#                                                batch_size=16,
#                                                target_size = (224,224))
```

```
Found 5003 images belonging to 5 classes.
Found 1249 images belonging to 5 classes.
```