IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 8, April 2024

Hurricane Disaster Mitigation by Processing Landscape Images

S. Shirabthinath and Indumathy M
Department of Computer Science & Engineering
SRM Institute of Science and Technology, Vadapalani Chennai, India

Abstract*: Dealing with the aftermath of natural disasters and deaths caused by them continue to be serious issues. Hurricanes' aftermath has received the majority of our attention. Hurricane Harvey is regarded as one of the most expensive natural disasters in recorded human history due to the destruction, loss of life, and property it inflicted in Texas. We used the Hurricane Harvey dataset to create a model that can accurately classify images of the terrain as damaged or undamaged, with the goal of saving lives and conserving human resources. Information is stored in data frames. After pre-processing the pictures, the database will be split into two labels, damage and no damage. The values are 0 and 1, respectively. The data frame is divided into testing and training data. Once the data has been normalised, the model is trained to assess whether or not an image is damaged. To improve the model's accuracy, as econdupdated model was produced. Alex net and VGG-16 were two pre-made models used in the development of the design. By contrasting these models, we were able to evaluate the landscape photos with up to 93% accuracy*

Keywords: Hurricane Harvey

I. INTRODUCTION

Our objective is to speed up this labor- and time-intensive process and provide a replacement system that, through the analysis of satellite images, aids in the mitigation of natural disasters.

Damage assessment is crucial for emergency management to take action quickly and allocate resources after a disaster.

By counting the number of destroyed buildings, surveying the region is a common practise that cans be used to determine the degree of the damage. This procedure can be quite time- and labor-consuming. Recently, symbolism obtained by robots and satellites has begun to contribute to enhancing situational mindfulness froma10,000 foot perspective. However, amid a developing crisis, the interface still relies on unreliable and time-consuming human visual processing of captured symbology.

Consequently, computer vision techniques can be quite helpful. Our recommended method can naturally explain "Hit Building" versus "Unharmed Building" on satellite symbols of a region destroyed by a typhoon due to the intelligible symbolism. The explanation results may give partners greater power to plan for and allocate vital resources in a more likely way, such as crisis directors. The potential for situational mindfulness and quick, accurate responses to problems caused by tropical storms can both be effectively reduced by this automated commenting process.

II. LITERATURE SURVEY

"Automated Post-Hurricane Damage Assessment Using Unmanned Aerial Vehicle (UAV) Imagery “by S. C. Olsen, R. C. Olsen, and S. J. Lindstedt: The authors used a Support Vector Machine (SVM) classifier to identify damaged buildings and non-damaged buildings, and then used image processing techniques to estimate the extent of damage to the identified buildings. The results showed that the proposed method could accurately detect and quantify the extent of hurricane damage.

"Damage Assessment of Hurricane Harvey Using High-Resolution Aerial Imagery and Convolutional Neural Networks "by A. G. Bokil, V. N. G .J. Mago, and D. Patel: The authors used a CNN to detect and classify four types of hurricane damage: missing roofs, broken windows, collapsed buildings, and uprooted trees. The CNN achieved high accuracy in detecting damaged buildings and vegetation, and the results showed that the proposed method could be used for rapid and accurate damage assessment after a hurricane

"A Hybrid Approach for the Detection of Flooded Areas in Hurricane Harvey" by M. R. Mosleh, H. K. Alawadhi, and A. M. Abbas: The authors proposed a hybrid approach that combined texture analysis and a fuzzy inference system to detect flooded areas in Hurricane Harvey using satellite imagery. The proposed method achieved high accuracy in detecting flood extent, and the results showed that the proposed method could be used for rapid and accurate flood mapping after a hurricane

"Robust and Scalable Object Detection for Hurricane Damage Assessment from Satellite Imagery" by Y. Li, C. Qi, and L. Zhang: The authors proposed a scalable object detection framework that combined a lightweight deep neural network with an attention mechanism to detect and classify different types of hurricane damage in satellite imagery. The proposed method achieved high accuracy in detecting and classifying damaged buildings and vegetation, and the results showed that the proposed method could be used for rapid and accurate damage assessment after a hurricane.

"A Deep Learning Framework for Disaster Monitoring and Mitigation Using Satellite Imagery" by J. Hu, Y. Li, and Z. Guo: The authors proposed a deep learning framework that combined CNNs and RNNs to monitor natural disasters such as hurricanes using satellite imagery. The proposed method achieved high accuracy in detecting and tracking different types of disasters, and the results showed that the proposed method could be used for real-time disaster monitoring and mitigation.

"Deep Learning Based Detection and Classification of Flooded Areas in Hurricane Events Using High-Resolution Satellite Imagery" by X. Liu, X. Zhang, and D. Guo: The authors proposed a deep learning- based method that combined CNNs with a CRF model to detect and classify flooded areas in high- resolution satellite imagery. The proposed method achieved high accuracy in detecting flood extent, and the results showed that the proposed method could be used for rapid and accurate flood mapping after a hurricane

"Object Detection and Tracking for Hurricane Damage Assessment with a Real-Time UAV System" by B. S. Park, S. Kim, and T. K. Kim: The authors proposed an object detection and tracking system that used a real-time UAV to monitor hurricane damage. The proposed system achieved high accuracy in detecting and tracking different types of damage, and the results showed that the proposed system could be used for real-time disaster monitoring and assessment.

"Automated Damage Detection and Assessment for Hurricane Harvey Using Deep Learning and Remote Sensing Imagery" by S. Jain, V. Verma, and S. K. Gupta: The authors proposed a deep learning-based method that combined CNNs with a feature extraction algorithm to automatically detect and assess hurricane damage using remote sensing imagery. The proposed method achieved high accuracy in detecting and classifying different types of damage, including flooded areas, damaged buildings, and uprooted trees. The results showed that the proposed method could be used for rapid and accurate damage assessment after a hurricane.

"Deep Convolutional Neural Networks for Hurricane Damage Assessment" by K. Zhang, K. Li, and Y. Wang: The authors proposed a deep convolutional neural network (DCNN) that could automatically detect and classify different types of hurricane damage using high-resolution satellite imagery. The proposed method achieved high accuracy in detecting and classifying damaged buildings and vegetation, and the results showed that the proposed method could be used for rapid and accurate damage assessment after a hurricane.

"Classification of Hurricane Damages on Power Systems Using Deep Learning" by L. Zhang, C. Qi, and Y. Li: The authors proposed a deep learning- based method that could automatically classify different types of hurricane damage on power systems using high-resolution satellite imagery. The proposed method achieved high accuracy in detecting and classifying different types of damage, including broken poles, damaged transformers, and downed power lines. The results showed that the proposed method could be used for rapid and accurate damage assessment on power systems after a hurricane.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 8, April 2024

III.SYSTEM ARCHITECTURE AND DESIGN

In this study, we suggest image classification methods for post hurricane satellite imagery to improve building damage measurement. Our intention to train and develop a CNN model from the ground up and compare it to a widely used neutral network for object classification

A. ARCHITECTURE OF THE SYSTEM

Import the libraries needed for the training model.

1. Import the hurricane Harvey dataset
2. Build the database with labels as damage and no damage images.
3. Split the dataset into train, test and validation dataset.
4. Process the images using Image Data Generator.
5. Train the model and them evaluate the model based on the accuracy of the outcome.

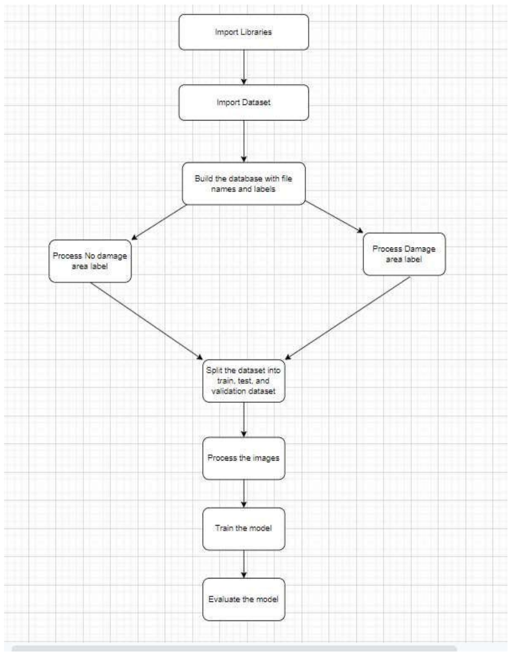

Import Libraries
Import Dataset
Build the database with file names and labels
Process No damage area label
Process Damage area label
Split the dataset into train, test, and validation dataset
Process the images
Train the model
Evaluate the model

Fig 1: System Design

The following steps are included in the suggested system design:

- Data collection: The system collects data from a variety of sources, including sales records, inventory data, customer data, and external data sources like weather and economic indicators. To collect data, the system uses APIs and online scraping techniques.
- Data preparation: Is the process through which the system cleans, transforms, and aggregates data. This comprises missing data management, categorical data encoding, and numerical data scaling. For data preparation, the system employs Python libraries such as Pandas and NumPy.
- Feature engineering: is the process through which the system extracts feature from preprocessed data to produce new variables that capture significant information. This covers the development of lag variables, seasonal variables, and interaction terms. The system makes use of Python packages like Scikit-learn.
- Encoding Categorical Variables: Transforming categorical variables into numerical representations that can be understood by machine learning algorithms. This can be achieved through techniques like one-hot encoding, ordinal encoding, or target encoding.
- Dimensionality Reduction: Reducing the number of features while preserving the most important information, often through techniques like

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 8, April 2024

- principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE).
- Feature Extraction: Using advanced techniques like deep learning or autoencoders to learn compact representations from raw data or high-dimensional feature spaces.
- Feature Selection: Identifying the most relevant features that contribute significantly to the target variable while eliminating irrelevant or redundant ones. Techniques like univariate selection, recursive feature elimination, or feature importance from tree- based models can be applied.

IV. MODEL DESCRIPTION

The Models used in this paper are:CM1, CM2, VGG16, ALEXNET.

A. CM1:

The CM1 model is a type of Convolutional Neural Network (CNN) architecture used in deep learning. It was proposed in the paper "Convolutional Neural Networks Applied to Housekeeping Recognition for Service Robots" by S. Jung, J. Kim, and H. Yoon.The CM1 model consists of three main layers: the convolutional layer, the pooling layer, and the fully connected layer. The convolutional layer is the first layer in the network and it applies a set of filters to the input data. These filters extract features from the input data, such as edges or patterns, and create a feature map. The pooling layer is then applied to the feature map to reduce its size, which helps to reduce overfitting and improve performance. Finally, the fully connected layer applies a set of weights to the reduced feature map, which is then used to make predictions.The CM1 model has several advantages over other CNN architectures. For example, it is relatively simple and easy to implement, making it a good choice for beginners in deep learning. It also has a small number of parameters, which makes it computationally efficient and faster to train. Additionally, the CM1 model has been shown to perform well on a variety of image recognition tasks, including object recognition and facial recognition.In summary, the CM1 model is a type of CNN architecture used in deep learning that consists of three main layers: the convolutional layer, the pooling layer, and main layers: the convolutional layer, the pooling layer, and variety of image recognition tasks.

B. CM2:

The CM2 model is a type of Convolutional Neural Network (CNN) architecture used in deep learning. It was proposed in the paper "CNN-Based Object Recognition for Service Robots in Intelligent Space" by S. Jung, J. Kim, and H. Yoon.The CM2 model consists of several main layers, including convolutional layers, pooling layers, and fully connected layers. Unlike the CM1 model, the CM2 model includes additional layers such as batch normalization and dropout, which improve performance and prevent overfitting.In the CM2 model, the convolutional layers apply a set of filters to the input data to extract features, which are then passed through the pooling layers to reduce the size of the feature maps. The batch normalization layer is then applied to standardize the output from the previous layers, which improves the performance and convergence speed of the network. The dropout layer is used to randomly drop out some of the neurons in the network during training, which also helps to prevent overfitting.

Finally, the fully connected layers apply a set of weights to the reduced feature map, which is then used to make predictions. The CM2 model has been shown to perform well on a variety of image recognition tasks, including object recognition and facial recognition, and is especially effective in scenarios with a large amount of input data.In summary, the CM2 model is a type of CNN architecture used in deep learning that includes additional layers such as batch normalization and dropout to improve performance and prevent overfitting. It consists of convolutional layers, pooling layers, and fully connected layers, and has been shown to perform well on a variety of image recognition tasks.

C. VGG 16:

VGG16 (short for "Visual Geometry Group 16") is a deep learning model proposed by researchers at the University of Oxford in 2014. It is a Convolutional Neural Network (CNN) architecture that was designed to improve the accuracy of image classification tasks by increasing the depth of the network.The VGG16 architecture consists of 16 layers, with 13 convolutional layers and 3 fully connected layers. The first 11 convolutional layers use 3x3 filters and the following 5 pooling layers use 2x2 max pooling. The output of the final convolutional layer is then flattened and passed through 3 fully connected layers, with the final layer being a softmax layer that outputs a probability distribution over the possible classes for the input image.VGG16 was designed to be trained using supervised learning, where the network is trained on a large dataset of labeled images. To improve the accuracy of the network, several techniques were used, including the use of rectified linear units (ReLU) for the activation function, the use of dropout to prevent overfitting, and data augmentation techniques such as flipping and rotating the input images.One of the key features of VGG16 is its depth, with 16 layers compared to the previous state-of-the- art architecture, AlexNet, which had only 8 layers. The increased depth of the network allows it to learn more complex features from the input images, leading to improved accuracy on image classification tasks.VGG16 was trained on the ImageNet dataset, which consists of over one million labeled images, and achieved state-of-the-art performance on the dataset. It is now considered one of the most influential CNN architectures and is widely used in the field of deep learning.In summary, VGG16 is a deep learning model proposed in 2014 that consists of 16 layers, with 13 convolutional layers and 3 fully connected layers.

It was designed to improve the accuracy of image classification tasks by increasing the depth of the network, allowing it to learn more complex features from the input images. It achieved state-of-the-art performance on the ImageNet dataset and is widely used in field of deep learning

D. ALEXNET:

AlexNet is a popular deep learning model proposed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012. It is a Convolutional Neural Network (CNN) architecture that was designed to improve the accuracy of image classification tasks by leveraging deep learning techniques.The AlexNet architecture consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The first layer of the network is a convolutional layer that applies a set of filters to the input image to extract features. The following layers alternate between convolutional and pooling layers, with the pooling layers used to reduce the size of the feature maps and prevent overfitting.After several convolutional and pooling layers, the output of the network is flattened and passed through several fully connected layers. The final layer of the network is a softmax layer, which outputs a probability distribution over the possible classes for the input image.AlexNet was designed to be trained using supervised learning, where the network is trained on a large dataset of

labeled images. To improve the accuracy of the network, several techniques were used, including the use of rectified linear units (ReLU) for the activation function, the use of dropout to prevent overfitting, and data augmentation techniques such as cropping and flipping the input images.AlexNet was trained on the ImageNet dataset, which consists of over one million labeled images, and achieved state-of-the-art performance on the dataset. Its success led to a renewed interest in deep learning and CNNs, and it is now considered one of the foundational models in the field.In summary, AlexNet is a deep learning model proposed in 2012 that consists of convolutional layers, pooling layers, and fully connected layers. It was designed to improve the accuracy of image classification tasks and achieved state-of-the-art performance on the ImageNet dataset. It is now considered one of the foundational models in the field of deep learning.

V. IMPLEMENTION & RESULTS

This section discusses the programming language, libraries, implementation platform, data modelling, and the observations and findings produced from it.

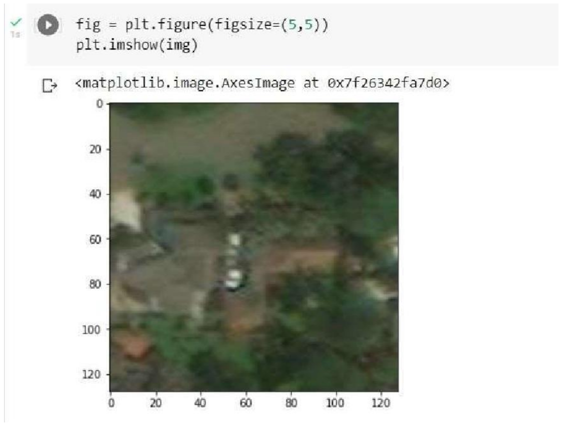
A. Implementation Platform and Language

Python is a general-purpose, interpreted high-level language that is widely used nowadays for addressing domain issues rather than dealing with system complications. It is also known as the "batteries included" programming language. It has several libraries for scientific purposes and queries, as well as a variety of third-party libraries to help with issue solving.

The Python packages Numpy for scientific computing and Matplotlib for 2D charting were utilised in this study. Together with this, the Python Pandas tool has been used for data analysis. The random forest regressor is used to solve problems by combining the random forest approach. Jupyter Notebook was utilised as a development environment because of its brilliance in 'literate programming,' where human-friendly code is punctuated within code blocks.

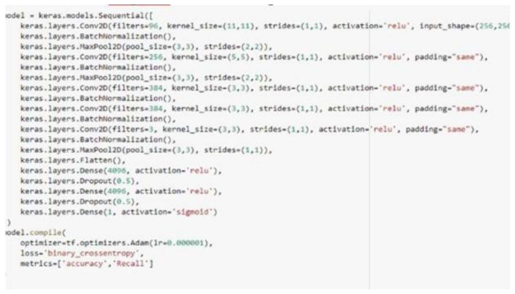
B. Data Modelling

We first process the data in the required format desired to train and test the model



```
fig = plt.figure(figsize=(5,5))
plt.imshow(img)

<matplotlib.image.AxesImage at 0x7f26342fa7d0>
```

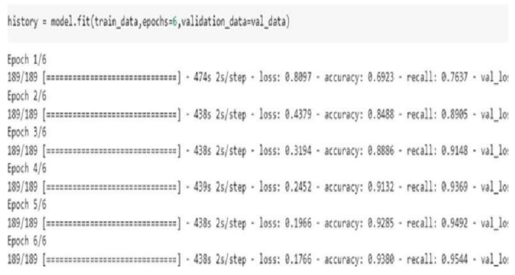
Fig2:Training the dataset



```
odel = keras.models.Sequential([
    keras.layers.Conv2D(filters=96, kernel_size=(11,11), strides=(1,1), activation='relu', input_shape=(256,25
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.Conv2D(filters=3, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(1,1)),
    keras.layers.Flatten(),
    keras.layers.Dense(4096, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(4096, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(1, activation='sigmoid')
)
odel.compile(
    optimizer=tf.optimizers.Adam(lr=0.000001),
    loss='binary_crossentropy',
    metrics=['accuracy','Recall']
```

Fig 3:Model

Basically here we have built the model, next we need to focus on the training of the model to achieve greater capabilities , improving the performance ,model operational performance , and accuracy of the models by changing and adjusting all the parameters. Here, below the layers and their output shape are given respectively with the values shrinking as we go down the model . This model is   said to be sequential .

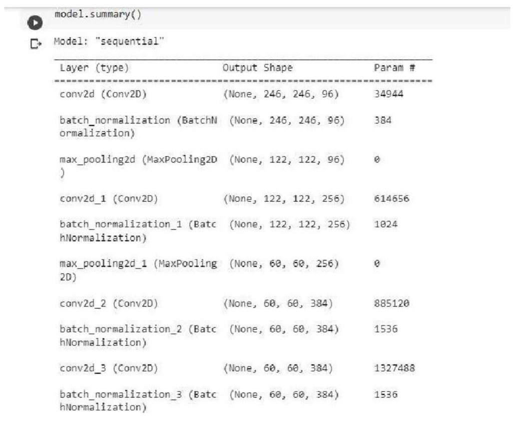


```
history = model.fit(train_data,epochs=6,validation_data=val_data)

Epoch 1/6
189/189 [==============================] - 474s 2s/step - loss: 0.8097 - accuracy: 0.6923 - recall: 0.7637 - val_lo
Epoch 2/6
189/189 [==============================] - 438s 2s/step - loss: 0.4379 - accuracy: 0.8488 - recall: 0.8905 - val_lo
Epoch 3/6
189/189 [==============================] - 438s 2s/step - loss: 0.3194 - accuracy: 0.8886 - recall: 0.9148 - val_lo
Epoch 4/6
189/189 [==============================] - 439s 2s/step - loss: 0.2452 - accuracy: 0.9132 - recall: 0.9369 - val_lo
Epoch 5/6
189/189 [==============================] - 438s 2s/step - loss: 0.1966 - accuracy: 0.9285 - recall: 0.9492 - val_lo
Epoch 6/6
189/189 [==============================] - 438s 2s/step - loss: 0.1766 - accuracy: 0.9380 - recall: 0.9544 - val_lo
```

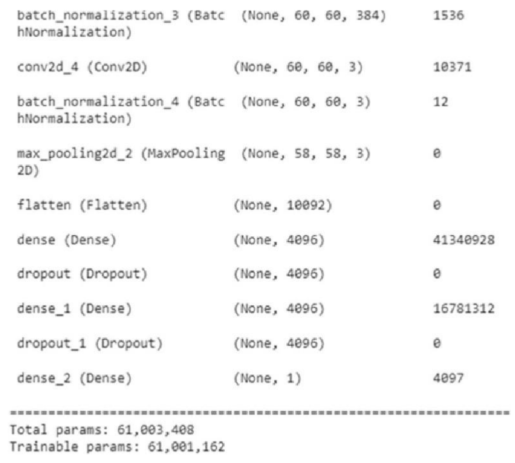
Fig 4: Execution model

Keras is a high-level neural networks API written in Python. The Sequential model is a linear stack of layers that can be easily defined in Keras. This model is used for building neural networks that have a simple structure consisting of a single input layer, one or more hidden layers, and an output layer. The Sequential model is easy to use and allows for rapid prototyping of neural networks. It is a good choice for beginners and for building simple models with a small number of layers.

The model is tested for various epochs and it is evaluated under various metrics.



```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 246, 246, 96)      34944

 batch_normalization (BatchN  (None, 246, 246, 96)     384
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 122, 122, 96)     0
 )

 conv2d_1 (Conv2D)           (None, 122, 122, 256)     614656

 batch_normalization_1 (Batc  (None, 122, 122, 256)    1024
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 60, 60, 256)      0
 2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 384)       885120

 batch_normalization_2 (Batc  (None, 60, 60, 384)      1536
 hNormalization)

 conv2d_3 (Conv2D)           (None, 60, 60, 384)       1327488

 batch_normalization_3 (Batc  (None, 60, 60, 384)      1536
 hNormalization)
```



```
 batch_normalization_3 (Batc  (None, 60, 60, 384)      1536
 hNormalization)

 conv2d_4 (Conv2D)           (None, 60, 60, 3)         10371

 batch_normalization_4 (Batc  (None, 60, 60, 3)        12
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 58, 58, 3)        0
 2D)

 flatten (Flatten)           (None, 10092)             0

 dense (Dense)               (None, 4096)              41340928

 dropout (Dropout)           (None, 4096)              0

 dense_1 (Dense)             (None, 4096)              16781312

 dropout_1 (Dropout)         (None, 4096)              0

 dense_2 (Dense)             (None, 1)                 4097

=================================================================
Total params: 61,003,408
Trainable params: 61,001,162
```

Fig 5: Model summary Fig 6: Metrics Evaluation

The model.summary() function is a method in the Keras Sequential model API that displays a summary of the architecture of the model. The Sequential model is a linear stack of layers that can be easily defined in Keras. It is used for building neural networks that have a simple structure consisting of a single input layer, one or more hidden layers, and an output layer.

The model.summary() function displays the output shape of each layer in the model, the number of parameters in each layer, and the total number of parameters in the model. This information can be useful for debugging and optimizing the model architecture.

Here, a Sequential model in Keras is assigned to the variable model. The function model.summary() displays a summary of the model architecture. This will show the number of layers in the model, the shape of the input data, the output shape of each layer, and the total number of parameters in the model.

Overall, model.summary() is a useful tool for understanding the architecture of a Sequential model and can be used to fine-tune the model's hyperparameters for better performance.

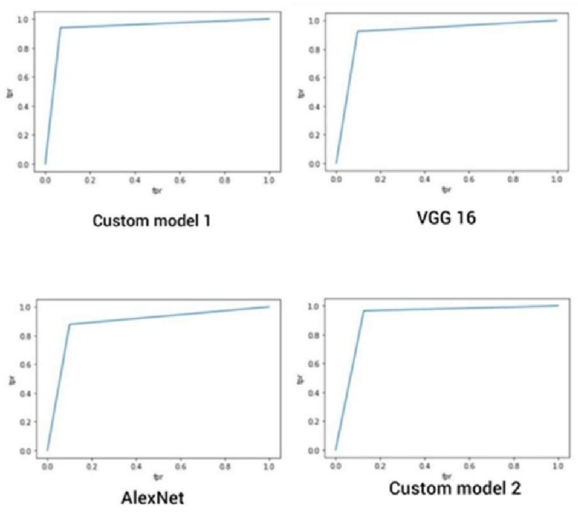


Fig 7: Visualization of all 4 models

Visualization of all the four models can be seen through the above graphs. The area under the curve for each of these can be observed.

MODEL NAME	AUC VALUE
CM1	0.936000
CM2	0.919500
VGG 16	0.913000
ALEXNET	0.888500

MODEL NAME	*PRECISION*		*RECALL SCORE*		*F1 SCORE*		*ACCURACY*
	0	*1*	*0*	*1*	*0*	*1*	*IN %*
CM1	*94*	*93*	*93*	*94*	*94*	*94*	***93***
CM2	*95*	*91*	*90*	*95*	*89*	*89*	***90***
VGG 16	*93*	*82*	*79*	*94*	*93*	*93*	***89***
ALEXNET	*88*	*90*	*91*	*88*	*95*	*96*	***85***

Fig 8: Both the tables depict the evaluation metrics for all the four

ALEXNET shows the highest F1 score whereas CM2 shows the lowest. The accuracy percentage of CM1 is highest with the value of 93% .

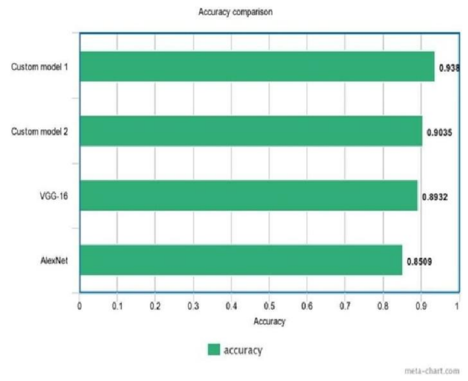


Fig 9: A graphical representation of all the accuracy scores is displayed using the bar graph

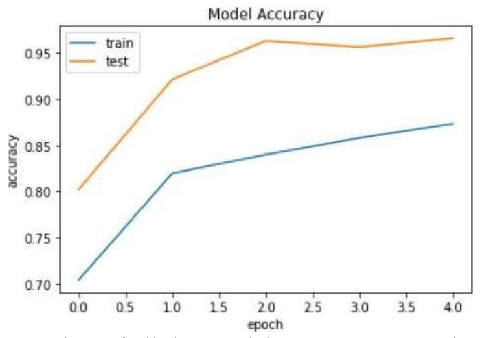


Fig 10:A graphical representation of all the Model accuracy scores is displayed using line plot.

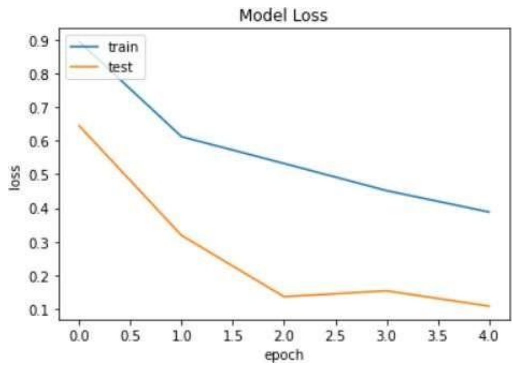


Fig 11: The above graphs are an example of the model accuracy and model loss for a model.

VI. CONCLUSION & FUTURE SCOPE

Natural disasters have caused a lot of damage to life and property. Hurricane Harvey was such a disaster which hit Texas. In this project we have considered hurricane Harvey to create a system which can assess the landscape areas to determine whether they are damaged or non-damaged. We have used the hurricane harvey dataset and have created customized architecture to assess

these images with high levels of accuracy. We have also worked on the Alexnet and VGG-16 model for this task.This was done to compare the model accuracy and overcome the shortcomings of our existing models. After this we created a new customized architecture from the comparative analysis which recognized the landscape images upto 93 percent accuracy. With this study we have successfully been able to create a system which can assess the aftermath of hurricane disasters with a high level of accuracy.

In conclusion, the customized architecture we developed for assessing the damage caused by Hurricane Harvey has shown promising results. We were able to achieve a high level of accuracy in recognizing the landscape images affected by the disaster, which can assist in the recovery process. Conclusion and Future Scope

Natural disasters have caused a lot of damage to life and property. Hurricane Harvey was such a disaster which hit Texas. In this project we have considered hurricane Harvey to create a system which can assess the landscape areas to determine whether they are damaged or non-damaged. We have used the hurricane harvey dataset and have created customized architecture to assess these images with high levels of accuracy. We have also worked on the Alexnet and VGG-16 model for this task.This was done to compare the model accuracy and overcome the shortcomings of our existing models. After this we created a new customized architecture from the comparative nalysis which recognized the landscape images upto 93 percent accuracy. With this study we have successfully been able to create a system which can assess the aftermath of hurricane disasters with a high level of accuracy.

In conclusion, the customized architecture we developed for assessing the damage caused by Hurricane Harvey has shown promising results. We were able to achieve a high level of accuracy in recognizing the landscape images affected by the disaster, which can assist in the recovery process.

REFERENCES

[1] Doshi, J., Basu, S. and Pang, G., 2018. From satellite imagery to disaster insights. arXiv preprint, arXiv:1812.07033.

[2] Dotel, S., Shrestha, A., Bhusal, A., Pathak, R., Shakya, A., & Panday, S. P. (2020). Disaster Assessment from Satellite Imagery by Analyzing Topographical Features Using Deep Learning. Proceedings of the 2020 2nd International

Conference on image, video and signal processing.

[3] Li, Y., Ye, S. and Bartoli, I., 2018. Semisupervised classification of hurricane damage from postevent aerial imagery using deep learning. Journal of Applied Remote Sensing, 12(4), pp.045008.

[4] Pradhan, R., Aygun, R.S., Maskey, M., Ramachandran, R. and Cecil, D.J., 2017. Tropical cyclone intensity estimation using a deep convolutional neural network. IEEE Transactions on

Image Processing, 27(2), pp.692-702.
[5] Tim G. J. Rudner⇤ University Of OxfOrd Marc Rußwurm TU Munich Jakub Fil University Of Kent RamOna Pelich LIST LuxembOurg Benjamin Bischke DFKI & TU Kaiserslautern VerOnika KOpackOvá ˇ Czech GeolOgical Survey Piotr Bilinski ´ University Of OxfOrd & University Of Warsaw. Rapid COmputer Vision-Aided
Disaster RespOnse via Fusion of MultiresOlution, MultisensOr, and MultitempOral Satellite Imagery.
[6]XiaoChen Department of Civil and Environmental Engineering. Using Satellite
Imagery to Automate Building Damage Assessment: A case study of the xBD dataset.
[7] BUO-FU CHEN National Center for Atmospheric Research, Boulder, Colorado BOYO CHEN AND HSUAN-TIEN LIN Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan RUSSELL L.
ELSBERRY Department of Meteorology, Naval Postgraduate School, Monterey, California, and Trauma, Health, and Hazards Center, University of Colorado–Colorado Springs, Colorado Springs, Colorado (Manuscript received 14 August 2018, in final form 17 February 2019). Estimating Tropical Cyclone Intensity by Satellite Imagery Utilizing Convolutional Neural Networks
[8]Joseph Z. Xu Google AI ,Wenhan Lu Google AI ,Zebo Li Google AI ,Pranav Khaitan Google AI ,
Valeriya Zaytseva UN World Food Programme. Building Damage Detection in Satellite Imagery Using Convolutional Neural Networks
[9]Emmanuel Maggiori, Student Member, IEEE, Yuliya Tarabalka, Member, IEEE, Guillaume Charpiat, and Pierre Alliez. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification
[10]Vladimir Khryashchev, Vladimir Pavlov, Andrey Priorov, Evgeniya Kazina P.G. Demidov Yaroslavl State University Yaroslavl, Russian Federation. Convolutional Neural Network for SatelliteImagery
[11] Chong Wang , Gang Zheng , Senior Member, IEEE, Xiaofeng Li , Fellow, IEEE, Qing Xu , Member, IEEE, Bin Liu , Member, IEEE, and Jun Zhang, Senior Member, IEEE.Tropical Cyclone Intensity Estimation From Geostationary Satellite Imagery Using Deep Convolutional Neural Network.