Project Report: UAV Disaster Damage Detection Using Semantic Segmentation

Authors: Bryant Sundell, Josiah Keime, and Jonathan Hansen

Affiliations: Indiana University – Luddy School of Informatics, Computing, and Engineering

1 INTRODUCTION

After natural disasters, a quick and accurate assessment of the damage is crucial for effective response and recovery. Traditional methods like manual ground surveys and crowdsourced data can be slow to come. Even with aerial images/satellite imagery, a human analyst must review the images to determine the extent of damage. This process is often time-consuming, labor-intensive, inconsistent, and potentially hazardous for ground survey crews. The core problem our project seeks to address is how to provide a faster and more reliable damage assessment of disaster areas. This project addresses those challenges by training a semantic segmentation model that could identify damaged buildings from aerial images and calculate a cost estimation of the damage.

This project performed fine-tuning on a DeepLabV3 model that has pretrained weights from training on the ImageNet dataset with 50 classes. The DeepLabV3 was fine-tuned on the RescueNet dataset [1]. The model performance was measured with the Intersection over Union (IoU) metric, aiming for over 90% segmentation accuracy on the test set. A secondary objective was to develop a model that translates the segmentation pixels to the economic cost of the damage, which provides actionable data for disaster relief agencies. The RescueNet dataset has 4,494 high-resolution images from Hurricane Michael that are labeled in 11 classes, which include the level of damage to a structure.

This report recounts our methodology, dataset preparation, segmentation model architecture, training, and damage estimates model and methodology. Included are quantitative and qualitative results from our findings of the model training and a discussion of limitations and future work that would further improve our findings.

2 BACKGROUND

The task of assigning a label to each pixel within an image, semantic segmentation, is a challenging problem in computer vision, but techniques and advancements in this area have grown by leaps and bounds recently. Automating areas that were historically manual processes has been growing in many different sectors, and semantic segmentation plays an important role in this for computer vision cases with pixel-level labeling and scene understanding. Along with the growing repertoire of

Unmanned Aerial Vehicle (UAV) high-resolution image datasets, there is great promise for the automation of damage classification and calculation for post-disaster relief efforts.

The RescueNet Dataset [1] provides a large-scale, high-resolution collection of UAV images that were captured after Hurricane Michael. This dataset has detailed pixel-level annotations across 11 classes that are particularly relevant for disaster scenarios. This dataset is helpful in training and evaluating the deep learning models used for post-disaster damage assessments and is the foundation for our project.

Our work builds on previous work by applying updated models like DeepLabV3+, which uses Atrous Convolutions for detailed results, and PSPNet [3], which captures global image features with pyramid pooling. The model uses these techniques to avoid losing spatial resolution and additionally uses a decoder module to analyze and further refine details in the segmentation.

Other studies, like the one by Tran et al. [4], show how deep learning can make damage mapping possible. Their work uses the DeepLabV3+ model for segmentation and estimating damage maps for wildfire disasters. Our project builds on these foundations by applying them to hurricane disaster areas. Our project also uses techniques like uncertainty modeling to make the model more reliable and flexible.

2.1 DATASET

Semantic segmentation is a powerful tool for understanding disaster damage from drone images. The RescueNet dataset [1], created by Rahnemoonfar et al., is specifically built for this kind of research, with detailed labels for damage levels. The dataset had the following attributes

- Source: High-resolution aerial UAS images of the aftermath of Hurricane Michael.
- Resolution: Image Size: 3000 pixels by 4000 pixels.
- Annotations: Pixel-level semantic labels for each image.
- Classes: 11 distinct classes:
    1. Background
    2. Water
    3. Building – No Damage
    4. Building – Minor Damage
    5. Building – Major Damage
    6. Building – Total Destruction
    7. Vehicle
    8. Road-Clear
    9. Road-Blocked
    10. Tree
    11. Pool
- Dataset Split: 80% training, 10% validation, 10% testing

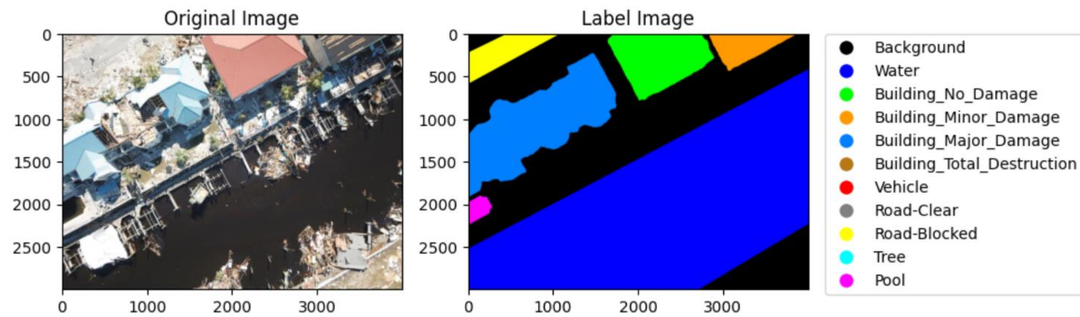

Figure 1: A side-by-side example of the RescueNet image (right) and its corresponding annotated image (middle). The legend (right) contains color-coated descriptions of the annotated labels.

3 METHOD

Our methodology includes image processing, semantic segmentation architecture and training, the evaluation metric, and the damage cost estimation model.

3.1 PREPROCESSING

The preprocessing of the images consisted of reading the image, converting it to RGB, normalizing it according to DeepLabV3 documentation [5], and converting it into tensors. The images were resized the images because of the computational cost associated with training a model on high-resolution images. But to limit the loss of information from image interpolation, the rescaling was incorporated into the model so it would be accounted for during back propagation.

3.2 MODEL ARCHITECTURE

For the core task of semantic segmentation, we implemented the DeepLabV3+ architecture with a ResNet-50 backbone. The team used the deepLabV3 pretrained model object from the PyTorch library and updated the model's head to have the necessary 11 prediction classes. Interpolation steps were added to the forward pass of the model to resize the image for faster inference. The interpolation steps happen before the model is called on the image to resize it to 500x800. The model's output is again interpolated back to the original size so it can be compared to the label image. The team chose to add these extra steps in the hope of balancing inference time and model accuracy.

3.3 MODEL TRAINING

To perform uncertainty modeling, we trained two models with different training regimens. Model 1 was trained for 3 epochs over all layers and 4 epochs on just the model's head. Model 2 was trained for 7 epochs just on the model's head. The different training regimes had a batch training time of 1.43 mins and 0.9 mins, respectively. Figure 2 shows the batch loss over 600 batches of training for models 1 and 2. As you can see from the figure, the loss of the two models is very similar and flat. The team suspects that the loss graph is flat because the model is pretrained and there shouldn't be many changes in the weights during fine-tune on the RescueNet dataset.

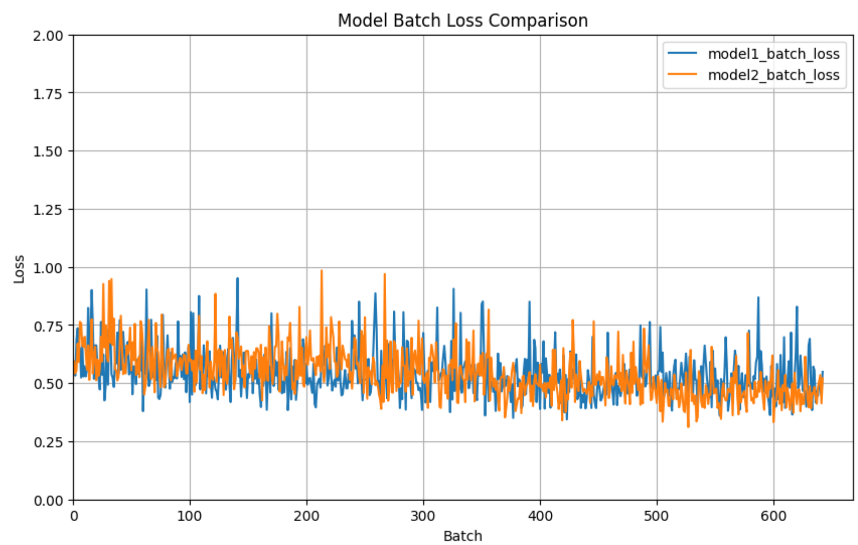

Figure 2: batch loss graph

Due to the large data content within each image and the number of images in RescueNet, the team needed to use special hardware to train the model. The team trained the model using Google Collab v2-8 TPU (Tensor Processing Unit). Even with the advanced hardware, the training was still restricted to a batch size of 25.

	Model 1	Model 2
Batch Size	25	25
Epochs	7	7
Training Regimen	3 epochs of training for all layers. 4 epochs of training to model head.	7 epochs of training of model head.
Batch Training Time	1.43 mins	.95 mins
Avg. Training Loss	0.541	0.544

Table 1: Models' Training Hyperparameters

3.4 Damage Cost Estimation

To output the economic estimates, an appropriate formula was developed:

$$\text{Total Cost} = \sum_{class}(Pixels_{class} \times \text{Area per Pixel} \times Cost\ Factor_{class})$$

Cost Equation Parameters:

- $\text{Pixels}_{\text{class}}$ – Is the number of pixels belonging to a specific class
- $\text{Area}_{\text{pixel}}$ - Estimates the ground area of a single pixel at 0.01 m^2.
- $\text{Cost Factor}_{\text{class}}$ – Is the rough estimated cost to fix or repair the damage per square meter.

  1. Background: $0/$m^2$
  2. Water: $0/$m^2$

3. Building – No Damage
4. Building – Minor Damage: $50/m^2
5. Building – Major Damage: $300/m^2
6. Building – Total Destruction: $1200/m^2
7. Vehicle: $0/m^2
8. Road-Clear: $0/m^2
9. Road-Blocked: $20/m^2
10. Tree: $0/m^2
11. Pool: $0/m^2

Cost factors will vary geographically, socioeconomically, and with various other factors. Some of the factors above do not indicate whether there is damage to the item. For example, we have an annotation for a vehicle, but we do not have an annotation for a vehicle that is damaged.

To use this formula, the segmentation mask is processed to count the pixels belonging to each damage class. These counts are then used in the formula above. We calculated a baseline cost for the test label image that gave an approximate total of $1.25 billion. This can be used as a reference for the total damage cost from our models.

3.5 EVALUATION METRICS

Segmentation: The main metric used to evaluate the segmentation performance was the Mean Intersection over Union (IoU). The IoU is calculated as the ratio of the area of intersection between the predicted segmentation and the ground truth to the area of their Union. The mean IoU is the average of all the scores across all classes. This measure provides a good indication of how the segmentation model is working.

$$IoU = \frac{True\ Positives}{True\ Positives + False\ Positives + False\ Negatives}$$

4 RESULTS AND DISCUSSION

4.1 SEGMENTATION MODEL:

Between the two models, Model 1 performed the best quantitatively. It achieved an IoU of 50.6% while Model 2 achieved 47.6%. The difference is not substantial and given the longer training time of Model 1, it is likely prudent to only train the model head to save on computing time.

Qualitatively, Model 1 performed much better than Model 2. As you can see from Figures 1 and 2, Model 1 produced a prediction image much closer to the label image. The general shape and class predictions are correct. Model 2 does have recognizable shapes, such as the road, but there is a lot of confusion about the building and its level of damage. Model 1 provides results that are more allied with our project's goals.

Our models are showing promise since we show identifiable objects only after 7 epochs of fine-tuning. But other research shows that training high-performing models (over 90% IoU) needs hundreds of epochs of training. These early results show the model works, but it isn't reliable enough for serious cost predictions. Misclassifications of major damage can hugely impact cost estimates.

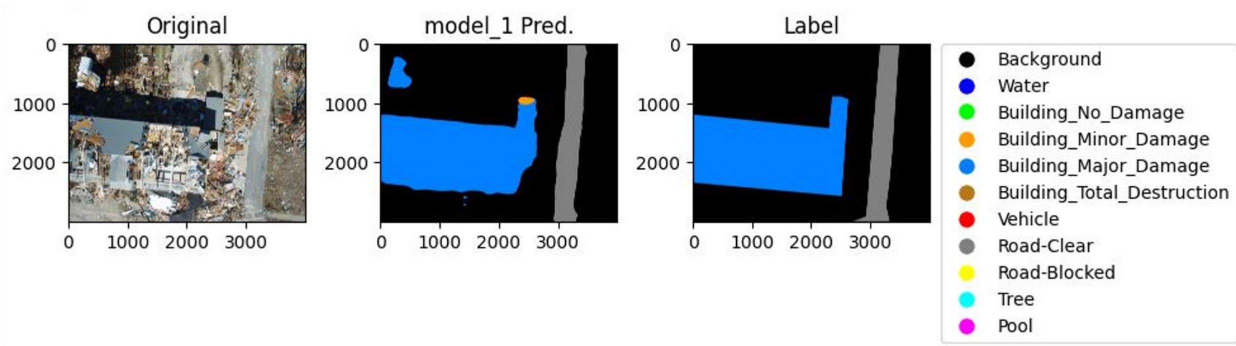

Figure 3: Model 1 evaluation Images

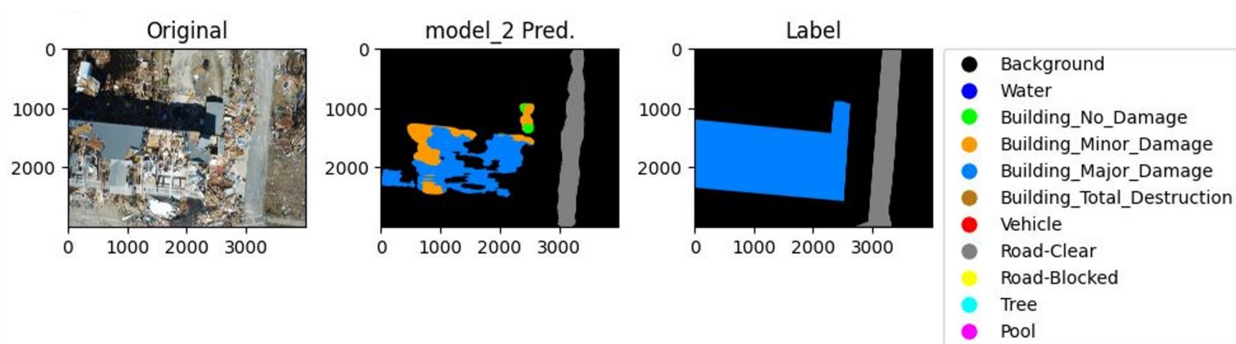

Figure 4: Model 2 evaluation Images

4.2 DAMAGE RESULTS

An evaluation of the cost estimation model was performed using the segmentation output from Model 1 on a set of 5 images from the test set. The results were:

- Pixel-wise accuracy: 81.65%
- Cost prediction error: 178.24%

The high-cost prediction error illustrates the sensitivity of the cost model to the segmentation inaccuracies. Even with small regions misclassified into an incorrect damage category can lead to extremely different outcomes in the final cost estimates. At the current segmentation performance level, the cost estimation model is not reliable.

5 DISCUSSION

5.1 RESULTS INTERPRETATION

The 50.6% IoU result from model 1 after 7 training epochs is indicative that the DeepLabV3+ model requires significantly more training to master the complexity of segmenting the RescueNet dataset. The model has clearly started to identify major features within the images and has a very significant improvement after 1 epoch of training. However, the accuracy shows that further training would improve convergence.

This result is understandable when given the context of standard practices for the training of deep segmentation models on large and complex datasets. Achieving very good results normally requires a much longer training period, often with hundreds of epochs. You can see below in Table 2 examples of models with significantly higher IoU results and much longer training durations.

Table 2: Examples of Training Durations for Semantic Segmentation Models

Model	Dataset	Performance (IoU)	Epochs	Reference
DeepLabV3+	Cityscapes	75 – 80%+	100 – 500+	[2]
PSPNet	Cityscapes	70 – 80%+	100 – 300 +	[3]

Comparing the results from these models to our model with training of 7 epochs shows the relatively good performance with a little amount of training time required. To see better results closer to our aimed >90%, it is recommended to have at least 100+ epochs of training.

The preliminary cost estimation results showing a 178.24% error on a small subset further show the dependence on good segmentation accuracy. The current unreliable estimates are a direct consequence of a lack of training time. Achieving a reliable cost estimation will rely primarily on first achieving a high segmentation score.

5.2 CHALLENGES WE HANDLED:

The choice to use Google Colab for model training was driven by its relatively inexpensive storage rates, ease of setup, and the team's familiarity with the products and tools. Colab GPU resources were critical in handling the large data size and training demands of the project.

However, the sheer scale of the dataset combined with the DeepLabV3+ model pushed the limits of the resources available on this platform. The constraints on the GPU usage restricted our ability to conduct very long training runs (i.e., hundreds of epochs).

Other than the environmental challenges, we also navigated:

- Resizing and interpolating large images without losing label quality
- Managing label inconsistencies
- Setting up a working training process with good initial hyperparameters

5.3 LIMITATIONS

Our primary limitations included some of the factors that stem from the above challenges, along with some others.

- Training duration: This was limited to 7 epochs per the above discussion.
- Simulated Cost Model: The cost estimation was very general in terms of area-per-pixel assumptions. Further data would be needed to give detailed information about the exact area-per-pixel.
- Dataset Generalization: Our dataset is from one specific disaster and one specific area. Segmentation results may not generalize easily to other areas or regions of the world.

5.4 FUTURE IMPROVEMENTS:

Based on the results, challenges, and limitations we have presented in this project, several key areas would greatly advance this damage assessment model.

1. More Training: Train the model for many more epochs to reach better accuracy. This would include securing more GPU resources to facilitate the longer training time needed for high segmentation results.
2. Optimize Speed: Make the model faster for use on drones or edge devices. This could include network pruning, knowledge distillation, or other techniques that would greatly decrease the model size and inference time.
3. Better Cost Factors: Use local economic data for more accurate damage costs.
    - Integrating and updating the local specific construction cost per region.
    - Obtaining flight metadata (altitude, camera parameters) to create more accurate ways of calculating the area-per-pixel.
    - Creating more ways to calculate the cost factors based on different classes. For example, car with minor damage versus a car with major damage, or a pool with minor damage versus a pool with major damage.
4. Explore Other Models: Try newer architectures like HRNet that might have better accuracy or efficiency.
5. 3D Analysis: Investigate ways to estimate volumetric damage. Potentially using LiDAR to move beyond 2D segmentation.
6. Generalization: Applying this model to other natural hurricane disaster sites to test how it generalizes to new regions.

6 CONCLUSION

This project investigated the practicality of automating post-disaster damage assessment by using deep learning and drone images. We fine-tuned a DeepLabV3+ model and got early signs of learning, but much longer training is needed to reach the goal of 90 %+ accuracy.

With more work, this kind of system could help first responders quickly understand damage after a disaster and make faster decisions. In the future, adding better economic models, faster processing, and 3D analysis could make it even more useful.

This project successfully established a baseline and highlighted some of the key challenges that remain. Ultimately, developing and deploying this model gives promise for enabling disaster response teams to obtain rapid data-driven assessments of damage after hurricanes.

7 ACKNOWLEDGEMENTS

The team would like to thank Professor Tang for her guidance and valuable feedback throughout the development of this project.

8 REFERENCES

1. Rahnemoonfar, M., Chowdhury, T., & Murphy, R. (2022). RescueNet: A high-resolution UAV semantic segmentation dataset for natural disaster damage assessment. *Scientific Data, 10*(1), 1-10. https://doi.org/10.1038/s41597-023-02799-4
2. Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 801-818). https://openaccess.thecvf.com/content_ECCV_2018/html/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.html
3. Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2881-2890). https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html
4. Tran, D. Q., Park, M., Jung, D., & Park, S. (2020). Damage-map estimation using UAV images and deep learning algorithms for disaster management system. *Remote Sensing, 12*(24), 4169. https://doi.org/10.3390/rs12244169
5. "Deeplabv3," PyTorch, https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/ (accessed Apr. 30, 2025).