Data Collection and Preprocessing Phase

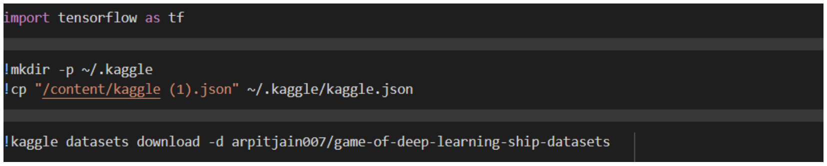
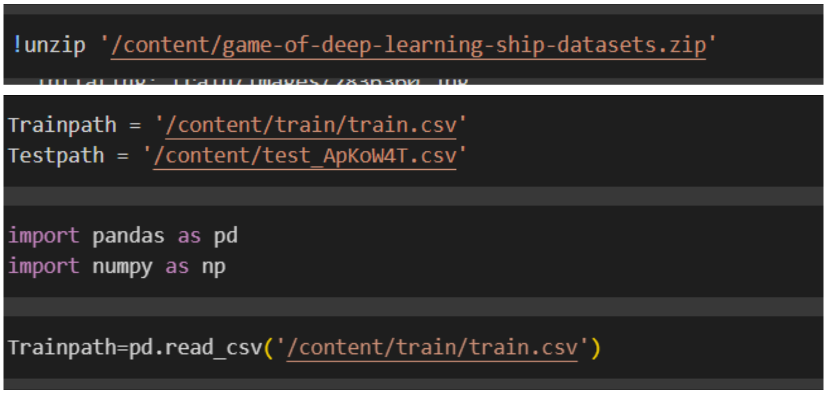
Date	18 April 2024
Team ID	738171
Project Title	Neural Network Ahoy:Cutting-Edge Ship Classification For Maritiome Mastery
Maximum Marks	6 Marks

Preprocessing Template

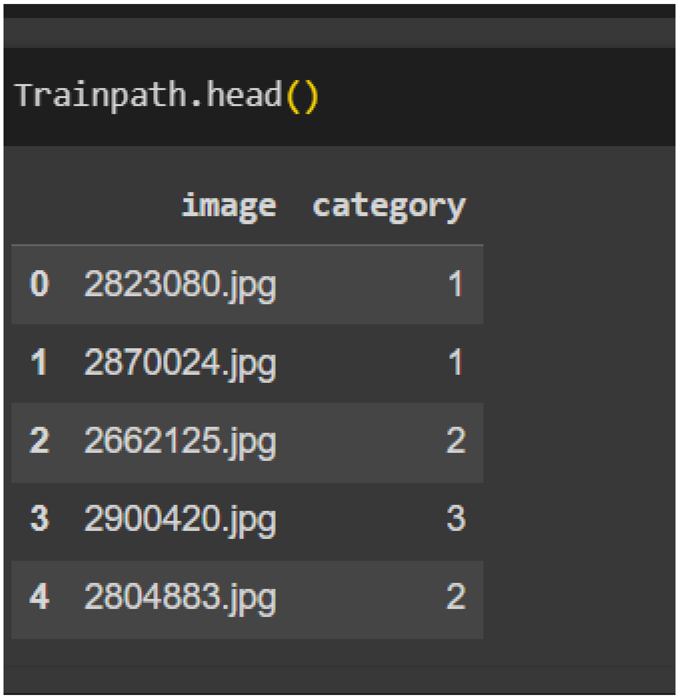
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network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

Section	**Description**
Data Overview	**Game of Deep Learning: Ship datasets** This Dataset is taken from Deep Learning The dataset for this machine learning project revolves around ships, structured within a "train" folder. This folder encapsulates the training data essential for model development, consisting of two key elements: an "Image" folder and a "train.csv" file. The "Image" folder hosts a collection of ship images, where each image serves as a vital input for training the model. Concurrently, the "train.csv" file provides metadata or annotations corresponding to these ship images.
Adding	Adding Ship Names in the dataset. The provided code snippet augments a dataset by enriching it with descriptive textual labels corresponding to numerical ship categories. In this transformation, each numerical category code, ranging from 1 to 5, is associated with a specific type of ship: "Cargo", "Military","Carrier", "Cruise", or "Tankers"

Categorize Train Images	As seen in the above image, we have used a dictionary to map the values to the corresponding category that the ship belongs to. Let us now organise our train images into folders representing their corresponding categories.
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	

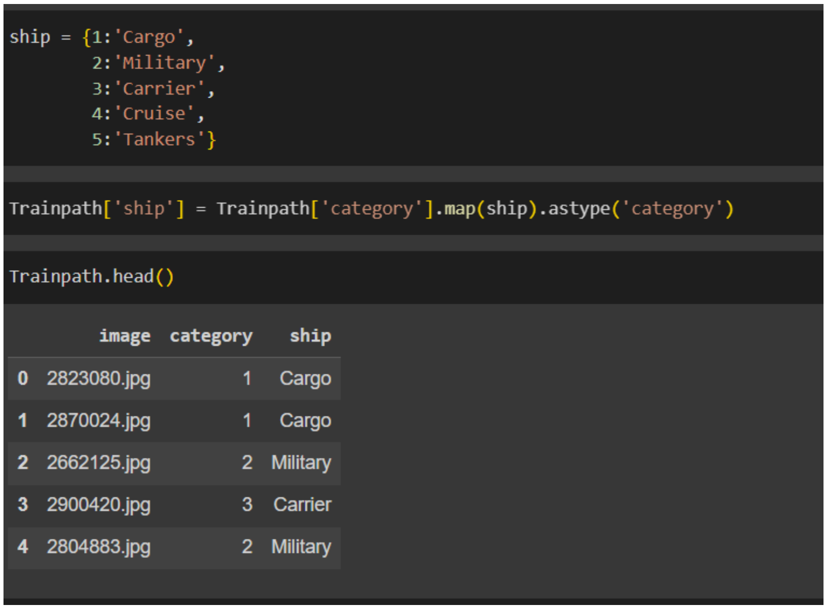
Loading Data	Load tensorflow,panda,numpy
	
	`import tensorflow as tf`
	`!mkdir -p ~/.kaggle`
	`!cp "/content/kaggle (1).json" ~/.kaggle/kaggle.json`
	`!kaggle datasets download -d arpitjain007/game-of-deep-learning-ship-datasets`
	
	`!unzip '/content/game-of-deep-learning-ship-datasets.zip'`
	`Trainpath = '/content/train/train.csv'`
	`Testpath = '/content/test_ApKoW4T.csv'`
	`import pandas as pd`
	`import numpy as np`
	`Trainpath=pd.read_csv('/content/train/train.csv')`

Adding



```
Trainpath.head()
```

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



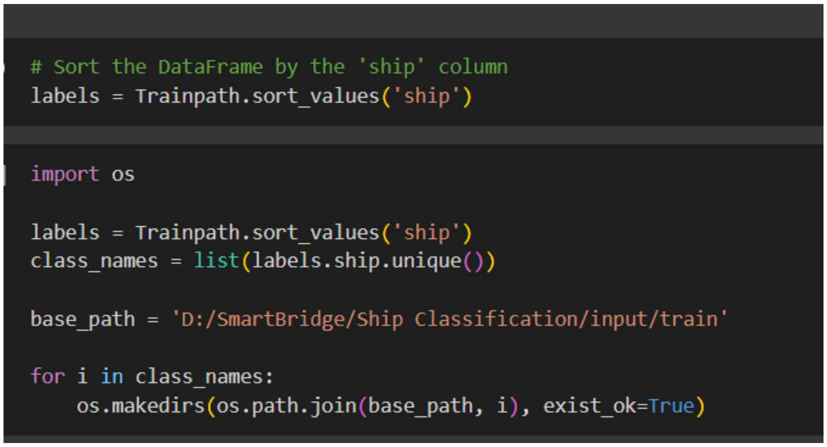
```
ship = {1:'Cargo',
        2:'Military',
        3:'Carrier',
        4:'Cruise',
        5:'Tankers'}
```

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

```
Trainpath.head()
```

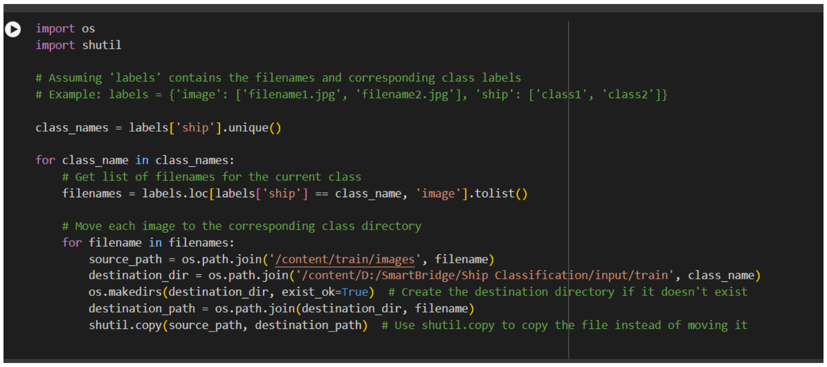
```
     image  category      ship
0  2823080.jpg     1     Cargo
1  2870024.jpg     1     Cargo
2  2662125.jpg     2  Military
3  2900420.jpg     3   Carrier
4  2804883.jpg     2  Military
```

Normalization



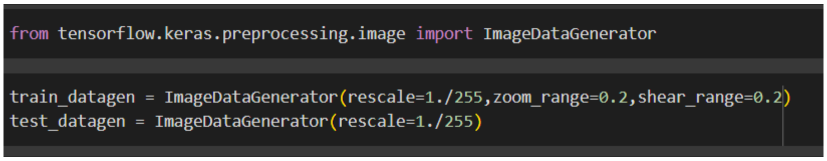
```python
# Sort the DataFrame by the 'ship' column
labels = Trainpath.sort_values('ship')
```

```python
import os

labels = Trainpath.sort_values('ship')
class_names = list(labels.ship.unique())

base_path = 'D:/SmartBridge/Ship Classification/input/train'

for i in class_names:
    os.makedirs(os.path.join(base_path, i), exist_ok=True)
```



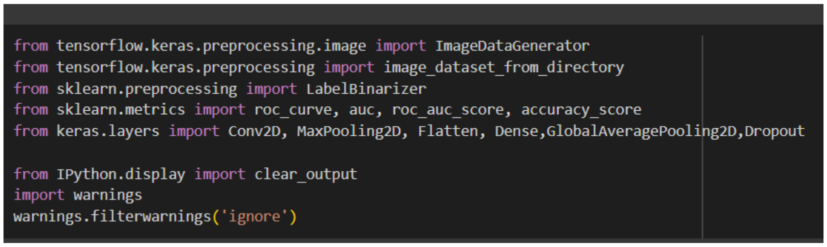
```python
import os
import shutil

# Assuming 'labels' contains the filenames and corresponding class labels
# Example: labels = {'image': ['filename1.jpg', 'filename2.jpg'], 'ship': ['class1', 'class2']}

class_names = labels['ship'].unique()

for class_name in class_names:
    # Get list of filenames for the current class
    filenames = labels.loc[labels['ship'] == class_name, 'image'].tolist()

    # Move each image to the corresponding class directory
    for filename in filenames:
        source_path = os.path.join('/content/train/images', filename)
        destination_dir = os.path.join('/content/D:/SmartBridge/Ship Classification/input/train', class_name)
        os.makedirs(destination_dir, exist_ok=True)  # Create the destination directory if it doesn't exist
        destination_path = os.path.join(destination_dir, filename)
        shutil.copy(source_path, destination_path)  # Use shutil.copy to copy the file instead of moving it
```

Import ImageDataGenerator Library And Configure It



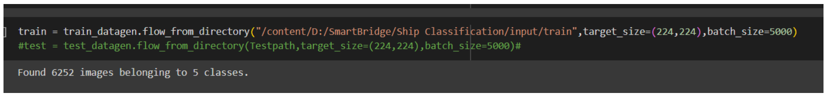
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2)
test_datagen = ImageDataGenerator(rescale=1./255)
```



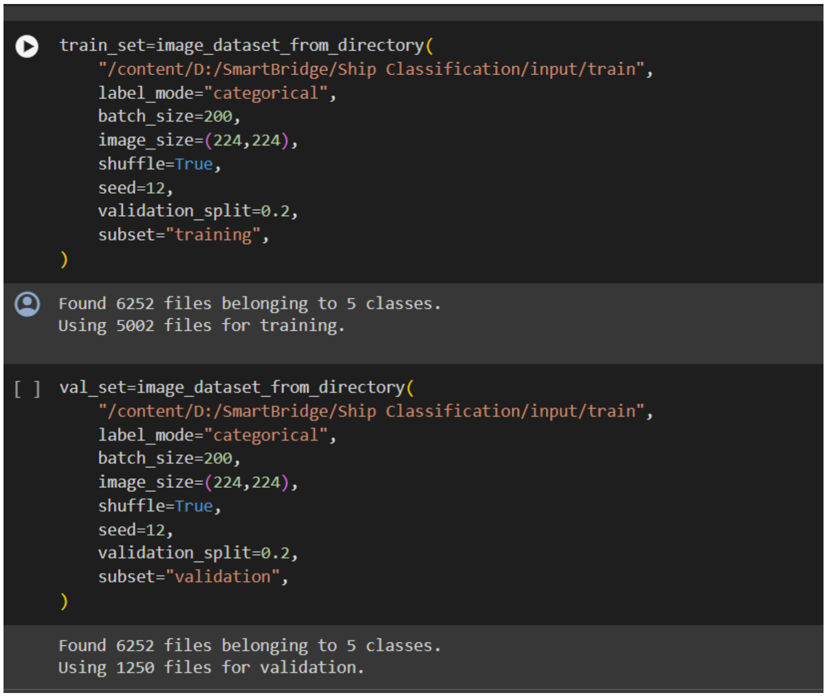
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image_dataset_from_directory
from sklearn.preprocessing import LabelBinarizer
from sklearn.metrics import roc_curve, auc, roc_auc_score, accuracy_score
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,GlobalAveragePooling2D,Dropout

from IPython.display import clear_output
import warnings
warnings.filterwarnings('ignore')
```

Apply ImageDataGenerator Functionality To Train And Test Set



```python
train = train_datagen.flow_from_directory("/content/D:/SmartBridge/Ship Classification/input/train",target_size=(224,224),batch_size=5000)
#test = test_datagen.flow_from_directory(Testpath,target_size=(224,224),batch_size=5000)#
```

```
Found 6252 images belonging to 5 classes.
```



```
train_set=image_dataset_from_directory(
    "/content/D:/SmartBridge/Ship Classification/input/train",
    label_mode="categorical",
    batch_size=200,
    image_size=(224,224),
    shuffle=True,
    seed=12,
    validation_split=0.2,
    subset="training",
)
```

```
Found 6252 files belonging to 5 classes.
Using 5002 files for training.
```

```
val_set=image_dataset_from_directory(
    "/content/D:/SmartBridge/Ship Classification/input/train",
    label_mode="categorical",
    batch_size=200,
    image_size=(224,224),
    shuffle=True,
    seed=12,
    validation_split=0.2,
    subset="validation",
)
```

```
Found 6252 files belonging to 5 classes.
Using 1250 files for validation.
```