Dual Convolutional Neural Networks and Regression Model-based Coral Reef Annotation and Localization

Rohit Raj Gunti[a], Abebe Rorissa[a,b]

[a,b] *University of Tennessee, Knoxville, UTK, USA*

Abstract

There has been an enormous interest in using deep learning to classify underwater substrate images to identify various objects like fishes, plankton, coral reefs, seagrass, submarines, and gestures of sea divers. This classification is essential for measuring water bodies' health and quality and protecting endangered species. In the previous study, we examined the effectiveness and flexibility of the CNN model using online individual coral and grouped datasets. Our observations from the previous study showed that our model predicted the substrates correctly. However, certain miscalculations led to lower recall scores. This year, we introduced a new approach and changed the platform to build a system that will improve the scores as well as predictions in an efficient manner. For the current study, we trained our model using high-level APIs of Keras and TensorFlow python libraries and produced improved predictions and accuracy.

Keywords

Image Classification, Coral Reef, Convolution Neural Network, Regression, Annotationand Localization

1. INTRODUCTION

In recent years, we have witnessed a massive growth in the interest in processing underwater images. Studying of the well-being and extent of corals and animals is beneficial marine biology, the economy, and biodiversity management. It can help in analyzing the differences in species and protecting endangered species. For example, plankton have a high sensitivity to changes in surroundings and the environment. Hence, the study of their well-being can detect any impending climatic events such as pollution and global warming. They are a crucial link in the ocean food chain and connect the ocean to the atmosphere. Plankton produce more than 80% of the world's oxygen. Hence, a low level of plankton is harmful. At the same time, an excessive number of plankton leads to toxins. Therefore, the level of plankton needs to be carefully controlled. Similarly, Posidonia Oceanica lives only in clean water and contributes to biodiversity, reduces erosion of beaches, and enhances water quality. Studying the well-being of coral substrates can help analyze the impact of global warming and excessive human activity on the water bodies and marine life [1][10], thus guiding preservation campaigns. Image processing can complement other techniques such as physiochemical analysis of water and sonar-based detection in achieving those goals. Therefore, the main objectives of the current study are:[1]

[1] CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
EMAIL: rgunti@albany.edu (A. 1); arorissa@albany.edu (A. 2)
ORCID: 0000-0002-5239-2419 (A. 1); 0000-0002-5300-617X (A. 2)

- To test the Neural Network logic on the TensorFlow platform using Python programming andcompare the results with those of our previous study [2] which were produced using MATLAB.
- To improve the performance, efficiency, and effectiveness of our predictions.
- To increase the accuracy of the Classification results.
- To effectively recognize plant images by using deep learning algorithms that improve the accuracy,kappa coefficient, and Jaccard coefficient values.

1.1. Approaches

In a previous study [2], we implemented the Convolutional Neural Network deep learning algorithm [3][4] through MATLAB and annotated the images uncategorized. Though the labeling part was done manual, the predictions and accuracy were impressive. However, unsupervised learning (the approach we took in the current study) promises to produce much-improved results compared to last year's semi-supervised approach[2] which had a number of downsides, including the inability to handle a large volume of data, incorrect classification results, and lack of adequate GPU memory [5].

On the other hand, this year, we followed two different models i.e., CNN Model [3][4] for predicting and Regression Model [6] for bounding boxes. The former model is rather a more straightforward categorical approach for Predicting, while the latter one is the complicated object detecting approach. Initially, we started training the images with ten coral substrates with ten categories. We assumed that instead of training just ten categories with hundreds of images, as most of them are online, we could train each image in ImageClef 2022 training dataset[1][10] with a label associated with a number of substrates present in that image and categorize each label with the provided annotations in "imageCLEFcoral2022_annotations_boxes_training" CSV file [1][10]. As there were more than two hundred combinations of coral classification in the annotations file, each image is in a respective directory with the manually created label of its respective annotation. Therefore, we categorized 228 distinct categories with the training images. For example, for the image ID "2018_0712_073450_057" there are nine different substrates annotated in the imageCLEFcoral2022_annotations_boxes_training CSV file. Hence, a directory with a label

"Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Hard_Coral_Mushroom_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge_and_Algae_Macro_or_Leaves" is created with a training image '2018_0712_073450_057.'

Similarly, the images in the Clef Coral 2022 training dataset were divided into 228 labels with each consisting of training images ranging from 1 to 98. The highest directory with the ninety-eight training images is categorized with the label in the combination of four substrates "Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge."

Since there are 228 categories of labels created manually with a significantly less number of training images in most of the directories, every image in the labeled directory required copying several times until it matched the total number of categories i.e., 228. For example, the training image 2018_0712_073450_057 which is labeled as

“Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Hard_Coral_Mushroom_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge_and_Algae_Macro_or_Leaves” is replicated 228 times to match the total number of labels created.

Next, to train it on the TensorFlow platform using high-level neural network libraries from Keras, the proposed model is introduced to overcome all the disadvantages that arose in the existing system. The proposed model is hoped to increase the accuracy of the results by classifying and predicting the precise bounding cox coordinates. It could enhance the performance and reliability of the overall classification results, identify underwater corals, and improve accuracy. The only downside to the proposed model isthat the computational speed for training the images is slow for large datasets. However, we could minimize the number of samples for batch training for each epoch (one training iteration).

1.2 Dataset

There are three datasets used in this study.

1. ImageClef Coral Annotation and Localization Training Dataset [1][10]
2. ImageClef Coral Annotation and Localization Testing Dataset [1][10]
3. Online coral substrate images from Google

Initially, the system was trained on online Images considering 11 substrates for training. The CNN model was tested for 11 substrates labeled as each category containing a minimum of eleven images and a maximum of hundred training images. To calculate the CNN model efficiency the conventional way was trained and tested.

However, after the ImageClef training dataset was made available for download, the image dataset was analyzed to design the CNN Model and regression model. After a thorough analysis, the 228 distinct categories of substrate combinations were spotted in the training dataset. Therefore, the dataset was created for each category with a given number of training images. In order to train the model, the system requires a minimum of 228 images in each category. Since to test the accuracy of the Model first the images have been replicated in every category of the dataset and used for training purposes. For training the model, we have created 228 labels with several replicates equivalent to the test data set created. Out of 228, we detected that only 96 categories are required to train the model for the predictions. Hence, the training dataset was eventually trained in 96 categories for predictions and confidence scores for 200 CoralClef Test dataset images.

We saved and divided all the predictions into twenty-one unique predictions together with respective test image IDs. We divided the 96 categories into twenty-one datasets for bounding boxes and coordinates and removed the replications inside each category. This means that each category was trained simultaneously with respective annotations provided in the training dataset. Finally, the classified test dataset was used to calculate the Bounding boxes for each image ID in all twenty-one folder.

2. SYSTEM ARCHITECTURE AND FLOW DIAGRAM

As the model is built on Python programming the System architecture, flow diagram, and several forms of UML diagrams are a better way to translate the code. The below diagrams, figures 1 – 7, help to analyze and fathom the structure and communication of the system.

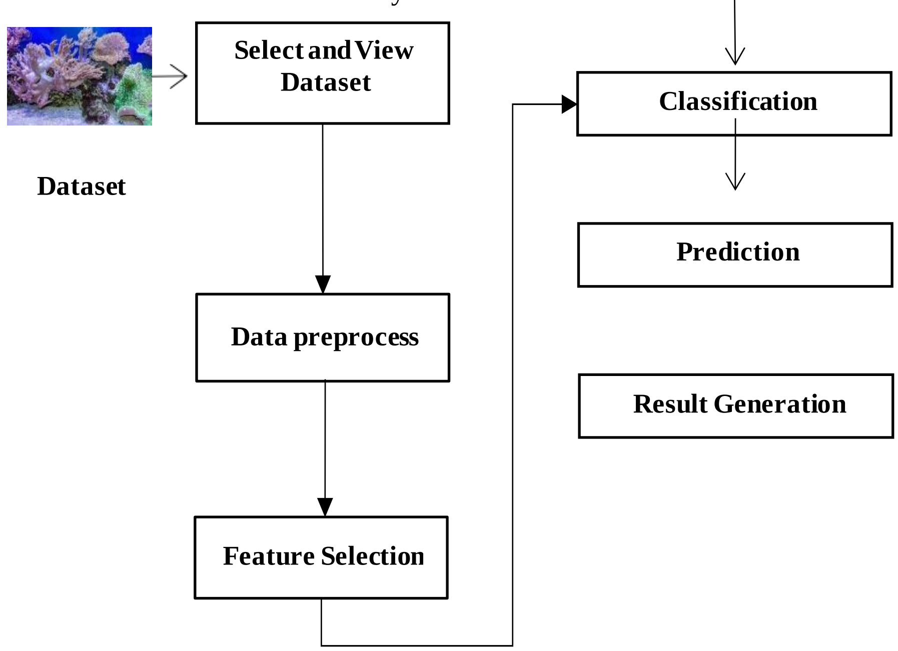


Figure 1: The above architecture of the system describes the five modules of the Convolutional Neural Network Model to classify an image into different substrates.

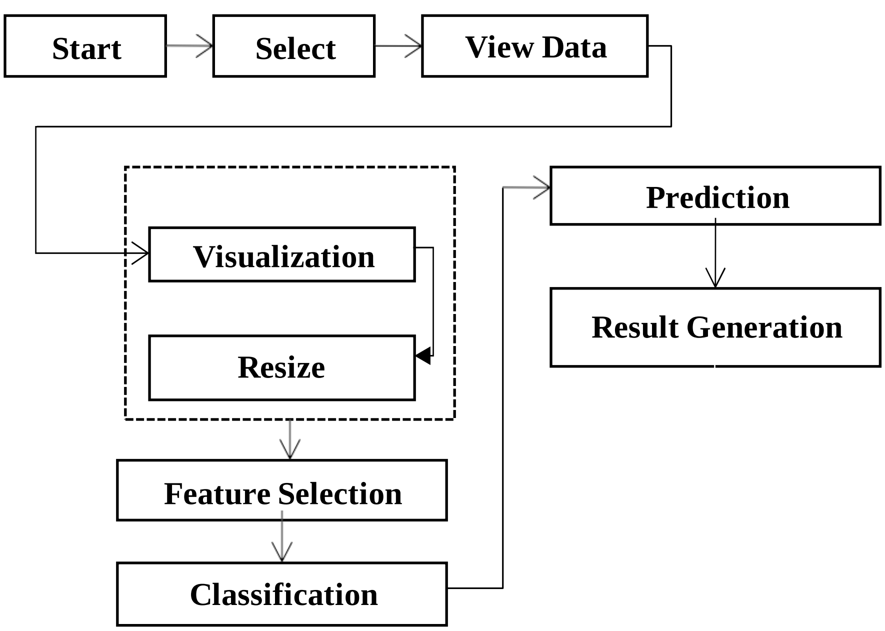


Figure 2: The above Flow Diagram of the CNN Model demonstrates the details of the process in each portrayed in Figure 1. Module One Select and View dataset and Module two Data preprocess are further divided into submodules as illustrated in the above flow diagram.

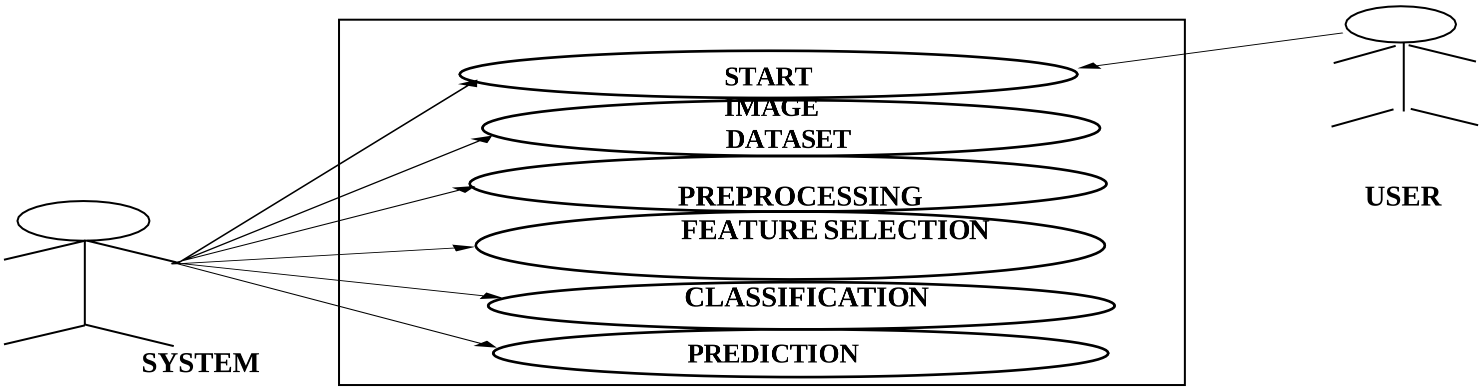


Figure 3: The above UML diagram shows the behaviors between the user and the modules within the system. As depicted the system has relationships at each module where it can take action at any stage whereas the user has only one relationship before loading the Image Dataset.

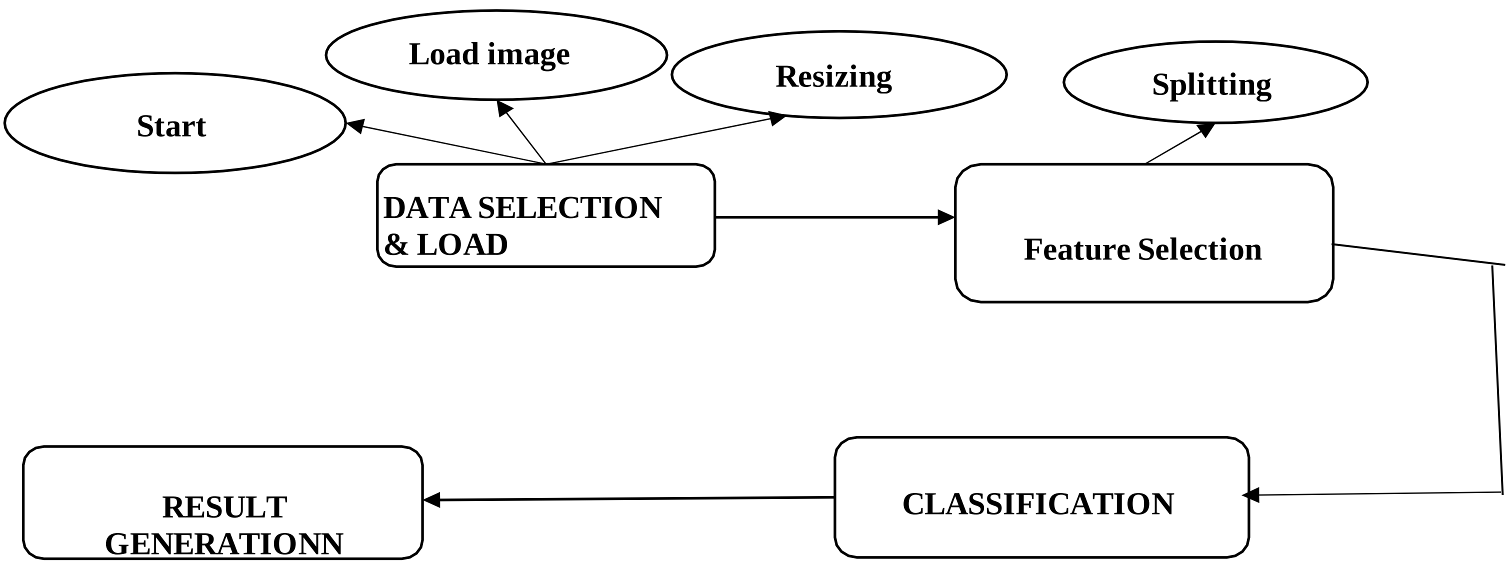


Figure 4: The above ER Diagram visualizes the data model design to demonstrate the requirements of each module.

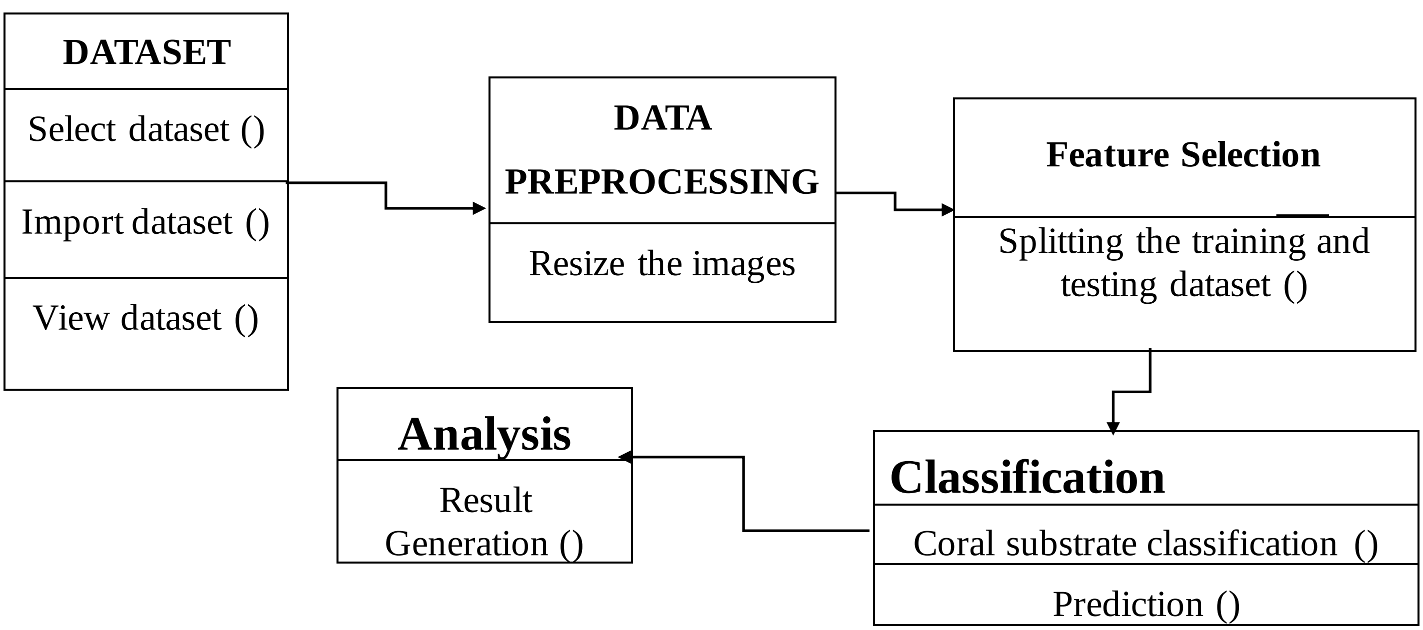


Figure 5: The above Class Diagram defines the structure of the model by defining the Module's classes and their operations.

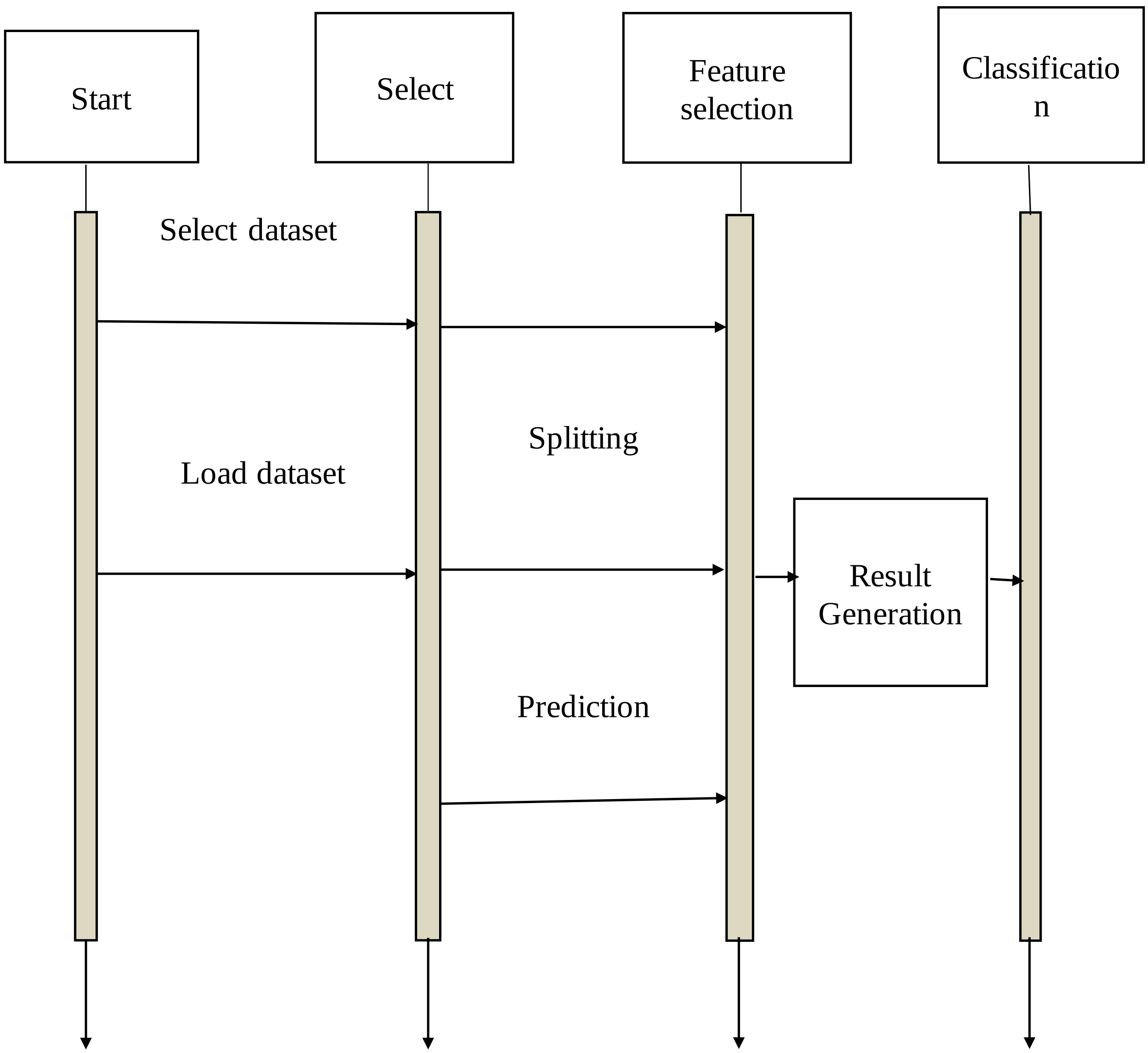


Figure 6: The above Sequence diagram describes the model design. The relationships are shown in figure 3 between the system and its modules, and the communication & interaction between the system and its modules are detailed in the above sequence diagram. All the interactions are depicted in chronological order. This diagram represents a better version of the architecture of the CNN Model (figure 1) and how different modules are interconnected and communicate.

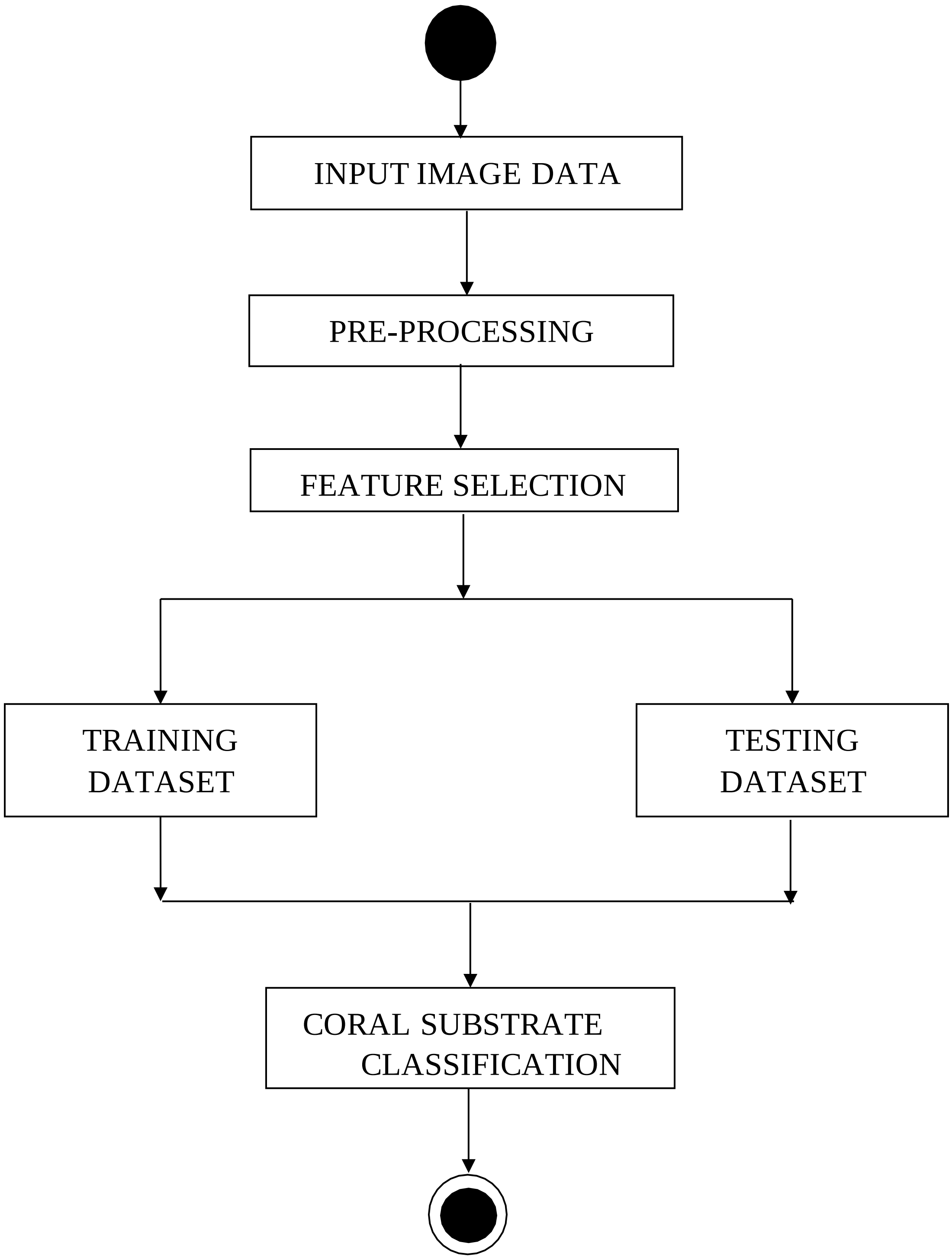


Figure 7: The above Activity Diagram is the flow diagram for the UML diagram represented in Figure 3. It shows the flow of description of brief activities and actions carried out in Figure 3 and Figure 5.

3. RESOURCES USED AND IMPLEMENTATION

We utilized a set of five modules, namely data selection and loading, data preprocessing, feature selection, classification, prediction, and result generation. Data selection is the process of selecting the Coral Images with a good blend of eleven substrates used for training in this study. Each substrate is evenly partitioned to train in the initial phases of the study to classify the type of substrate. The dataset contains twelve coral substrates like Hard Coral Boulder, Hard Coral Branching, Hard Coral Encrusting, Hard Coral Foliose, Hard Coral Mushroom, Hard Coral Submassive, Hard Coral Table, Soft Coral, Soft Coral Gorgonian, Sponge, Sponge Barrel, Micro Algae Leaves. Hence, two datasets were used in the entire study. Before the release of datasets for the competition, the model's validity was calculated using online datasets downloaded from Google. Many of the images were downloaded from the web and trained with more than fifty samples.

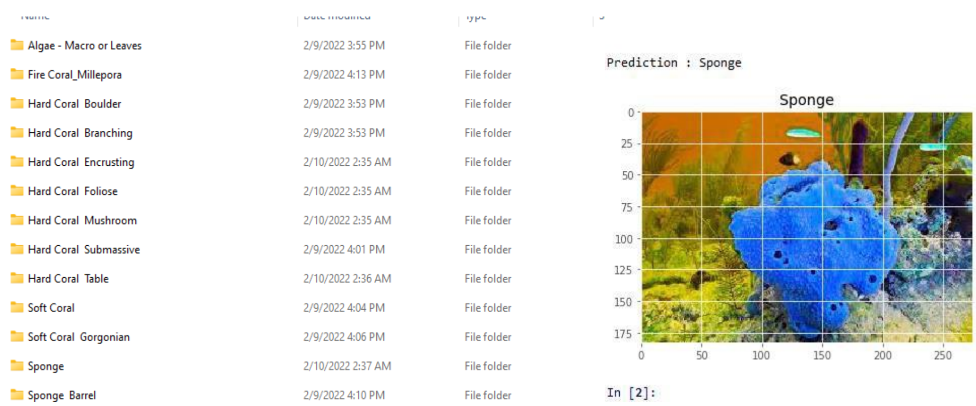


Figure 8: A sample prediction based on an online Google dataset

In the second phase, after the competition began, we trained the model with ImageClef Coral 2022 dataset. However, we changed the training process as we didn't have images with single substrates. Therefore, all the images in the dataset have been categorized and validated in 228 categories, considering the annotations in the annotation file in the training dataset.

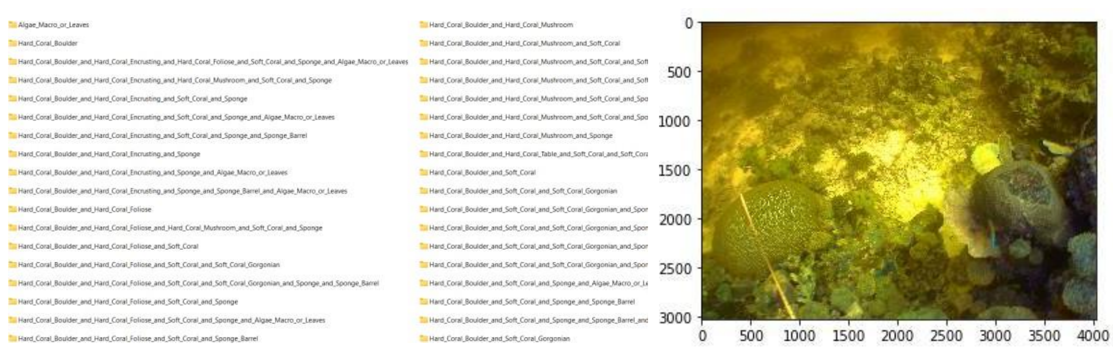


Figure 9: Unique categorical labels on the left and a sample image from the Clef Coral 2022 training set

Image Data pre-processing is the process of getting rescale data from the dataset. It involves obtaining the data and resizing images in the dataset that involves rescaling the size of the remote sensing scene dataset images to 50. Categorical data is defined as variables with a finite set of rescaled values. That most deep learning algorithms require an array of input and output variables. In the current study, resizing was done mainly for the initial phase, where we trained individual images with more than fifty images for each substrate. Hence there were different datasets involved before downloading the actual Coral 2022 dataset. The resizing for every image with different shapes and dimensions is monitored explicitly as shown in Figure 9.

Data splitting is the act of partitioning available data into two portions, usually for cross-validate purposes. A portion of the data is used for developing a predictive model while the other portion is used to evaluate the model’s performance. Separating image data into training and testing sets is also an important part of evaluating image processing models. Typically, when one separates a data set into a training set and testing set, most of the image data is used for training, and a smaller portion of the data is used for testing. When training most of the substrates individually, the training is set to 80 percent of the dataset and the labels of each substrate. The remaining 20 percent is validated for testing as demonstrated in Figure 10. As each substrate contained images in the range from fifty to one hundred fifty, our results were very encouraging.

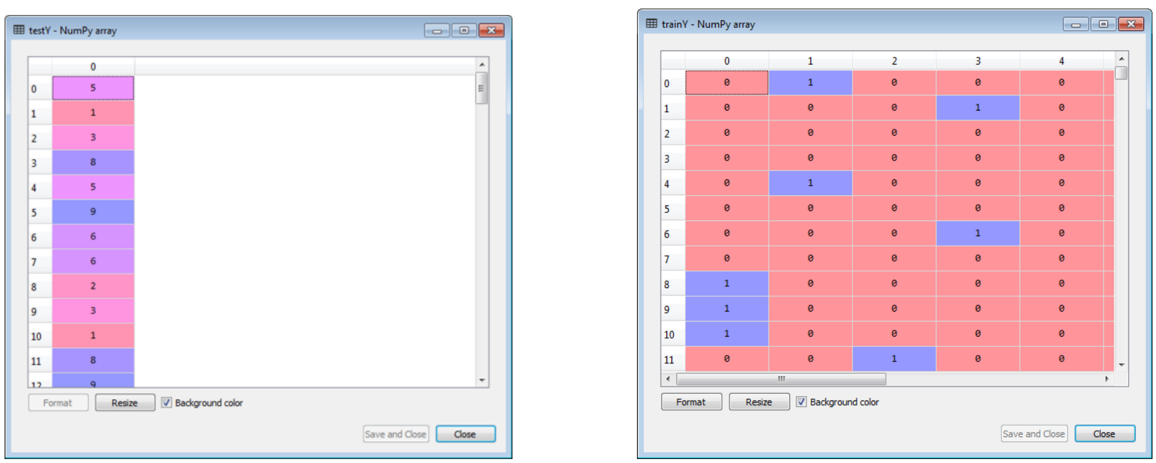


Figure 10: The validation for the ImageClef dataset was divided into eighty percent for training and twenty percent for testing. All the categories were split into four datasets depending on different locations from Indonesia, Seychelles, and the Caribbean. The model was programmed to train with an equal or a greater number of images in each folder with a total number of labels. Since there are 228 combinations of annotations from all locations in the training dataset, we have six locations partitioned training process into six rounds where we trained N categorical folders with each containing more than or equal to N image samples. However, the test set is from a single location K1, and we primarily trained 96 categorical folders with each containing more than or equal to 96 samples. Since some folders have very few sample images, we duplicated the samples to make the total sample images in the folder add up to at least 96. The above figure displays the validation set when classifying the ten substrates initially. The right side showsthe Boolean values trainY array that stores the true labels wherever the predicted labels, for all 96 categories, match the true labels and False everywhere else. On the left, we have the number of correct predicted labels for each iteration through which we generate the classification report and calculate the Cohen Kappa Coefficient, Jaccard similarity Coefficient, and confusion matrix

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks most commonly applied to analysing visual imagery[3][4]. They are used in image and video recognition applications, recommender systems, image classification, medical image analysis, natural language processing, brain-computer interfaces, and financial time series[3][4]. CNNs are regularized versions of multilayer perceptron. Multilayer perceptron is usually a fully connected network, that is, each neuron in one layer is connected to all neurons in the next layer [3][4]. The “full connectedness” of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach toward regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNN’s fall on the lower scale. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted visual field region known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

Prediction is a process of recognizing the substrate from the dataset by using a deep learning model. In our project, we attempted to effectively predict the classification of images from the dataset by enhancing the performance of the overall prediction system. To do that, we programmed two models—one for prediction and another for bounding box visualization. In the prediction model, out of 200 test data images, the prediction model used a batch size of 32 samples with 25 epochs (iterations). This model was compiled using a categorical cross-entropy loss evaluator and Adam optimizer and trained with 96 pre-processed class labels with NumPy categorical to convert the array of labeled data to vector as shown in figure 11.

```
(trainX, testX, trainY, testY) = train_test_split(data, labels, test_size=0.2, random_state=42)
# Preprocess class labels
trainY = np_utils.to_categorical(trainY, 96)

#================================Classification================================
'''Convolutional Neural Network'''

print('Convolutional Neural Network')
model = Sequential()

model.add(Convolution2D(32, (2, 2), activation='relu', input_shape=(HEIGHT, WIDTH, N_CHANNELS)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(32, (2, 2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(96, activation='softmax'))
```

```
This image 2018_0712_073241_021 most likely belongs to Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge with a 0.41 Confidence Score.
```

```
print(model.summary())

model.fit(trainX, trainY, batch_size=32, epochs=25, verbose=1)
print("Accuracy of the CNN is:",model.evaluate(trainX,trainY)[1]*100, "%")
history=model.history.history
#Plotting the accuracy
train_loss = history['loss']
train acc = history['acc']
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	416
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 31, 31, 32)	4128
max_pooling2d_1 (MaxPooling2	(None, 15, 15, 32)	0
dropout (Dropout)	(None, 15, 15, 32)	0
flatten (Flatten)	(None, 7200)	0
dense (Dense)	(None, 128)	921728
dense_1 (Dense)	(None, 96)	12384

Total params: 938,656
Trainable params: 938,656
Non-trainable params: 0

Figure 11: Convolutional Neural Network Model for Clef Coral training Dataset 2022

After testing with 200 image samples in the Clef Coral dataset, the predictions were classified into twenty-one unique labels as shown in figure 12. For each label, a new bounding box regression model [6] was compiled using an mse loss evaluator and Adagrad optimizer and trained with VGG16 network ensuring the head FC layers. All the predicted values were then scaled on image dimensions to draw the bounding boxes. However, there has been an overlap between the substrates for some images and some images predicted only single substrates. We assigned eleven different colors for eleven substrates.

Categories	Number of Images
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral	8
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge	104
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge_and_Sponge_Barrel	24
Hard_Coral_Boulder_and_Soft_Coral	1
Hard_Coral_Boulder_and_Soft_Coral_and_Sponge	9
Hard_Coral_Boulder_and_Soft_Coral_and_Soft_Coral_Gorgonian	2
Hard_Coral_Branching_and_Soft_Coral_and_Sponge	1
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Hard_Coral_Mushroom_and_Soft_Coral_and_Sponge	24
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Hard_Coral_Mushroom_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge	3
Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge	6
Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Table_and_Soft_Coral_and_Sponge	4
Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Soft_Coral_and_Sponge_and_Algae_Macro_or_Leaves	1
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge_and_Algae_Macro_or_Leaves	1
Hard_Coral_Branching_and_Hard_Coral_Foliose_and_Soft_Coral_and_Sponge_and_Sponge_Barrel	1
Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Soft_Coral_and_Sponge_Barrel	1
Hard_Coral_Branching_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge	1
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge	4
Hard_Coral_Branching_and_Hard_Coral_Boulder_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Algae_Macro_or_Leaves	1
Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Hard_Coral_Mushroom_and_Soft_Coral_and_Soft_Coral_Gorgonian_and_Sponge_and_Algae_Macro_or_Leaves	1
Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Table_and_Hard_Coral_Foliose_and_Soft_Coral_and_Sponge_and_Algae_Macro_or_Leaves	1
Hard_Coral_Branching_and_Hard_Coral_Submassive_and_Hard_Coral_Boulder_and_Hard_Coral_Foliose_and_Soft_Coral_and_Sponge_and_Algae_Macro_or_Leaves	2

Figure 12: A sample of 200 test dataset prediction categories/labels classified into twenty-one unique categories. The number of images per category is demonstrated in the above figure.

We have encountered some images with a couple of mismatchings as shown in figure 13.

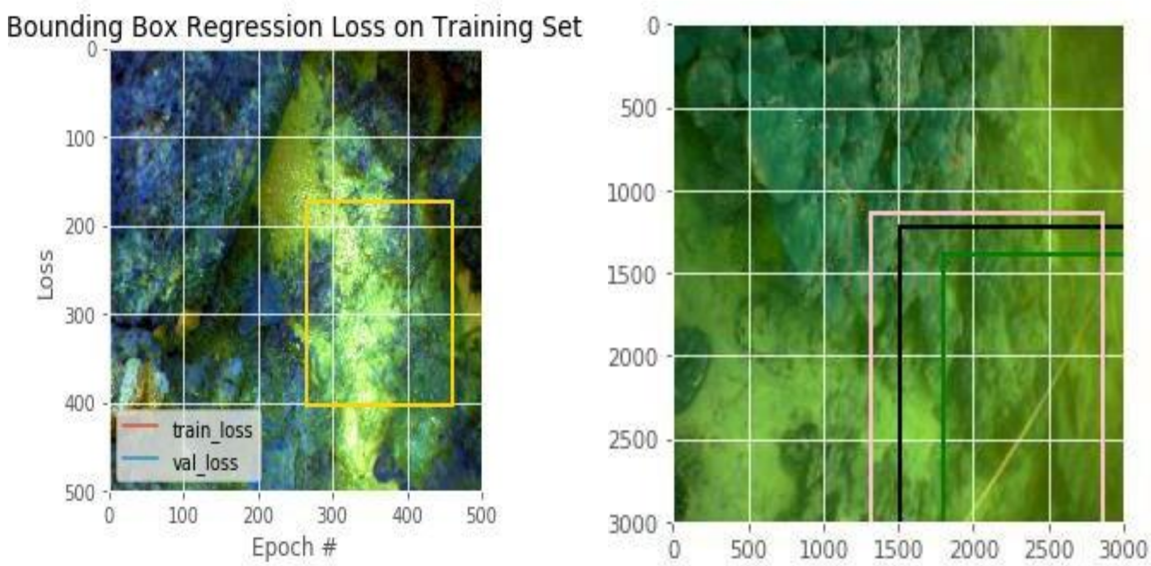


Figure 13: Measuring errors and overlapping in Bounding Box Visualization for two samples in test dataset

4. RESULTS

We generated the Final Result[10] based on overall classification and prediction. The performance of this proposed approach is evaluated using some measures such as accuracy, precision, recall, F1-measure, Kappa Coefficient, and Jaccard Coefficient[7][8] as shown in figure 14 and figure 15.

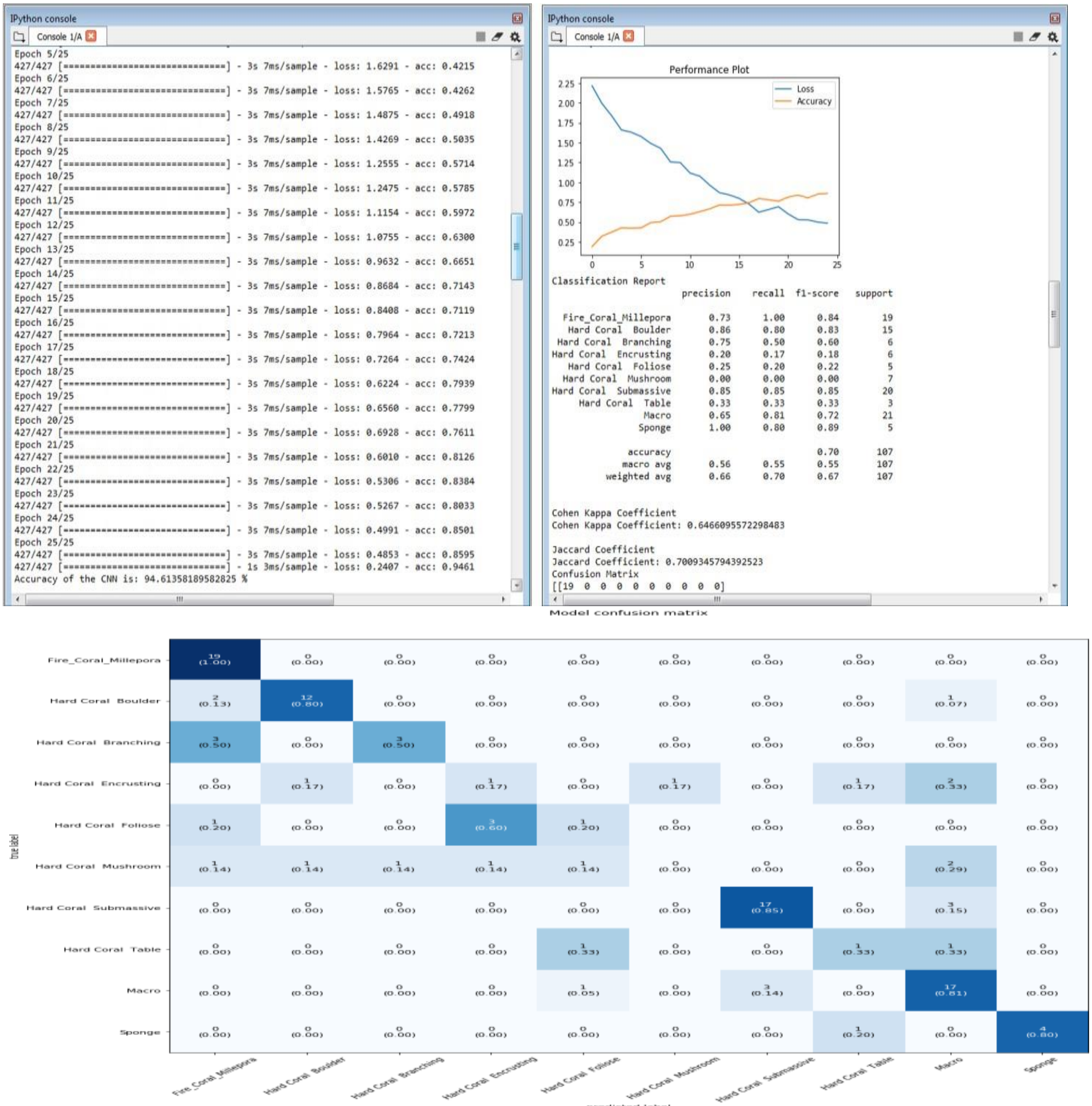
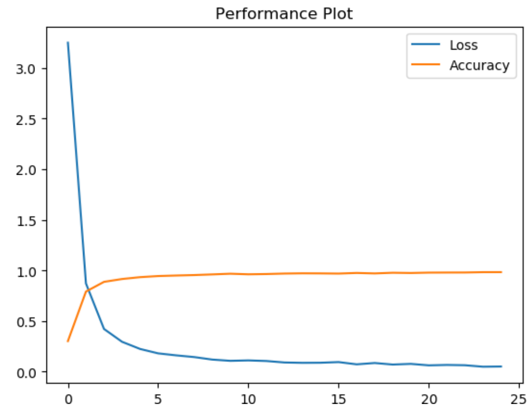


Figure 14: Above figure demonstrates the Accuracy (on left), precision(on right), recall(on right), f1-score(on right), and confusion matrix(on bottom). The confusion matrix portrayed above is for the initial dataset downloaded online. It portrays the number of true labels for each coral substrate throughout the training dataset and their predicted labels.




```
Cohen Kappa Coefficient
Cohen Kappa Coefficient: 0.9967666665613402

Jaccard Coefficient
Jaccard Coefficient: 0.9968017057569296
Confusion Matrix
[[23  0  0 ...  0  0  0]
 [ 0 20  0 ...  0  0  0]
 [ 0  0 20 ...  0  0  0]
 ...
 [ 0  0  0 ... 19  0  0]
 [ 0  0  0 ...  0 19  0]
 [ 0  0  0 ...  0  0 21]]
```

Figure 15: Performance Measures for ImageClef Coral Annotation and Localization training dataset

Our initial plan was to submit as many runs as possible, considering the previous run scores and observations. Therefore, each unique prediction was trained again for the boxing visualization with the training dataset and annotations with the same category taken from ninety-six image categories of the training dataset and training annotation file. We used the callback function for every iteration to get the maximum number of predictions with a difference in accuracy to get the most accurate prediction[8]. As saving results of the predictions in the developed Model from the callbacks require high GPU memory[5], the bounding box model was considered training on lower batch levels and higher epochs(iterations). This model was developed to take input image shapes to resize and reshape them into four-dimensional 224 x 224 image sizes and then gives the most desirable predictions from the VGG16 trained model. As mentioned earlier, the dataset itself trained in this model belongs to the similar category of the input image that predicts most predictions from the trained model to improve the recall and precision. To calculate the coordinates, the NumPy array value in test data is multiplied by the exact dimensions of the input image for each saved NumPy array prediction and visualized the coordinates onto the input image.

To submit the runs, the task was divided into two parts. One for the prediction and confidence score using the prediction Model followed by the bounding cox coordinates using the bounding box model. The precision and recall were tested in the first run by training the system without the prediction callbacks and fewer iterations with 32 batch sizes for every unique dataset. In the second run, the system was developed, and the boxing regression model was retrained with more iterations, a batch size equal to the total number of training dataset images, and iterations equal to the total number of trained annotations. Hence, the observations from the first run gave a lead in improving the model, and the second run results were improved. Though we planned to retrain the system to submit more runs to increase the scores, we decided to postpone it for future campaigns. Though we wanted to retrain both models with more accurate results, we only managed to train Bounding Box Regression Model[6] for the next run. However, one can see the difference in the scores.

Primary run: Recall: 0.001; Precision: 0.001

Soft Coral – 200 test dataset images with an average of 2000 annotations

Hard Coral Boulder- 197 test dataset images with an average of 985 annotations

Sponge – 187 test dataset images with an average of 561 annotations

Hard Coral Branching -186 test dataset images with an average of 558 annotation

Secondary Run: Recall: 0.002; Precision: 0.003

Even though there is an increase for the second run, the score was not satisfying. So, to verify and validate the score the submitted predictions were converted into XML file to visualize the coordinates on the images. We see a huge overlapping of images when we visualized the annotations as shown in figure 16. The actual reason is because the displacement of (Xmin1, Ymin1) coordinates with (width, height) in the submission format.

```
[image_ID];[substrate1] [[confidence1,1]:][width1,1]x[height1,1]+[xmin1,1]+[ymin1,1],
[[confidence1,2]:][width1,2]x[height1,2]+[xmin1,2]+[ymin1,2],...;[substrate2] ...
```

Above is the actual submission formation that should be for ClefCoral 2022. However, all our coordinates [Xmin1,1] and [ymin1,1] are misplaced with [width1,1] and [height1,1]

For example:

Our actual result for a test dataset sample while running on Spyder

```
algae_bbox=[2259,2051,302,267]
gorgorian_bbox=[1885,1986,299,229]
sponge_bbox,sponge2_bbox,sponge3_bbox=[1942,1743,242,215],[1773,1844,264,221],[1885,1884,268,242]
branching_bbox,branching2_bbox,branching3_bbox,branching4_bbox,branching5_bbox = [1881,1112,844,313],[1859,1642,793,
soft_bbox,soft2_bbox,soft3_bbox,soft4_bbox,soft5_bbox,soft6_bbox,soft7_bbox,soft8_bbox,soft9_bbox,soft10_bbox = [190
submassive_bbox,submassive2_bbox,submassive3_bbox = [1892,1690,279,180],[1743,1920,309,177],[1888,1784,260,197]
foliose_bbox,foliose2_bbox = [1978,1613,209,249],[1920,2056,308,250]
mushroom_bbox,mushroom2_bbox,mushroom3_bbox = [1758,1851,260,231],[1967,1678,218,232],[1815,1896,269,226]
```

Below are our coordinates misplaced in the submission text file

```
2018_0712_073320_027;c_hard_coral_branching
0.7:844x313+1881+1112,0.7:793x195+1859+1642,0.7:516x160+1842+1617,0.7:442x154+1732+1823,0.7:331x180+2034+1726;c_hard_coral_submassive
0.6:279x180+1892+1690,0.6:309x177+1743+1920,0.6:260x197+1888+1784;c_sponge
0.6:242x215+1942+1743,0.6:264x221+1773+1844,0.6:268x242+1885+1884;c_hard_coral_foliose
0.8:209x249+1978+1613,0.8:308x250+1920+2056;c_hard_coral_mushroom 0.7:260x231+1758+1851,0.7:218x232+1967+1678,0.7:269x226+1815+1896;c_soft_cor
0.7:244x231+1900+1782,0.7:213x219+1870+1628,0.7:285x223+1870+1951,0.7:287x226+1847+1954,0.7:206x228+1878+1590,0.7:287x241+1938+1961,0.7:252x23
,0.7:254x240+1898+1810,0.7:250x252+2082+1806,0.7:247x225+1746+1748;c_soft_coral_gorgonian 0.9:299x229+1885+1986;c_algae_macro_or_leaves
0.6:302x267+2259+2051
```

When we visualized the correct coordinates by converting them into an XML file below are the results.

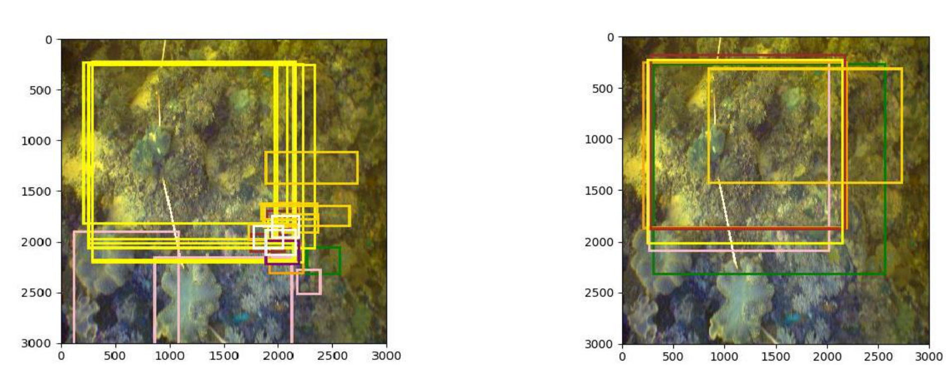


Figure 16: left image displays the actual annotations, while the right image displays the submitted annotations in the text file

5. Discussion and Concluding Remarks

In this study, the deep learning classifier analyzed the different types of coral substrates. Two models were developed for different purposes, and the coral images were taken as input data and applied to the pre-processing method. The images were resized and converted into an array in the pre-processing method. Then it was processed into the feature selection method; in this method, the dataset is split into a training dataset and a testing dataset. However, the partitioning of data was prioritized differently in both Models. Since the first model was developed for predictions and confidence scores, it was partitioned into categorical labels. In contrast, the second model was developed to calculate the bounding box coordinates and visualize them on the input test image. All the images are resized and converted into an array in both models. Finally, the first model that demonstrates the deep learning algorithm of CNN was implemented and predicted the result based on accuracy, precision, recall f1-measure, kappa coefficient, and Jaccard coefficient [7]. In contrast, the second model was the bounding box regression model, which visualizes and enables predict continuous values.

Our findings indicate that it will be possible to improve prediction accuracy as we increase the number of iterations and submit the annotations in the required format. Since all the configuration changes are only made on the Bounding Box Regression Model[6] for the second run, we see the improvement. We strongly believe that retraining both models for this study would improve the Accuracy and Precision as we might see the difference in prediction levels. We utilized and built an unsupervised system that can be used in any field of science. The unsupervised training algorithm trains the images to detect the sensing scene. It will also enhance Graphical User Interface to find the underwater coral type from the image. This approach of training will provide valid predictions, and it will enhance the performance and increase the accuracy, and visualizes multiple annotations[9].

6. REFERENCE

[1] Jon Chamberlain, Antonio Campello, Jessica P. Wright, Louis G. Clift, Adrian Clark and Alba García Seco de Herrera, "Overview of the ImageCLEFcoral 2022 Annotation and Localisation Task," Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022), September 2022, (CEUR-WS.org), Bologna, Italy.

[2] Gunti R.R, Rorissa A, A convolutional based neural networks based coral reef annotation and localization (2021), CEUR Workshop Proceedings, 2936:1229-1238.

[3] Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE transactions on medical imaging, 35(5), 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

[4] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2012).

[5] D. Strigl, K. Kofler, and S. Podlipnig, "Performance and Scalability of GPU-Based Convolutional Neural Networks," Euromicro Conference on Parallel, Distributed, and Network-Based Processing (2010), pp. 317-324.

[6] A. Rosebrock, Object detection: Bounding box regression with Keras, TensorFlow, and Deep Learning (2020). URL: https://pyimagesearch.com/2020/10/05/object-detection-bounding-box-regression-with-keras-tensorflow-and-deep-learning/

[7] Brownlee J. Machine Learning Mystery: How to Calculate Precision, Recall, F1, and More for Deep Learning Models?, Deep Learning (2019). URL: https://machinelearningmastery.com/how-to-calculate-precision-recall-f1-and- more-for-deep-learning-models/

[8] Hossini A. Custom Callback for transformers: How to quantify F1-Score when fine-tuning pre-trained transformers for NLP tasks?, Towards Data Science (2021). URL: https://towardsdatascience.com/custom-callback-functions- for-transformers-ae65e30c094f

[9] D. Cireşan, et al., "Flexible, High-Performance Convolutional Neural Networks for Image Classification," in International Joint Conference on Artificial Intelligence (2011), pp.1237-1242.

[10] Bogdan Ionescu, Henning Müller, Renaud Péteri, Johannes Rückert, Asma Ben Abacha, Alba García Seco de Herrera, Christoph M. Friedrich, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Serge Kozlovski, Yashin Dicente Cid, Vassili Kovalev, Liviu-Daniel Ștefan, Mihai Gabriel Constantin, Mihai Dogariu, Adrian Popescu, Jérôme Deshayes-Chossart, Hugo Schindler, Jon Chamberlain, Antonio Campello, Adrian Clark, Overview of the ImageCLEF 2022: Multimedia Retrieval in Medical, Social Media and Nature Applications, in Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022), Springer Lecture Notes in Computer Science LNCS, Bologna, Italy, September 5-8, 2022**.**