

# AUTOMATED LARGE-SCALE DAMAGE DETECTION ON HISTORIC BUILDINGS IN POST-DISASTER AREAS USING IMAGE SEGMENTATION

J. Kallas, R. Napolitano\*

Pennsylvania State University, United States of America - (jkallas, nap)@psu.edu

**KEY WORDS:** Disaster Recovery, Buildings Inspection, Image Segmentation, Wall Collapse, Computer Vision, Machine Learning.

## ABSTRACT:

This research aims to investigate the application of computer vision and machine learning for the automatic detection of wall collapse damage in historic buildings caused by natural and man-made disasters. Given the complexities involved in inspecting damaged buildings, particularly in post-disaster scenarios, this research aims to establish a foundation for creating an automated assessment process. Our findings demonstrate the successful automatic detection of various shapes of wall collapse on damaged buildings from the Beirut explosion of 2020, as well as from other damaged buildings obtained from the internet, thereby highlighting the transferability of our method. This research paves the way for the development of a more robust machine learning model capable of detecting a broader range of damages, which can significantly enhance the efficiency and accuracy of post-disaster assessment of historic structures. The paper presents a novel approach for damage detection and quantification, which underscores the potential of structural health monitoring in improving disaster response and recovery efforts.

## 1. INTRODUCTION

Natural and man-made disasters on historic buildings pose a significant threat to their structural integrity and durability. To reduce disaster risk at cultural heritage sites, international organizations, such as UNESCO, are engaged in ongoing efforts to identify and implement multiple measures (UNESCO, 2015). The inspection and assessment of damaged historic buildings is a complex and time-consuming process, even in the case of a single building. However, in the event of a disaster, the scope of impact often extends beyond individual buildings or clusters, and may encompass entire cities, states, or countries. Traditional assessment and surveying techniques, while efficient, can be further challenged by the aftermath of a disaster. The timely implementation of emergency interventions is critical to the preservation of damaged historic buildings, and this requires a rapid and efficient assessment process. Thus, the exploration of alternative, more efficient techniques for the assessment of damaged historic buildings in post-disaster scenarios is of utmost importance.

The use of digital technologies has been shown to significantly enhance the speed and efficacy of disaster recovery processes. This was demonstrated in the aftermath of the August 4th, 2020 explosion in Beirut, where the author conducted a comprehensive digitization project of damaged historic buildings using photogrammetry (UNESCO, 2022); (J. Kallas, M. Silver, O. Vileikis, 2020). This generated detailed 3D models of the structures, allowing for rapid implementation of emergency interventions and restoration planning (Figure 1). The application of computer vision techniques, such as image segmentation, can also play a crucial role in disaster relief efforts. Image segmentation, a key aspect of image processing and computer vision, has numerous applications such as object detection, face recognition, and satellite image analysis (Shervin Minaee, Yuri Boykov, et al., 2020). Other successful applications of image segmentation include the automated detection of cracks in roads and bridges,

\* Corresponding author



**Figure 1.** 3D photogrammetric model conducted by the author of one of the damaged historic buildings in Beirut following the explosion of August 4th, 2020.

crucial for infrastructure monitoring and assessment (Chen, C., Seo, H., Jun, C. et al., 2022).

The current study endeavors to investigate the application of computer vision and machine learning in automatically identifying damage on historic buildings in post-disaster zones. This approach has significant practical implications, as it facilitates a rapid damage assessment process by automatically localiz-

ing damages on impacted buildings. Moreover, integration with photogrammetric 3D modeling allows for automated quantification of damages, potentially reducing the time and effort required from experts and inspectors on site. Studies focused on damage detection have been widely reported in the literature and approached from various perspectives. In the 1990s, Wu et al. (X. Wu et al., 2003) and Masri et al. (S. F. Masri et al., 1996) conducted experiments to classify undamaged and damaged structures using neural networks. More recently, some studies have utilized multi-resolution convolutional neural networks for image classification of damaged buildings in the aftermath of natural disasters such as earthquakes (D. Duarte et al., 2018). Fujita et al. employed a convolutional neural network (CNN) to detect damaged regions following a tsunami, by identifying completely disappeared buildings using pre- and post-disaster aerial images (A. Fujita et al., 2017). Ma et al. (H. Ma et al., 2020) applied a Geographic Information System to extract building damage information post-disaster and then utilized an improved convolutional neural network to classify the degree of damage for building groups. As previously highlighted, the majority of studies regarding post-disaster damage detection have primarily utilized satellite imagery and focused on determining general building damage rather than specific types of damage. The use of satellite imagery, however, presents several limitations, such as the potential for cloud cover to severely reduce the effectiveness of real-time damage detection systems and the vertical orientation of the images which limits damage recognition to only the roofs of buildings.

Studies examining crack detection as a specific damage type have been widely explored in the literature as well (A. Mohan, S. Poobal, 2018). Alam et al. (S. Y. Alam et al., 2015) proposed a novel approach combining digital image correlation and acoustic emission, which allows for precise measurement of surface displacements and determination of crack openings and spacing. Shan et al. (B. Shan et al., 2016) developed a stereovision-based method for crack width detection, utilizing two cameras to recover crack edge coordinates. Fujita et al. (Y. Fujita et al., 2011) designed an automatic crack detection system for noisy concrete surfaces, which involves two pre-processing steps and two detection steps, including the use of a multi-scale line filter with the Hessian matrix to emphasize cracks and a combination of probabilistic method and adaptive threshold algorithm for detecting cracks. Pereira et al. (F. C. Pereira et al., 2015) suggested using an Unmanned Aerial Vehicle for autonomous inspection of building pathologies, incorporating various image processing algorithms for crack detection. Zou et al. (Q. Zou et al., 2012) introduced Deep-Crack, a trainable deep convolutional neural network model, capable of detecting cracks in pavement images by leveraging high-level features such as ridges. Loverdos et al. (D. Loverdos et al., 2022) employed various deep learning networks for automated crack detection in brick masonry, and found that the convolutional neural network outperformed traditional image-processing functions even when applied to clean images.

This study focuses on detecting wall collapse damage, which poses unique challenges compared to crack detection that has already been extensively researched. Wall collapse lacks a specific or uniform shape, unlike cracks, and can occur partially or completely, following the masonry joints or breaking through the masonry. Detecting wall collapse damage is also complicated by the fact that it creates an opening into the building's interior, potentially misleading algorithms that analyze surrounding surfaces. The study employs an experimental machine

learning model, trained using datasets collected after the Beirut explosion, to evaluate the effectiveness of image segmentation for detecting wall collapse damage. The Mask R-CNN method and the ResNet50 architecture are used in the experiment (W. Abdulla, 2017), with the Mask R-CNN structure involving two steps: processing input images to generate region suggestions and confirming and classifying the target object while creating bounding boxes and masks.

## 2. RESEARCH AIM

The objective of this research is to develop a deep learning and artificial intelligence-based approach for automating the detection of large-scale damages, such as wall collapse, on damaged historic buildings, using the case of the Beirut blast as a case study. The proposed approach aims to establish a foundation for a complete automated damage assessment process that can be deployed in post-disaster scenarios, thereby reducing the time required for inspectors to be physically present on site and expediting the recovery of damaged structures. This research has the potential to contribute to the broader field of disaster management and resilience, and offers practical insights for future applications.

## 3. MATERIALS & METHODS

Image segmentation is the process of assigning each pixel in an image to a specific category or class. There are several methods of image segmentation available, but two methods that have gained prominence in the field of Deep Learning are Semantic Segmentation and Instance Segmentation. In Semantic Segmentation, all pixels of an object belonging to a specific class are given the same label or color value. This method is widely used for various computer vision tasks, such as autonomous driving and medical image analysis, as it allows the model to identify and classify different objects in an image. On the other hand, Instance Segmentation assigns a unique label or color value to each pixel of every object in a class, providing a more precise identification of individual objects. This method is often used in applications such as robotics and object detection. The choice of image segmentation method depends on the specific task and available resources. In this study, Semantic Segmentation was utilized to detect wall collapse in damaged buildings, which provided an efficient means of identifying damaged regions in the structures of existing buildings. As computer vision technology continues to advance, image segmentation is expected to become more widely used, facilitating the accurate analysis of complex visual data for a variety of applications.

The methodology employed in this experiment utilized the open-source implementation of the Mask R-CNN method developed by Matterport (W. Abdulla, 2017), which is based on the Feature Pyramid Network (FPN) (T. Y. Lin et al., 2017) and the ResNet50 backbone (K. He et al., 2017). Typically, the backbone network of the Mask R-CNN (Regional Convolutional Neural Network) adopts ResNet101, with 101 network layers. However, for our experiment, which involves a relatively small dataset for detecting wall collapse damage, a lower number of network layers is sufficient to meet the requirements of the study. Therefore, to further enhance the algorithm's running speed, this paper implemented the ResNet50 backbone. Mask R-CNN is an instance segmentation model that consists of two steps. In the first step, the input images are processed to generate region proposals that may contain the target object.

In the second step, these proposals are validated, and the target object is classified, along with the creation of bounding boxes and masks. The model is trained end-to-end on the wall collapse damage dataset to learn and detect the specific damage patterns. The implementation of Mask R-CNN in this study demonstrated a successful workflow in detecting wall collapse damage in images of damaged buildings, highlighting the effectiveness of the model in the context of structural health monitoring and damage identification. The dataset utilized for this experiment comprised 100 aerial images of damaged buildings taken following the Beirut blast of August 4th, 2020. The images were captured using a DJI Phantom 4 Pro drone (DJI, 2020), and included a range of building types exhibiting various degrees of wall collapse damage. The dataset was split into training data (80 images) and validation data (20 images), with the former accounting for 80% of the dataset. To facilitate effective learning, we included images that depicted the damage from various viewpoints, including close-up shots and those taken from a distance, which showed the damage in the context of the post-disaster urban environment.

To mitigate the small size of our dataset, we employed the "imgaug" library (Imgaug Python, 2020) to perform data augmentation during the training process. Specifically, we applied the random flip method, which included horizontal and vertical flipping, with a probability of 0.85. This increased the size of the dataset by 85%. In addition, to enhance the model's robustness, we also performed random rotations on the input images, adjusted their contrast, and added Gaussian noise and filtering operations. This ensured that the model could detect wall collapse damage accurately, even when presented with images that were not included in the original dataset.

The creation of a thoroughly labeled and annotated image dataset is crucial for training a supervised computer vision model. In this experiment, the VGG Image Annotator (VIA) (A. Dutta et al., 2018), an open-source, web-based tool developed by researchers at Oxford University, was utilized to manually create detailed and precise annotations of the wall collapse damage present in all images of the dataset, including both the training and validation sets. This process involved outlining the damaged areas of each image with precise polygons and assigning appropriate labels, resulting in a comprehensive dataset for training and supervising the computer vision model (Figure 2). The annotations were then exported and saved in ".json" file format (json, 2001), enabling seamless integration with the training process.

In order to optimize the training process of the wall collapse damage detection model, a series of modifications were made to the dataset and optimization algorithms. The original aerial images in our dataset were 5472 x 3648 pixels, which were down-sized to 512 x 512 pixels during the training for the color image and color mask pipelines, and further down-sampled to 28 x 28 pixels for the binary mask pipeline. These modifications were necessary to ensure efficient model training speed and minimize the use of GPU memory and space. To optimize the optimization algorithm, we utilized the Stochastic Gradient Descent (SGD) (Keras, 2018a) optimization algorithm with a learning rate of 0.001 and learning momentum of 0.9, which demonstrated strong robustness in selecting hyper parameters. Given the small dataset size, we trained our model for only 50 epochs with 100 steps per epoch to prevent overfitting. The experiment was conducted using TensorFlow 1.14 (Tensorflow, 2019) and Keras 2.2.4 (Keras, 2018b), and trained on an Intel(R) UHD Graphics 630 GPU (Intel, 2016) with 8 GB memory.



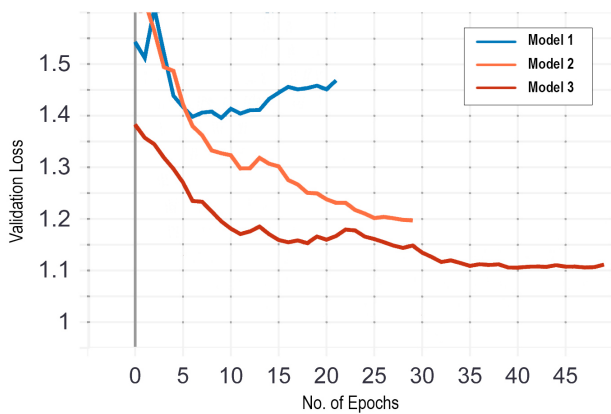
**Figure 2.** Labeling of the images using the VGG Image Annotator (VIA) tool.

In this experiment, we leveraged the pre-trained weights from MS COCO (T. Y. Lin et al., 2015), made available by TensorFlow [28], as a transfer learning approach to initialize our network. Developing a deep CNN from scratch can be an arduous task for several reasons. Primarily, optimal performance for deep CNN architectures demands a substantial dataset, which may prove challenging to obtain in some cases. Additionally, training a deep model can be computationally expensive, requiring significant computational power and leading to extended convergence times. Even with an adequate dataset and computational resources, pre-trained neural networks may still outperform neural networks trained from scratch (M. Ouqab et al., 2014); (S. Ahmed et al., 2021). This has spurred the widespread adoption of transfer learning in many applications, including the present study.

#### 4. RESULTS & DISCUSSIONS

As previously noted, the ResNet50 backbone was utilized in our experiment instead of the more recent and common ResNet101. This decision was not arbitrary, but rather a result of our concern that the ResNet101 backbone could cause overfitting of our model due to the limited size of our dataset. To ensure the optimal choice of backbone, we trained three distinct Mask-R-CNN models utilizing the aforementioned pipelines and evaluated their performance. The first model was trained with the ResNet101 backbone, which exhibited overfitting prior to reaching the final epochs, as illustrated by the validation loss curve in Figure 4. For the second model, we maintained the same parameters and implemented the ResNet50 backbone, resulting in improved performance, as indicated by the validation loss curve in Figure 3. However, in an effort to further enhance overall model performance, we utilized the ResNet50 backbone for the third model, while modifying parameters such as weight

decays, validation steps, and the number of epochs. Table 1 presents the selected parameters for each of the three models.



**Figure 3.** Validation Loss curve

Training Parameters	Models		
	Model 1	Model 2	Model 3
Backbone	ResNet101	ResNet50	ResNest50
Learning Rate	0.001	0.001	0.001
Learning Momentum	0.9	0.9	0.9
Number of Classes	2	2	2
Number of Epochs	30	30	50
Steps per Epoch	100	100	100
Validation Steps	50	50	200
Weights Decay	0.0001	0.0001	0.0005

**Table 1.** Table showing the parameters we used for each of the three models we trained.

When evaluating the accuracy and performance of a machine learning model in a classification task, a confusion matrix is frequently utilized (S. V. Stehman, 1997). The confusion matrix provides a tabular representation of the predicted versus actual labels for a dataset, where each column corresponds to an actual category to which the instance belongs. Within the confusion matrix, TP (True Positive) represents the number of instances that are correctly predicted as belonging to the given category, while FP (False Positive) represents the instances that belong to other categories but are mistakenly classified as belonging to the given category. Similarly, FN (False Negative) corresponds to instances that belong to the given category but are mistakenly classified as belonging to another one. Lastly, TN (True Negative) refers to instances that are correctly classified as belonging to other categories. In our study, we generated a confusion matrix for each of the three models at the conclusion of each training iteration, for both the training and validation sets. These matrices provide a detailed account of the performance of the models and are presented in Figure 4. Through the analysis of these matrices, we are able to evaluate the classification performance of our models, and more specifically, identify any instances where our model may have struggled to accurately predict the category of a given instance.

In order to compare the three models in terms of their performance, several commonly used evaluation metrics were utilized for both the training and validation sets. These metrics provide a comprehensive understanding of the models' performance in terms of overall accuracy, detection, and segmentation performance. Specifically, each model was compared based on precision, recall, and F1 scores. Precision is defined as the proportion of relevant instances, out of the total instances that the

model retrieved. In contrast, the recall score is defined as the proportion of correctly assigned instances, such as the percentage of "Wall Collapse" images that were correctly classified as damaged. The F1 score, on the other hand, represents the harmonic mean of Precision and Recall. To calculate these metrics, the generated confusion matrices were utilized. The detailed results of each of the three trained models are presented in Table 2, providing a clear and concise comparison of their performance in terms of these important evaluation metrics. Through this analysis, we are able to determine which model performed the best overall and identify specific strengths and weaknesses of each model for future improvement.

Upon analyzing the results presented in Table 2, it can be inferred that the ResNet50-based Model 2 outperforms Model 1, which utilizes ResNet101 as its backbone. However, Model 3, which also employs ResNet50 as its backbone, exhibits superior performance compared to both the aforementioned models. The noteworthy enhancement in Model 3's performance can be attributed to the fine-tuning of training parameters, namely, an increase in decay weights, validation steps, and training epochs, as illustrated in Table 1.

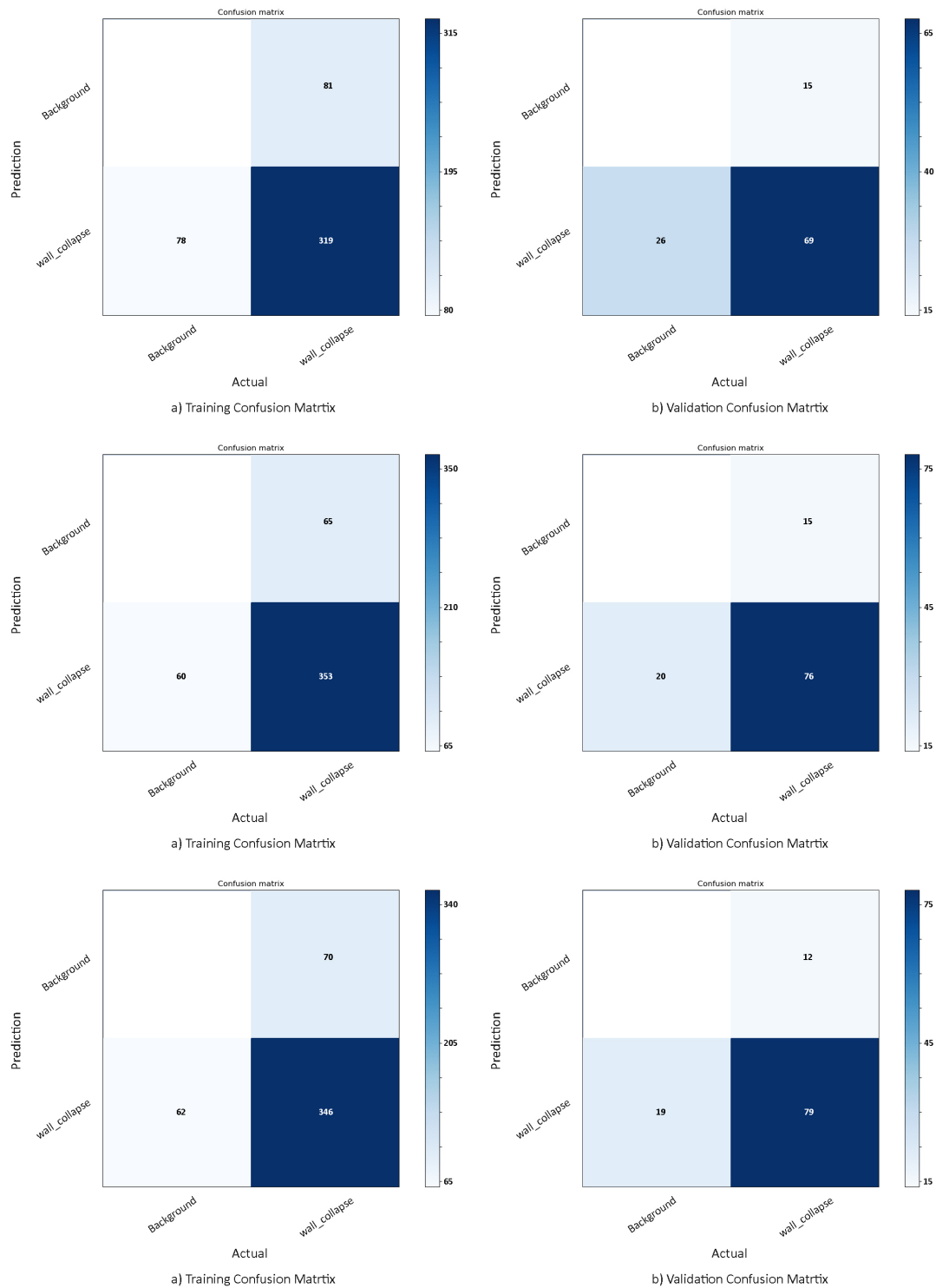
The predictions made using the results of model 3 are presented in this section. The wall collapse damage detection accuracy achieved by the model was 72.38% for the training set and 71.81% for the validation set, indicating that the model was able to correctly identify the majority of wall collapse damages in the dataset (Figure 5). The precision of wall collapse detection was found to be 83.17% for the training set and 86.81% for the validation set. Similarly, the recall of wall collapse damage detection reached 84.80% and 80.61% for the training set and the validation set, respectively. The F1 score of model 3 was found to be 83.96% for the training set and 83.59% for the validation set, indicating that the model had a balanced performance in terms of precision and recall. However, misdetections were observed in some cases due to the presence of trees that were partially covering the wall collapse damage or the presence of wooden and metallic scaffolding inside the space where a wall damage occurred. In some cases, wrong detections were also made due to the similarity of demolished windows with actual wall collapse damage. These issues can be addressed in the future by increasing the size of the training dataset and improving the model's ability to learn the relevant features. Visual examples of the misdetections and wrong detections are presented in Figure 6 and Figure 7, respectively.

To verify the efficacy of our proposed method, we conducted experiments on a diverse set of wall collapse images, which were randomly downloaded from the internet. Our evaluation demonstrated that the method yields satisfactory results in detecting wall collapses, indicating its effectiveness in a real-world setting (Figure 8). Moreover, the good detection results we achieved in the experiments validates the transferability of the proposed approach to different datasets, highlighting its potential for a wide range of applications. In summary, our findings suggest that the proposed method holds promise for the efficient detection of wall collapses, paving the way for its application in various domains such as structural health monitoring, and disaster response.

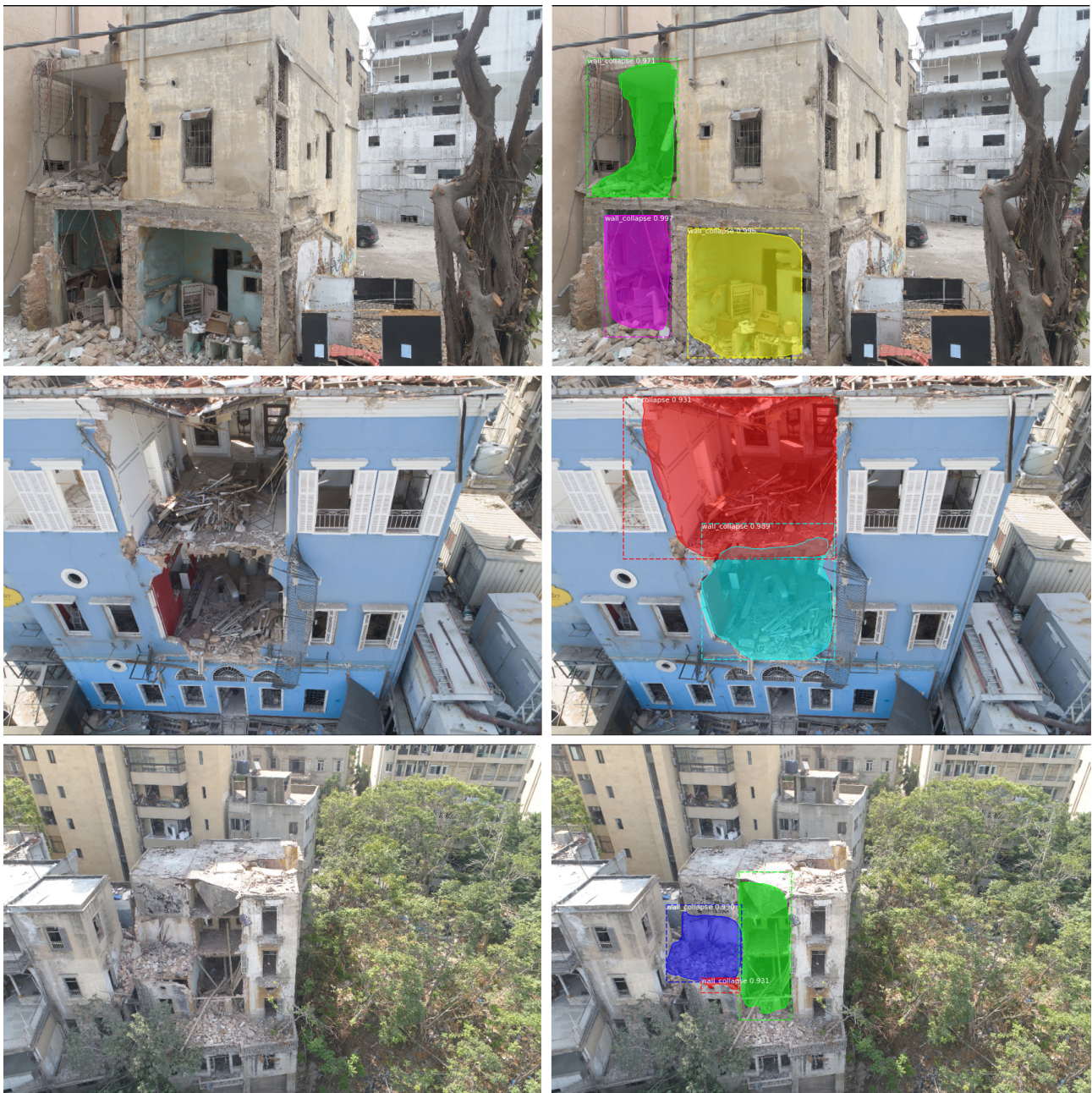
Despite not achieving an overall accuracy level over 80% or 90%, our experiment can be deemed successful in validating the main concept of our study, which aims to demonstrate the feasibility of auto-detecting large-scale and complex damage types using image segmentation. To date, image segmentation has

Evaluation Metrics	Model 1		Model 2		Model 3	
	Training	Validation	Training	Validation	Training	Validation
Accuracy score	66.73%	62.72%	<b>73.84%</b>	69.00%	72.38%	<b>71.91%</b>
Precision score	79.75%	82.14%	<b>84.44%</b>	83.51%	83.17%	<b>86.81%</b>
Recall score	80.03%	72.63%	<b>85.47%</b>	79.16%	84.80%	<b>80.61%</b>
F1 score	79.88%	77.09%	<b>84.95%</b>	81.36%	83.96%	<b>83.59%</b>

**Table 2.** Table showing the results of the different evaluation metrics we used to evaluate the performance of the three models we trained.



**Figure 4.** Confusion Matrices of both the Training and Validation sets for the three different models we trained. Row 1: Model 1; Row 2: Model 2; Row 3: Model 3.



**Figure 5.** Photos showing the wall collapse damage detection results on random photos taken after the Beirut explosion of August 4th, 2020.



**Figure 6.** Photos showing the misdetection of some of the wall collapse damage in areas partially covered by trees or by wooden and metallic scaffolding on the inside.



**Figure 7.** Photos showing the wrong detection of the wall collapse damage due to the similarity of demolished windows with actual wall collapse damage.



**Figure 8.** Photos showing the wall collapse damage detection results on random photos of damaged buildings taken from the internet.

been limited to detecting cracks or overall damaged buildings, as demonstrated in the first part of this paper, rather than detecting specific complex structural damage. We anticipate that future works with larger datasets will yield better results.

## 5. CONCLUSION

In conclusion, while neural networks have been widely used for detecting damaged buildings in the aftermath of disasters, most

studies have focused on classifying buildings into general damage categories, with limited attention given to sub-classifying specific types of damage. In this research, we employed Mask-RCNN and ResNet50 backbone to detect wall collapse damage in images collected from Beirut following the devastating explosion of August 4th, 2020. We trained multiple models with varying parameters and evaluated their performance using various metrics, including accuracy, precision, recall, and F1 scores. The best performing model was then used for im-

age segmentation, resulting in good detection of wall collapse damage in different scenarios. Our future work will focus on expanding the dataset to include larger samples of wall collapse and other typical damages that historic buildings experience during unexpected disasters. This will enable the development of a more robust model capable of detecting multi-scale damages. Additionally, we plan to explore novel techniques for transferring segmentation onto 3D point clouds, which will facilitate a complete automated damage assessment process for deployment in post-disaster scenarios. By minimizing the time and need for physical inspections on site, our approach has the potential to expedite the recovery of damaged structures, thus aiding disaster response and recovery efforts.

## FUNDING

This material is based upon work supported by the National Science Foundation under Grant IIS-2123343 and Grant CMMI-2222849. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- A. Dutta et al., 2018. Vgg image annotator (via).
- A. Fujita et al., 2017. Damage detection from aerial images via convolutional neural networks. 15th IAPR International Conference on Machine Vision Applications (MVA).
- A. Mohan, S. Poobal, 2018. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal* 57, pp. 787–798.
- B. Shan et al., 2016. A stereovision-based crack width detection approach for concrete surface assessment. *KSCE Journal of Civil Engineering* 20, pp. 803–812.
- Chen, C., Seo, H., Jun, C. et al., 2022. A potential crack region method to detect crack using image processing of multiple thresholding. *SIVIP* 16, pp. 1673–1681.
- D. Duarte et al., 2018. Multi-resolution feature fusion for image classification of building damages with convolutional neural networks. *Remote Sensing* 10, pp. 1–26.
- D. Loverdos et al., 2022. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*.
- DJI, 2020. Dji phantom 4 pro drone.
- F. C. Pereira et al., 2015. Embedded image processing systems for automatic recognition of cracks using uavs. *IFAC-PapersOnLine* 48, pp. 16–21.
- H. Ma et al., 2020. Improved cnn classification method for groups of buildings damaged by earthquake, based on high resolution remote sensing images. *Remote Sensing*.
- Imgaug Python, 2020. Image augmentation library.
- Intel, 2016. Intel uhd graphics 630.
- J. Kallas, M. Silver, O. Vileikis, 2020. Beirut's shattered heritage: An emergency documentation.
- json, 2001.
- K. He et al., 2017. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition*.
- Keras, 2018a. Sgd optimizer from keras.
- Keras, 2018b. version 2.2.4.
- M. Ouqab et al., 2014. Learning and transferring mid-level image representations using convolutional neural networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 1717–1724.
- Q. Zou et al., 2012. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters* 33, pp. 227–238.
- S. Ahmed et al., 2021. Transfer learning approach for classification of histopathology whole slide images. *Sensors*.
- S. F. Masri et al., 1996. Neural network approach to detection of changes in structural parameters. *Journal of Engineering Mechanics*.
- S. V. Stehman, 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62, pp. 77–89.
- S. Y. Alam et al., 2015. Use of the digital image correlation and acoustic emission technique to study the effect of structural size on cracking of reinforced concrete. *Engineering Fracture Mechanics* 143, pp. 17–31.
- Shervin Minaee, Yuri Boykov, et al., 2020. Image segmentation using deep learning: A survey. *arXiv Archives*.
- T. Y. Lin et al., 2015. Microsoft coco: Common objects in context. *Computer Vision and Pattern Recognition*.
- T. Y. Lin et al., 2017. Feature pyramid networks for object detection. *Computer Vision and Pattern Recognition*.
- Tensorflow, 2019. version 1.14.
- UNESCO, 2015. Reducing disasters risks at world heritage properties.
- UNESCO, 2022. 60,000 images to save beirut's architectural heritage.
- W. Abdulla, 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow.
- X. Wu et al., 2003. Use of neural networks in detection of structural damage. *Computers Structures* 42, pp. 649–659.
- Y. Fujita et al., 2011. A robust automatic crack detection method from noisy concrete surfaces. *Machine Vision and Applications* 22, pp. 245–254.