Prediction of Post-Hurricane Damage Status Using Deep Learning Algorithms

Jamie Atiyah
Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, UK

***Abstract*—Assessing the damage status of houses post-hurricane can allow for quicker emergency aid and response. In this paper, an automated classification system to detect the damage status of houses is built by training transfer learning models on satellite imagery. Two models, VGG16 and ResNet50, were trained on 10,000 images of houses after hurricane Harvey with 5000 images consisting of houses that had sustained damage, and 5000 images consisting of houses that had not. The models were analysed on performance for varying learning rates, optimisers and epochs. Coupled with model enhancement techniques early stopping and learning rate schedulers, the VGG16 model with a learning rate of 0.001, stochastic gradient descent optimiser, and 76 epochs was the best performing model, with an accuracy of 80.85% and F_1 score of 0.8341.**

***Index Terms*—Hurricane Harvey, Post-Disaster, Damage Assessment, deep learning.**

I. Introduction

The natural phenomenon of hurricane occur across the world every year. They are immensely powerful and destructive with very strong, sustained winds of at least 74 m.p.h that can tower at five to six miles high [1]. Hurricanes cause great damage to infrastructure and properties, whilst also resulting in the loss of life. Out of all the different weather disaster types, hurricanes cause the most damage, both in terms of the number of deaths and the monetary cost to repair damaged infrastructure [2]. The damage of hurricanes is exasperated for low-income and minority communities as buildings within these communities are more susceptible to being damaged [3]. The ever growing thread of climate change will result in hurricanes being more destructive, by increasing the intensity and decreasing the speed at which they travel [4].

The use of Deep Learning models, such as Convolutional Neural Networks (CNNs), have shown promising performances in image classification based problems relating to hurricanes. In particular, classification systems can detect whether a property has sustained damage. The automatic and rapid nature of such classification systems provides enormous benefit to emergency managers, providing an alternative to previously employed methods of assessing damage field surveys and damage reports [5] which are highly manual and measured processes.

The aim of this research is to build an automated classification system to detect the damage status of houses by analysing satellite images using Deep Learning. The objective of this research is attained using the satellite images from Texas after Hurricane Harvey which has images of houses which have sustained damage as a result of a hurricane and houses which have not sustained any damage. The models used within this research were identified and selected with justification and critical appraisal from relevant literature. In particular, the models used for this research are the VGG-16 [6] and ResNet50 [7]. The models were trained on the data set with their performances being evaluated through the use of evaluation metrics.

II. Background

Researchers have proposed the use of Machine learning algorithms to detect damage sustained after natural disasters. Whilst promising results have been found, manual feature selection and engineering is required [8] [9] [10]. In comparison, Deep Learning models are able to extract the features automatically from the data itself. In this project, transfer learning architectures are used for damage detection. Transfer learning employs pre-trained models on classifying images for this project's context. This removes the need to develop a network from scratch which demands significant computational resources and is also time intensive.

A. Convolutional Neural Networks

Multilayer perceptrons networks (MLP) consist of multiple layers of fully connected neurons. Whilst traditional MLP models can achieve high accuracy in classifying images, these results are dependent on the manual feature extraction [11]. Furthermore, the high dimensional inputs are required to be flattened into a vector which can lead to MLP's neglecting localised information [12]. In comparison, CNN models require less computational time [13] [14], and are also easily parallelized across GPU cores as the mathematical operation of convolution is easy to parallelize [15]. CNN models are able to process images consisting of three channels: red, green, and blue. CNNs are also able to effectively use the translation invariance of images, enabling CNNs to be able to identify objects within images located at different positions and orientations. Localised information is captured by CNNs by applying a kernel to localised regions of images which leads to higher classification accuracy [16] [17]. Convolutional layers take two parameters of the input and a kernel and uses convolution to create a highly local output. Kernels learn from the data. Convolutions are able to perform tasks such as detecting edges and lines, blurring images, and sharpening them.

1) Padding: Multiple layers of convolutions decrease the dimensions of the original image, potentially leading to the removal of the outer boundaries of the image and any information that exists at the image boundaries. To prevent this, padding is used which adds extra pixels around the image boundary. In the 2-d space, where n denotes the input shape, k the kernel shape, h the rows, w the columns and p the padding, the addition of padding changes the output dimension from 1 to 2.

$$(n_h - k_h + 1) \times (n_w - k_w + 1) \tag{1}$$

$$(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1) \tag{2}$$

2) Stride: To decrease dimensions of the input, for the purposes of decreasing the required computation or for subsampling, stride can be implemented. Stride changes the steps convolutional windows traverse per slide. In the 2-D space, adding stride changes the output dimension from 2 to 3.

$$\frac{(n_h - k_h + p_h + 1 + s_h)}{s_h} \times \frac{(n_w - k_w + p_w + 1 + s_w)}{s_w} \tag{3}$$

3) Pooling: Pooling operations are used in CNNs to prevent overfitting, computation time and accuracy [18]. Pooling aggregates local information within a window to return an output of smaller dimension. The returned value is often either the maximum value within the pooling window (max-pooling) or the average value (average pooling).

4) Activation Functions: The convolutional operation are linear Therefore, the CNN would become one convolutional layer. To make the CNN non-linear, an activation function is used. However, some activation functions can be the cause of the vanishing gradient problem. Therefore, the most common activation function used is the ReLu activation function 4.

$$ReLU(x) = max(x, 0) \tag{4}$$

B. Related Literature

Cheng et al. [19] employed a stacked convolutional neural network architecture where one network is used to distinguish between buildings and none building, and an additional network to assess the damage sustained of the property. As the damage is split into ordinal classes, the square of earth mover's distance loss. The stacked architecture is trained on a video data set which is created using web mining. The video data set comprised of five videos, each with a duration of 301 seconds. The building classification model was developed using RetinaNet [20]. The model that classifies the damage class was trained through transfer learning on MobileNet [21]. The trained model achieved a 65.6% accuracy in identifying buildings and achieved a 61% classification accuracy in assessing the damage class. Berezina and Liu [22] similarly investigated the use of multi-class classification for damage assessment as an alternative to binary classification through the use of a coupled deep learning model. The model was learned on 22686 labeled satellite images relating to Hurricane Michael in 2018. The coupled model uses a U-Net [23] model coupled with a ResNet50 model [7]. The coupled CNN model achieved a 86.3% accuracy.

Kaur et al. [24] tested the capability of a variety of transfer learning modes, namely VGG16 [6], MobileNetV2 [25], InceptionV3 [26], and DenseNet121 [27]. The models were trained and tested on 23,000 Hurricane Harvey satellite images. The VGG16 model obtained the best results, with an accuracy of 75%. An increase in accuracy to 78 % for the VGG model was found by changing the optimiser to the RMSprop optimiser [28]. Bao and Emedom-Nnamdi [29] also assessed the performance of CNN models for classifying post-hurricane satellite imagery for varying optimisers. The models used with varying parameters were trained on 5000 images of damaged and 5000 images of undamaged buildings. They found that the stochastic gradient descent led to better model performance.

Reducing the effects of hurricane's is not limited to post event analysis. Hurricane tracking can be of vital importance in providing an early warning to regions expected to be impacted by an approaching hurricane. Alshaye et al. [30] use 60500 images generated using 100 Global Navigation Satellite Systems Reflectometry (GNSS-R) receivers and 21 GNSS satellites. The images are interpolated using the Nearest Neighbor interpolation algorithm and trained on a CNN. The CNN model achieves 96.5% accuracy in predicting the coordinates of a hurricane.

There are various sources where relevant images can be sourced from. Satellite images [22] [29] [24], videos sourced from unmanned UAVs [19], and images shared on social media [31] [32] have all been used to train deep learning models to predict damage incurred from a hurricane. One benefit of opting to use satellite imagery is the large spatial area covered. All three sources are real-time, allowing for a prompt response. Using social media as a primary source has its negatives with digital exclusion being unevenly split upon demographics [33], potentially making the feed of shared images on social media biased against digitally excluded demographics.

C. Selected Model I: VGG-16

The first model selected is the visual geometry group network (VGGNet) [6]. VGG architectures utilize VGG blocks. Each block consists of convolutions and pooling layers. The convolutions use a 3×3 kernel and the pooling type is max pooling. Different configurations of the VGG architecture refer to the depth of the network, with the commonly used VGG-16 having 16 layers. Figure 1 and 2 show the architectures of VGG-16 and VGG-19 respectively. VGG-16 architecture consists of 13 convolutional layers, with 5 pooling layers and then 3 fully-connected layers. The VGG-16 model has 134,277,186 trainable parameters whilst the VGG-19 model has 139,589,442 trainable parameters. The model used in this project is the VGG-16 model due to the lower amount of trainable parameters which helps reduce the computational cost in comparison to the VGG-19 model. Furthermore, the

VGG-16 model has been shown to provide highly accurate results in damage assessment after a hurricane [24].

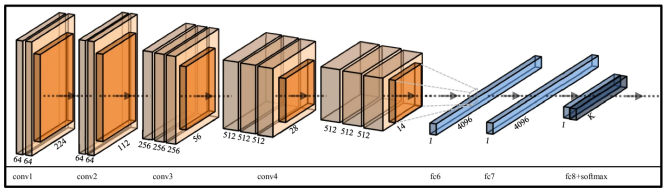

Fig. 1. VGG-16 Architecture [34]

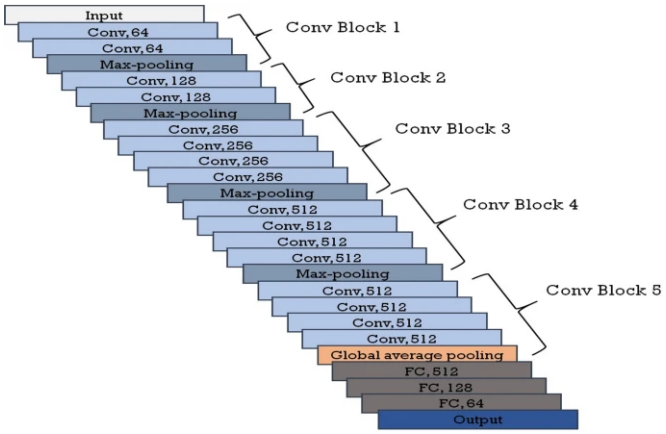

Fig. 2. VGG-19 Architecture [35]

D. Selected Model II: ResNet50

The second model selected is residual network (ResNet) [7]. The ResNet architecture utilises nested function layers. The network is made up of residual blocks that are stacked on top of each other. Residual connections allow the model to overcome vanishing gradients, allowing the model to be deeper. These residual connections allow for the model's performance to not be hindered when adding additional layers as none useful layers will be skipped over. The ResNet architecture is suitable for the classification task for this project as the residual connections of the network allow for a deep structure which will help provide accurate results whilst avoiding computational expense. The different configurations of the ResNet models refer to the depth. ResNet50 consists of 50 layers. Within ResNet50 there are 48 convolutional layers and 2 pooling layers: one max pooling layer and one average pooling layer. The architecture of ResNet50 is displayed in figure 3.

III. Data Set Description and Analysis

The data set used for this project is publicly accessible from Kaggle. The data set consists of images of houses which have sustained damage as a result of a hurricane Harvey and houses which have not sustained any damage [37]. In total, there are 14000 images which are separated into train, test and validation folders. There are two classes: "damage" and "no_damage". The train folders consists of 10000 images,

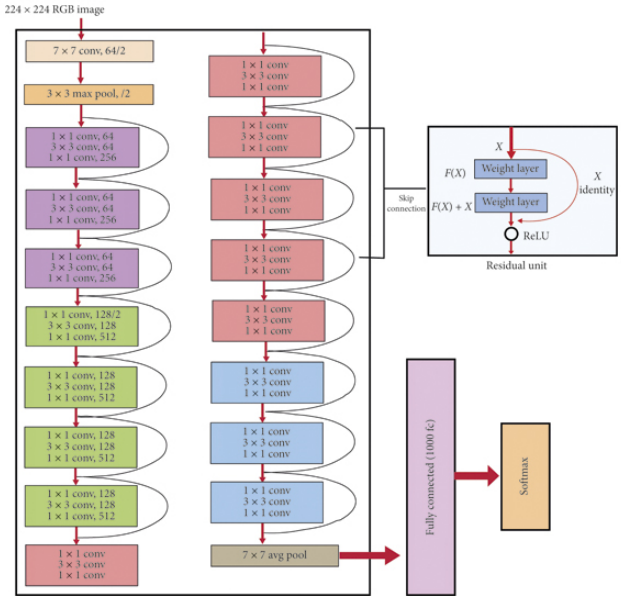

Fig. 3. ResNet50 Architecture [36]

with 5000 of those images being classed as "damaged" and 5000 images are classed as "no_damage". Test and validation folders both consist of 2000 images each, with both having 1000 images classed as "damage" and 1000 images classed as "no_damage". Validation and test data sets allow for the generalisation of the models to be tested. The distribution of classes for the train, validation and test data sets are shown in figure 4. Figure 5 shows a sample of images labelled by their respective class.

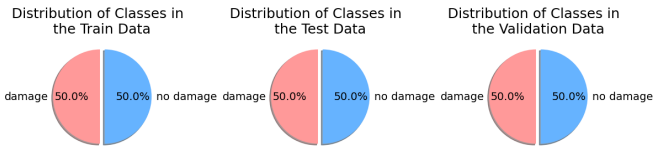

Fig. 4. Class Distributions

A. Image Augmentation

Image augmentation techniques such as scaling, cropping, horizontally flipping and standardisation can improve the accuracy of deep learning models [38]. In this project, the images in the train and validation were augmented by the following. Images were scaled to a square of 40 pixels in both height and width as high resolution images lead to an overload of computation [39]. The images are then randomly scaled from 64% to 100% the size and are cropped to a 32×32 pixel image. Randomly cropping images allows the model to obtain the best learning efficiency and increase model stability [40]. Images are randomly flipped horizontally and are transformed to a tensor. Randomly flipping images creates noise within the data and helps the model overcome the problem of overfitting

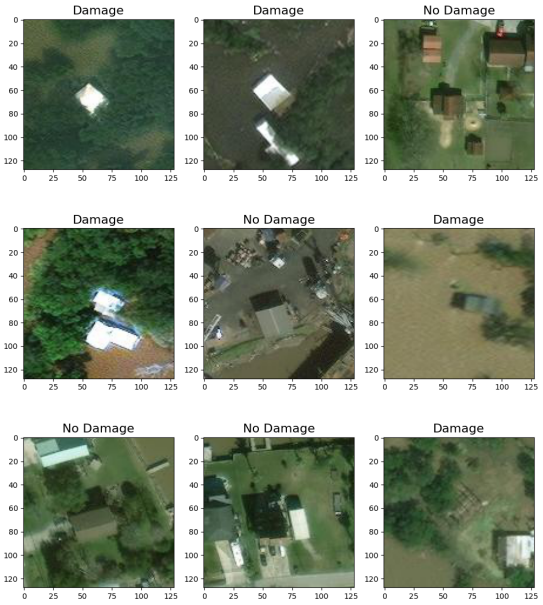

Fig. 5. Damaged and Undamaged Properties

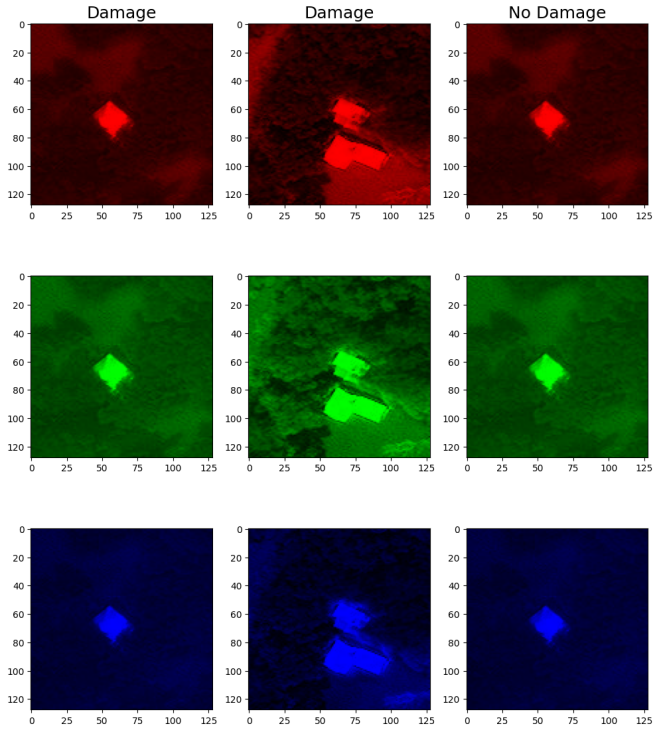

Fig. 6. Damaged and Undamaged Properties Split over RGB channels

[41]. Finally, as images are made up of three RGB channels, as shown in figure 6, the images are normalised around the mean and standard deviation for each RGB channel. Images within the test data were transformed to a tensor and normalised around the mean and standard deviation for each RGB channel.

IV. Experimental Methodology

A transfer learning approach was used for this project. In particular, the two pre-trained Deep Learning models used in this project were ResNet50 and VGG-16. The weights of both networks were trained using ImageNet [42] and then transferred from the vast ImageNet classification problem to the smaller scaled hurricane damage classification problem.

A. Model Parameters

To optimise the performance of the transfer learning models, the hyper-parameters can be fine tuned. Different optimisers can be explored to find the optimal choice that reduces the loss function the best. From the related literature II-B, researchers have had success in predicting post-hurricane damage using the RMSprop and stochastic gradient descent optimisers. These optimisers, along with the Adam optimiser, were tested to find the best optimiser. In addition, the optimal learning rate was explored. The hyper-parameter learning rate, controls how much the model's weights are adjusted when fine-tuning to minimise the loss function. Too small a learning rate vastly increases the computational expense whilst too high a learning rate leads to an increase in the loss. The models were trained multiple times, with the optimiser and learning rate being varied with every iteration. For each combination of optimiser and learning rate, the models were trained up to 5 epochs. A small amount of epochs was utilised when finding the optimiser and learning rate to save on computational expense. The chosen optimiser and learning rate For the models was selected by finding the lowest loss in the validation data set with the criterion set to the cross entropy loss function. For the VGG-16 model, the optimiser that yielded the best performance was the stochastic gradient descent, with a learning rate of 0.0001. When finding the best optimiser and learning rate for the ResNet50 model, a learning rate scheduler with a 1 cycle learning rate policy is employed to set the learning rate across the 6 parameter groups. For the ResNet50 model, the optimiser and learning rate that yielded the best performance was the Adam optimiser with a learning rate of 0.0001.

B. Model Training

Model enhancement techniques such as a learning rate scheduler and early stopping can help performance and computational efficiency. The models were trained on 100 epochs with a learning rate scheduler and early stopping. The learning rate scheduler decreases the learning rate by a specified factor if the validation loss does not increases over a specified number of 'patience' epochs which was set to 5 and the factor for which the learning rate will decrease by was set to 0.5. Early stopping is a regularisation technique which stops the model training when the validation loss has

not improved over a specified number of 'patience' epochs. For this project, the number of 'patience' epochs was set to 5. The hidden layer parameters of the models were frozen so the pre-trained weights can be utilised.

C. Model Evaluation

1) Confusion Matrices & Performance Metrics: Confusion matrices were created. As this is a binary classification problem, the confusion matrix was a 2 x 2 matrix. Table I shows a confusion matrix for a binary classification problem. The predicted values are compared to the ground truth. The values that lie in the upper-left diagonal, TP (true positive) and TN (true negative), are correctly predicted whilst the values FP (false positive) and FN (false negative) are incorrectly predicted.

TABLE I
CONFUSION MATRIX

		PREDICTION	
		Positive	Negative
ACTUAL	Positive	**TP**	FN
	Negative	FP	**TN**

The following model performance metrics can be calculated. The accuracy (5) of the model is calculated by dividing the correct predictions by the total number of predictions. Accuracy is a useful measure of model performance when the predictions are symmetric (i.e. the error rate of the model in predicting both classes is similar). The precision (6) is calculated by dividing the number of true positives against the total number of observations that were predicted as being positive. The sensitivity (7) of the model is calculated by dividing the number of true positives against the total number of positive observations. The specificity (8) is calculated by dividing the number of true negatives against the total number of negative observations. The F_1 (9) score can be calculated from the precision and sensitivity. F_1 scores are a suitable measure of model performance when the distribution of predictions are asymmetrical.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{8}$$

$$F_1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{9}$$

2) ROC & AUC: Receiver operating characteristic (ROC) plot the true positive rate against the false positive rate. Higher values of true positive rate and lower values of false positive rate are desired. The area under the curve (AUC) measures the performance of the model across both classes to be predicted. A higher AUC indicates better predictive output of the model.

V. EVALUATION

A. Accuracy & Loss

The accuracy and losses when predicting the training and validation data sets for the VGG-16 model are displayed in figures 7 and 8. The training was stopped early at epoch 81. The best accuracy for the validation set to be achieved at epoch 81. The loss for the validation set was minimised at epoch 76.

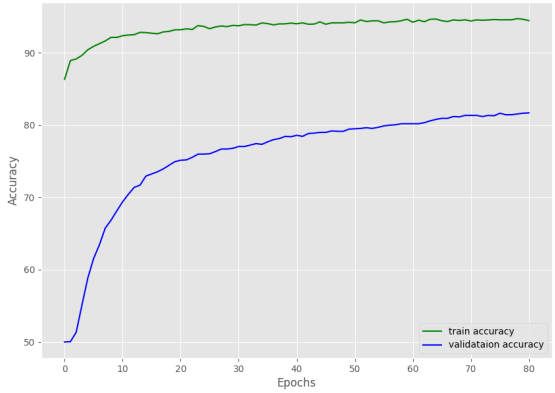

Fig. 7. VGG-16 Accuracy for Train and Validation sets

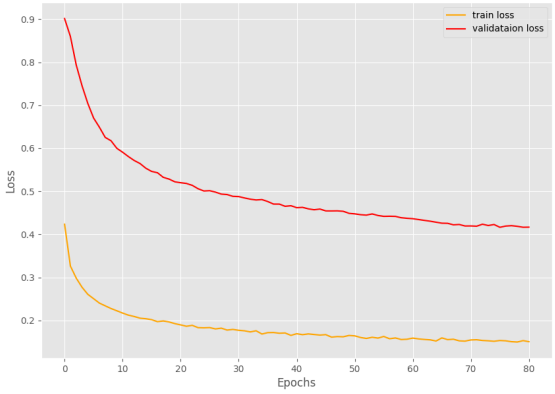

Fig. 8. VGG-16 Losses for Train and Validation sets

The accuracy and losses when predicting the training and validation data sets for the ResNet50 model are displayed in figures 9 and 10. The training of the ResNet model was stopped a lot earlier than the VGG16 model, as the training was stopped at epoch 27. The best accuracy was achieved for the validation data set at epoch 21. The loss was minimised at

epoch 22. Both models performed better on the training data set in comparison to the validation data set.

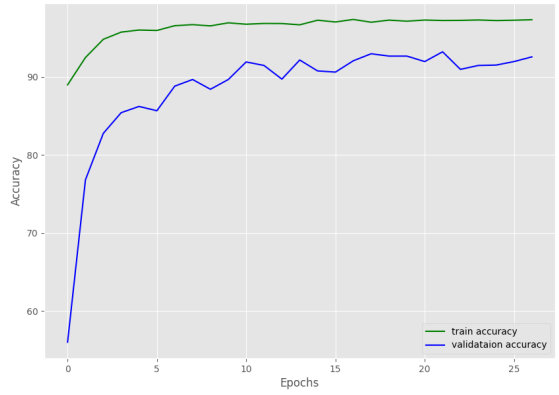

Fig. 9. ResNet50 Accuracy for Train and Validation sets

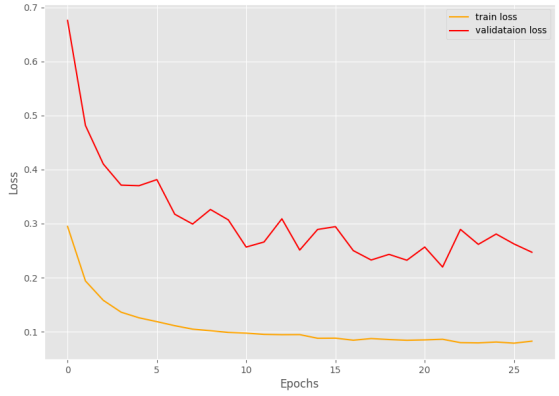

Fig. 10. ResNet50 Losses for Train and Validation sets

B. Confusion Matrices

Confusion matrices for the predictions on the test data set are displayed in figure 11 for the VGG16 model and figure 12 for the ResNet model. Both models are shown to perform highly in predicting the 0 class (damaged properties) with 96% and 97% of the images that are of damaged properties being correctly predicted by the VGG16 and ResNet models respectively. Therefore, the VGG16 model has a higher chance of causing a type II error as more images of damaged houses are being predicted as being undamaged. However, the number of occurrences of type II errors are low, with only 37 and 29 instances for the VGG16 and ResNet50 model respectively. The low number of occurrences of type II errors is encouraging, as damaged houses predicted as undamaged may not be assistance for emergency responders. The VGG16 model is shown to better predict the 1 class (undamaged properties) with 65% of images that are of undamaged properties being correctly predicted by the VGG16 model whilst only 61% are correctly predicted by the ResNet model. Therefore, the type I error for the ResNet model is higher as undamaged houses are being predicted as damaged. The high number of type I errors may lead to the misuse of emergency resources, as undamaged houses that are being predicted as damaged may be assigned nonessential assistance. Non-normalised confusion matrices for the test data are displayed in figure 13 and 14 and can be used to calculate performance metrics.

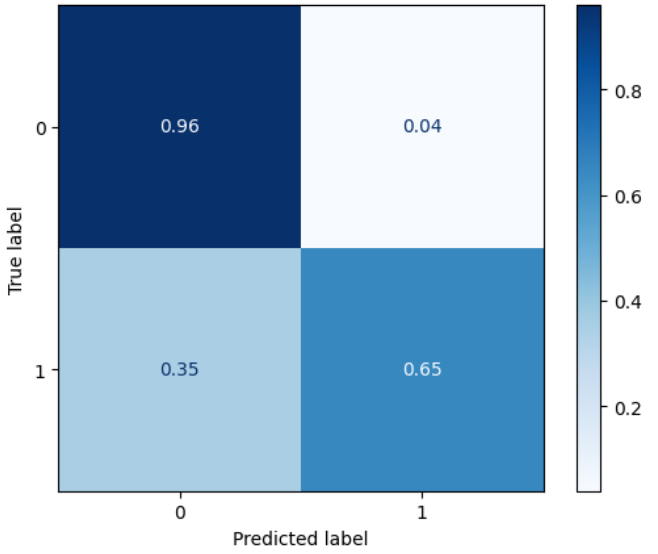

Fig. 11. Normalised Confusion Matrix for the VGG16 model for the test data

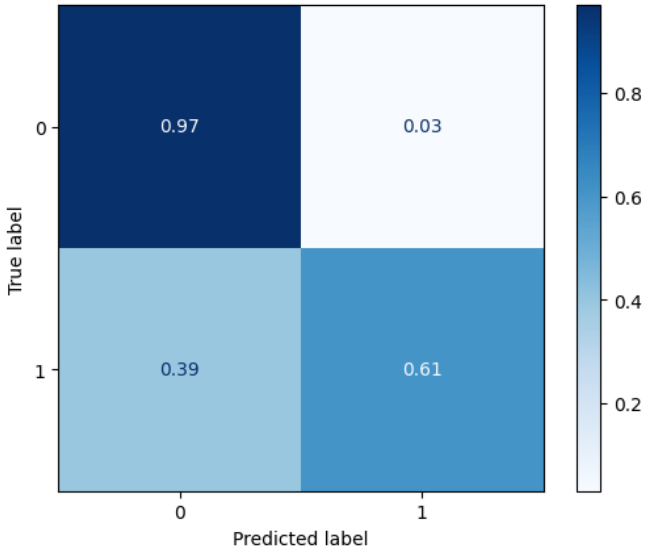

Fig. 12. Normalised Confusion Matrix for the ResNet model for the test data

C. Performance Metrics

Performance metrics are displayed in table II. The VGG16 model outperforms the ResNet50 model on all metrics except

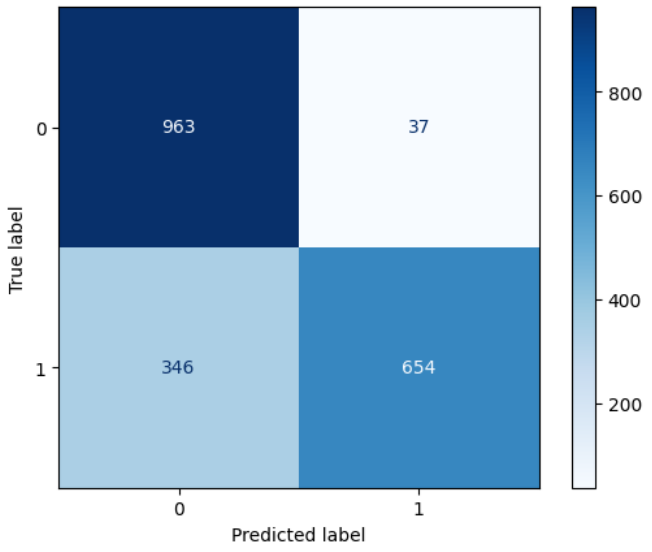

Fig. 13. Non-Normalised Confusion Matrix for the VGG16 model for the test data

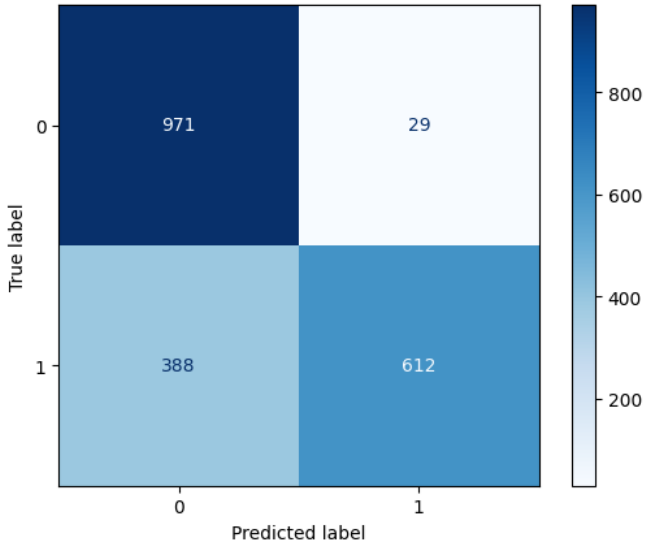

Fig. 14. Non-Normalised Confusion Matrix for the ResNet model for the test data

the sensitivity. The sensitivity metric displays the accuracy of the models in predicting the damaged houses. The F_1 score, which combines the sensitivity and the precision, is higher for the VGG16 model.

TABLE II
PERFORMANCE METRICS OF MODELS ON TEST DATA SET

'Model	Accuracy	Precision	Sensitivity	Specificity	F_1 Score
VGG16	0.8085	0.7357	0.963	0.654	0.8341
ResNet50	0.7915	0.7145	0.971	0.612	0.8203

D. ROC-AUC plots

Figures 15 and 16 contain the ROC-AUC plots on the test data for the VGG16 and ResNet50 models respectively. The micro-average curves represents the averaging of the true and false positive rates across both classes. The macro-averaging curve is the average true and false positive rates per class. The curves in figure 15 are closer together, indicating a consistent performance across the different classification metrics. The areas under the curve (AUC) are higher across all classification metrics for the VGG16 model, indicating that a better overall performance compared to the ResNet50 model.

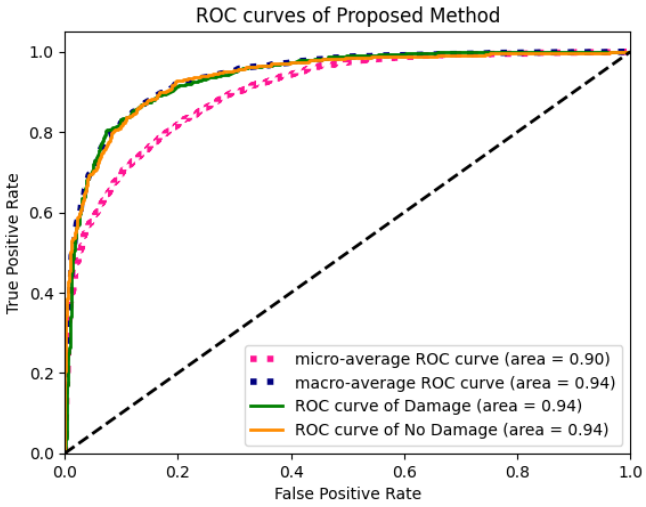

Fig. 15. ROC-AUC plot for the VGG16 model

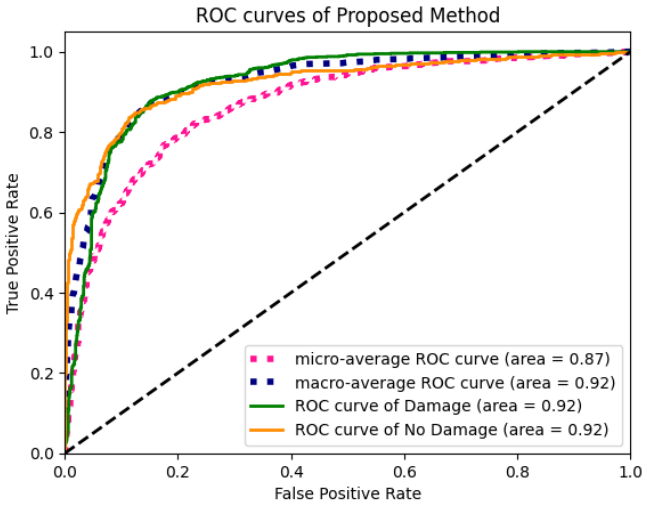

Fig. 16. ROC-AUC plot for the ResNet50 model

E. Computational Expense

With the implementation of the early-stopping model enhancement technique, the required computation for each model was different. As shown in figures 9 and 10, the

training of the ResNet50 model was stopped early at epoch 21. In comparison, as shown in figures 7 and 8, the training of the VGG16 model stopped at epoch 81. Therefore, the computational expense of training the VGG16 model was higher than training the ResNet50 model.

F. Limitations

One area of limitation was the satellite imagery used. Satellite images can be affected by objects such as clouds and fog which may affect the line of sight from the satellite to the house. To make the automated detection systems less susceptible to varying weather conditions, alternative methods of imagery can be utilised, such as unmanned UAVs [19].

VI. Conclusion

In this project an automated classification system to detect the damage status of houses post-hurricane by analysing satellite images was built. Transfer learning models VGG16 and ResNet50 were trained and evaluated. The VGG16 model caused less type I errors but caused slightly more type II errors compared to the ResNet50 model. The VGG16 model outperformed the ResNet50 model on performance metrics accuracy, precision, specificity, and F_1 score. Furthermore, the AUC from the ROC plot (figures 15 and 16) was higher across the micro-average and macro-average, damage, and no damage ROC curves. Therefore, VGG16 can be concluded as being the best performing model.

VII. Project Prototype

The code used for this project is available at https://colab.research.google.com/drive/174pUWL8lW11M9DN9OgBJmz88DqSzwY4h?usp=sharing

References

[1] M. Office, "Hurricanes," https://www.metoffice.gov.uk/research/weather/tropical-cyclones/hurricane, accessed: 2022-04-18.

[2] N. O. for Coastal Management, "Hurricane costs," https://coast.noaa.gov/states/fast-facts/hurricane-costs.html, accessed: 2022-04-18.

[3] E. Krause and R. Reeves, "Hurricanes hit the poor the hardest," https://www.brookings.edu/blog/social-mobility-memos/2017/09/18/hurricanes-hit-the-poor-the-hardest/, accessed: 2022-04-18.

[4] C. for Climate and E. Solutions, "Hurricanes and climate change," https://www.c2es.org/content/hurricanes-and-climate-change/, accessed: 2022-04-18.

[5] T. D. of Homeland Security Federal Emergency Management Agency, "Damage assessment manual," 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] B. Xie, J. Xu, J. Jung, S.-H. Yun, E. Zeng, E. M. Brooks, M. Dolk, and L. Narasimhalu, "Machine learning on satellite radar images to estimate damages after natural disasters," in *Proceedings of the 28th international conference on advances in geographic information systems*, 2020, pp. 461–464.

[9] H. A. Al-Najjar, B. Pradhan, B. Kalantar, M. I. Sameen, M. Santosh, and A. Alamri, "Landslide susceptibility modeling: An integrated novel method based on machine learning feature transformation," *Remote Sensing*, vol. 13, no. 16, p. 3281, 2021.

[10] A. Kumaraswamy, B. N. Reddy, and R. Kolla, "Richter's predictor: Modelling earthquake damage using multi-class classification models," in *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*. IEEE, 2020, pp. 1–5.

[11] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *2016 International conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016, pp. 1–8.

[12] Q. Zhao, Z. Zhou, S. Dou, Y. Li, R. Lu, Y. Wang, and C. Zhao, "Rethinking the zigzag flattening for image reading," *arXiv preprint arXiv:2202.10240*, 2022.

[13] C. Xu, "Applying mlp and cnn on handwriting images for image classification task," in *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*. IEEE, 2022, pp. 830–835.

[14] M. Kumar, K. Namrata, and N. Kumari, "Hyper-parametric improved machine learning models for solar radiation forecasting," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 23, p. e7190, 2022.

[15] E. Aghapour, Y. Zhang, A. Pathania, and T. Mitra, "Pipelined cnn inference on heterogeneous multi-processor system-on-chip."

[16] B. J. Kim, H. Choi, H. Jang, D. G. Lee, W. Jeong, and S. W. Kim, "Dead pixel test using effective receptive field," *Pattern Recognition Letters*, vol. 167, pp. 149–156, 2023.

[17] H. Xu, X. Su, and D. Wang, "Cnn-based local vision transformer for covid-19 diagnosis," *arXiv preprint arXiv:2207.02027*, 2022.

[18] N. Akhtar and U. Ragavendran, "Interpretation of intelligence in cnn-pooling processes: a methodological survey," *Neural computing and applications*, vol. 32, no. 3, pp. 879–898, 2020.

[19] C.-S. Cheng, A. H. Behzadan, and A. Noshadravan, "Deep learning for post-hurricane aerial damage assessment of buildings," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 6, pp. 695–710, 2021.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] P. Berezina and D. Liu, "Hurricane damage assessment using coupled convolutional neural networks: a case study of hurricane michael," *Geomatics, Natural Hazards and Risk*, vol. 13, no. 1, pp. 414–431, 2022.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[24] S. Kaur, S. Gupta, S. Singh, V. T. Hoang, S. Almakdi, T. Alelyani, and A. Shaikh, "Transfer learning-based automatic hurricane damage detection using satellite images," *Electronics*, vol. 11, no. 9, p. 1448, 2022.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[28] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[29] J. Bao, "Utilizing post-hurricane satellite imagery to identify flooding damage with convolutional neural networks," *arXiv preprint arXiv:2209.02124*, 2022.

[30] M. Alshaye, F. Alawwad, and I. Elshafiey, "Hurricane tracking using multi-gnss-r and deep learning," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2020, pp. 1–4.

[31] F. Alam, F. Ofli, M. Imran, T. Alam, and U. Qazi, "Deep learning benchmarks and datasets for social media image classification for disaster response," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2020, pp. 151–158.

[32] S. Kaur, S. Gupta, and S. Singh, "Hurricane damage detection using machine learning and deep learning techniques: a review," in *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1. IOP Publishing, 2021, p. 012035.

[33] P. Serafino, "Exploring the uk's digital divide," *Office for National Statistics*, 2019.

[34] H. Tabrizchi, S. Parvizpour, and J. Razmara, "An improved vgg model for skin cancer detection," *Neural Processing Letters*, pp. 1–18, 2022.

[35] N. Habib, M. M. Hasan, M. M. Reza, and M. M. Rahman, "Ensemble of chexnet and vgg-19 feature extractor with random forest classifier for pediatric pneumonia detection," *SN Computer Science*, vol. 1, pp. 1–9, 2020.

[36] A. Shabbir, N. Ali, J. Ahmed, B. Zafar, A. Rasheed, M. Sajid, A. Ahmed, and S. H. Dar, "Satellite and scene image classification based on transfer learning and fine tuning of resnet50," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–18, 2021.

[37] Q. D. Cao and Y. Choe, "Detecting damaged buildings on post-hurricane satellite imagery based on customized convolutional neural networks," 2018. [Online]. Available: https://dx.doi.org/10.21227/sdad-1e56

[38] J. Wang, L. Perez *et al.*, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, vol. 11, no. 2017, pp. 1–8, 2017.

[39] J. Bobulski and M. Kubanek, "Deep learning for plastic waste classification system," *Applied Computational Intelligence and Soft Computing*, vol. 2021, pp. 1–7, 2021.

[40] C. Zhang, W. Zhong, C. Li, and H. Deng, "Random walk-based erasing data augmentation for deep learning," *Signal, Image and Video Processing*, pp. 1–8, 2023.

[41] M. J. Cardoso, W. Li, R. Brown, N. Ma, E. Kerfoot, Y. Wang, B. Murrey, A. Myronenko, C. Zhao, D. Yang *et al.*, "Monai: An open-source framework for deep learning in healthcare," *arXiv preprint arXiv:2211.02701*, 2022.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.