

A Deep Learning Framework for the Detection of Tropical Cyclones from Satellite Images

Aravind Nair*, K S S Sai Srujan*, Sayali R. Kulkarni*, Kshitij Alwadhi[†], Navya Jain[‡], Hariprasad Kodamana^{§||}, S. Sandeep^{*||}, and Viju O. John[¶]

Abstract—Tropical cyclones (TCs) are the most destructive weather systems that form over the tropical oceans, with 90 storms forming globally every year. The timely detection and tracking of TCs are important for advanced warning to the affected regions. As these storms form over the open oceans far from the continents, remote sensing plays a crucial role in detecting them. Here we present an automated TC detection from satellite images based on a novel deep learning technique. In this study, we propose a multi-staged deep learning framework for the detection of TCs, including, (i) a detector - Mask Region-Convolutional Neural Network (R-CNN), (ii) a wind speed filter, and (iii) a classifier - CNN. The hyperparameters of the entire pipeline is optimized to showcase the best performance using Bayesian optimization. Results indicate that the proposed approach yields high precision (97.10%), specificity (97.59%), and accuracy (86.55%) for test images.

Index Terms—remote sensing, Tropical cyclone (TC), Deep learning, Detectron, Mask R-CNN, CNN, DenseNet

I. INTRODUCTION

Tropical Cyclones (TCs) are some of the most devastating extreme weather events that form over the warm tropical oceans and have a high socio-economic impact. On a global scale, an average of 90 TCs form annually over the tropical warm waters [1]. The trajectory of a TC is important to understand the areas it can affect. The destructive power of TCs is increasing in response to global warming [2]. The current TC data archives suffer from uncertainty in the data collection by manual methods [3]. The satellite data archive spans a period of more than four decades, and it can be used to extract a long term homogeneous TC dataset.

An early automated tracking technique uses the pattern correlation coefficient from two consecutive IR images [4], [5]. They defined clouds as a connected set of pixel values and by applying area and temperature threshold, clouds can be tracked by overlapping between these pixels in successive images [6]–[8]. An automatic algorithm to detect and track in time the tropical mesoscale convective systems from infrared image series through a 3-D segmentation is proposed by [9]. Piñeros et al. proposed an approach to Detecting tropical cyclone

genesis from remotely sensed IR image data [10]. Related works based on fluxes of the gradient vectors of brightness temperature and fitting spiral features within the IR images are also used for to fix the center position of TCs [11], [12]. [13] presented as an approach for locating the typhoon center by using satellite and microwave scatterometer data.

Multiple meteorological agencies do a post-season analysis of TC tracks which will be useful for forecast verification and trend analysis known as “best tracks”. As this process is subject to manual errors, this data is prone to uncertainty. For example, the best tracks data from the Joint Typhoon Warning Center (JTWC) show an increase in stronger TCs over the Western North Pacific (WNP) [14]. However, the TC data from Japanese and Hong Kong meteorological agencies show no such trends [15]. The development of a TC dataset using an automated algorithm from satellite images can reduce these uncertainty.

The identification of TCs from satellite images is based on their size, position, status, and intensity [16]. This form of identification, based on pattern recognition, was pioneered by Dvorak and is based on human judgement, hence requiring an expert eye. A semi-automated approach to this had been proposed by [17], which used of Elliptic Fourier Descriptors (EFD) and Principle Component Analysis (PCA) on visible and infrared images for classification. Detection of eye of a TC also has been the focus of much research since the formation and categorization of the TC depends on it. Synthetic Aperture Radar (SAR) technology has also been extensively used in helping with detection of the eye of TC’s due to its ability of cloud penetration. SAR is also used in [18] for a semiautomatic center location method in cases when a TC is imaged without its eye.

A Quadratic Discriminant Analysis (QDA) has also been proposed for classification of IR images as images with eye and without eye in [19]. Support vector machines (SVMs), Random Forest (RF) and Decision Trees have also been shown to be effective at detection of TC formation [20]. Deep learning algorithms have also been used used for identification and classification purposes. This has been demonstrated in the use of an ensemble on CNNs classifiers on simulated outgoing longwave radiation (OLR) for classifying as TCs and their precursors in [21]. The CNNs were trained with 50,000 images containing TCs and their precursors and 500,000 non-TC data for binary classification, showing success in WNP region. Four different state-of-the-art U-Net models were developed in [22] for detection of Regions of Interests (ROIs) for tropical and extratropical cyclones. [23] implemented a deep fusion model

* Centre for Atmospheric Sciences, Indian Institute of Technology Delhi, New Delhi 110016, India. [†] Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India. [‡] Department of Chemical Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India. [§] School of Artificial Intelligence, Indian Institute of Technology Delhi, New Delhi 110016, India. [¶] EUMETSAT, 64295 Darmstadt, Hessen, Germany. ^{||} Corresponding authors. Email: kodamana@iitd.ac.in, sandeep.sukumaran@cas.iitd.ac.in. The authors gratefully acknowledge funding received from the Ministry of Earth Sciences, India, with grant number: NCPOR/2019/PACER-POP/OS-04

built to use the TC track data and 3D reanalysis as input.

In this study, we present a novel implementation of ML technique to detect tropical cyclones from high resolution satellite images by considering only the shape of the clouds in the images and maximum sustained surface wind speeds from JTWC. Each detection is also provided with a segmentation, allowing any initial analysis based on shape and size of the detected TC. The pipeline consists of a state-of-the-art mask R-CNN detector, a wind speed filter, and a CNN classifier. The ML pipeline can be used for subsequent timestamps to generate a time series of the predicted segmentation.

II. DATA

A. Data Extraction

The level 1.5 Meteosat Visible Infra-Red Imager (MVIRI) IR satellite images from Meteosat 5 and Meteosat 7 Indian Ocean Data Coverage (IODC) at a six-hourly frequency was considered for the analysis from 2001-2007 and 2007-2016 respectively. The Meteosat 5 and Meteosat 7 during their IODC coverage were located at a sub longitude of 63° and 57° respectively providing data for the full disk coverage, but we have considered only the Asian region (44.5°E - 105.5°E & 10°S - 45°N) for the study. The 8-bits measurement counts of IR channel are calibrated and converted into brightness temperatures as described in [24].

B. Data Preparation

The Microsoft Common Objects in Context (MS COCO) dataset is employed in the deep learning framework where the annotations are stored in JSON files. The annotations specified for object detection consist of the following information for each image: (i) image_id to uniquely identify a specific image from the dataset; (ii) category_id to uniquely identify each category in the dataset; (iii) segmentation consisting of a list of vertices in polygonal or RLE format; (iv) area for the area of Segmentation; (v) bounding_box drawn around the segmentation; (vi) is_crowd to specify if the segmentation is for a collection of objects or a single object. In case of a single object, Polygon method is used to specify the segmentation. In order to make the JSON file, usual procedure involves getting segmentation masks manually.

The compiled dataset of images was further processed to represent it in COCO dataset format. Each segmentation mask was also colour coded according to wind speed. Following the WMO criteria for classification of TCs in the Northern Indian Ocean region, any time stamp with wind speed ≥ 34 knots is

TABLE I
OPTIMIZED PARAMETERS OF THE DETECTOR (MAXIMIZING ACCURACY)

Parameter	Range	Optimal Value
Images Per Batch	[1, 15]	7
Base LR Value	[0.0001, 0.01]	0.00161
Maximum Iterations for the LR Scheduler	[100, 600]	335
Batch size per Image	[50, 170]	50
Test Score Threshold	[0.4, 0.95]	0.95

marked as a tropical cyclone [25]. In the segmentation masks, this was represented by filling the cloud segmentation with green colour. Otherwise, the cloud segmentation was filled with red colour. The images and masks were also randomly separated into the following three sets: training (60%)/validation (20%)/testing (20%). For each image, segmentation mask was used to get the list of vertices to generate annotations as per the COCO dataset format using another code. The program extracted the following data for each image: (i) width, height and image id from the satellite image; (ii) annotations with image id, category id, segmentation (in polygon format), area of segmentation and bounding box coordinates (in x, y width, height format where x and y are coordinates of the top left corner of the bounding box) from the segmentation mask. Fig. 1a shows a sample satellite image for TC Nanauk. Fig. 1b shows the corresponding segmentation mask generated, which is used for getting annotations for the corresponding satellite image.

III. METHODOLOGY

A. Proposed ML Pipeline

The the proposed ML pipeline functions in the following manner: (i) An input satellite image is passed to the detector (Mask R-CNN R50 FPN model). The detections are obtained for the input image and recorded in detectron2's output format; (ii) If one or more bounding boxes are detected, the wind speed of the corresponding timestamp is checked. If the wind speed is less than 34 knots, the predictions for the image are discarded; (iii) If there are more than one predictions made for an image, the classifier (DenseNet169) is then provided images cropped from the input satellite image using the bounding box coordinates; (iv) If more than one of the cropped images are classified as cyclones, then the one with the highest confidence score from the classifier is chosen as the correct prediction.

1) *Detector*: The detector is a Mask R-CNN R50 FPN model with 1 x LR Scheduler available in detectron2's model zoo. The model used was prepared by training it on the training

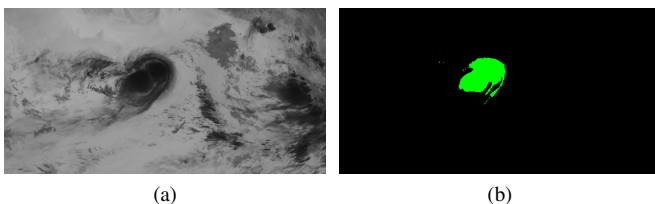


Fig. 1. Figures (a) and (b) show a satellite image and a prepared segmentation mask for Cyclone Nanauk (at timestamp 2014-06-12 0000-0030)

TABLE II
OPTIMIZED PARAMETERS OF THE CLASSIFIER (MAXIMIZING ACCURACY)

Parameter	Range	Optimal Value
Learning Rate	[1E-06, 0.4]	4.61E-04
Momentum	[0, 1]	Not applicable
Weight Decay	[1E-06, 1]	5.26435
Optimizer	[SGD, Adam, Adagrad, Adadelta, RMSProp, Rprop]	Adagrad
Number of Epochs	[10, 50]	43
Batch Size	[1, 50]	13

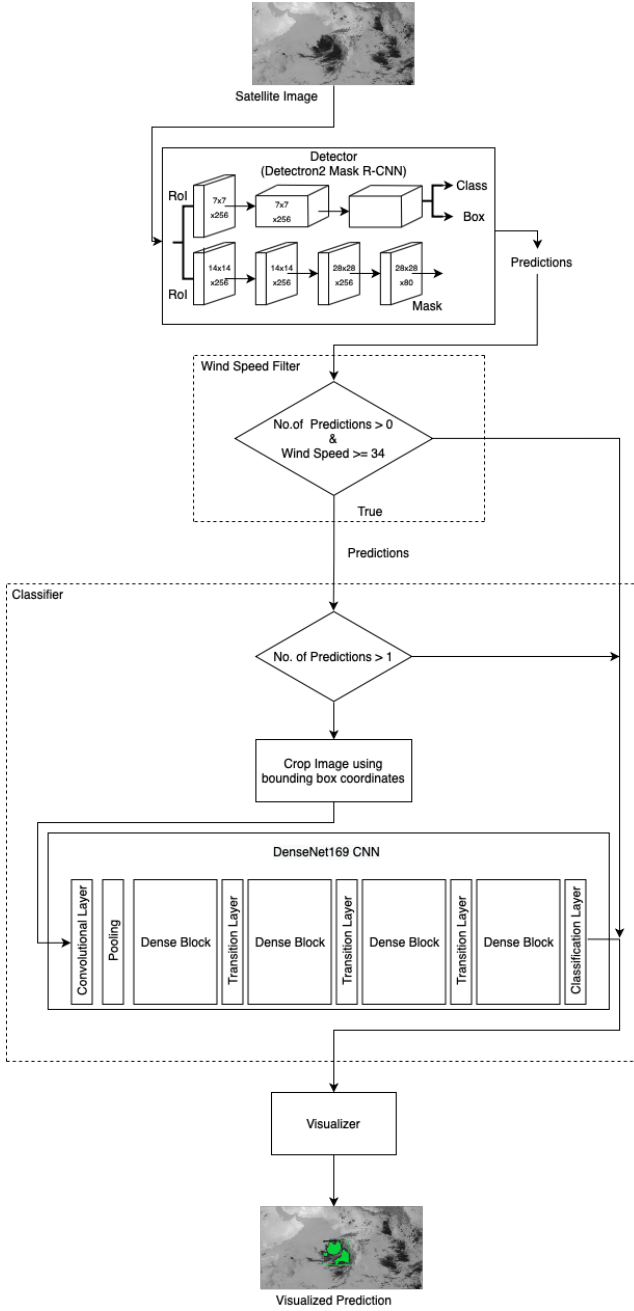


Fig. 2. Proposed ML Pipeline

set after loading it with pretrained weights available in the model zoo. The detector is used to provide predictions about the location of TC in the input image along with its segmentation. It was trained on RGB images of size $688 \times 1339 \times 3$. Models in detectron2 also use the annotations stored in JSON files in the COCO dataset format during the training phase.

Mask R-CNN is an extension of Faster R-CNN used for object image segmentation [26]. It consists of two stages. The first stage is a Region Proposal Network which proposes candidate bounding boxes. In the second stage, it performs two tasks in parallel: (i) extract features from proposed bounding boxes for classification and bounding box regression; (ii) generate a binary mask for each Region of Interest (RoI).

The binary masks represent an object's spatial layout in the image. For each RoI, a $m \times m$ mask is predicted by using pixel-to-pixel correspondence provided by convolutions. The RoI features, developed as small feature maps, are aligned by using RoIAlign to produce pixel-accurate masks. Considering the feature maps as a grid, it utilizes bilinear interpolation of each sampling point with the nearby grid points of the feature map and the results are aggregated.

2) *Wind Speed Filter*: One of the methods used to filter out some of the false positives was by using wind speed. For each timestamp, the wind speed was compared and following the WMO criteria for Northern Indian Ocean Region. If the wind speed ≥ 34 knots threshold for the timestamp then it is accepted as a TC, otherwise it is discarded. This helps remove any predictions made on timestamps just before or after the cyclone is classified as a TC.

3) *Classifier*: The classifier is a CNN model which has been trained to classify a given image as a TC or not. The current classifier model is a DenseNet169 model which was obtained by optimizing its parameters using Bayesian Optimization for maximizing accuracy. Also, this classifier did not require any layer freezing and used the Adagrad optimizer and had been initialised with pretrained weights available in torchvision before being trained for TC classification. In the current pipeline, it is used when the detector has detected more than one bounding boxes. In such cases, the bounding box coordinates provided by the detector are used to crop the images and these cropped images are sent to the CNN model. After classifying these images, if still more than one of them are classified as aTC, then the one for which the classifier has the highest confidence score is chosen as the correct prediction.

For training the classifier, bounding boxes obtained from segmentation masks drawn for images with TCs and with disturbances was used to crop on their corresponding images. These cropped images are sent to the classifier after being labelled as 'tc' (for images with TC) and 'not_tc' (for images without TC). Initially a list of 18 models was made from the PyTorch documentation based on the recorded accuracy values¹. These belong to the following architectures: (i) AlexNet; (ii) VGG; (iii) ResNet; and (iv) DenseNet and the best performing one was DenseNet169.

The labelling of the prediction as a true positive or false positive depends on the intersection over union (IOU). It is calculated by taking the area of overlap and dividing it with the area of union of the ground truth and the prediction bounding box. A prediction is considered a true positive if the value of the IOU is greater than or equal to a specified threshold. In case more than one predictions satisfy this criteria, the prediction with the highest confidence score is labelled as a true positive and the rest are labelled as false positives.

B. Hyperparameter Optimization

Hyperparameter optimization of the entire ML pipeline is achieved via Bayesian optimization technique [27]. For hyperparameter tuning, the inputs for the surrogate model are con-

¹<https://pytorch.org/vision/stable/models.html>

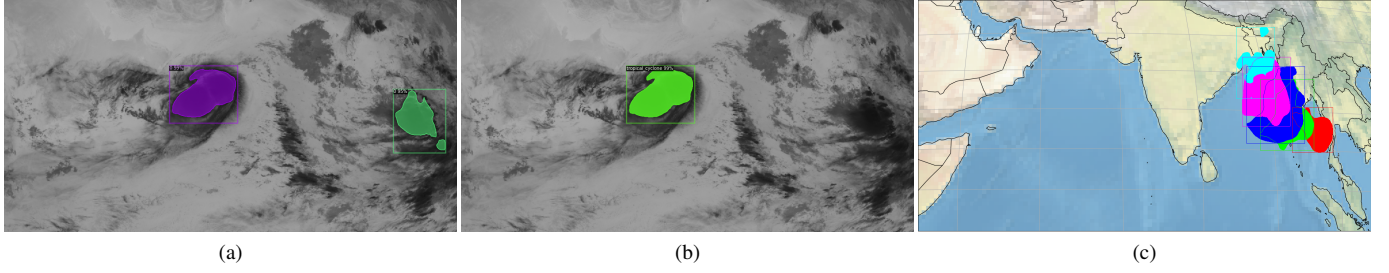


Fig. 3. (a) and (b) show the Visualized predicted segmentation after using detector & wind speed filter and after using classifier for TC Nanauk (at timestamp 2014-06-12 0000 - 0030) respectively; (c) shows visualised predictions of a TC path as a time series of segmented images for TC Hudhud

TABLE III
ML PIPELINES COMPARISON

Metrics	ML Pipelines			
	Detector ¹ + CNN ¹	Detector ² + CNN ²	Detector ² + Wind Speed	Detector ² + Wind Speed + CNN ²
Precision	48.29	74.03	93.24	97.10
Recall	81.61	65.52	79.31	76.14
Specificity	41.98	76.16	94.32	97.59
Accuracy	57.79	70.76	86.86	86.55
F1 Score	60.68	69.52	69.52	85.35

¹ - optimized for F1 score, ² - optimized for accuracy

sidered to be the hyper-parameter values and its output is the metric to be maximized. Here, the surrogate model is produced by mapping hyper-parameter values to the performance metric values using Gaussian process.

For detector, initially, F1 score was chosen as the metric to be optimized with the purpose of getting a model with the best precision-recall values. This resulted in model which had higher rate of false positives. Later to reduce the False Positives, F1 Score was replaced by accuracy as the optimizing parameter. While, for classifier, F1 score was used as the metric to be maximized in this case. The resultant pipeline was better than using only the detector but still suffered from high number of false predictions. Hence the method was slightly changed to obtain a detector and a classifier optimized separately for optimal accuracy and used together in the pipeline. The parameters, along with their ranges and values obtained after optimization for accuracy are tabulated in Table I and Table II for the detector and the classifier, respectively.

C. Implementation framework

The Mask R-CNN models used were provided in detectron2, a PyTorch-based modular object detection library. The detectron2 model zoo consists of high-quality implementations of object detection algorithms such as DensePose, Faster R-CNN, RetinaNet and Mask R-CNN models developed by Facebook Artificial Intelligence Research. These models train on GPU by default and is highly customizable through a configuration system. The config system is a key-value system to obtain standard and common behaviours. Using the config system, the model can be configured with specific hyperparameter values

to adapt to any custom dataset. The Mask R-CNN models selected expect the data to be either provided in detectron2's format or in COCO dataset format. Hyperparameter optimization was also performed using Ax (Adaptive Experimentation Platform). For our purposes, it has been used for hyperparameter optimization using Bayesian optimization method. It iteratively explores the given parameter space for identifying the set of best parameter values. Bayesian optimization in Ax is implemented through Botorch, a PyTorch-based library for Bayesian optimization.

IV. RESULTS AND DISCUSSIONS

We have employed the proposed ML pipeline to test 171 high resolution satellite images with 88 images with TCs. Fig 3a shows the visualization of predictions after the satellite image has been passed through the detector and the wind speed filter for TC Nanauk at a timestep 2014 June 12 0030UTC. In this case, the detector makes predictions of two TC detection on the satellite image. It provides the output with class, score, predicted segmentation mask and its bounding box for each detection. After checking for the number of predictions on the image, the wind speed filter is used to check the wind speed in the current timestep. Since the wind speed for this timestep is 55 knots, it satisfies the wind speed criteria. The image is then passed to the classifier since it has multiple outputs. In this scenario, the cropped images are generated from coordinates of each bounding box and passed one at a time to the DenseNet169 model. The model classifies each output as either "TC" or "not_TC". If more than one cropped images are classified as "TC", the prediction with the highest confidence score from the CNN is chosen as the correct prediction. The corresponding detector output is accepted as the valid detection for that timestep, as shown by the visualization of the final output in Fig 3b. Fig 3c shows the results from performing detections on satellite images for TC Hudhud form 2014. The results are presented as a time series for 2014 October 08 to 12.

During the study, various pipelines were tested and compared to each other on test data sets. The results from these are provided in Table III. The metrics in the table are provided as percentages. The proposed ML pipeline has a high true positive rate (also called recall) of 76.14% and a high true negative rate (also called specificity) of 97.59%. It also has a high accuracy of 86.55% for detection of TCs from the satellite images. Out of the 88 images with TCs, correct detections

were made in 67 images. The proposed pipeline is also able to avoid false predictions and has successfully avoided making false predictions on 81 images without TCs.

During the experiments we have observed that Mask R-CNN models were optimized for F1 score and suffered from high number of false positives. This was observed even when the number of outputs reported for each image after filtering from the classifier was limited to one. Higher success obtained with optimizing the Mask R-CNN and CNN models for accuracy. It yielded a model with higher true positives and true negatives. It is found that the use of CNNs as classifiers in the pipeline reduces the number of positive detections and contributes to the negative detections. This is because a number of positives being reclassified as 'not_tc', which leads to a reduction in the false positives as well as true positives. Use of wind speed filter helped reduce the number of false positives and increasing true negatives. Hence, combining both of these approaches as per the proposed pipeline has given the most optimal results by removing predictions that do not satisfy the wind speed criteria, and by removing false predictions when multiple predictions are passed using CNN.

V. CONCLUSION

In this study, a novel deep learning framework has been developed for TC detection in high resolution satellite images. The framework uses a mask R-CNN model as a detector to provide the segmentation and wind speed filter and CNN classifier to further refine the predictions by filtering out possible false predictions. The proposed ML Pipeline uses satellite images taken at an interval of 6 hours and is able to detect a TC for most of its life cycle. The study has also generated annotated dataset with segmentation masks for every satellite image and is made publicly available at the GitHub repository², along with a simple python tool for generating the JSON files as per COCO dataset format and the code for the pipeline, along with the mask R-CNN and CNN model, has also been made available in the repository². It not only shows the potential of the pipeline in automating the task of TC detection from satellite images, but also in possible application as a predictive tool for TCs.

REFERENCES

- [1] K. A. Emanuel and D. Nolan, "Tropical cyclone activity and global climate," in *26th Conf. on Hurricanes and Tropical Meteorology, Miami, FL, Amer. Meteor. Soc.*, 2004, pp. 240–241.
- [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. 11 975–11 980, 2020.
- [3] C. W. Landsea, G. A. Vecchi, L. Bengtsson, and T. R. Knutson, "Impact of duration thresholds on atlantic tropical cyclone counts," *Journal of Climate*, vol. 23, no. 10, pp. 2508 – 2519, 2010.
- [4] J. A. Leese, C. S. Novak, and B. B. Clark, "An automated technique for obtaining cloud motion from geosynchronous satellite data using cross correlation," *Journal of Applied Meteorology*, vol. 10, no. 1, pp. 118–132, 1971.
- [5] J. Schmetz, "Cloud motion winds from meteosat: performance of an operational system," *Global and Planetary Change*, vol. 4, no. 1, pp. 151 – 156, 1991.
- [6] W. L. Woodley, C. G. Griffith, J. S. Griffin, and S. C. Stromatt, "The inference of gate convective rainfall from sms-1 imagery," *Journal of Applied Meteorology*, vol. 19, no. 4, pp. 388–408, 1980.
- [7] M. Williams and R. A. Houze, "Satellite-observed characteristics of winter monsoon cloud clusters," *Monthly Weather Review*, vol. 115, no. 2, pp. 505–519, 1987.
- [8] Y. Arnaud, M. Desbois, and J. Maizi, "Automatic tracking and characterization of african convective systems on meteosat pictures," *Journal of Applied Meteorology*, vol. 31, no. 5, pp. 443–453, 1992.
- [9] 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 and Remote Sensing*, vol. 51, no. 7, pp. 4302–4315, 2013.
- [10] M. F. Piñeros, E. A. Ritchie, and J. S. Tyo, "Detecting tropical cyclone genesis from remotely sensed infrared image data," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 826–830, 2010.
- [11] N. Jaiswal and C. M. Kishtawal, "Automatic determination of center of tropical cyclone in satellite-generated ir images," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 460–463, 2010.
- [12] —, "Objective detection of center of tropical cyclone in remotely sensed infrared images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 1031–1035, 2013.
- [13] T. Hu, X. Wang, D. Zhang, G. Zheng, Y. Zhang, Y. Wu, and B. Xie, "Study on typhoon center monitoring based on hy-2 and fy-2 data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2350–2354, 2017.
- [14] P. J. Webster, G. J. Holland, J. A. Curry, and H.-R. Chang, "Changes in tropical cyclone number, duration, and intensity in a warming environment," *Science*, vol. 309, no. 5742, pp. 1844–1846, 2005.
- [15] M. Wu, K. Yeung, and W. Chang, "Trends in western north pacific tropical cyclone intensity," *Eos, Transactions American Geophysical Union*, vol. 87, no. 48, pp. 537–538, 11 2006.
- [16] 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," *IEEE*, 2019.
- [17] I. Dutta and S. Banerjee, "Elliptic fourier descriptors in the study of cyclone cloud intensity patterns," *SN Applied Sciences*, 2013.
- [18] S. Jin, S. Wang, X. Li, L. Jiao, J. A. Zhan, , and D. Shen, "A salient region detection and pattern matching-based algorithm for center detection of a partially covered tropical cyclone in a sar image," *IEEE*, 2016.
- [19] R. Demaria, G. Chirokova, J. Knaff, and J. Dostalek, "P 189 machine learning algorithms for tropical cyclone center fixing and eye detection," 2015.
- [20] 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, 2019. [Online]. Available: <https://www.mdpi.com/2072-4292/11/10/1195>
- [21] D. Matsuoka, M. Nakano, D. Sugiyama, and S. Uchida, "Deep learning approach for detecting tropical cyclones and their precursors in the simulation by a cloud-resolving global nonhydrostatic atmospheric model," *Progress in Earth and Planetary Science*, 2018.
- [22] C. Kumler-Bonfanti, J. Stewart, D. Hall, and M. Govett, "Tropical and extratropical cyclone detection using deep learning," *Journal of Applied Meteorology and Climatology*, vol. 59, no. 12, p. 1971–1985, Dec 2020. [Online]. Available: <http://dx.doi.org/10.1175/JAMC-D-20-0117.1>
- [23] 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. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fdata.2020.00001>
- [24] V. O. John, T. Tabata, F. Rüthrich, R. Roebeling, T. Hewison, R. Stöckli, and J. Schulz, "On the methods for recalibrating geostationary long-wave channels using polar orbiting infrared sounders," *Remote Sensing*, vol. 11, no. 10, p. 1171, 2019.
- [25] M. Mohapatra, G. S. Mandal, B. K. Bandyopadhyay, A. Tyagi, and U. C. Mohanty, "Classification of cyclone hazard prone districts of india," *Natural Hazards*, 2011.
- [26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.
- [27] V. Nguyen, "Bayesian optimization for accelerating hyper-parameter tuning," *IEEE*, 2019.

²<https://github.com/AravindNair430/TCTracker.git>