International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

BBCNN: Bounding Box Convolution Neural Network for Cyclone Prediction and Monitoring

Mamatha N.P [1], Mohmad Umair Bagali[2], Thangadurai N[3]

Submitted: 14/11/2022 **Accepted:** 15/02/2023

Abstract*:* The correct identification of tropical cyclones is essential for the prevention and preparation for catastrophic events that they cause. The prediction of a cyclone's surge, as well as the monitoring of its severity, is an important part of the predictive model. Despite the significant efforts made each year, a significant number of individuals still lose their lives as a consequence of cyclones. More appropriate predictive methods need to be developed to reduce the severity of this harm. Deep learning techniques provide perks in detecting challenges since they can increase the prediction algorithm's stability and efficiency. The method discussed here uses artificial neural networks to analyse MOSDAC satellite imagery. With the help of satellite data, a several-layer neural-net model was trained to predict cyclones, much like biological visual perception. The findings suggest that the method has the potential to be further refined into an efficient instrument for cyclone track predictions by making use of a variety of different kinds of remote sensing imagery and information.

Keywords: *Cyclone Detection, Cyclone Intensity, Deep Learning, Predictive Model, Satellite imagery*

1. Introduction

Tropical cyclones (TC) are among the most destructive types of severe weather catastrophes. They originate over the warm waters of the tropics and significantly influence the global economy. These types of storms, when they reach their peak intensity, are referred to as hurricanes in the Atlantic Ocean, typhoons in the western North Pacific, and cyclones in the Bay of Bengal. On a worldwide scale, an annual average of ninety tropical cyclones form over the warm waters of the tropics [1]. Knowing the path that a TC will take is essential for figuring out which places it will impact. As a result of climate change, tropical cyclones are becoming more powerful in terms of their destructive potential [2].

Fig.1. The Formation of a Cyclone Eye (Hurricane Kate on October 4, 2003)

Surface winds in a TC can reach 80 m/s or more in its centre, and the circulation of the entire system can span hundreds of kilometres. As a result of the Coriolis effect, the cyclone circulation moves in a counter-clockwise direction in the northern hemisphere and a clockwise one in the southern part [3]. Meteorologists of hurricanes can greatly benefit from knowing whether or not an eye has formed, as this is frequently a precursor to the substantial development of tropical cyclones [4].

Many people are at risk from cyclones because they occur most frequently in tropical seas, where population densities are also the highest [5]. Cyclones that have made headlines include the ones that impacted Bangladesh on November 12, 1970, and May 24, 1985, both of these cyclones crossed the shoreline at Chittagong, as well as Hurricane Camille, which made landfall in the United States on August 17, 1969; Cyclone Tracy, which swept over the Australian coast on December 25, 1974. Every one of these cyclones has resulted in several deaths and extensive property loss.Table 1 presents some of the recent tropical cyclones formed in the Bay of Bengal and the Arabian sea region along with their related details

Table 1. Most violent cyclones in India in the Bay of Bengal and Arabian Sea region since 2019

Name	**Time**	**Year**	**Region**	**Affected sub-regions**	**Saffir-Simpson scale**
Fani	April 25 to May 4	2019	Bay of Bengal	Eastern	Category 4
Bulbul	October 28 to November	2019	Bay of	Eastern	Category

[1] *Research Scholar, Department of ECE, Jain University, Bangalore*
[2] *Jain University, Bangalore -562112, ORCID ID : 0000-0001-5342-4067*
[3] *Center for Research and Innovation, Sankalchand Patel University, Gujarat – 384315, ORCID ID : 0000-0002-2149-9440*

	11		Bengal	North Eastern	1
Amphan	May 15-21	2020	Bay of Bengal	Eastern North Eastern	Category 4
Nisarga	May 31 to June 4	2020	Bay of Bengal	Western Central	Category 1
Nivar	November 22-27	2020	Bay of Bengal	Southern	Category 1
Burevi	November 29 to December 5	2020	Bay of Bengal	Southern	Tropical storm
Yaas	May 24-27	2021	Bay of Bengal	Eastern	Tropical storm
Asani	May 7-11	2022	Bay of Bengal	Southern	Tropical storm
Sitrang	October 23-24	2022	Bay of Bengal	North Eastern	Tropical storm
Vayu	June 8-18	2019	Arabian Sea	Southern Western	Category 1
Hikka	September 20-26	2019	Arabian Sea	Did not hit	Category 1
Kyaar	October 22 to November 3	2019	Arabian Sea	Did not hit	Category 4
Maha	October 28 to November 11	2019	Arabian Sea	Western	Category 3
Tauktae	May 14-19	2021	Arabian Sea	Western Northern	Category 4
Gulab	September 24 to October 4	2021	Arabian Sea	Southern Eastern Central Western	Category 1

Nevertheless, frequent but less intense cyclones cause human losses and significant economic damage in tropical regions. These cyclones can have less wind speed and inferior surge levels. The study, forecasting, and promotion of socioeconomic adaptation for cyclonic storms and cyclone-generated disasters have received considerable funding on a global scale. This research work aims to enhance the accuracy of early warning systems by automating the process of predicting cyclones.

The present TC data archives are plagued with ambiguity as a result of the data-collecting process being carried out manually. Most of the time, it has been seen that the dataset size is not ample enough. Even if some datasets have many images, they have countless unrelated data. The irrelevant pictures end up being a reason for garbage input. When it comes to deep learning procedures, the improved quality of the inputted data, the greater the likelihood that the output produced will also be of high quality. This relationship poses a significant difficulty to analytics teams when it comes to determining the appropriate data to assist in resolving business issues. This is why a dataset with relevant satellite images and a large number of image data is required to design and develop a machine learning-based image processing algorithm for cyclone prediction and monitoring**.**

2. Literature Review

Piñeros et al. described a method that may be used to objectively tell apart cloud clusters forming from those not amid cyclogenesis [6]. In order to measure the axisymmetry of a cloud cluster within a certain diameter, a statistical test of the direction of the luminance-temperature gradient is carried out. Jaiswal et al. proposed a new approach for automated detection of the centre of tropical cyclones by isolating the spiralling patterns inside it using infrared (IR) pictures captured by satellites [7]. Fiolleau et al. experimented with monitoring mesoscale convective systems (MCS) using satellite IR data in the tropics to solve the problems of representing convection at the 'real' size and of presenting cohesive MCS phases [8].

It is crucial to know where a cyclone's eye is located in order to observe its progress and make accurate predictions. Later in 2013, an empirical method was given by Jaiswal et al. for locating the exact centre of the cyclone in similar kinds of pictures [9]. Employing EFD and PCA approaches, Dutta et al. conducted a semi-automatic computational approach to the cyclone storm strength development trend in precompiled visible (VIS) and improved infrared (EIR) satellite pictures [10]. Later, Jin et al. worked on similar issues and proposed an approach for locating salient regions, with the salient region atlas primarily including the rain belts of a tropical hurricane, as shown in SAR pictures. The method provided by Kar et al. for pinpointing an image's region of focus involves measuring the distance from the image's centre of gravity to any other noteworthy features [11]. The method involved determining the centre of mass and the mean distance across significant sites based on the Euclidean and Manhattan range measurements. Kim et al. asserted that traditional ML methods are superior to Linear Discriminant Analysis (LDA) for identifying cyclones [12]. Kar et al. presented an image-processing-based approach for estimating tropical cyclone strength using satellite imagery. As part of the classification process, a perceptron system used geometric information included in tropical cyclone images [13].

In order to tackle climatic pattern recognition issues, Liu et al. created a deep CNN categorisation technique and showed that Deep Learning is effective for finding trends in climate-related detection problems [14]. Matsuoka et al. presented a strategy based on ensemble learning to foresee the onset of tropical cyclones [15]. In order to combine historical trajectory information with meteorological imagery, Giffard-Roisin et al. devised a neural network algorithm. The 24-hour tracking prediction was based on data collected from a region of interest that moves in relation to the cyclone's eye [16].

3. Problem Definition and Objective

3.1. Problem Definition

Accurate identification of tropical cyclones (TC) is critical for preventing and controlling natural catastrophes linked with tropical cyclones. In the process of predicting TCs, it is necessary to anticipate storm surges and identify their intensities. The ability to accurately anticipate and identify the severity of tropical cyclone intensity is essential for reducing the number of lives lost and the amount of property that can be damaged as a result of these storms. Detection and further monitoring of tropical cyclones can be done using machine learning and deep learning techniques, but the efficient implementation of these models on real-time data is essential. The detection model uses satellite infrared image data for analysing and predicting the cyclone intensity. On the other hand, it is challenging to attain good precision in the situation of limited training instances in image recognition and other domains. Therefore, deep learning models often need a huge number of training examples.

3.2. Objective

So, the aim of this study is to develop a novel technique based on deep learning methods using satellite infrared image data to detect the cyclone intensity at the cyclone's eye.

So, the objective of this research work is as follows:

- To develop an algorithm from satellite images by using a hybrid deep learning Methodology for the prediction and monitoring of cyclones.

4. Source of Data Collection

The image dataset used here is INSAT3D Infrared & Raw Cyclone Imagery (2012-2021). It includes all of the infrared and raw cyclone images that Indian National Satellite-3D (INSAT3D) took over the Indian Ocean, Bay of Bengal and Arabian sea from 2012 to 2021, together with the intensity of each cyclone image in Knots. The Raw Data came from the Meteorological and Oceanographic Satellite Data Archival Centre (MOSDAC) server. The images were collected by KALPANA 1 satellite. Each image was labeled by finding the timestamp and its appropriate coordinates in the intensity-time plot of every cyclone subdirectory. Figure 2-4 are the visualizations of sample data present in this dataset. Figure 2 depicts the sample set of raw image data captured using INSAT3D satellite. Figure 3 shows the infrared versions of the raw image data. Figure 4 shows a four-by-four display Grid (Raw Image, Brigtness temperature Contours, Infrared Image and Cloud height).

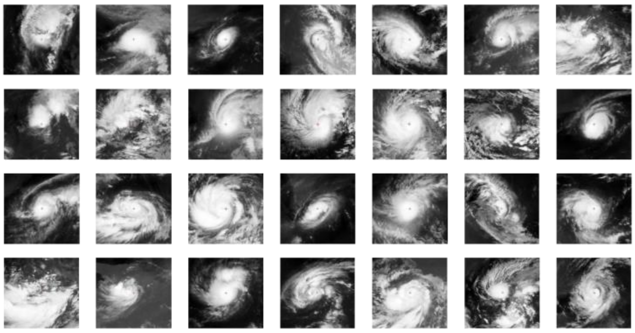


Fig.2. Set of Raw data Images (INSAT-3D TIR1)

We used the infrared images present in this dataset (figure 3) for the cyclone category prediction. The intensity of each corresponding image file name was stored in a '.csv' file for users.

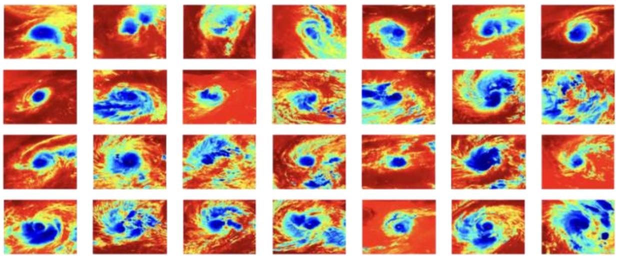


Fig.3. Set of Infrared images

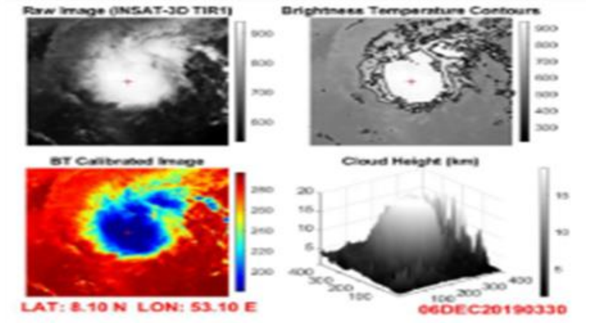


Fig.4. Sample of different types of Images in the dataset: Raw Image, Brightness temperature Contours, Infrared Image and Cloud height

5. Methodology

5.1. Methodology Workflow

The platform and tools utilized in this experimental study are presented below.

- Python coding and execution platform for the research work: Google Collaboratory
- Graphical plotting and Image Generation: Matplotlib
- Data Preprocessing: Python and TensorFlow Keras
- Model Design and Execution: TensorFlow Keras

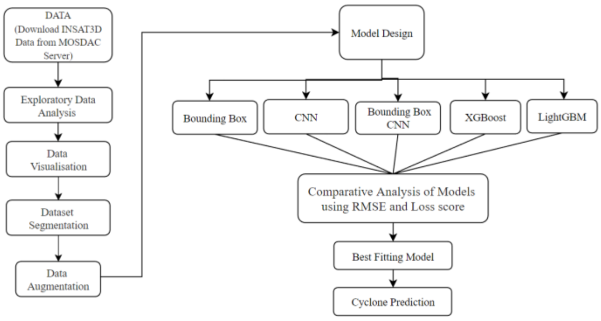


Fig.5. The Workflow of this Research

The workflow of this research work is presented in the flowchart shown in figure 5. We aim to predict cyclone intensity for forecasting and monitoring cyclones. We collected the image data (INSAT3D dataset) required for this study from the MOSDAC server. We did an exploratory data analysis on this dataset to know detailed statistics. The analysis is then presented via data visualization. Then we split the dataset into training and testing subsets. After this, data augmentation was implemented on this dataset. Then we designed different models for cyclone prediction. We intend to test the input data on these four models, which are XGBoost, LightGBM, Bounding-Box, Custom CNN, and Bounding Box-CNN ensemble, for this prediction purpose. The results from those models were then compared using RMSE, Loss scores and accuracy parameters for model assessment. After this step, the best-fitting model was selected for this prediction work.

5.2. Data Preprocessing and Visualisation

Since tropical cyclones are powered by the latent heat generated by condensation, they can only operate over water and quickly dissipate if they encounter ground. In addition to possessing surface winds that regularly surpass 100 knots, a core ocean-level pressure of 900 MB or below is a defining characteristic of this system. They are strongest over the tropical seas and decrease as they approach inland.

Table 2. The categories of cyclones according to the Saffir– Simpson scale and Categories used in this study

Saffir-Simpson Scale Category	**Proposed Study Category**	**Category Code**	**knots (kn)**
5	Severe Cyclone	3	≥ 137 kn
4	Severe Cyclone	3	113–136 kn
3	Severe Cyclone	3	96–112 kn
2	Severe Cyclone	3	83–95 kn
1	Severe Cyclone	3	64–82 kn
TS	*TS*	2	*34–63 kn*
TD	*TD*	1	*≤ 33 kn*

Due to the fact that tropical cyclones can range in intensity, we can group them into distinct categories. We used the Saffir–Simpson scale for the classification of tropical cyclones. The highest sustained wind velocity is used as the sole criterion for a cyclone's classification on the 1–5 Saffir–Simpson Scale (table 2). Other than those five categories, there are two other classes of tropical cyclones which fall below category one on the basis of intensity. All these categories are described below from lowest intensity to highest one in figure 6. We used three categories of cyclones based on their intensities. These three categories are as follows:

- Tropical Depression (TD) - a tropical cyclone with the highest sustained winds of up to 62 kilometres per hour (kph) or less than 33 nautical miles per hour (knots)
- Tropical Storm (TS) - a tropical cyclone with a wind speed range of 63 to 118 kph or 34 - 63 knots.
- Severe cyclone - a tropical cyclone with a wind speed range of 64 up to more than 136 knots.

Figure 7 depicts the distribution of different Saffir-Simpson categories present in the INSAT3D dataset. According to figure 7(a), the infrared images are arranged in an ascending manner in this database. The pie chart presented in figure 7(b) depicts that most of the image data belong to the tropical depression class. It shows that the frequency of tropical depressions is the highest as it is not severe case of cyclonic storm

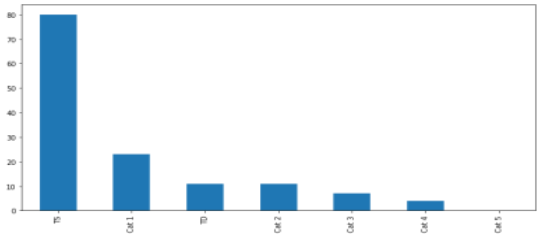
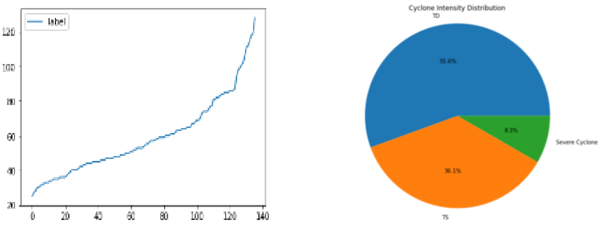


Fig .6. The distribution of different classes in the dataset based on the Saffir-Simpson Scale



(a) (b)

Fig.7. The data Distribution -(a) The Distribution of Intensity with Image Count; (b) Distribution of Three Cyclone Classes Used in this Study

We split the whole dataset into training and test subsets in an 8:2 ratio. This brings 108 raw images for training and 28 raw images for model testing. As the amount of data in different classes is imbalanced, we implemented data augmentation to the sub-datasets. The process of augmenting picture data entails generating new copies of pictures in the train set that have the same classification as the source material. We implemented 'shear_range' for slanting the image data, which is done by stretching the image when an axis is fixed. Zoom range was applied to obtain a number of random zoomed versions of the input data. The rotational function was also applied to generate rotated images. A brightness range from 0.3 to 1 was introduced as well to bring variety to the image dataset on the basis of brightness.

Table 3. Parameters Implemented for Data Generation

Parameters	Values
Rescale	1./255
Shear Range	0.1
Zoom Range	0.1
Rotation Range	15
Brightness range	[0.3,1]
Fill Mode	"nearest"
Horizontal Flip	True
Vertical Flip	True

The images were converted into a NumPy array so that they could be used in the model training. Each image in the dataset was in a different size format. So, we resized all of them into a 224×224 size. The infrared images are in 'RGB' format, so they have three different colour channels. So, the input image size of every single picture is 224 × 224 × 3. The '.csv' file containing each image's training and test intensity values is loaded into a data frame and then converted to a NumPy array.

5.3. Deep Learning Model Design

5.3.1 XGBoost Model

XGBoost refers to the Extreme Gradient Boosting algorithm. In order to maximise the effectiveness and reliability of the learning side, XGBoost continuously brings additional tree structures to fit a variable with several residual rounds [17]. In contrast to the gradient boosting introduced by Friedman [18], XGBoost employs a Taylor series expansion to approximatively simulate the loss. It can help in balancing the model in the case of bias and variability. We employed the XGBoost classifier to the training image dataset and its labels. In an XGBoost model, each indicator receives a score that is then used as input into the decision tree's prediction algorithm. The next decision tree is loaded with the variables that the first tree incorrectly anticipated. Together, these independent predictors form a robust and accurate framework. There is an integrated cross-validation technique in XGboost implementations. Due to this reason, the system can avoid overfitting issues whenever the input dataset size is limited. As this is a multiclass categorisation, 'softprob' hyperparameter was used here. The 'entropy' was set as the criterion parameter for this model. The model is trained for 500 iterations. The maximum tree depth was kept at 4, and the learning rate was 0.01.

5.3.2 LightGBM Model

LightGBM is a decision-tree-based gradient boosting system that improves the quality of the model with less memory consumption. It employs two cutting-edge methods—Gradient-based Single-Side Sampling and Exclusive Feature Bundling —to overcome the shortcomings of the histogram-based methodology employed in most gradient-boosting methods. In contrast to previous boosting algorithms, LightGBM divides the tree along its leaves rather than its levels (figure 8). For optimal growth, it selects the leaflet with the highest delta loss. Given that the leaf is immutable, the loss of the leaf-wise approach is smaller than that of the level-wise one. A more sophisticated model and possible over fitting in short datasets are also risks associated with growing a tree from its leaves outward. Vector Art

In order to preserve the figures' integrity across multiple computer platforms, we accept files in the following formats: .EPS/.PDF/.PS. All fonts must be embedded or text converted to outlines in order to achieve the best-quality results.

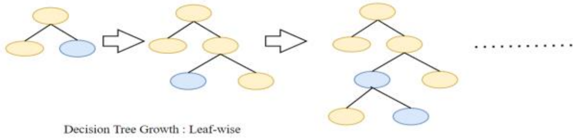


Fig.8. The Leaf-wise approach in the LightGBM architecture

The depth of the tree was limited to 4 in this case study for LightGBM model construction. The model was iterated for 100 iterations. The learning rate for this classifier was set at 0.001. As the value of 'n_jobs' was set at '-1', it indicates that all of the parallel threads were put to use for training. It sets the number of physical cores to be in use for efficient model performance.

5.3.3 Bounding Box Model

We employed a bounding-box-based neural network for this detection task as a third model. The bounding box is a hypothetical box in the shape of a rectangle that encompasses both an entity and a collection of data points. When discussing image processing techniques, the term "bounding box" refers to the dimensions of a picture's boundary along the X and Y axes [19]. They are utilized to identify a target, act as a baseline for object identification, and produce an impact area for the item in question [20].

We used Tensorflow Keras for model building. The whole model design is presented in the flow diagram in figure 9. We employed the Xception model from the Keras applications library as a baseline model. We also tried DenseNet121, Efficientnet, Mobilenet, ResNet, and VGG models in this step, but only the Xception model produced good outcomes, so we chose to go with the Xception model [21].

Xception is a convolutional neural network that is 71 layers deep [22]. The weights assigned here were incorporated from ImageNet. ImageNet has become the de facto standard for the categorisation of images. Since a system that has been trained on the ImageNet architecture would already have some understanding of fundamental forms, it is helpful to load some weights [23]. After then, the subsequent layers are kept trainable so that the model may be tailored to the given image data. The input tensor size is 224×224×3.

After this, the defined classification layer was implemented here by giving flattened output from the Xception network. Here, we added three dense layers to the architecture. The first layer has 64 neurons with 'relu' activation. The second layer has 32 neurons with 'relu' activation. The third layer has one neuron, as we are predicting a single class here, which is the intensity of the cyclone eye.

After this, we compiled the model. While compiling the model, we used the 'Adam' optimizer. The learning rate applied here is 0.001. The loss function running in this compiling step is a mean absolute error. The evaluation metric for the training is the root mean squared error. The training went on for 50 epochs.

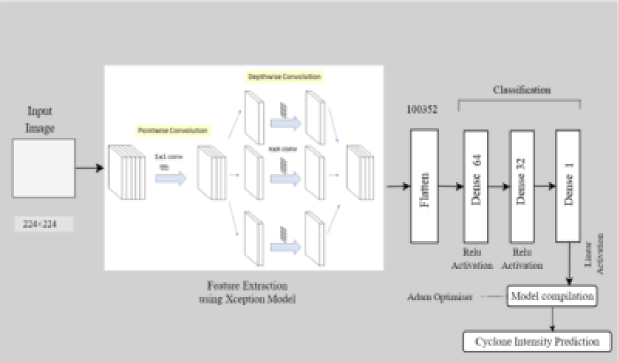


Fig.9. The Bounding Box Design Flow Diagram

5.3.4 CNN Model

Convolutional neural network (CNN) is a form of deep learning technique used to analyse data organised in a grid structure. They are a form of deep learning approach utilised to analyse the information with a temporal or spatial component. Using a convolutional layer stack gives CNN an extra degree of complexity compared to other types of neural nets. There are three main types of layers in a CNN framework. They are the convolutional layer, pooling layer, and fully connected layer [24]. Convolutional layers use a series of filters to process a picture [25]. After the filters are implemented to the input picture in this layer, a feature map is produced. The more complicated the model is, the more comprehensive characteristics it can acquire from the pictures it is fed. This is accomplished by stacking convolutional layers. The CNN model presented in this study has a total of 4 convolutional layers present in it. Each of these layers has been implemented with a 'relu' activation function.

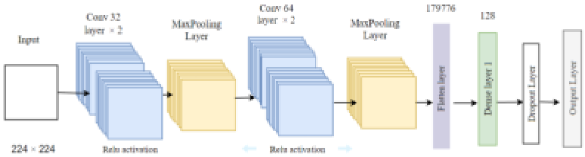


Fig.10. The CNN Model Design

The spatial dimension of the image data is reduced by pooling layers, rendering it more manageable and reducing the amount of storage space needed [26]. Furthermore, the number of variables can be decreased and training time shortened by pooling. Maximum and average pooling are the two most used methods of these kinds of layers. Maximum pooling takes the highest value from each feature map. Pooling layers are often placed after convolutional layers to minimize the data size. We utilized two max-pooling layers in this CNN framework. After the second pooling layer, the input was flattened. Figure 10 represents the model design.

At the very end of a convolutional neural network, when forecasts need to be made based on the characteristics learnt in earlier layers, completely connected layers are employed. At this stage, all of the neurons in the underlying layer are linked to one another. We used three such layers, of which the last is the decision-making layer on the predictions to be made. We employed a dropout layer in the framework with a dropout value of 0.2. Dropout is a method whereby a certain number of neurons are neglected arbitrarily throughout the training phase. This results in a reduction of the network's sensitivity to individual neuronal weights. This leads to a structure that can generalize more effectively and is less prone to over fitting its training dataset. After the dropout stage, the dense output layer makes the prediction, which has an activity reguliser with a value of 0.01 and a kernel reguliser with a value of 0.0001.

5.3.5 Novel BB-CNN Model

Stacked Generalisation, abbreviated 'Stacking,' is an ensemble ML technique. It is similar to combining and boost-up techniques, combining the forecasts from several different ML and deep learning models run on the same input data. The structure of a stacking system consists of more than one base model, which is frequently described as a level-0 system, and a meta-model that integrates the projections of the base classifiers, which is described as a level-1 system [27]. This meta-model learns from the inferences of the underlying models applied to the input dataset [28]. In other words, the level-0 systems are given information that was not utilized for training them, and the projections they make, combined with the predicted outcomes, form the input and outcome combinations of the training data employed to fit the meta-model. Its strengths become apparent when implemented in a wide variety of inefficient learning models, each of which contributes something unique to the meta-model as a whole.

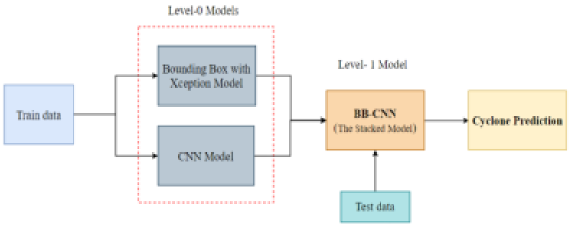


Fig11. The BB-CNN Model

Figure 11 depicts the model design of this Bounding Box-CNN (BB-CNN) approach. We used Keras to stack these two models which we designed before. Then we ensembled those two models as a meta-structure for cyclone prediction. References

6. Results and Discussion

All the designed models were saved using Pickle for further usage. Various metrics were used for the assessment of the different models we presented here. A confusion matrix is a special kind of matrix that can be used to see how well a technique performs when applied to the issue of classification tasks in pattern recognition and ML. It consists of various parameters generated from the classifier outputs. These include true and false positives and true and false negatives. Various assessment scores can be generated from them. The most renowned ones are precision, recall, F1-score, and accuracy.

In statistics, precision is calculated as the proportion of true predictions generated by an algorithm. The recall percentage can be calculated by dividing the entire amount of true positives and false negatives by only the amount of true positives. Measurements for precision and recall can be helpful for gaining insight into a method's efficacy and for ensuring that output meets specifications. However, determining which method best fits the data while evaluating many methods trained on identical information sets is challenging when using just these values to make the comparison. This is why F1-score is essential. The F1 score can be referred to as the harmonized average of the accuracy and recall scores. The following equations (1-4) are the mathematical formulae of precision, recall, F1-score and accuracy.

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (1)$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (2)$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative} \quad (4)$$

We also used RMSE scores for the model assessment. When analyzing the predictive performances, one of the metrics that is utilized most frequently is known as the RMSE, which is also sometimes referred to as the root mean square deviation. It does this by calculating the Euclidean distance between the actual true values and the predicted ones. Equation 5 represents the mathematical formula of the RMSE score.

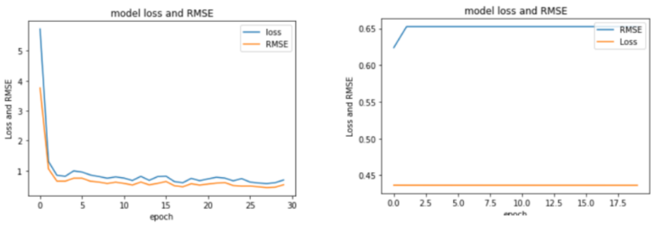
$$Root-mean-square\ Error=\sqrt{\frac{\sum_{i}^{n}\left|y_i-\bar{y}_i\right|^2}{n}} \quad (5)$$

Here, yi is the ith value, n indicates the number of data points and is the related prediction of that value.

Initially, we tried predicting cyclones using the XGboost method. It led us to an accuracy of 46%. The model was only capable of recognizing the tropical storm category. The precision and recall, in this case, were 0.50 and 0.87, respectively. We tried hyper parameter tuning using the entropy criterion and the number of iterations. The highest accuracy achieved remained at 46%. The RMSE score, in this case, was 0.7319.

As XGBoost was not giving us satisfactory outcomes, we ran a trial using a modified approach that was better than XGBoost. We implemented the LightGBM technique to the input data. lightGBM improved the accuracy along with the other evaluation parameters. The precision and recall increased to 0.54 and 1.00. The tropical storm recognizing model reached an F1-score of 0.70. The RMSE value for this case was 0.6813.

In the Bounding Box model, the loss function reduced from 26.80 to 1.70144 in the 48th epoch. After this, the loss function did not improve. The root-mean-square error got reduced from 35.1709 to 2.3677. So, we modified the model by increasing the number of epochs and making the activation function 'linear' in the last dense layer. It led us to an accuracy of 60.71%. The loss value decreased from 2.8822 to 0.4959 in the 14th epoch, and the RMSE score was 0.676. The gradual shift in Loss score and RMSE value is presented in figure 12(a).



(a) (b)

Fig.12. Changes in RMSE Value and Loss Score- (a) Bounding Box Model; (b) CNN Model.

In the CNN model, the accuracy increased a lot. The model accuracy reached a value of 71.43% in 20 epochs. The RMSE score reached a value of 0.6526. The loss function achieved a score of 0.4364. The precision and recall, in this case, are 0.7143 and 1 for the tropical storm category. Figure 12(b) depicts the changes in loss and RMSE score for the CNN model prediction.

The last model we designed and implemented on the INSAT3D dataset is the Bounding-Box- CNN ensemble model. This model was formed considering the outputs and metric values generated in the CNN and Bounding Box models. Utilizing Keras's average technique, we were able to generate outputs which reached an accuracy of 75%. This model could detect severe cyclonic storms, which previous models could not achieve. The changes in loss score and RMSE value are presented in figure 13.

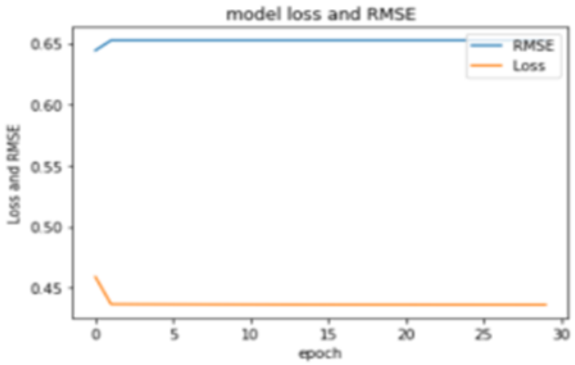


Fig.13. The changes in Loss Function Value and the RMSE score in BB-CNN Model

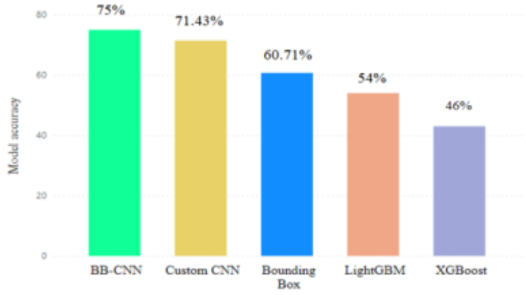


Fig.14. Comparison of Accuracy of Different Models

Figure 14 shows the comparison of accuracies among the five models described in this article. It shows that the BBCNN model has the highest accuracy among them, but accuracy is not a good metric for model assessment. This is why we have to consider the confusion matrix and its related parameters. So, we made the assessment based on the F1-score, precision and recall values from each model. Figure 16 shows that although the CNN and LightGBM models have their advancements in the recall section, the BB-CNN technique is a well-rounded method. The precision, F1-score and recall values are not so far apart in this case. It leads to the conclusion that the BB-CNN model is an efficient approach for cyclone prediction. Table 4 shows the results obtained from different model tested in this research work on the INSAT3D dataset. Figure 15 presents a graphical comparison of tropical cyclonic storm precision, recall and F1-score for different models.

Table 4. Results Obtained from different models

Machine Learning Algorithms	Class	Precision	Recall	F1-score	Accuracy
XGBoost	1	0.0000	0.0000	0.0000	46%
	2	0.5000	0.8700	0.6700	
	3	0.0000	0.0000	0.0000	
LightGBM	1	0.0000	0.0000	0.0000	54%
	2	0.5400	1.0000	0.7000	
	3	0.0000	0.0000	0.0000	
Bounding Box Model	1	0.1667	0.5000	0.2500	60.71%
	2	0.7273	0.8000	0.7619	
	3	0.0000	0.0000	0.0000	
CNN	1	0.0000	0.0000	0.0000	71.43%
	2	0.7143	1.0000	0.8333	
	3	0.0000	0.0000	0.0000	
BBCNN (Proposed Model)	**1**	**0.5000**	**0.2000**	**0.2967**	**75.00%**
	2	**0.8300**	**0.8600**	**0.8400**	
	3	**0.3300**	**0.0000**	**0.5000**	

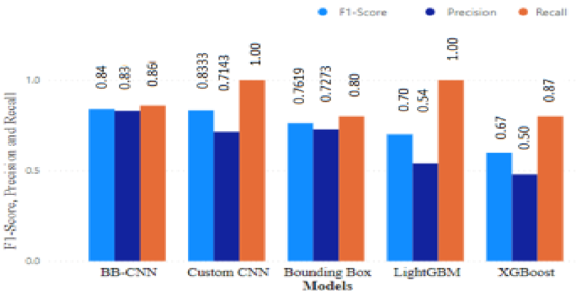


Fig.15. Comparison of Tropical Cyclonic Storm Precision, Recall and F1-Score for Different Models

7. Conclusion

Detection and further monitoring of tropical cyclones can be done using machine learning and deep learning techniques. The detection model uses satellite infrared image data for analysing and predicting the cyclone intensity. So, the aim of this study is to develop a novel technique using deep learning methods using satellite infrared image data to detect the cyclone intensity at the cyclone eye. Our team accessed the MOSDAC server to retrieve the necessary picture data (INSAT3D dataset) for our investigation. In order to get insight into the statistics included in this dataset, we conducted an exploratory study. Data visualisation is then used to illustrate the results of the analysis. We then divided the data set into training and test sets. Next, data augmentation was applied to the resulting dataset. After that, we set out to develop several cyclone forecasting models. With this prediction in mind, we plan to send the input data through four different models (bounding box, XGBoost, LightGBM, CNN and BB-CNN). Model performances were evaluated by using RMSE and Loss scores along with accuracy, precision, recall, and F1-scores to initial findings. At this point, we picked the most appropriate model for making these forecasts. The best score was given by the BB-CNN method in this study. In the future, we can increase the model's efficiency by implementing transfer learning to the dataset. This can lead us to the improvement in precision, recall, as well as F1-score of the said model.

Acknowledgement

This research was supported/partially supported by ISRO. We thank our colleagues from Jain University and Brindavan college of engineering, Bangalore who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. We thank Dr. Ezhilarasan G, Research Coordinator, Jain University for assistance with CNN technology, methodology, and Mrs. Ananya, Assistant professor, Brindavan college of engineering for comments that greatly improved the manuscript.

Author contributions

Mohmad Umair Bagali: Conceptualization, Methodology, Software, Field study **Thangadurai. N:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Mamatha N.P:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

All authors declare no conflicts of interest.

References

[1] K. Emanuel and D. S. Nolan, "Tropical cyclone activity and the global climate system," in 26th conference on hurricanes and tropical meteorolgy, 2004.

[2] J. P. Kossin, K. R. Knapp, T. L. Olander, and C. S. Velden, "Global increase in major tropical cyclone exceedance probability over the past four decades," *Proceedings of the National Academy of Sciences,* vol. 117, no. 22, pp. 11975-11980, 2020.

[3] A. Persson, "The Coriolis Effect," *History of Meteorology*, vol. 2, pp. 1-24, 2005

[4] R. DeMaria, "Automated tropical cyclone eye detection using discriminant analysis," Colorado State University, 2015.

[5] R. S. Cerveny and L. E. Newman, "Climatological relationships between tropical cyclones and rainfall," *Monthly Weather Review,* vol. 128, no. 9, pp. 3329-3336, 2000.

[6] M. F. Piñeros, E. A. Ritchie, and J. S. Tyo, "Detecting tropical cyclone genesis from remotely sensed infrared image data," *IEEE Geoscience Remote sensing letters,* vol. 7, no. 4, pp. 826-830, 2010.

[7] N. Jaiswal and C. M. Kishtawal, "Automatic determination of center of tropical cyclone in satellite-generated IR images," *IEEE Geoscience Remote Sensing Letters,* vol. 8, no. 3, pp. 460-463, 2010.

[8] T. Fiolleau and R. Roca, "An algorithm for the detection and tracking of tropical mesoscale convective systems using infrared images from geostationary satellite," *IEEE transactions on Geoscience Remote Sensing,* vol. 51, no. 7, pp. 4302-4315, 2013.

[9] N. Jaiswal and C. M. Kishtawal, "Objective detection of center of tropical cyclone in remotely sensed infrared images," IEEE Journal of Selected Topics in Applied Earth Observations Remote Sensing, vol. 6, no. 2, pp. 1031-1035, 2013.

[10] I. Dutta and S. Banerjee, "Elliptic fourier descriptors in the study of cyclone cloud intensity patterns," International Journal of Image Processing, vol. 7, no. 4, pp. 402-417, 2013.

[11] C Kar, A. Kumar, D. Konar, and S. Banerjee, "Automatic region of interest detection of tropical cyclone image by center of gravity and distance metrics," in 2019 Fifth International Conference on Image Information Processing (ICIIP), 2019, pp. 141-145: IEEE.

[12] M. Kim, M.-S. Park, J. Im, S. Park, and M.-I. Lee, "Machine learning approaches for detecting tropical cyclone formation using satellite data," Remote Sensing, vol. 11, no. 10, p. 1195, 2019.

[13] C. Kar, A. Kumar, and S. Banerjee, "Tropical cyclone intensity detection by geometric features of cyclone images and multilayer perceptron," *SN Applied Sciences,* vol. 1, no. 9, pp. 1-7, 2019.

[14] Y. Liu et al., "Application of deep convolutional neural networks for detecting extreme weather in climate datasets," 2016.

[15] D. Matsuoka, M. Nakano, D. Sugiyama, and S. Uchida, "Detecting precursors of tropical cyclone using deep neural networks," in *The 7th International Workshop on Climate Informatics, CI*, 2017.

[16] S. Giffard-Roisin, M. Yang, G. Charpiat, C. Kumler Bonfanti, B. Kégl, and C. Monteleoni, "Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data," *Frontiers in big Data,* vol. 3, p. 1, 2020.

[17] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1-4, 2015.

[18] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics,* pp. 1189-1232, 2001.

[19] H. Ibrahem, A. Salem, and H.-S. Kang, "LEOD-Net: Learning Line-Encoded Bounding Boxes for Real-Time Object Detection," *Sensors,* vol. 22, no. 10, p. 3699, 2022.

[20] M. A. Islam *et al.*, "Comprehensive Analysis of CNN and YOLOv5 Object Detection Model to Classify Phytomedicine Tree's Leaf Disease," 2022.

[21] R. Chaganti, V. Ravi, T. D. J. J. o. I. S. Pham, and Applications, "Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification," vol. 69, p. 103306, 2022.

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801-818.

[23] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," 2018.

[24] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1-6: Ieee.

[25] R. Chauhan, K. K. Ghanshala, and R. Joshi, "Convolutional neural network (CNN) for image detection and recognition," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 278-282: IEEE.

[26] X. Jiang, Y. Wang, W. Liu, S. Li, and J. Liu, "Capsnet, cnn, fcn: Comparative performance evaluation for image classification," *International Journal of Machine Learning Computing,* vol. 9, no. 6, pp. 840-848, 2019.

[27] X. Cheng and H. Lei, "Remote sensing scene image classification based on mmsCNN–HMM with stacking ensemble model," *Remote Sensing,* vol. 14, no. 17, p. 4423, 2022.

[28] Z. Ouyang, X. Sun, J. Chen, D. Yue, and T. Zhang, "Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things," *IEEE Access,* vol. 6, pp. 9623-9631, 2018.