

Deep Learning for Post-Hurricane Damage Detection with SAR-based Analysis using DenseNet201 and SVM

Neha Kumari¹, Poonam Moral², Debjani Mustafi³, Abhijit Mustafi⁴, Shamama Anwar⁵ and Sudip Kumar Sahana⁶

¹⁻⁶Birla Institute of Technology Mesra, Ranchi, India

Email: phdcs10057.23@bitmesra.ac.in

Email: {phdcs10051.21, debjani.mustafi, abhijit, shamama, sudipsahana}@bitmesra.ac.in

Abstract—In the face of escalating natural disasters, particularly hurricanes, swift and accurate post-disaster assessment is imperative for effective humanitarian assistance and disaster response (HADR). This study delves into the challenges of employing machine learning and deep learning techniques tailored explicitly for post-hurricane damage detection. Leveraging advanced technologies such as synthetic aperture radar (SAR) and remote sensing, the research proposes an innovative methodology. Deep features are extracted using DenseNet201, and dimensionality reduction through Principal Component Analysis (PCA) is applied. A Support Vector Machine (SVM) classifier discerns damage from satellite images. The model is meticulously developed and rigorously tested on the Hurricane Harvey dataset. Results exhibit a promising accuracy of 97.10, laying a foundation for post-disaster damage identification. The comparison is also done with other classifiers based on performance metrics such as Accuracy, confusion matrix, etc., However, the study acknowledges areas for future exploration, including real-time implementation, scalability, interpretability of deep learning models, integrating human expertise, and addressing biases. Tackling these challenges promises advancements in disaster response, aiding timely and effective actions to mitigate the impact of hurricanes and similar calamities.

Index Terms— Hurricane Satellite images, Deep Learning, Classification, DenseNet201, Principal Component Analysis.

I. INTRODUCTION

In recent years, there has been a concerning upward trend in the frequency of natural disasters, which is directly proportional to the increase in the number of people who are prone to disaster [1]. Among these disasters, hurricanes stand out as one of the most common and perilous. Strong winds and heavy rainfall cause inland flooding and storm surge flooding [2]. The U.S Geological Survey (USGS) discovered in terms of casualties that floods are the most dangerous natural disaster [3]. At the time of hurricane disaster, situational awareness is the most important part for the managers of emergency that they have to face before they respond to the situation. There are some possible ways to evaluate the current situation and damage, i.e., depending on the response provided by the volunteers who drive around the area that is affected, which is commonly known as a windshield survey. Another way to evaluate hurricane damage is flood detection with the help of SAR (synthetic aperture radar) images [4], floods stand as one of the most devastating natural calamities globally [5]. Consequently, swift, and precise actions in Humanitarian Assistance and Disaster Response (HADR) become imperative during

such crises, playing a vital role in safeguarding thousands of lives [6]. For this response, some ground planning, preparations, and analysis must be performed on the damaged resources. This is a very important step in rescuing and helping people with food and resources on time, but it is time-consuming and laborious.

Many research has been done to overcome this issue i.e., reduce data collection time and apply deep learning in predicting the damage after and before a hurricane. Remote sensing technology has been rapidly increasing in recent years which provides high-resolution satellite images for the assessment of damages in hurricanes [7]. Traditionally this assessment is done by the experts through field reconnaissance and volunteers record all information related to damages, which is time-consuming if the field is large. For flood detection, Synthetic-aperture radar (SAR) images have been utilized. It helps in mapping features, roughness patterns, and texture [8]. The satellite image capture from this radar from various countries in the past provides a large amount of data which is useful for research. On these huge Hurricane-damage data, Artificial Intelligence (AI) can be used to analyze. There are extremely large events in these datasets which is a major challenge in research. To tackle hurricane damage analysis, the Deep learning technique, which is based on a machine learning approach has attracted much attention. Over the recent years, there have been notable and impressive advancements in the field of deep learning (DL) [9]. Deep learning-based methods, such as convolutional neural networks are in the limelight between the researchers. Features extracted from images are done by the CNN because it is very effective in detecting patterns in input images [10]. CNN pre-trained model is designed for feature extracting and binary class SVM is employed for the classification process to achieve higher accuracy.

1. In section 2, we thoroughly examined the difficulties in utilizing machine learning and deep learning for post-hurricane damage detection, exploring challenges specific to these domains.
2. The datasets utilized in this study were meticulously curated, and rigorous data preprocessing techniques were applied to ensure high-quality inputs for the model's analysis and interpretation, which is explained in section 3.
3. Section 4 describe our proposed model and its implementations and it will demonstrate different performance matrices and comparisons.

Finally, Sections 5 and 6 convey the results obtained, and the paper is consolidated with the direction for future research.

II. LITERATURE SURVEY

In recent years, the damage caused by hurricanes detected automatically with deep learning methods which are very effective and accurate. This includes analysis of damaged buildings, assessment, and segmentation which is cheery on top topics in HADR [11], [12].

Satellite imagery serves as a rich source of information about our planet, providing invaluable insights that are pivotal for global decision-making. These images play a key role in addressing a myriad of critical challenges, spanning from climate change [11] and sustainable water management [12] to ensuring food security [13], guiding urban planning [14], monitoring land use [15], and facilitating disaster response [16], among other applications. Their versatility extends to vital domains like fire detection. In the context of damage assessment, the procedure can be bifurcated into two distinct tasks: identifying structures and classifying the extent of damage. Building detection has been a subject of intensive research recently. Within the scholarly literature, this task is commonly approached as a semantic segmentation challenge. Satellite imagery analysis not only provides invaluable data but also presents an opportunity for innovation and advancements in various fields. Its potential to enhance our understanding of the world and contribute to informed decision-making is immeasurable. Using Optical remote sensing as follows: band rationing, visual interpretation, rule-based classification, and object-based classification [19]

In recent investigations, machine learning methodologies have been employed to evaluate the extent of post-hurricane damage through the analysis of satellite imagery. One research initiative concentrated on identifying submerged roads by contrasting satellite images before and after a particular event, albeit with limitations to specific types of damage.[17] Furthermore, two companies specializing in satellite imagery introduced unsupervised algorithms designed to identify flooded regions by analyzing the spectral characteristics of contaminated water [18]. Before the emergence of deep learning, an approach employing pattern recognition templates was implemented for detecting hurricane-related damages in multi-spectral images; however, it proved unsuitable for our pan-sharpened image dataset. With the continuous enhancement of algorithms and computational power, in comparison to other supervised learning methods, CNN has become more effective and popular in the classification of images. Different types of CNN algorithms have been effectively and accurately

applied in the processing of remote sensing images [20] as follows: classification of aerial sensing images [21][22], object detection in unmanned aerial vehicle imagery [21], and classification of SAR images [22].

CNN, a potent deep learning algorithm, processes images by assigning weights and biases to distinct features, distinguishing and extracting intricate patterns. At its core lies the pivotal convolutional layer, housing filters that generate feature maps upon convolving with the input image. Additionally, the pooling layer operates on these maps individually, enhancing efficiency. The introduction of the Rectified Linear Unit (ReLU) layer imparts non-linearity, enriching the network's capacity for complex relationships. Lastly, the fully connected layer establishes comprehensive connections between neurons and previous activations, culminating in a sophisticated and adaptive neural network structure [23][24].

Predicting image features using transfer learning embraces leveraging pre-trained deep learning models to extract relevant features from images, which can then be used for various tasks such as classification, object detection, or segmentation. Transfer learning helps in transferring knowledge learned from one task or dataset to another, allowing models to generalize better on new, unseen data. In image processing, transfer learning involves using pre-trained neural networks, such as VGG, ResNet, DenseNet, or Inception, as feature extractors. These networks undergo training on extensive datasets such as ImageNet, followed by fine-tuning on more specialized tasks using smaller datasets. This approach leverages the learned features to improve performance on new, related tasks.

This paper delves into the realm of transfer learning by focusing on DenseNet201 for fine-tuning feature extraction. DenseNet201 serves as the foundational framework, providing an excess of learned features from a diverse range of images.

III. DATA PREPARATION

A. Dataset

In this paper, Hurricane Damage satellite Images are used as a dataset, which is extracted from IEEE Data port [21]. The Department of Industrial and Systems Engineering at the University of Washington took the lead in acquiring satellite imagery during Hurricane Harvey in 2017 within the Houston region. The primary objective of this endeavor was to elevate the accuracy and effectiveness of methods employed in detecting damage. The dataset amassed as part of this effort comprises 21,000 satellite images formatted in JPEG.

The images are classified into two classes i.e., damaged, and no_damage (see **Error! Reference source not found.**) In which damage data indicates a building affected by a hurricane disaster whereas no_damage data indicates an unaffected building. The size of the images is 224*224 pixels. The actual dataset is categorized into four folders: train_another, validation_another, test, and test_another, but in this paper, two datasets are used i.e., train_another and test which is used for the processing and prediction of damages:

- Training Set: It contains the 5000-training data and 5000 images of each class.
- Testing Set: It contains 2000 test data images in which 1000 of damaged and 1000 no_damaged classes.



Figure. 1 shows damaged and no-damaged satellite images in 128×128 pixels by comparison

B. Data Pre-processing

It is already acknowledged that when we train our deep neural networks, it results in a slow training process, and this happens due to the occurrence of gradient diffusion and vanishing gradient. To overcome this problem pre-processing of images is the most important stage. It improves the features of satellite images and rejects the

unwanted and noisy data present in the images [22]. In this paper, data processing methods are used, i.e., Normalization and Data rescaling by using the function TensorFlow. Keras.preprocessing.image.

After importing the dataset, Normalization is the most crucial step of preprocessing because it can easily maintain the numerical stability of the model of CNN. By using this method, the CNN model can quickly learn and make its gradient descent stable. In normalization, the picture values are set in the range of 0-1[23]. After Normalizing the data next step is to rescale it and it was done by the multiplication of 1/255 with the pixel values in the model.

IV. PROPOSED APPROACH

For modelling high-complexity data through nonlinear multi-layer transformations, the non-linear approximation function, i.e., the deep neural network, is used [24]. In deep learning networks, CNN is the effective algorithm for image format data, as recently, many researchers have analyzed breakthrough performance [25]. For extracting useful features, fine-tuning, and pretraining the models, Transfer learning is very effective and accurate.

A. Extracted features using DenseNet201

Extracting features from extensive datasets can be susceptible to human biases, potentially affecting the quality of features, which, in turn, impacts the classification task [26]. In this work, we address these challenges by utilizing a pre-trained and fine-tuned CNN classifier known as DenseNet201 for feature extraction [27], comprising 201 layers, to tackle the image category classification problem efficiently. Leveraging a pre-trained network on large datasets like ImageNet enhances the extraction of meaningful features from Hurricane Satellite images, removing the necessity for an extensive dataset. Within the DenseNet201 framework, the convolutional layers are distributed across four blocks termed 'dense blocks,' with 6, 12, 48, and 32 layers, respectively. These dense blocks incorporate bottleneck layers involving 1×1 and 3×3 convolutions to reduce the number of feature maps. The output size of each block is ultimately reduced from 112×112 to 7×7 pixels through a series of convolutional operations and pooling within the dense blocks.

The processing of Satellite images H_s involves a network of $L(L+1)/2$ direct connections, indicating that the L th layer receives feature maps from all previous layers, denoted as $f_0 \dots f_{L-1}$, as input while also utilizing its feature maps as input for all subsequent layers. The transition layers connecting the dense blocks supervise the execution of convolution and pooling operations. Each layer employs a non-linear transformation involving convolution, pooling, rectified linear units, and batch normalization, culminating in the composite function $C_L(\cdot)$. The inputs to $C_L(\cdot)$ are consolidated into a 1D tensor, simplifying the implementation. The fusion of feature maps across layers $0 \dots L-1$ is represented as $[f_0, f_1, \dots, f_{L-1}]$, with the L^{th} layer's output denoted as f_L , as expressed in 1.

$$f_L = C_L * ([f_0, f_1, \dots, f_{L-1}]) \quad (1)$$

After processing the images, 1920-D feature vectors are extracted by capturing the output immediately before the classification layer using the Global Average Pooling layer. This approach effectively captures spatial features and addresses the vanishing gradient issue, ultimately enhancing the performance of the model.

B. Dimension reduction of deep features using PCA

In the context of high-dimensional data, it is commonly noted that features exhibiting minimal variance tend to be redundant [28]. Thus, the main aim of dimension reduction methods is to eliminate these redundant features from the higher-dimensional dataset while preserving vital information within the reduced feature vector. PCA [29] plays a pivotal role in dimensionality reduction for the deep Convolutional Neural Network (CNN) features employed in this study. This process starts with the computation of the covariance matrix, which summarizes feature interrelationships. Subsequently, PCA computes eigenvectors and eigenvalues from this matrix, with eigenvectors representing the directions of maximum data variance and eigenvalues quantifying the variance along these directions. The eigenvectors with the largest eigenvalues, known as principal components, contribute to forming the reduced feature vector.

C. Classification using Support Vector Machine

The core aim of the Support Vector Machine (SVM) algorithm is to identify the optimal hyperplane within an N-dimensional space. Serving as a decisive boundary, this hyperplane proficiently partitions data points into discrete classes within the feature space. SVM accomplishes this by optimizing the margin, denoting the distance between the hyperplane and the nearest data points from each category. The pursuit of this optimal hyperplane

by SVM guarantees the most efficient segregation of classes in a high-dimensional space, establishing it as a potent tool across diverse machine-learning applications.

In our experiment, the problem is binary classification with two classes, damage, and no_damage which are labeled as +1 and -1. After training our model (DenseNet201 with PC) having inputs features vectors X and their corresponding class label Y . The equation for this linear hyperplane is given in “2”.

$$W^T x + b = 0 \quad (2)$$

Where W denotes the normal vector to the hyperplane which explain the direction perpendicular to the hyperplane, b is the offset or distance of the hyperplane along with vector W .

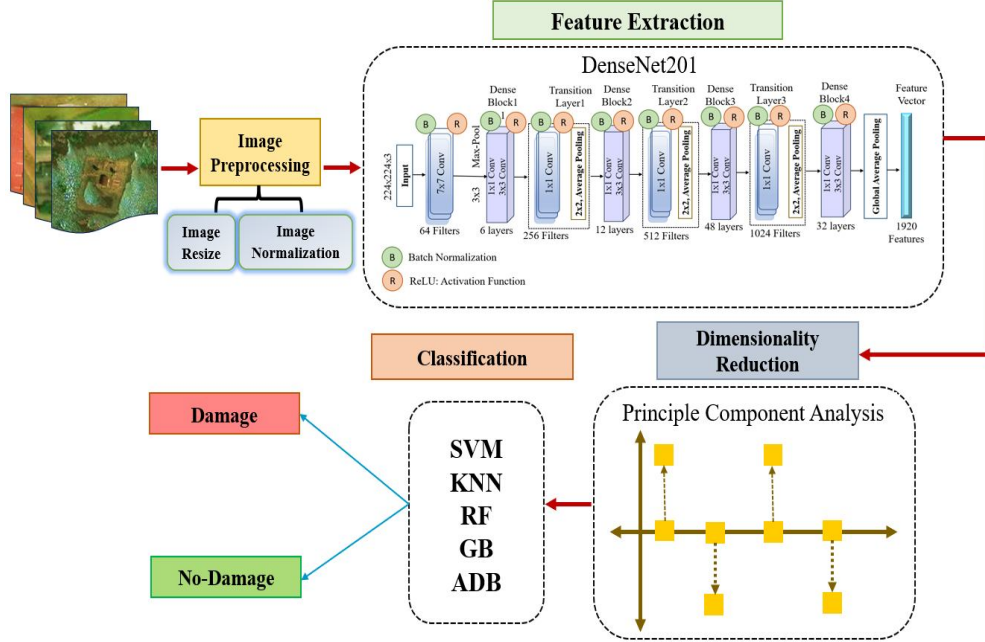


Figure 2. Flow chart of proposed approach

D. Implementation

In the Anaconda Navigator's platform utilizing the Jupyter Notebook editor, we conducted an experiment detailed in the flow chart (see Figure. 2). The implementation was carried out in Python, leveraging key libraries including sci-kit learn [30], TensorFlow, and Keras [31]. The dataset used was the Hurricane Harvey dataset, as discussed in section III.A. In the large size of the dataset, we performed image preprocessing involving normalization and re-scaling using ImageDataGenerator. The images were resized to (224,224), and the dataset was categorized into two classes: 'damage' and 'no_damage', leading to the use of binary class_mode.

For feature extraction, as explained in section IV. A, we employed the DenseNet 201 pre-trained model. Specifically, we selected intermediate layers for feature extraction, focusing on the block labeled "conv5-block32-con0cat". The resulting extracted features had a shape of (10000, 7, 7, 1920), representing a 4-dimensional NumPy array. This format implies that it encompassed features extracted from 10,000 images, with each image represented as a 7x7 grid of 1920 features. Recognizing the potential complexity introduced by the 4D array, we opted for dimensionality reduction of the extracted features using Principal Component Analysis (PCA), as discussed in section 4.2. PCA was applied to the flattened 2D array of extracted features, and the parameter num_components was utilized to specify the number of principal components (features) to retain after the reduction.

Subsequently, we integrated a Support Vector Machine (SVM) classifier to achieve significant accuracy with reduced computational demand. To evaluate performance, we compared the SVM classifier with alternative classifiers such as K-Nearest Neighbors (KNN), Adaboost, Gradient Boosting, and Random Forest. This comparative analysis aimed to discern the strengths and weaknesses of each classifier in the context of the implemented model.

In this experiment, several evaluation matrices were used to evaluate the performance of the model. These are the following:

Confusion Matrix: The Confusion Matrix serves as a pivotal indicator of the classification model's accuracy in predicting instances from different categories. Comprising two axes, the first denotes the expected label, while the second signifies the actual label [32].

Here, is what each term represents:

True Positive (TP): The model correctly identifies positive cases.

False Positive (FP): The model incorrectly identifies negative cases as positive.

True Negative (TN): The model accurately identifies negative cases.

False Negative (FN): The model mistakenly identifies positive cases as negative.

1. Accuracy: Accuracy: It is one of the simplest classifications of metrics, which is used to determine the number of correct predictions to the total number of predictions as shown in “3”.

$$Accuracy = \frac{TP(True\ Positive)+TN(True\ Negative)}{TP(True\ Positive)+TN(True\ Negative)+FP(False\ positive)+FN(False\ Negative)+FN} \quad (3)$$

2. Precision: To overcome the limitation of the Accuracy metric, the precision metric is used. It can be calculated as the True Positive or predictions that are true to the total positive predictions. 4 is the formula of precision.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Positive(FP)} \quad (4)$$

3. Recall: It is used to calculate the proportion of an actual positive which identified incorrectly. The equation is shown in “5”

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Negative(FN)} \quad (5)$$

4. F-Score: It is calculated as the harmonic mean of both Recall and precision. It assigns equal weight to each other. As shown in “6”

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (6)$$

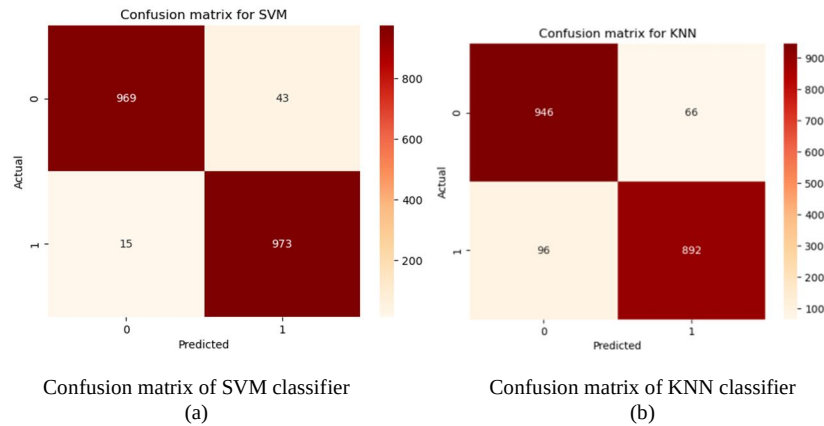
5. Specificity: It is used to measure the model’s ability to correctly identify negative outcomes. The formula is given in “7”

$$Specificity = \frac{True\ Negative(TN)}{True\ negative(TN)+False\ Positive(FP)} \quad (7)$$

6. AUC (Area Under the Curve)-ROC: AUC stands for Area Under the ROC curve. It calculates the two-dimensional area under the whole ROC curve.

V. RESULT ANALYSIS

The results of the study showcase a highly promising approach to post-hurricane damage identification utilizing machine learning and deep learning techniques, particularly tailored for efficient humanitarian assistance and disaster response (HADR).



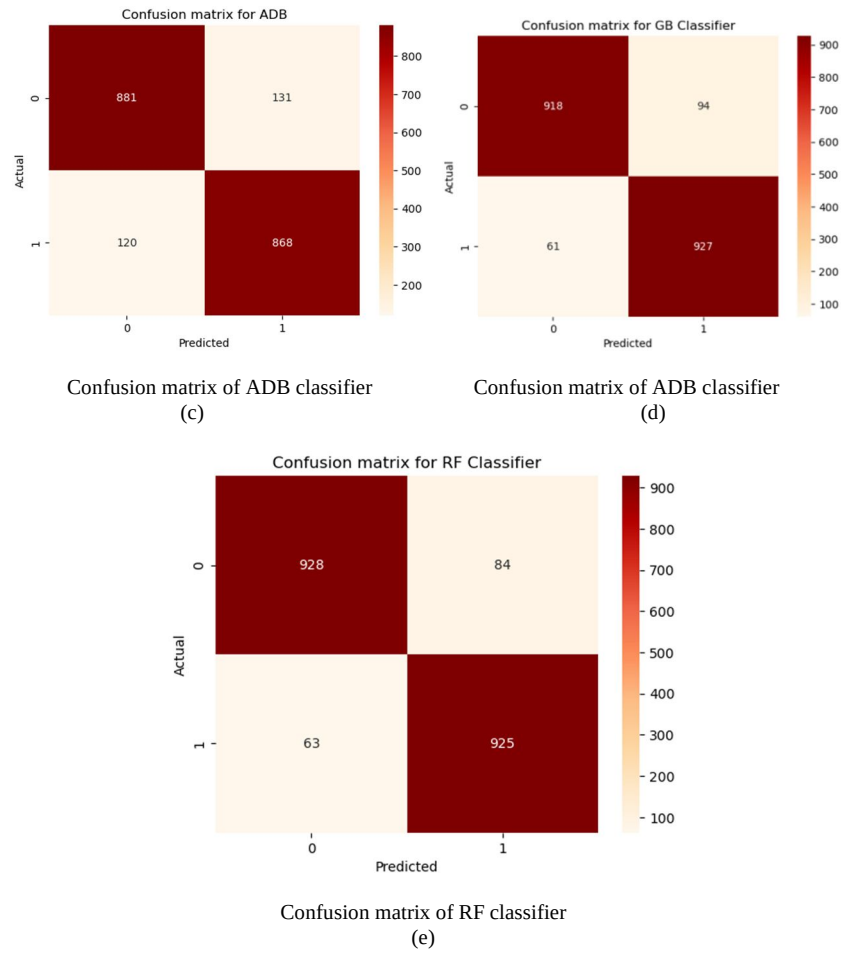


Figure 3. Confusion Matrix for different classifiers: (a) SVM (b) KNN (c) ADB (d) GB (e) RF

Illustrating the confusion matrix of all classifiers, through which the comparison is in transparency has been (see in Figure.3).

TABLE I. PERFORMANCE METRICS OF DIFFERENT CLASSIFIERS

PCA-Components	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
100	SVM	97.10	95.76	98.48	97.10	95.75
	KNN	91.90	93.11	90.28	91.67	93.47
	ADB	87.45	86.88	87.85	87.36	87.05
	GB	92.25	90.79	93.82	92.28	90.71
	RF	92.65	91.67	93.62	92.63	91.69
250	SVM	96.55	96.76	96.47	96.61	96.62
	KNN	93.40	95.31	91.58	93.41	95.29
	ADB	89.35	91.14	87.67	89.37	91.10
	GB	93.65	94.52	92.95	93.73	94.37
	RF	93.65	94.26	93.24	93.75	94.06
500	SVM	95.80	94.93	96.65	97.78	94.96
	KNN	92.45	93.18	91.39	92.28	93.47
	ADB	89.10	90.44	87.14	88.76	91.01
	GB	92.70	92.69	92.51	92.60	92.89
	RF	92.90	92.64	93.01	92.82	92.79

Table I presents the classification report results for our proposed model, employing various classifiers including SVM, KNN, Adaboost, Gradient Boosting (GB), and Random Forest (RF). The evaluation metrics considered include accuracy, precision, F1-score, recall, and specificity. Notably, the SVM classifier stands out as the top

performer, achieving the highest accuracy of 97.10% in our DenseNet201 with the PCA model, surpassing the performance of the other class

This section includes the results performed on the different number of components used in PCA for dimension reduction and with different performance metrics such as accuracy, F1-score, recall, precision, AUC, and confusion matrices. As discussed earlier in section IV C we must set several components differently to get the best result for the model. At first num_components = 250 we got 96.55 accuracy in our proposed model. similarly, num_components = 100 and 500. The best result we get in 100 components is 97.10. Our proposed approach (DenseNet201, PCA, and SVM) outperforms and starts calculating other performance metrics.

After setting the perfect Components in PCA, we compare our proposed model with other classifiers such as KNN, Adaboost, Gradient Boosting, and random forest. And the results we got are (see Figure.4).

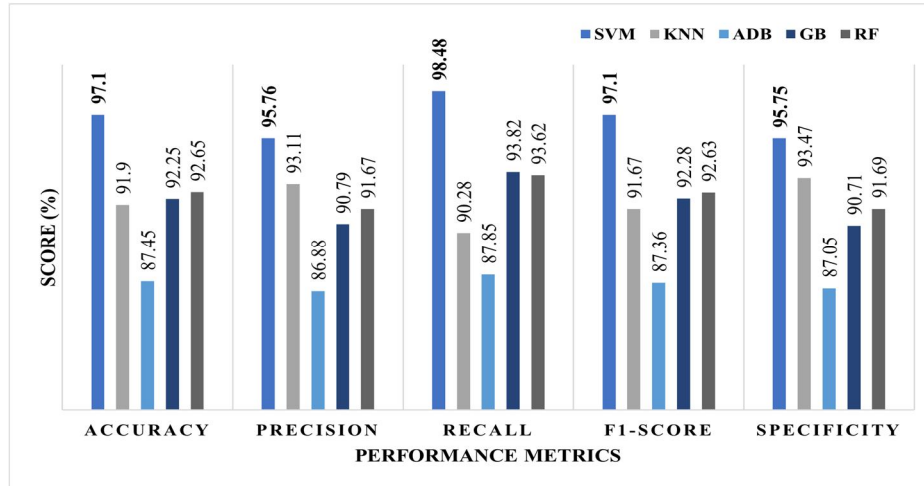


Figure. 4. Histogram of different classifiers with their performance metrics

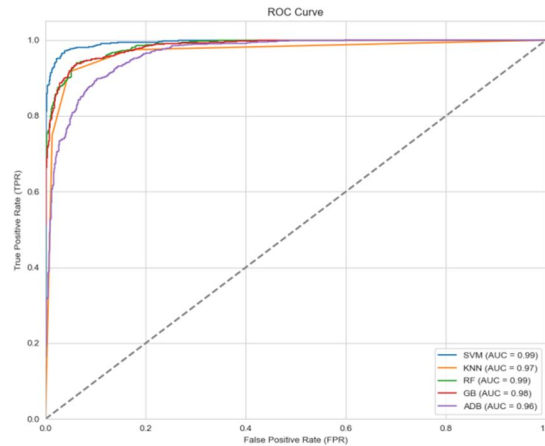


Figure. 5. AUC curve of Different classifiers

Receiver Operating Characteristic (ROC) curves are useful for evaluating the performance of classification algorithms, especially when dealing with imbalanced datasets. As depicted in Figure 5, the SVM classifier exhibits superior performance with an AUC value of 0.99, while the KNN classifier achieves an AUC of 0.97. Additionally, the Random Forest (RF) model, Gradient Boosting (GB) model, and AdaBoost (ADB) model attain AUC of 0.99, 0.98, and 0.96 respectively.

VI. CONCLUSION

This article addresses the challenges in utilizing machine learning and deep learning techniques for post-hurricane damage detection, focusing on the specific domains of hurricane disasters. The study emphasizes the

importance of immediate and accurate responses in humanitarian assistance and disaster response (HADR) to save lives during and after hurricanes. To achieve this, the research proposes an approach that combines deep learning, feature extraction using DenseNet201, dimensionality reduction using PCA, and classification using a support vector machine (SVM) to detect hurricane damage from satellite images. Based on evaluation metrics such as Accuracy, Confusion matrix, and AUC, SVM obtained the best result of 97.10 accuracy from the other classifiers Such as KNN, RF, GB, and Adaboost. The study does not address real-time implementation, where immediate response during a disaster is crucial. Future research should focus on developing algorithms and models that can provide real-time analysis of satellite images for timely disaster response.

ACKNOWLEDGMENT

This research work is part of the project supported by DST under the PURSE 2022 scheme.

REFERENCE

- [1] Y. Pi, N. D. Nath, and A. H. Behzadan, "Convolutional neural networks for object detection in aerial imagery for disaster response and recovery," *Advanced Engineering Informatics*, vol. 43, p. 101009, 2020.
- [2] Dataset link: "<https://www.noaa.gov/education/resource-collections/weather-atmosphere/hurricanes>"
- [3] Perry, Charles A. Significant floods in the United States during the 20th century: USGS measures a century of floods. Vol. 24. US Department of the Interior, US Geological Survey, 2000.
- [4] Dartmouth Flood Observatory (DFO). <http://floodobservatory.colorado.edu/> (2020).
- [5] T. Bui et al., "A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high-frequency tropical storm area," *Sci. Total Environ.*, vol. 701, p. 134413, 2020.
- [6] N. Pierdicca et al., "Triple collocation to assess classification accuracy without a ground truth in case of earthquake damage assessment," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 485–496, Jan. 2018
- [7] K. A. Duda and B. K. Jones, "Usgs remote sensing coordination for the 2010 haiti earthquake," *Photogrammetric Engineering & Remote Sensing*, vol. 77, no. 9, pp. 899–907, 2011.
- [8] R. S. Westerhoff et al., "Automated global water mapping based on wide-swath orbital synthetic-aperture radar," *Hydrol. Earth Syst. Sci.*, vol. 17, no. 2, pp. 651–663, 2013.
- [9] S. Kaur et al., "Detection of Alzheimer's disease using deep convolutional neural network," *Int. J. Image Graphics*, vol. 22, no. 3, pp. 2140012, 2022.
- [10] Q. D. Cao and Y. Choe, "Building damage annotation on posthurricane satellite imagery based on convolutional neural networks," *Natural Hazards*, vol. 103, no. 3, pp. 3357–3376, Sep. 2020.
- [11] L. Dong and J. Shan, "A comprehensive review of earthquake-induced building damage detection with remote sensing techniques," *ISPRS J. Photogramm. Remote Sens.*, vol. 84, pp. 85–99, Oct. 2013.
- [12] R. Anniballe et al., "Earthquake damage mapping: An overall assessment of ground surveys and VHR image change detection after L'Aquila 2009 earthquake," *Remote Sens. Environ.*, vol. 210, pp. 166–178, Jun. 2018.
- [13] B. Kalantar, N. Ueda, H. A. H. Al-Najjar, and A. A. Halin, "Assessment of convolutional neural network architectures for earthquake-induced building damage detection based on pre- and post-event orthophoto images," *Remote Sens.*, vol. 12, no. 21, p. 3529, Jan. 2020.
- [14] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.
- [15] G. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.
- [16] Y. Liu, Y. Zhong, and Q. Qin, "Scene classification based on multiscale convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2018.
- [17] Y. Bazi and F. Melgani, "Convolutional SVM networks for object detection in UAV imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3107–3118, 2018.
- [18] S. K. Vanama, Y. S. Rao, and C. M. Bhatt, "Change detection based flood mapping using multi-temporal Earth Observation satellite images: 2018 flood event of Kerala, India," *European Journal of Remote Sensing*, vol. 54, no. 1, pp. 42–58, 2021.
- [19] M. Rahnemoonfar et al., "Floodnet: A high resolution aerial imagery dataset for post flood scene understanding," *IEEE Access*, vol. 9, pp. 89644–89654, 2021.
- [20] R. Al-Saffar, R. K. Mohammed, W. M. Abed, and O. F. Hussain, "Detecting Damaged Buildings on Post-Hurricane Satellite Imagery based on Transfer Learning," *NeuroQuantology*, vol. 20, no. 1, p. 105, 2022.
- [21] M. Scannell et al., "Deep-learning-based preprocessing for quantitative myocardial perfusion MRI," *J Magn Reson Imaging*, vol. 51, no. 6, pp. 1689–1696, 2020.
- [22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016, [Online]. Available: <http://arxiv.org/abs/1607.06450>.
- [23] Y. Lecun et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

- [24] K.-C. Peng and T. Chen, "A framework of extracting multi-scale features using multiple convolutional neural networks," in 2015 IEEE International Conference on Multimedia and Expo (ICME), 2015.
- [25] Rasti et al., "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," IEEE Geoscience and Remote Sensing Magazine, vol. 8, no. 4, pp. 60-88, 2020.
- [26] Latif et al., "Content-based image retrieval and feature extraction: a comprehensive review," Mathematical problems in engineering, vol. 2019.
- [27] J. Abadicio et al., "Ground-level Post-Disaster Image Classification using DenseNet201 for Disaster Damage Assessment," in 2023 International Conference On Cyber Management And Engineering (CyMaEn), pp. 132-137, IEEE, 2023.
- [28] S. Kundu and S. Ari, "P300 detection with brain-computer interface application using PCA and ensemble of weighted SVMs," IETE Journal of Research, vol. 64, no. 3, pp. 406-414, Aug. 2017.
- [29] H. Zhao et al., "Convolutional neural network based heterogeneous transfer learning for remote-sensing scene classification," International Journal of Remote Sensing, vol. 40, no. 22, pp. 8506-8527, 2019.
- [30] Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
- [31] Chollet et al., "Keras," 2015, [Online]. Available: <https://keras.io>.
- [32] Alyami, A. Alharbi, and I. Uddin, "Lifelong machine learning for regional-based image classification in open datasets," Symmetry (Basel), vol. 12, no. 12, pp. 1-17, Dec. 2020.