Hurricane Classification using Ensemble Techniques

Preprint · May 2023

DOI: 10.13140/RG.2.2.34248.49921

2 authors:

Kanchan Maurya
Yeshiva University
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Reiyo Reiyo
Yeshiva University
5 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Hurricane Classification using Ensemble Techniques

Kanchan Subhashchandra Maurya Reiyo
Yeshiva University
kmaurya@mail.yu.edu rreiyo@mail.yu.edu

1. Abstract

Artificial intelligence (AI) and machine learning (ML) have become powerful tools for solving complex problems in various fields, including meteorology and weather forecasting. Accurate and timely classification of hurricanes is essential for taking proactive measures and mitigating their impact on human life, property, and infrastructure. However, manual classification of hurricanes using satellite images is a time-consuming and error-prone task.

This project aims to develop a deep learning model for hurricane classification using Ensemble Techniques on the Hurricane and Severe Storm Sentinel (HS3) Naval Research Laboratory (NRL) Tropics Satellite Data V1. The dataset contains over 9,000 unlabelled satellite images, making labelling and classification accuracy a challenge. The project proposes using object detection models to label the images and ensemble techniques to combine the predictions of a handcrafted model and a pre-trained model to improve accuracy.

Evaluation metrics such as accuracy , error and loss will be used to evaluate the performance of the models, and cross-validation will be performed to prevent over-fitting. The model will be deployed for real-time hurricane classification, integrated with existing weather forecasting systems, to mitigate the impact of hurricanes on human life and infrastructure.

2. Introduction

There is a need to develop an automated system that can accurately classify hurricanes using satellite images. This project aims to address this need by developing a deep learning model for hurricane classification using Ensemble Techniques on the Hurricane and Severe Storm Sentinel (HS3) Naval Research Laboratory (NRL) Tropics Satellite Data V1.

The proposed model will combine a handcrafted model and a pre-trained model, and ensemble techniques will be used to improve accuracy. The model will be evaluated using various metrics, and once it is deemed satisfactory, it will be deployed for real-time hurricane classification, integrated with existing weather forecasting systems, to mitigate the impact of hurricanes on human life and infrastructure.

This project focuses on developing a deep learning model to classify hurricanes using the unlabelled images data. With more than 9,000 images, this dataset poses a challenge in terms of labelling and classification accuracy. In this project, we have taken on the challenge of developing an automated system to accurately classify hurricanes using satellite images. To accomplish this, we manually labeled 3,000 images, with 1,500 images featuring hurricanes and 1,500 without. We then used YOLOv5 to label an additional 6,000 images, checking to ensure the output was correctly separated into two folders.

To improve the accuracy of the model, we are using combinations of different pre-trained models such as ResNet50, ResNet101, EfficientNet, and GoogleNet. To augment the data, we applied various data augmentation techniques, including random rotation, horizontal and vertical flips, and random resized crop, to the images. We then used normalization to make sure the input data had zero mean and unit variance.

Figure 1. The architecture of various ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Once the model is developed, it will be evaluated using various metrics, and ensembling techniques will be applied to improve its accuracy further.

Finally, we will deploy the model for real-time hurricane classification, integrating it with existing weather forecasting systems to help mitigate the impact of hurricanes on

Figure 2. The architecture of Efficient

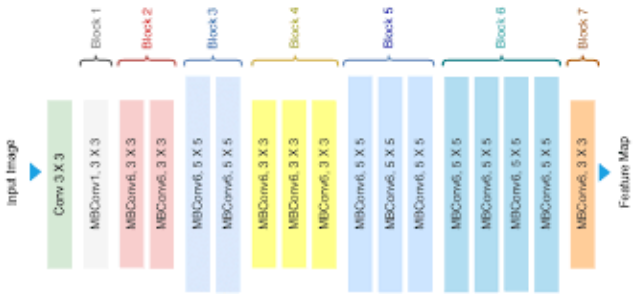
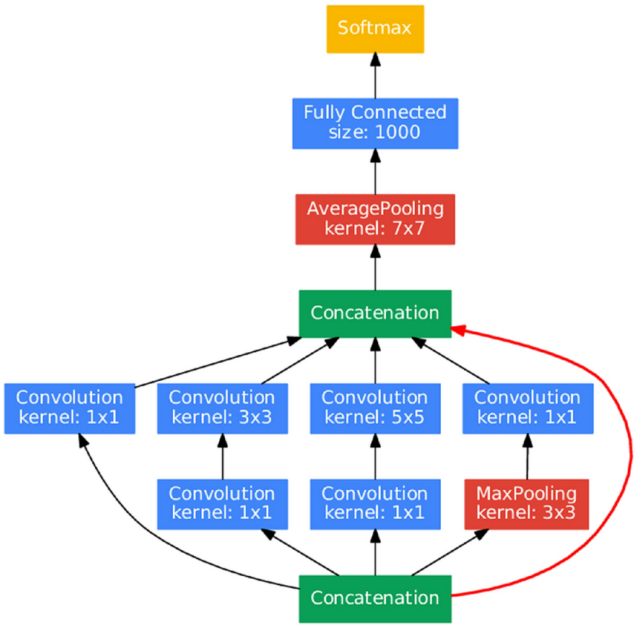


Figure 3. The architecture of Google Net



human life and infrastructure.

This project aims to develop a robust deep learning model for hurricane classification that can help in forecasting the trajectory of hurricanes and reducing their impact on human life and property.

3. Problem Statement

The problem addressed in this project is the accurate and timely classification of hurricanes using satellite images. Manual classification of hurricanes is a time-consuming and error-prone task, and developing an automated system for hurricane classification using deep learning models can significantly improve the efficiency and accuracy of the classification process. The dataset used for this project poses a challenge in terms of labelling and classification accuracy, with more than 9,000 unlabelled images. The aim of this project is to develop a robust deep learning model for hurricane classification using ensembling techniques on the Hurricane and Severe Storm Sentinel (HS3) Naval Research Laboratory (NRL) Tropics Satellite Data V1, which will be evaluated using various metrics such as accuracy, error, and loss. The model will be deployed for real-time hurricane classification, integrated with existing weather forecasting systems, to mitigate the impact of hurricanes on human life and infrastructure.

3.1. Data Acquisition and Labelling

The dataset used in this project was obtained from NASA's HS3 NRL Tropics Satellite. This dataset comprises high-resolution satellite images of clouds captured by the HS3 NRL Tropics Satellite during the hurricane season. The dataset consists of over 9,000 unlabeled JPEG images of clouds. Before using the images for training the deep learning model, they were preprocessed and resized to a standardized size.

The unlabeled images need to be labelled before they can be used for training the model. The dataset's large collection of images provides a diverse range of cloud patterns and formations, making it suitable for developing accurate and robust models for hurricane classification.

To develop the deep learning model, we used a combination of labeled and unlabeled data. Initially, we manually labeled 3,000 images, with 1,500 images featuring hurricanes and 1,500 without. We ensured that the labeling was accurate and consistent to avoid any biases during training. To increase the dataset size, we used YOLOv5 to label an additional 6,000 images, which were checked to ensure they were correctly separated into two folders.

Figure 4. Dataset Distribution for Labelling

LABELLING DATASET METHOD	MANUALLY	YOLOV5
NUMBER OF IMAGES	3000	6000

The final dataset consisted of 9,000 images of clouds, with each image resized to a standardized size before being used for training and testing the models. To improve the accuracy and robustness of the model, we are using ensembling techniques and a combination of different pre-trained models, including ResNet50, ResNet101, EfficientNet, and GoogleNet, along with a handcrafted model.

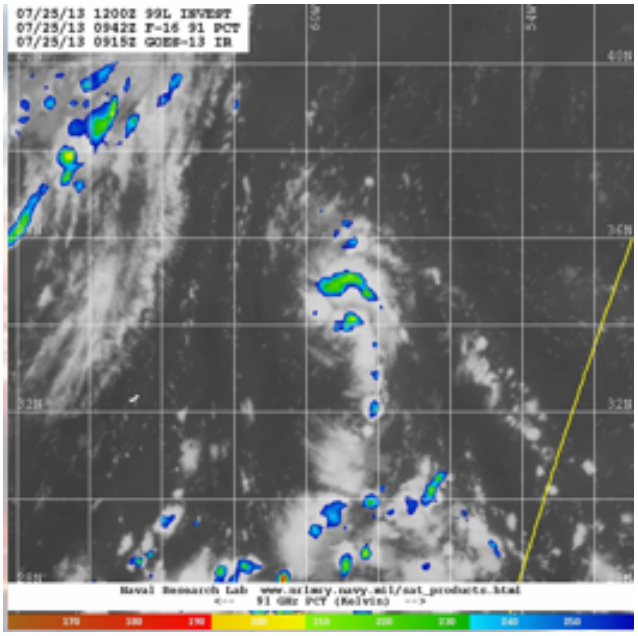
Figure 5. The Dataset for Training and Testing

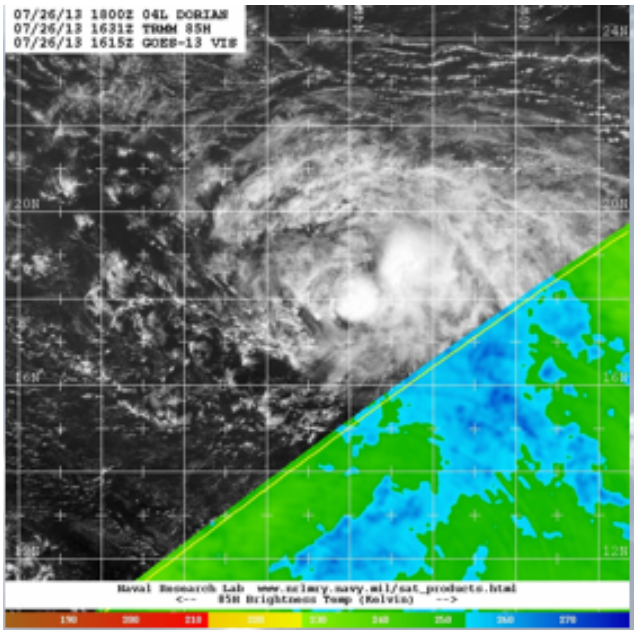
DATASET	TRAIN	TEST
NUMBER OF IMAGES	6000	3000

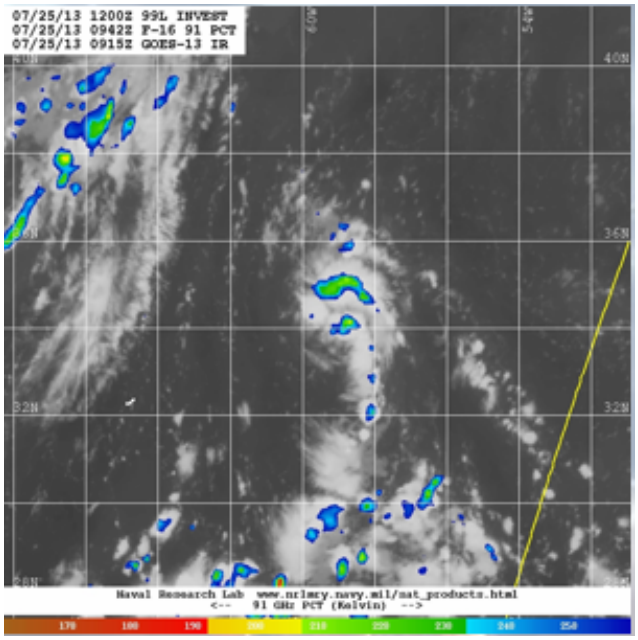
The models will be fine-tuned on the labeled dataset and evaluated using various metrics to select the best performing model.

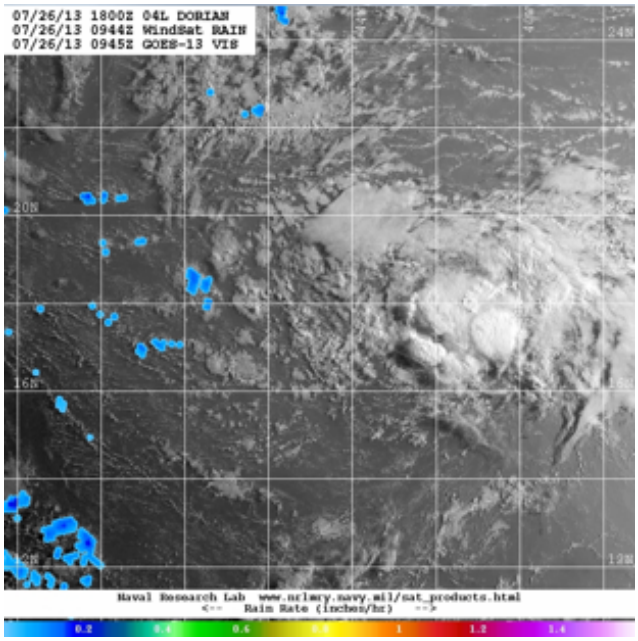
3.2. Sample Results: Yolo V5 result labeling hurricanes- PRESENT

The Images that were trained by YOLO V5 and depicted the presence of Hurricanes.





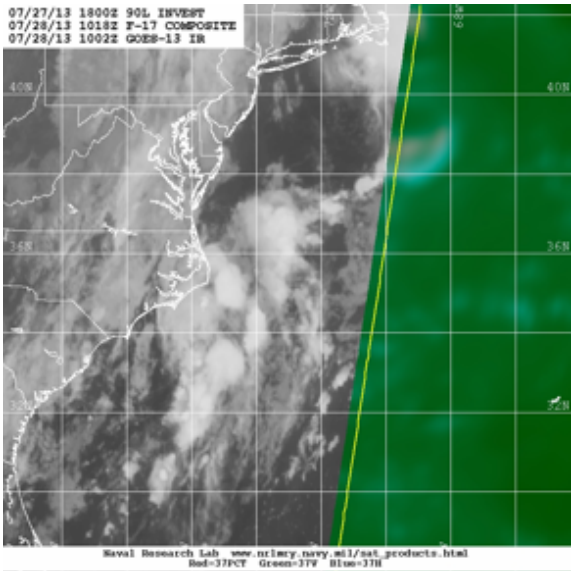


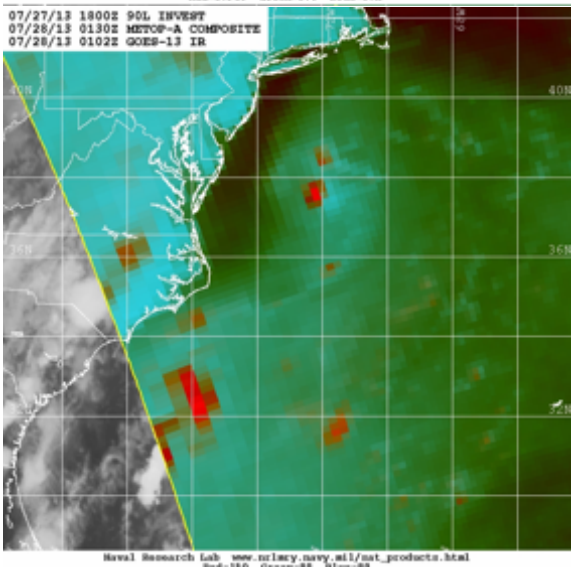


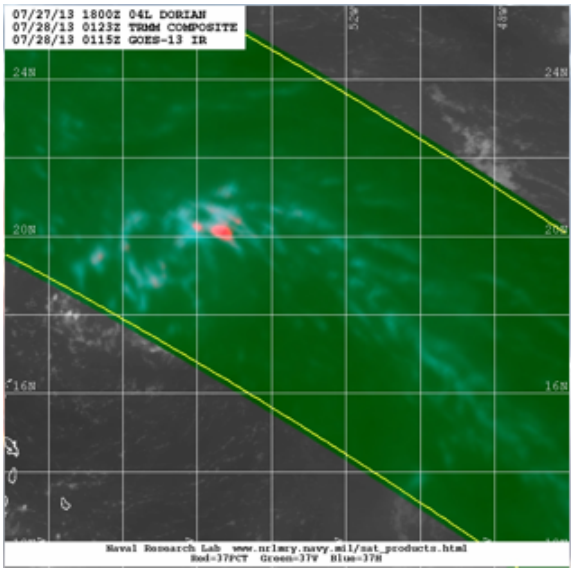
These result depict that model has classified correctly.

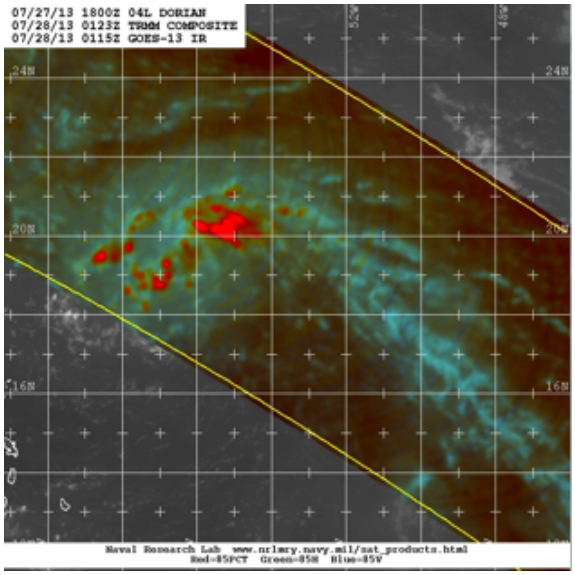
3.3. Sample Results: Yolo V5 result labeling hurricanes- ABSENT

The Images that were trained by YOLO V5 and depicted the absence of Hurricanes.









These result depict that model has classified correctly.

4. Related Work

In recent years, deep learning has shown great potential in solving complex problems in various fields, including meteorology and weather forecasting. Several studies have been conducted to develop deep learning models for hurricane classification using satellite images.

The paper of Dawood et al. proposed Deep-PHURIE, a deep learning-based model for hurricane intensity estimation using infrared satellite imagery. The model used a convolutional neural network (CNN) to extract features from the images and achieved a mean absolute error of 2.6 knots in hurricane intensity estimation. [4]

In another study,Pradhan et al. (2017) also proposed a CNN-based model for tropical cyclone intensity estimation using infrared satellite images. Their model used a residual network architecture and achieved a mean absolute error of 4.3 knots in intensity estimation. [6]

The paper of Zhang et al. (2021) proposed a model for tropical cyclone intensity classification and estimation using infrared satellite images and deep learning. They used a combination of CNN and long short-term memory (LSTM) networks and achieved an accuracy of 96.3 percent in intensity classification. [9]

The Neptune AI blog post [2]was a valuable resource for my project as it helped us to understand the differences between various object detection algorithms and libraries. It also aided in selecting and understanding the most appropriate algorithm for our specific project requirements, considering its performance and suitability. Apart from the rest mentioned resources used.[1] [3] [7] [8] [5]

These studies show the potential of deep learning in improving hurricane forecasting and intensity estimation, and motivate our work to develop a deep learning model for

hurricane classification using ensemble techniques.In this project, we aim to develop a deep learning model for hurricane classification using ensemble techniques to improve accuracy and robustness.

5. Methods

The given code aims to tackle class imbalance in a dataset through data augmentation and random sampling. The first step involves defining data augmentation transforms that randomly rotate, flip, resize, and normalize the images. These transforms are then applied to the hurricane images and saved to a new directory.

Next, both the hurricane and no hurricane images are loaded, and the number of samples is defined to balance the classes. Random sampling is used to randomly select a subset of hurricane and no hurricane images with the same number of samples. These balanced images are then moved to new directories.

To prepare the dataset for training and testing, the code creates train and test folders to store the data. The dataset is split into training and validation sets with a specified proportion of images to use for validation. The training and validation files are then copied to their respective directories.

Lastly, the code defines data transforms that resize, convert, and normalize the images to tensors and creates data loaders to store the training and testing image batches. Overall, the code provides a comprehensive approach to tackling class imbalance in a dataset, ensuring that both classes are represented equally in the training and testing sets

We propose to use ensembling techniques for hurricane classification. Ensembling techniques are a popular way to improve the accuracy of machine learning models. For hurricane classification, we propose to use ensembling techniques that combine a handcrafted model and pre-trained models such as ResNet50, ResNet101, EfficientNet, and GoogleNet.

The first model we will use is a handcrafted Convolutional Neural Network (CNN) with multiple layers for feature extraction from the images. This model will be trained on the hurricane image dataset to learn the features that are specific to hurricane images.

The second set of models we will use are pre-trained CNNs such as ResNet50, ResNet101, EfficientNet, and GoogleNet. These models are trained on a large dataset of images, typically the ImageNet dataset, and are capable of extracting generic image features that are useful for a wide range of image classification tasks.

To use these pre-trained models for hurricane classification, we will replace their last fully connected layer with a new layer that is specific to hurricane classification. This is because the last layer of a CNN is typically a classification layer that is specific to the dataset on which the model was trained.

To combine the predictions of these models, we will use an ensemble method such as majority voting. This means that we will take the predictions from each model and select the prediction that receives the most votes. We may also use more advanced ensemble techniques such as stacking, where the predictions of the individual models are used as input to a meta-model.

ResNet is a widely used convolutional neural network that has achieved state-of-the-art performance on a variety of computer vision tasks. It introduces the residual block, which allows for deeper networks to be trained more easily by mitigating the vanishing gradient problem. ResNet50 and ResNet101 are variations of the original ResNet architecture, with 50 and 101 layers, respectively. EfficientNet is another popular architecture that achieves high accuracy with fewer parameters than other models. It uses a compound scaling method to optimize the scaling of depth, width, and resolution in a principled way. GoogleNet, also known as Inception v1, is an early architecture that introduced the idea of inception modules, which are parallel pathways of convolutional layers with different filter sizes.

Now, The ensemble method of majority voting. In this method, each model makes a prediction for the input, and the class with the most number of votes is chosen as the final prediction. For example, if there are three models, and each one predicts the classes A, B, and C for an input, respectively, then the majority voting method would choose the class with the most votes, which in this case is class A. This method is simple and easy to implement, and can often lead to improved performance compared to using a single model. However, it may not be as effective when the models have high correlation or when the classes are imbalanced.

In addition to majority voting, there are other ensembling methods such as stacking, which was briefly mentioned in the original statement. In stacking, the predictions of the individual models are used as input to a meta-model, which then makes the final prediction. This method can be more effective than majority voting when the individual models have different strengths and weaknesses, and can be optimized further by tuning the hyperparameters of the meta-model.

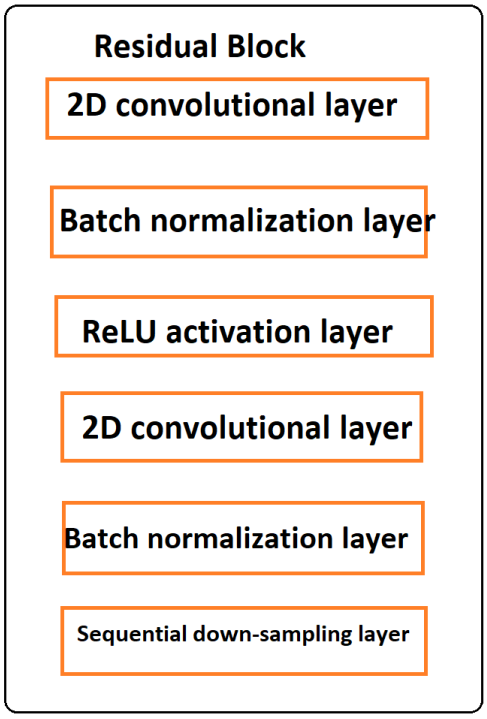
Overall, using an ensemble of handcrafted and pre-trained models can help improve the accuracy of hurricane classification, especially when combined with ensembling techniques. By leveraging the strengths of each model and combining their predictions, we can achieve more accurate and robust hurricane classification.

5.1. Handcrafted CNN with Residual Blocks

The architecture provided is a Handcrafted CNN with Residual Blocks. It is used for image classification tasks

and consists of multiple layers for feature extraction from the images. The architecture includes residual blocks that help the model to learn more complex features by skipping some layers in the neural network. The Residual Block class is defined with the init and forward methods.

Figure 6. The architecture of Residual Block



In the init method, the ResidualBlock class takes in channels, out channels, stride, and downsample as input parameters. It initializes two convolutional layers, batch normalization layers, and a ReLU activation function. The downsample parameter is used for downsampling the input if the stride is not equal to 1 or the number of input channels is not equal to the number of output channels.

The CovNet class is defined, which is a child class of the nn.Module class. It initializes the first convolutional layer, batch normalization layer, and ReLU activation function. It then initializes the residual blocks using the make layer method, which takes the number of output channels, the number of residual blocks, and the stride as input parameters. The make layer method creates a list of residual blocks using the ResidualBlock class.

In the forward method, the input image is passed through the CovNet model. It first goes through the convolutional layer, batch normalization layer, and ReLU activation function. It then goes through each of the residual blocks in order, which extract more complex features from the image. Finally, the output is passed through the fully connected layers, which perform the classification task.

It then trains the model on the training dataset for the specified number of epochs using the optimizer and loss criterion provided. During each epoch, the function computes the running loss and accuracy of the model and prints them. Finally, the function returns the trained model and plots the training loss and accuracy graphs.

The Adam optimizer is a popular optimization algorithm for training deep neural networks. It is an extension of stochastic gradient descent and uses adaptive learning rates for each parameter in the model. Adam works by computing a running average of both the gradients and the second moments of the gradients and then uses this information to update the parameters.

$$\text{Cross Entropy Loss} = \sum_{c=1}^{M} (y_{o,c} log(p_{o,c}),$$

Cross Entropy Loss is a loss function commonly used in classification tasks in deep learning. It measures the difference between the predicted probability distribution and the true probability distribution.

In the context of neural networks, it is commonly used with softmax activation functions to compute the probability distribution over the classes. The goal is to minimize this loss function during training, which improves the model's ability to correctly classify inputs into their respective classes.

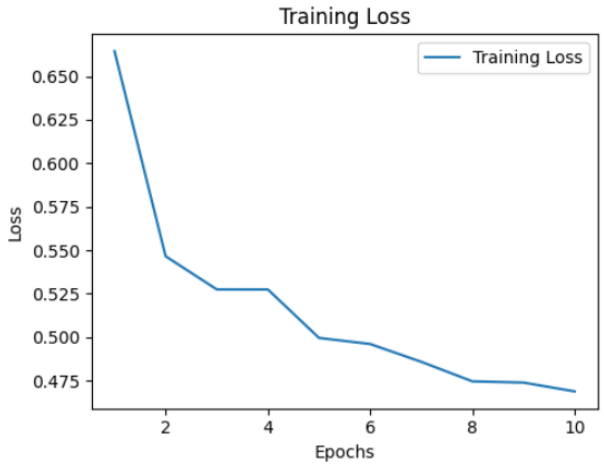
Overall, the architecture is designed to extract more complex features from the input images and classify them with high accuracy. The use of residual blocks helps in achieving this by learning complex features using fewer layers.

6. Results

The Plot shows how the training loss changes over time, where the x-axis represents the number of times the model has gone through the dataset (epochs) and the y-axis represents the loss. The loss measures how well the model is doing at predicting the correct output, so lower values are better. The training loss measures the difference between the predicted output and the actual output, with lower values indicating better performance.

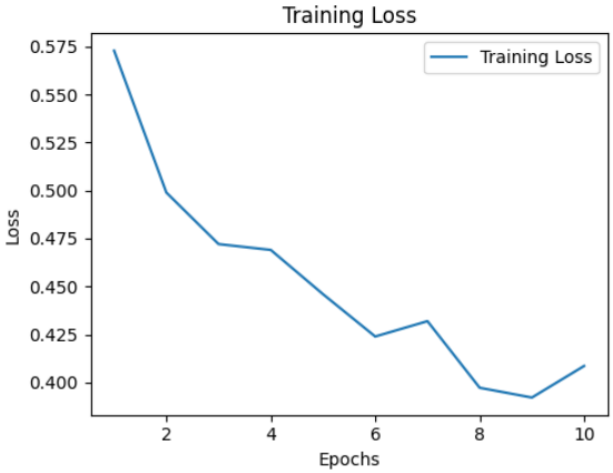
The graph indicates that the training loss decreases as the number of epochs increases, which suggests that the model is improving over time.

The Handcrafted Model- Train Loss

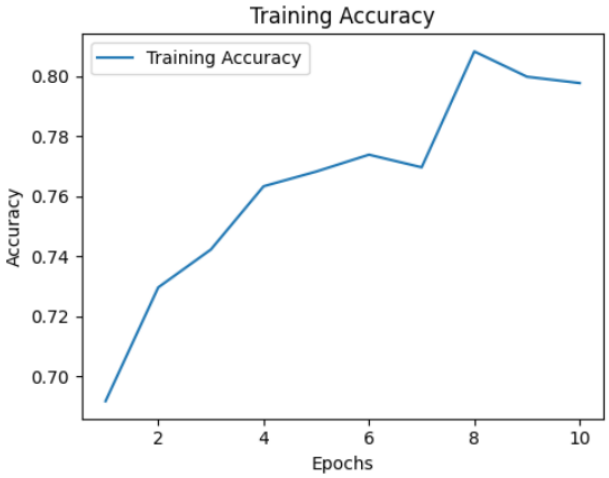


The Handcrafted Model- Train Accuracy

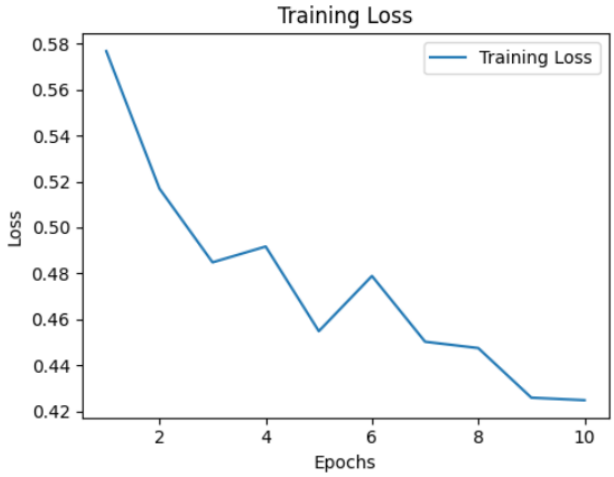
The training process for the hand-crafted CovNet model is shown in the output. The model is trained for 10 epochs, and the loss and accuracy are reported for each epoch.



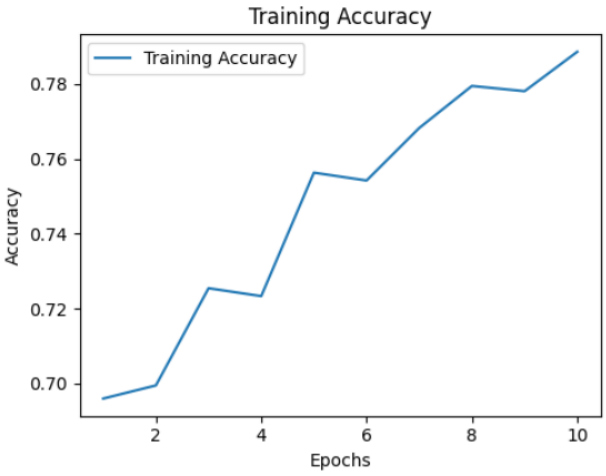
The Resnet50 Model- Train Loss



The Resnet50 Model- Train Accuracy



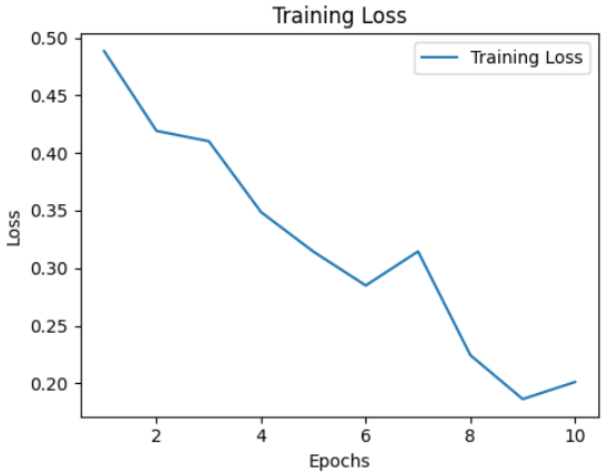
The Resnet101 Model- Train Loss



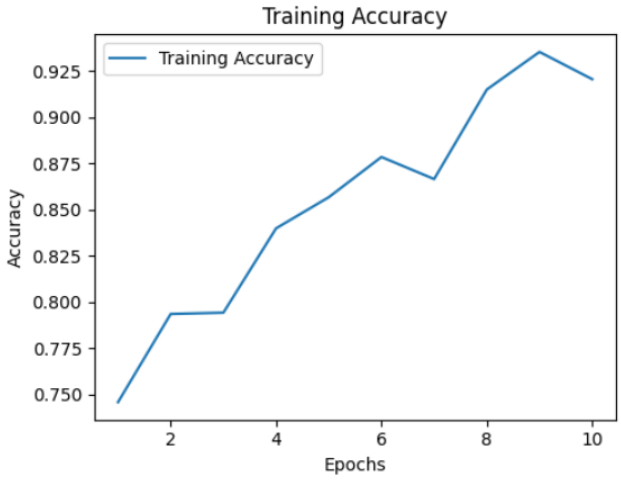
The Resnet101 Model- Train Accuracy

The training loss decreases as the model is trained, indicating that the model is learning to better fit the training data. The training accuracy also improves, indicating that the model is getting better at classifying the training data correctly.

The Google Net Model- Train Loss



The Google Net Model- Train Accuracy



At the end of the training process, the handcrafted model's final accuracy is 0.7538. These metrics give an indication of how well the model has learned to classify the data.

The second plot shows how the training accuracy change over time. The x-axis still represents the number of epochs, but the y-axis represents the accuracy of the model in percentage terms. The accuracy measures how well the model is correctly predicting the output, so higher values are better.

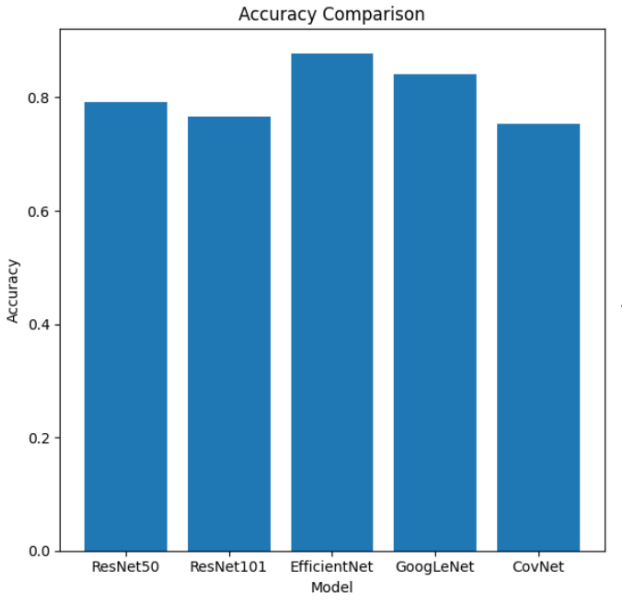


Figure : The Comparison of Models- Accuracy

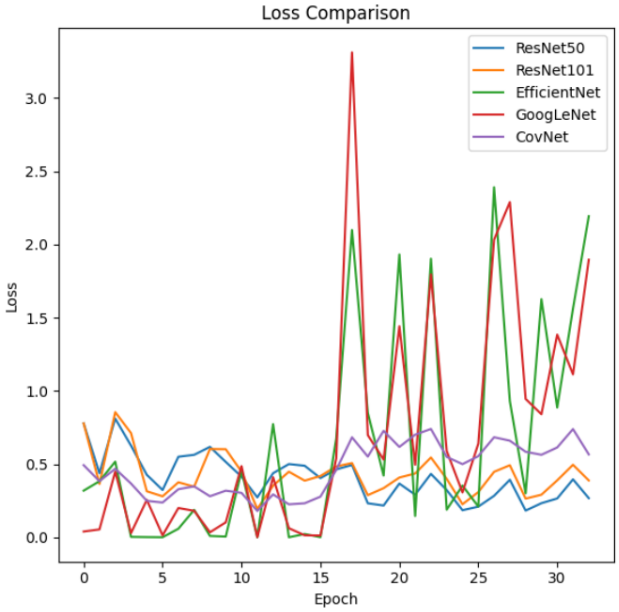


Figure : The Comparison of Models- Loss

We have achieved the below result after running the models for 10 epochs :

The accuracy of the ensemble model is 0.8068, indicating that the combination of different models is effective in improving accuracy. Among the individual models, EfficientNet achieved the highest accuracy of 0.8769, indicating that it was the most effective model in this particular task. GoogLeNet had an accuracy of 0.8409, followed by ResNet50 with an accuracy of 0.7917. ResNet101 had an accuracy of 0.7670, while CovNet had the lowest accuracy of 0.7538 among all the models used.

Overall, the statistics suggest that the ensemble model is more accurate than any individual model, with EfficientNet being the most effective individual model.

Its notably on how handcrafted model is able to achieve accuracy as good as Resnet 101 model.

7. Discussion

The model describes an approach for tackling class imbalance in a dataset through data augmentation and random sampling. The first step involved defining data augmentation transforms that randomly rotate, flip, resize, and normalize the images, which are then applied to the hurricane images and saved to a new directory. Next, the number of samples is defined to balance the classes, and random sampling is used to randomly select a subset of hurricane and no hurricane images with the same number of samples. These balanced images are then moved to new directories. Finally, in the model we defined data transforms that resize, convert, and normalize the images to tensors and creates data loaders to store the training and testing image batches.

The approach also proposes using ensembling techniques for hurricane classification. Ensembling techniques are a popular way to improve the accuracy of machine learning models. For hurricane classification, the approach proposes using ensembling techniques that combine a handcrafted model and pre-trained models such as ResNet50, ResNet101, EfficientNet, and GoogleNet. These models are trained on a large dataset of images and are capable of extracting generic image features that are useful for a wide range of image classification tasks.

To use these pre-trained models for hurricane classification, the approach replaces their last fully connected layer with a new layer that is specific to hurricane classification. This is because the last layer of a CNN is typically a classification layer that is specific to the dataset on which the model was trained. To combine the predictions of these models, the approach uses an ensemble method such as majority voting. This means that it takes the predictions from each model and selects the prediction that receives the most votes. The approach may also use more advanced ensemble techniques such as stacking, where the predictions of the individual models are used as input to a meta-model.

Overall, using an ensemble of handcrafted and pre-trained models can help improve the accuracy of hurricane classification, especially when combined with ensembling techniques. By leveraging the strengths of each model and combining their predictions, more accurate and robust hurricane classification can be achieved. The approach provides a comprehensive solution to tackle class imbalance in a dataset and ensures that both classes are represented equally in the training and testing sets, which is crucial for achieving accurate results.

7.1. Conclusion

In conclusion, the proposed approach for tackling class imbalance in a hurricane dataset through data augmentation and ensembling techniques is a comprehensive solution to improve the accuracy and robustness of hurricane classification. The first step involves defining data augmentation transforms and randomly sampling the images to balance the classes. Then, pre-trained models such as ResNet50, ResNet101, EfficientNet, and GoogleNet are used for hurricane classification. The approach replaces the last fully connected layer with a new layer that is specific to hurricane classification and uses ensemble methods such as majority voting to combine the predictions of these models.

The use of ensembling techniques for hurricane classification is a popular way to improve the accuracy of machine learning models. By leveraging the strengths of each model and combining their predictions, more accurate and robust hurricane classification can be achieved. The approach also ensures that both classes are represented equally in the training and testing sets, which is crucial for achieving accurate results.

Overall, the proposed approach provides a comprehensive solution to tackle class imbalance in a hurricane dataset and improve the accuracy and robustness of hurricane classification. This approach can be applied to other image classification tasks where class imbalance is an issue and can serve as a useful reference for researchers and practitioners working in the field of machine learning and computer vision.The model can be integrated with existing weather forecasting systems to provide accurate and timely information about hurricanes.

References

[1] https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/. 4

[2] https://neptune.ai/blog/object-detection-algorithms-and-libraries. 4

[3] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. 4

[4] Muhammad Dawood, Amina Asif, and Fayyaz ul Amir Afsar Minhas. Deep-phurie: deep learning based hurricane intensity estimation from infrared satellite imagery. *Neural Computing and Applications*, 32, 2020. 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[6] Ritesh Pradhan, Ramazan S Aygun, Manil Maskey, Rahul Ramachandran, and Daniel J Cecil. Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Transactions on Image Processing*, 27(2), 2017. 4

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent

Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 4

[8] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 4

[9] Chang-Jiang Zhang, Xiao-Jie Wang, Lei-Ming Ma, and Xiao-Qin Lu. Tropical cyclone intensity classification and estimation using infrared satellite images with deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2021. 4