

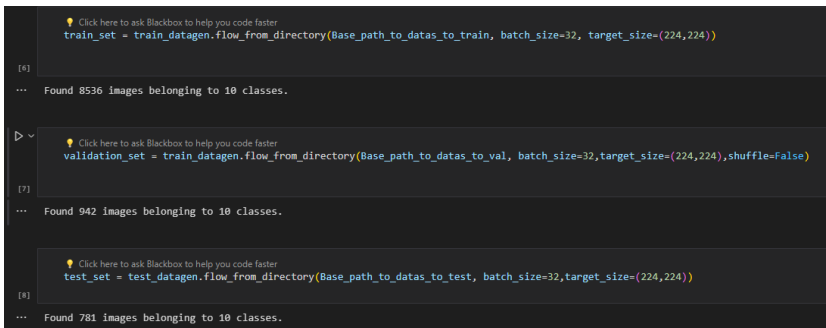
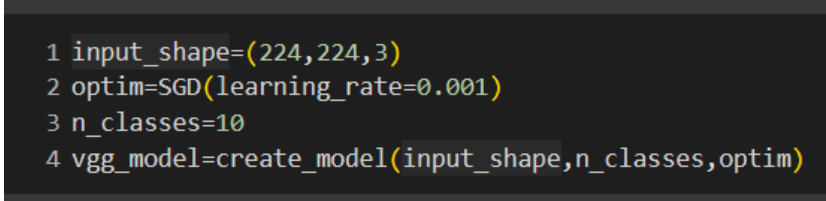
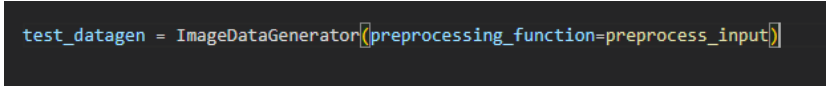
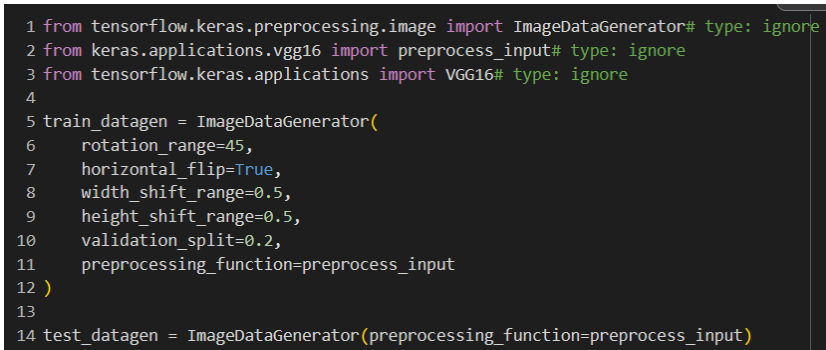
Data Collection and Preprocessing Phase

Date	15 April 2024
Team ID	Team-738165
Project Title	Neural Networks Ahoy: Cutting-edge Ship Classification for Maritime Mastery
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, and batch normalizing/ 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	The dataset for this project consists of a collection of images of different ships.
Resizing	Resized images to a specified target size (224,224).
Normalization	Normalize pixel using preprocessing_input function to convert image from RGB to BGR, as the standard format of imagenet
Data Augmentation	rotation_range, horizontal_flip, width_shift_range, height_shift_range, validation_split and preprocessing_function.
Denoising	NA
Edge Detection	NA

Color Space Conversion	NA
Image Cropping	NA
Batch Normalization	Apply batch normalization to the input of each layer in the neural network.
Data Preprocessing Code Screenshots	
Loading Data	 <pre> Click here to ask Blackbox to help you code faster train_set = train_datagen.flow_from_directory(Base_path_to_datas_to_train, batch_size=32, target_size=(224,224)) [6] ... Found 8536 images belonging to 10 classes. Click here to ask Blackbox to help you code faster validation_set = train_datagen.flow_from_directory(Base_path_to_datas_to_val, batch_size=32, target_size=(224,224), shuffle=False) [7] ... Found 942 images belonging to 10 classes. Click here to ask Blackbox to help you code faster test_set = test_datagen.flow_from_directory(Base_path_to_datas_to_test, batch_size=32, target_size=(224,224)) [8] ... Found 781 images belonging to 10 classes. </pre>
Resizing	 <pre> 1 input_shape=(224,224,3) 2 optim=SGD(learning_rate=0.001) 3 n_classes=10 4 vgg_model=create_model(input_shape,n_classes,optim) </pre>
Normalization	 <pre> test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input) </pre>
Data Augmentation	 <pre> 1 from tensorflow.keras.preprocessing.image import ImageDataGenerator# type: ignore 2 from keras.applications.vgg16 import preprocess_input# type: ignore 3 from tensorflow.keras.applications import VGG16# type: ignore 4 5 train_datagen = ImageDataGenerator(6 rotation_range=45, 7 horizontal_flip=True, 8 width_shift_range=0.5, 9 height_shift_range=0.5, 10 validation_split=0.2, 11 preprocessing_function=preprocess_input 12) 13 14 test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input) </pre>
Denoising	NA

Edge Detection	NA
Color Space Conversion	NA
Image Cropping	NA
Batch Normalization	<pre>top_model=Flatten(name="flatten")(top_model)</pre>