That's a Spicy Cloud: A Comparative study to approaches for Tropical Storm Intensity Estimation from Satellite images.

Balajee Devesha Srinivasan

basrini@iu.edu

Shubhkirti Prasad

shubpras@iu.edu

Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington, Bloomington, Indiana

Abstract

In the past several decades or so, we have seen a steep deterioration in the global climate which has led to an increase in many disastrous natural calamities. We have seen a steep incline in the number of tropical storms and hurricanes, causing widespread destruction of property and loss of lives. "Tropical cyclone intensities globally are projected to increase on average by 1 to 10% according to model projections for a 2-degree Celsius global warming"[1]. Accurate prediction of the intensity and path of these storms can act as critical information for practical disaster management efforts. With the advent of technology, the aspect of satellite imagery and associated climate prediction has been proven to be an effective tool for the same. Raw and Infrared satellite images can be used to estimate storm intensities by analyzing various meteorological parameters such as cloud temperature, wind speed, water vapor, and precipitation.

In this study, we want to determine the relative strengths and weaknesses of various methods of intensity estimation and suggest improvements for better accuracy in tropical storm intensity estimation using satellite images.

1. Introduction

In our attempt to study the approaches to Tropical storm intensity using satellite images, we present a comparative approach of using various Deep learning architectures for the regressive approach to the prediction of the target raw satellite image of the cloud formation. The dataset used for the same has various forms i.e., Infrared Temperature satellite images, Water Vapor analysis images, and Raw images of the cyclone.

1.1. Dataset

For our analysis we have tried to procure the dataset for each category of images but primarily we have obtained only the Raw images format with the labels based on the NASA dataset of 600 storms curated hosted by Radiant ML Hub [2]. For the Infrared Dataset, we procured some sample images from the INSAT IR3D dataset available on Kaggle [3].

The former consisted of 600 individually documented storms with their cloud formation progression images and corresponding wind intensity values split into test and training sets. The train set consisted of 70257 images and 44377 images for testing. The latter consisted of 136 images of INSAT-captured storms with their corresponding intensities. There is a clear lack of infrared images for any clear experimentation in the limited scope of the project hence much of the modeling and the results are based on the Raw image data.

2. Methodology

As the first attempt for the regression with the dataset we set up a small CNN regression model that simply took the provided training and validation split dataset and tried to perform the regression on the dataset however we quickly realized that this method would not be suitable for the dataset due to various reasons we will discuss further. As a baseline model of pure regression, we chose the RMSE as the key metric for evaluations. The baseline model for pure regression scored an RMSE value of 11.1878. To understand the plausible cause of these discrepancies we analyzed the dataset to discover the following key underlying problems in the data and approach.

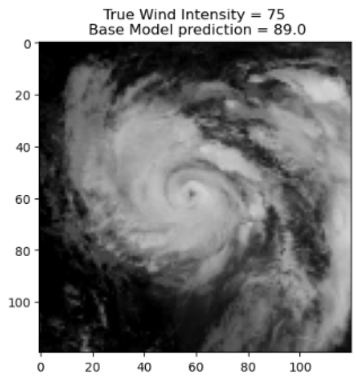

Fig 2.1. Output of baseline regression model

2.1. Dataset Bias

The most important underlying cause of the poor performance of our regression model was the Dataset bias but that isn't directly applicable to a regression problem during our literature review we had come across Tropical Storm classification based on the wind speed intensity and this led us toward the problem of Dataset bias.

The wind intensities of a tropical storm are not something that is static and can vary in a wide range and for the purpose of our dataset which estimates the intensity in knots we decided to check the classes mentioned in [4].

We decided to use 7 Classes for the representation of the intensities as per Table 1 and created bins in the dataset to segregate the individual training images into the classes.

Based on our segregation the Class imbalance in the dataset was fairly obvious as below:

Class 0: train images = 29864, test images = 21405
Class 1: train images = 22450, test images = 14173
Class 2: train images = 9747, test images = 5162
Class 3: train images = 6184, test images = 2887
Class 4: train images = 1888, test images = 735
Class 5: train images = 118, test images = 15
Class 6: train images = 6, test images = 0

Clearly, there is a severe bias towards the Class 0 and 1 in the dataset and it naturally falls inline with the natural expectation in nature. It is not often that a cyclone would evolve gradually into Super Typhoons each time and certainly, it isn't something we go around expecting either. This leads us to the current situation of Class imbalance.

However, you might say all these have got to do nothing with our Original problem statement of performing a regression on the Image data. This is because our method of using exclusively a CNN regressor for the problem at hand is not a suitable solution to the problem statement. Each class represented in the bins has some characteristics that need to be recognized for providing a more accurate estimate, Therefore, we propose a pipeline of 2-stage processing of the images by splitting the problem at hand using Classification and then Regression based on the class.

This is the basic premise of the project i.e. proposing a suitable model for the accurate classification of the images into one of the 7 classes and then using the classification to pass data to the exact or a combined average of the individual class regressors for better performance.

Typhoon Level	**Wind Speed (kt)**	**Wind Speed (m/s)**	**Assigned Labels**
-	< 38.9	< 10.8	0
Tropical depression	38.9 -61.6	10.8–17.1	1
Tropical storm	61.7–87.8	17.2–24.4	2
Severe tropical storm	87.9–117.4	24.5–32.6	3
Typhoon	117.5–149.0	32.7–41.4	4
Severe typhoon	149.1–183.2	41.5–50.9	5
Super typhoon	≥183.3	≥51.0	6

Table 2.1. Typhoon intensity standard level assigned labels

2.2. Classification

Our next problem was to check the performance of classifiers on the dataset.

Based on the information we obtained from [4], we found that VGG16 was one of the best classifiers for our dataset.

Using an inference-based model, we used the pre-trained weights stored in the Keras library to design a simple transfer learning model. Overall, we saw the validation categorical accuracy of: 0.4459, which meant a misclassification error of 66.5%.

To verify the misclassification error rate, we chose to deploy a VGG16 model on our training data so that the model weights are trained on the nuances of our dataset. We obtained similar results from our manual VGG16 model.

We then evaluated the performance of more recent deep learning architectures, namely ResNet on our dataset. Our results as obtained on the validation set are listed in Table 2.2.

Model Architecture	**Validation Categorical Accuracy**
VGG16 (Transfer)	0.4459
VGG16 (Manual)	0.4199
ResNet (Transfer)	0.5894

Table 2.2. Validation Categorical Accuracy values of different models for our dataset.

2.3. Spicy Cloud Model

After using the transfer learning models we decided to implement a CNN model to evaluate its performance on our training data.

The model was loosely based on the VGG16 model but, had a few significant differences:

- The number of convolutional layers is increased from 13 to 16.
- The number of filters in each convolutional layer is smaller than in VGG16.
- The kernel size and padding of some layers are different.
- The number of fully connected layers is reduced from 3 to 2, with fewer units.
- The activation function used in the fully connected layers is ReLU instead of Sigmoid.

Overall, our model was deeper than VGG16, with more convolutional layers, but has fewer filters per layer. This resulted in a more complex model that achieved higher accuracy but also required more training time. The modified fully connected layers with ReLU activation also helped to reduce the risk of vanishing gradients during training.

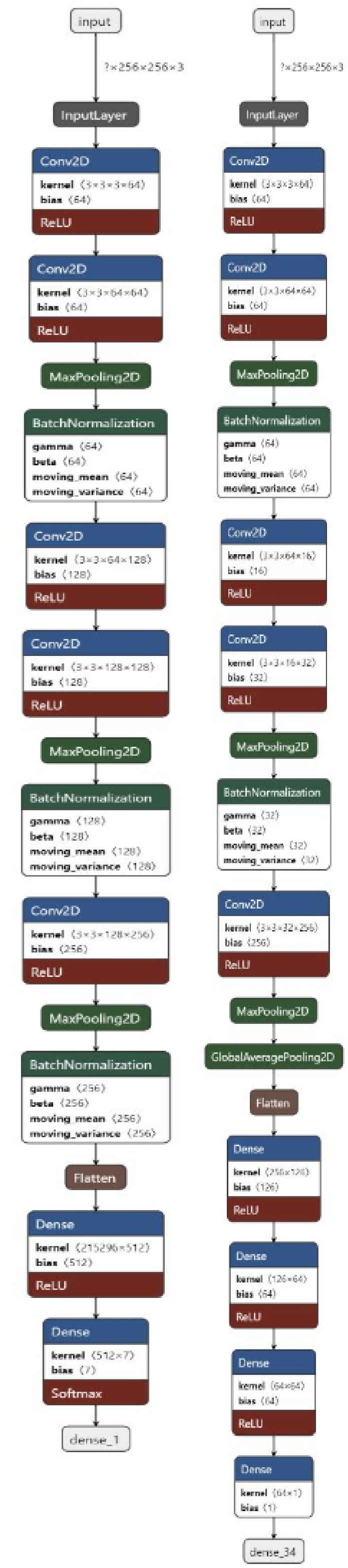

Fig 2.3. Representation of the Spicy Cloud classification model and regressor

Our model performed significantly better than the other models that we tested above.

2.4. Regression

After establishing a baseline performance for the classification problem, using our model, the next problem to tackle is that of the in-class regression. This is the key step that gives us the desired result. Now since we have established there are going to be 7 classes this also means that we need to make 7 regressors, and all of them should be equally reliable as they are going to be fed data of similar characteristics and may also need to deal with the occasional misclassification.

CNNs are primarily renowned for their extensive usage in classification problems however there are surveys and some papers that also promote the use of CNNs for the purpose of regression [5,6]. Thus, we decided to utilize CNNs and see if they would be able to perform better in the same task, with the results from classification in memory we decided to try a few modified transfer learning models for the different input resolutions which we talk about in the augmentation and Bias mitigation section.

After a set of similar testing with models we chose to use a similar model to the one used in the classification and to our surprise, it again surpassed the other models significantly in the base class 0 and 1 regressions. The results are displayed in Table 2 with their RMSE. As we can see the RMSEs have an incremental trend in relation to the classes. Kind of obvious if we check again with the lack of data for the higher intensity classes, nevertheless the model far outperforms estimates from the baseline regression-only model as below.

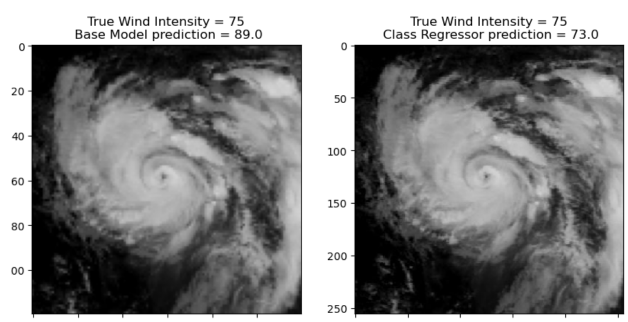

Fig 2.4. Base regression model vs Spicy cloud regression

Based on our training data we got really good results for the regression overall however, there was severe oversimplification for cases with high bias and a severe lack of estimate for the higher intensities. This requires bias mitigation approaches.

2.5. Addressing Bias and Augmentation

As we already have mentioned several times there is an innate bias towards Classes 0 and 1 in the dataset in order to mitigate this the easiest way in a general case would be to gather data for the other classes and match the sample sets for each class. However, that is not possible in our case as the dataset comprises highly precise segmented images of hurricanes with corresponding labels for the same.

To gather more data would either mean we need more historic data which may be blurry or with less reliable labels or wait for more (god-forbid) cyclones of the higher classes. Either of these was currently out of scope for the study timeline. Another solution would be scraping the data from HURSAT [7] dataset. Which essentially means segmenting cyclones and storms from an image of the entire globe and correcting the orientation of the cyclone for a better fit as input. Again, out of the scope of the current study.

What we decided is to use augmentation from the ImaegDataGenerator modules but even so, there are only limited forms of augmentation that would be useful as many of the augmentations may adversely affect the features as they may distort (confirmed via trial and error) the actual features of the cloud formation which are essential for the regression. Thus we only used rescaling, Vertical and horizontal flips for base augmentation, and additional zoom_range and width_shift_range for the input images and the test images were only rescaled. with 0.8 to 0.2 train validation split. We used 3 channel images for the future scope of using Infrared images.

Image dimensions also mattered significantly in the process and we tried with multiple dimensions, the original dataset is in 366 x 366 dimensions, unfortunately, the training was done on local hardware on a laptop and the full-size image was not able to be processed however we tried 120 x 120, 128 x 128, 200 x 200, 256 x 256, etc.

Out of these, we saw the best results with the 256 x 256 and thus we think that supplying higher res images would improve the results further at each stage as a form of future improvements. Another possible approach for the same using SOTA methods would be using cycle-GANs to generate an artificial dataset that would match the characteristics of the higher-intensity cyclones and then perform a semi-supervised version of the same process we have followed here in our study.

Thus, in our study, the primary approach for bias mitigation is the use of an in-class regression ensemble and mild data augmentation at different dimensions.

3. Edge Cases/Loss Hypothesis

Apart from the innate Class bias of the dataset there are some other factors that contribute towards the misclassification of the dataset and also towards the regression RMSE. These edge cases that we observe arise from the hard classification and then using the corresponding regression model. To elaborate further, let us look at one such result from a class 2 classification.

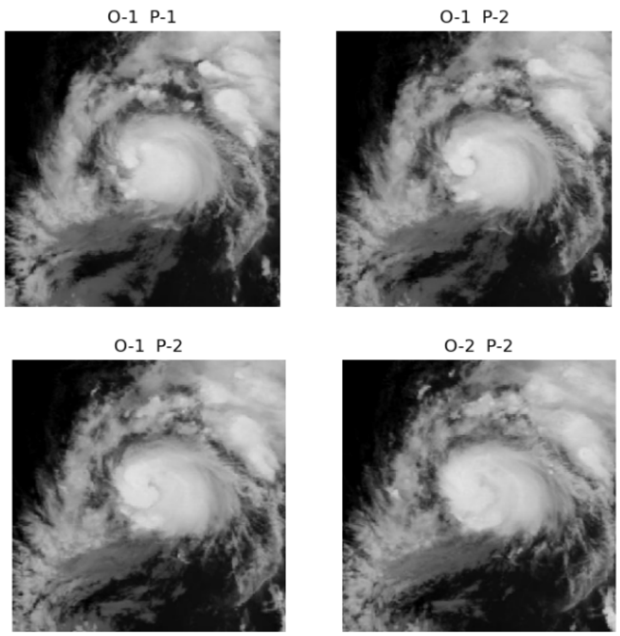

Fig 3 Progression from Class 1 to Class 2 edge case

From Fig 3 we can see the progression of an active storm from the intensities of Class 1 to Class 2 numerically however in the case of the classification, it does not have the contextual info of a hard set boundary it just assesses the image data to infer the weights and essential features from them and predict a class probability for the image. Thus, whenever a storm is moving into or away from a class it is difficult for it to determine the change and change the classification. This in turn also means that regression which has been trained exclusively on the threshold segregated values would be unable to predict accurately a transitional image.

To mitigate this the proposed methodology would be either using the ensemble of individual class regressors or the inclusion of a small percentage of class bordering images for each cyclone for the class before and after thus allowing for the accommodation of the edge cases.

4. Results

Overall our model achieved good results in both the classification and regression as listed in Table 4.1.

Class of Storm	Train RMSE	Test RMSE
0	4.355	4.501
1	6.384	6.627
2	7.330	7.248
3	7.742	7.717
4	7.114	6.778
5	125.188	115.0461
6	-	-

Table 4.1. RMSE values for individual class regressors

Individual model results :

(Key: O - True/Observed Labels, P- Predicted Labels)

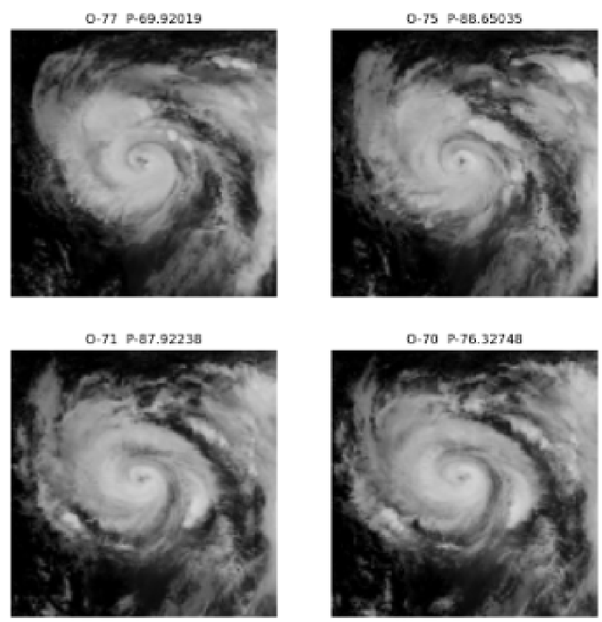

Fig 4.1. Predicted VS True wind speed values for base regressor

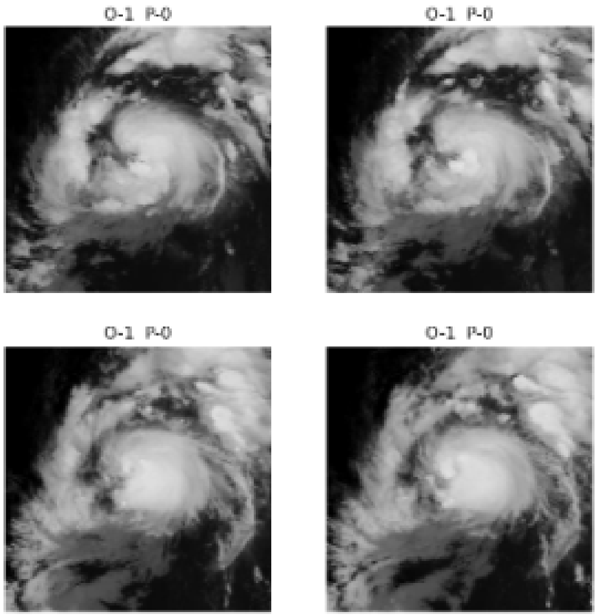

Fig 4.2. Predicted VS True labels for VGG16 Transfer learning model

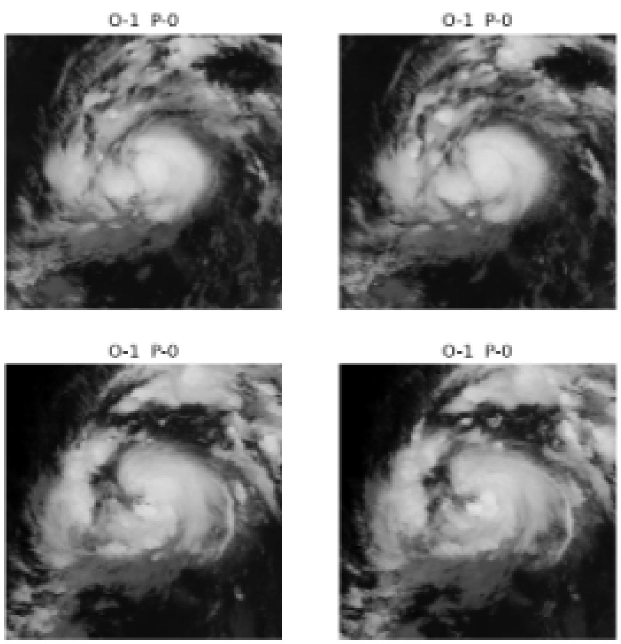

Fig 4.3. Predicted VS True labels for VGG16 manual implementation

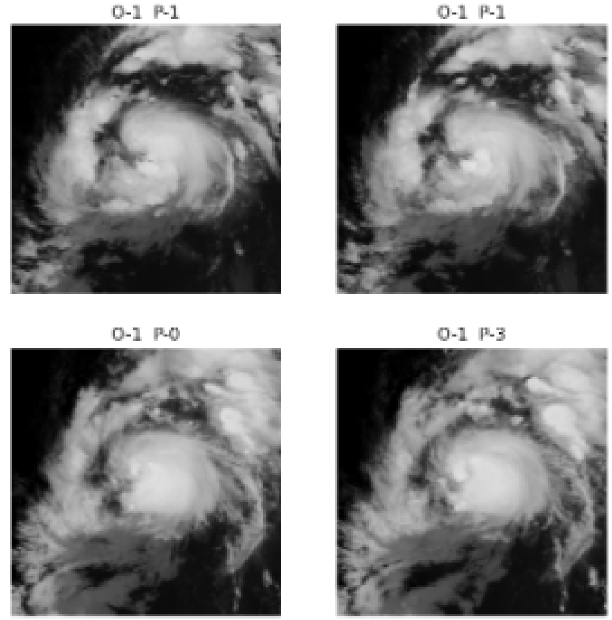

Fig 4.4. Predicted VS True labels for ResNet Transfer learning model

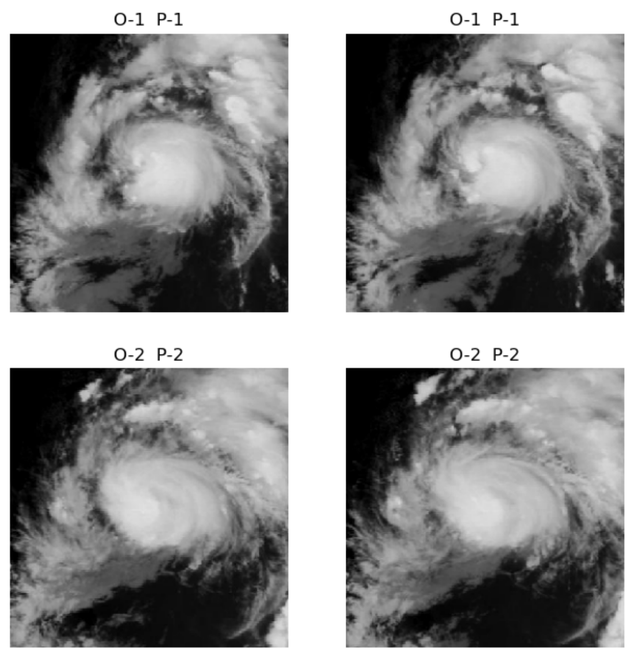

Fig 4.4. Predicted VS True labels for ResNet Transfer learning model

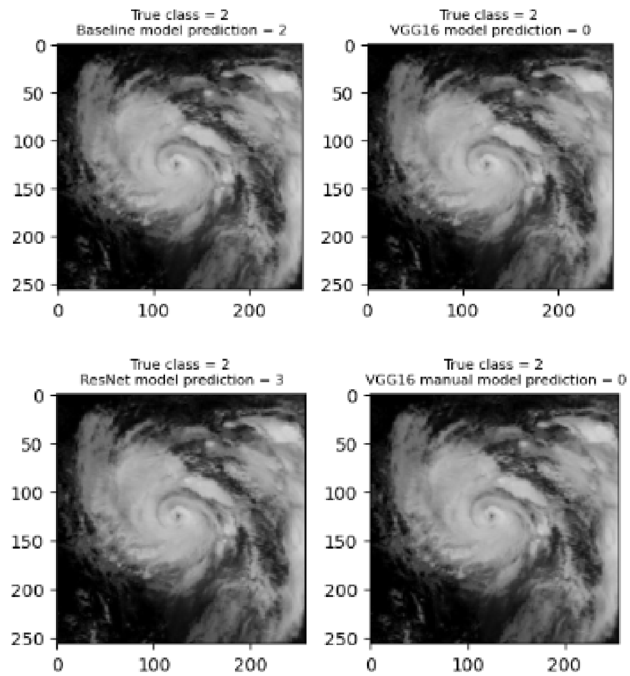

Fig 4.5. Predicted VS True labels for all the implemented classification models

Individual class regressor results:

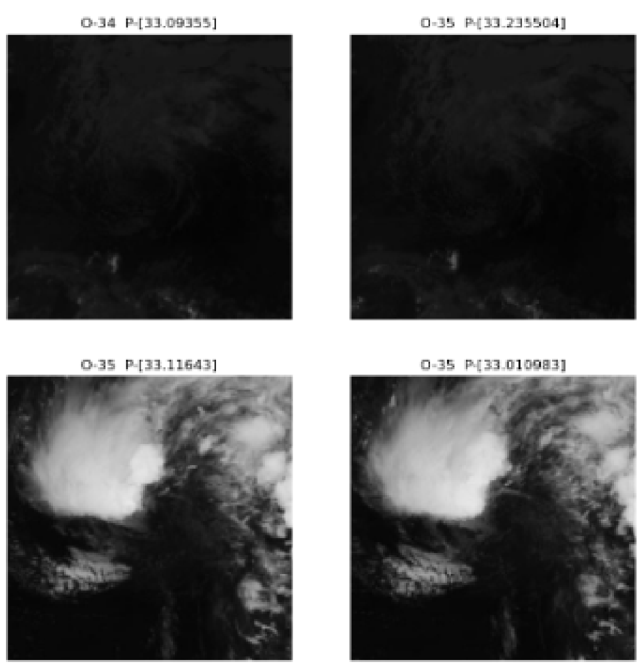

Fig 4.6. Predicted VS True wind speed values for Class : 0

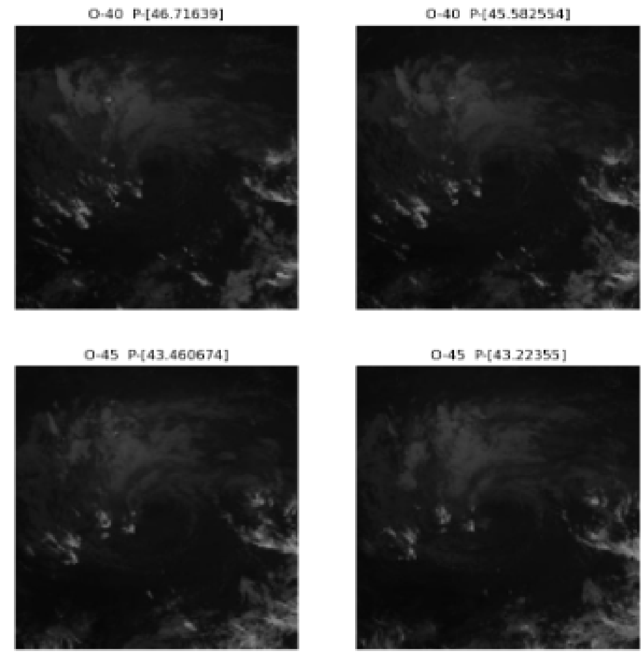

Fig 4.7. Predicted VS True wind speed values for Class: 1

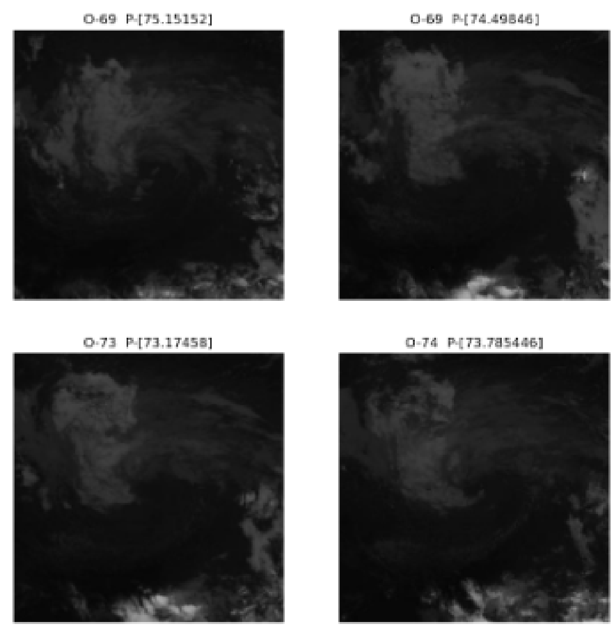

Fig 4.8. Predicted VS True wind speed values for Class: 2

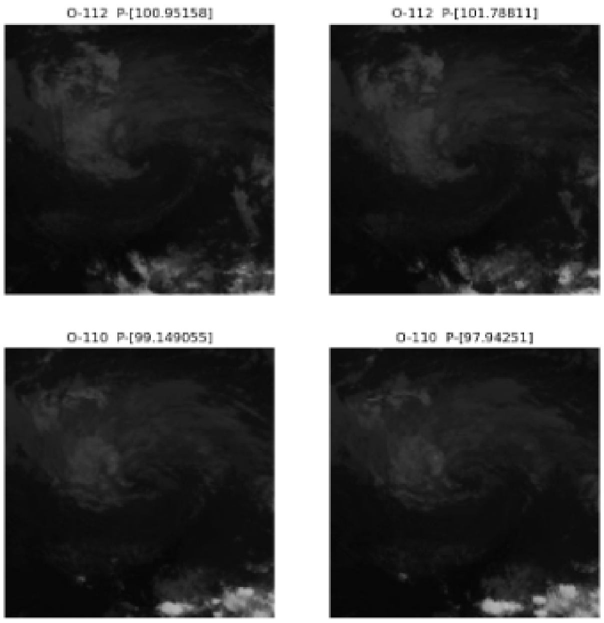

Fig 4.9. Predicted VS True wind speed values for Class: 3

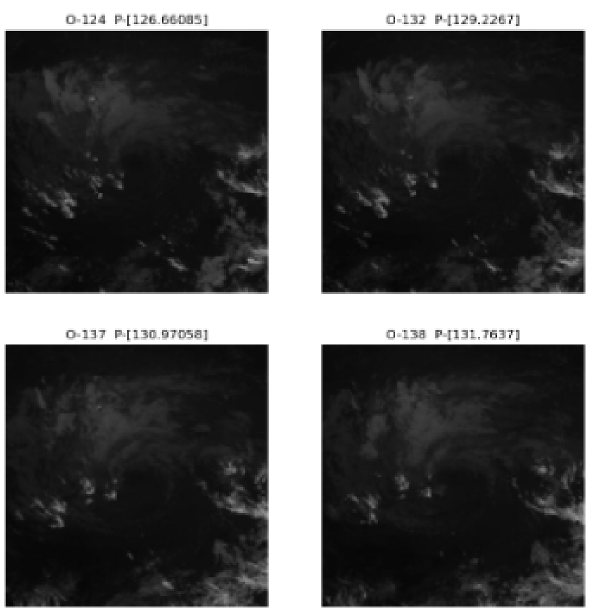

Fig 4.10. Predicted VS True wind speed values for Class: 4

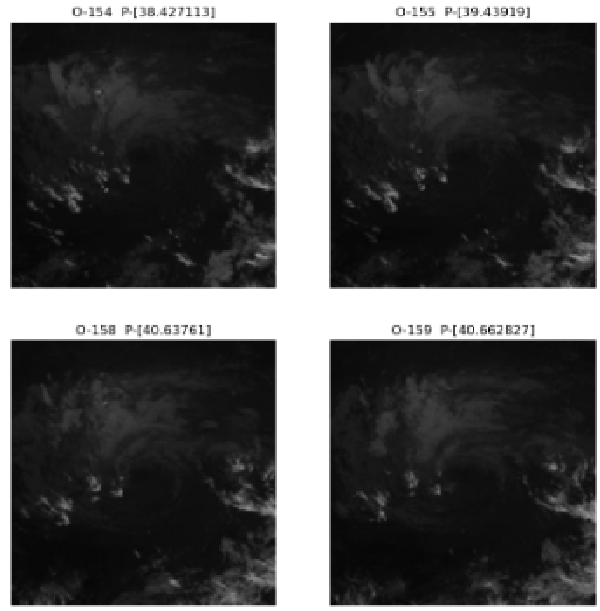

Fig 4.11. Predicted VS True wind speed values for Class: 5

Final Output:

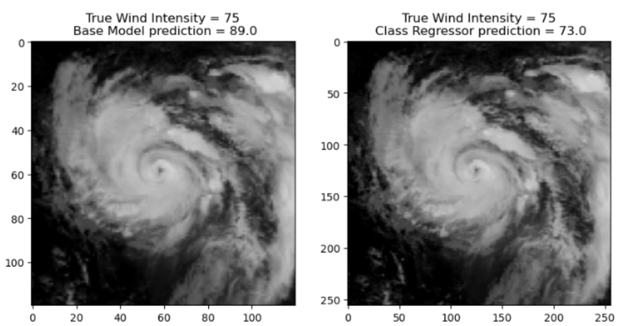

Fig 4.12. Predicted VS True wind speed for base regressor VS Spicy cloud model

5. Future Work

We have already talked about a lot of modifications that would increase the scope of the study. Reiterating some the most useful ones again:-

1. Usage of cGAN for artificial dataset generation and then semi-supervised learning for the same.
2. Better Edge case inclusion strategies into in-class regressors.
3. Better classification models to improve the performance upstream like trying out the

XGBoost + CNN classification models or Network in Network Models.
4. Usage of weighted ensemble method for the Regression output by utilizing the softmax results from classifiers.

These were some of the more obvious ones that we have come up with just from the basic study conducted.

Now, to significantly improve the performance and to simultaneously mitigate bias we suggest the following:

5. A probabilistic belief propagation approach to the estimation of intensity using the cloud formation pattern based on the Dvorak technique [8-10] to use the cloud formation gradients to estimate the class and intensity of the cyclone.
6. Training an image segmentation network for scraping HURSAT dataset and acquiring cyclone images for the missing class cyclones from the global satellite images to compensate for class bias.
7. Similar segmentation and scraping for Infrared images from IR and Water vapor Satellite images.

References

[1] Knutson, T. R., McBride, J. L., Chan, J., Emanuel, K., Holland, G., Landsea, C., ... Sugi, M. (2010). Global Warming and Hurricanes. Geophysical Fluid Dynamics Laboratory.

[2] M. Maskey, R. Ramachandran, I. Gurung, B. Freitag, M. Ramasubramanian, J. Miller (2020) "Tropical Cyclone Wind Estimation Competition Dataset", Version 1.0, Radiant MLHub. [Date Accessed] https://doi.org/10.34911/rdnt.xs53up.

[3] Sshubam Verma. (2022). INSAT3D Infrared & Raw Cyclone Imagery (2012-2021) [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/3868589.

[4] Jiang S, Tao L. Classification and Estimation of Typhoon Intensity from Geostationary Meteorological Satellite Images Based on Deep Learning. Atmosphere. 2022; 13(7):1113. https://doi.org/10.3390/atmos13071113.

[5] Qin, Z., He, Z., Li, Y., Supat, S., & Koike, Y. (2022). A CW-CNN regression model-based real-time system for virtual hand control. Frontiers in Neurorobotics, 16, Article 1072365. https://doi.org/10.3389/fnbot.2022.1072365.

[6] Jernelv, I. L., Hjelme, D. R., Matsuura, Y., & Aksnes, A. (2018). Convolutional neural networks for classification and regression analysis of one-dimensional spectral data. arxiv. https://doi.org/10.48550/arXiv.2005.07530

[7] Knapp, K. R. and J. P. Kossin, 2007: New global tropical cyclone data from ISCCP B1 geostationary satellite observations. Journal of Applied Remote Sensing, 1, 013505. doi:10.1117/1.2712816

[8] Velden, Christopher; Bruce Harper; Frank Wells; John L. Beven II; Ray Zehr; Timothy Olander; Max Mayfield; Charles “Chip” Guard; Mark Lander; Roger Edson; Lixion Avila; Andrew Burton; Mike Turk; Akihiro Kikuchi; Adam Christian; Philippe Caroff & Paul McCrone (September 2006). "The Dvorak Tropical Cyclone Intensity Estimation Technique: A Satellite-Based Method That Has Endured For Over 30 Years" (PDF). Bulletin of the American Meteorological Society. 87 (9): 1195–1214.

[9] Dvorak, V. F., 1975: Tropical Cyclone Intensity Analysis and Forecasting from Satellite Imagery. Mon. Wea. Rev., 103,420–430, https://doi.org/10.1175/1520-0493(1975)103<0420:TCIAAF>2.0.CO;2.

[10] Maskey, M., Ramachandran, R., Ramasubramanian, M., Gurung, I., Freitag, B., Kaulfus, A., ... & Miller, J. (2020). Deepti: Deep-learning-based tropical cyclone intensity estimation system. IEEE journal of selected topics in applied Earth observations and remote sensing, 13, 4271-4281.

[11] V. Dvorak, "A technique for the analysis and forecasting of tropical cyclone intensities from satellite pictures", pp. 1-19, 1973.

[12] Piñeros, M. F., Ritchie, E. A., & Tyo, J. S. (2011). Estimating Tropical Cyclone Intensity from Infrared Image Data, Weather and Forecasting, 26(5), 690-698. doi: https://doi.org/10.1175/WAF-D-10-05062.1

[13] https://github.com/23ccozad/hurricane-wind-speed-cnn

[14] https://github.com/miniufo/besttracks

[15] https://github.com/Zahlii/tropical-cyclone-speed-cnn/tree/master/process

[16] Individual Code references are provided in the code itself.