Road Network Extraction via Aerial Image Segmentation

Andrei Solodin, Eli Schlossberg, Jack Swanberg and Kyle Anthes

***Abstract*—We present a computer vision approach to extracting road networks from aerial imagery, with a focus on two key applications: disaster recovery and road detection in underdeveloped areas. For disaster recovery, the goal is to identify usable roadways in the aftermath of natural disasters, such as hurricanes, to aid in rescue and relief efforts. In underdeveloped areas, where roadways are often less defined, we aim to detect roads that are primarily gravel or dirt paths. To achieve this, we will explore three prominent models in semantic segmentation tasks: UNet, Segformer, and ResUNet, each with unique strengths in handling complex image features. Our training data will be sourced from the National Aerial Imagery Program (NAIP), with road masks derived from the Minnesota Department of Transportation's (MnDoT) road centerlines. This approach will provide a comprehensive method for extracting road networks in various terrains and scenarios, demonstrating both the robustness and generalizability of the models.**

I. Introduction

The ability to accurately identify and extract road networks from aerial imagery is a critical task with numerous real-world applications. From urban planning to disaster recovery, the extraction of road networks has the potential to improve decision-making and resource allocation. For instance, in the aftermath of natural disasters such as hurricanes, quick and reliable access to information about usable roads is essential for effective emergency response. Similarly, in underdeveloped regions where road networks are often less distinct, the ability to identify even unpaved roads can assist with infrastructure development and planning.

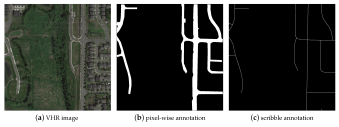

Fig. 1. Example of Roadway Extraction from Aerial Imagery

Advances in computer vision, particularly in the domain of semantic segmentation, provide powerful tools for addressing these challenges. By training models to recognize road networks in aerial imagery, we can automate the extraction of roadways and make these systems more efficient and scalable. In this proposal, we focus on three models commonly used for segmentation tasks: UNet, Segformer, and ResUNet. UNet, known for its success in biomedical image segmentation, is a well-established architecture for segmentation tasks. Segformer, a transformer-based model, introduces an attention mechanism that improves performance in dense prediction tasks. Finally, ResUNet builds on UNet by incorporating residual connections, making it particularly suited for handling remotely sensed data.

To train these models, we will use aerial imagery from the National Aerial Imagery Program (NAIP), with road masks generated from Minnesota's Department of Transportation (MnDoT) road centerlines. Additionally, the RescueNet dataset, which contains imagery from Hurricane Ilene, will allow us to test the models in disaster scenarios, evaluating their ability to generalize to challenging and dynamic environments. This proposal outlines our approach and expected outcomes, including the potential for these models to contribute to disaster relief, urban planning, and navigation systems.

II. Related Work

Semantic segmentation of road networks in aerial imagery has become an increasingly important area of research in computer vision. However, accurately extracting the road networks has numerous challenges such as occlusions caused by buildings, trees, shadows, debris, or flooding. This literature review explores recent advancements in algorithm designs and data set creation relevant to our segmentation problem.

One of the most popular architectures that has been used extensively for segmentation tasks over the past ten years is U-Net, originally proposed Ronneberger, Fischer and Brox in 2015 for medical imaging segmentation [1]. A major drawback that previous architectures had faced was that there was a trade off on what the model paid more attention due depending on what patch size was selected. A larger patch size allowed the model to pay more attention to context surrounding a pixel, at the cost of not being able to extract as much high level detail information from the pixel of interest and its immediate neighbors. The symmetric encoder-decoder approach proposed by U-Net, shown in figure **??**, allows for the model to combine the localized information extracted from the high-resolution image with the contextual information extracted from the lower resolution representation of the image after it has been reduced through multiple convolution and max-pooling steps.

Since the original U-Net was introduced, it has seen multiple different adaptations and improvements utilizing the latest innovations. One such evolution is ResUnet, a deep residual variation of U-Net that was explicitly designed and tested for the task of road extraction [2]. This added residual is an identity mapping that is added with the output of the neural unit and has been shown to improve training accuracy and avoid degradation as the network gets deeper [3].

Traditional methods for extracting road networks rely on incredibly labor-intensive pixel-wise manual annotations to

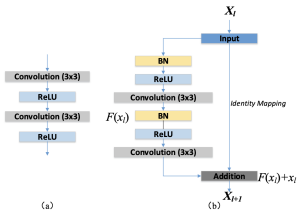

Fig. 2. Example of (a) neural unit used in the original U-Net (b) residual unit with identiy mapping used in ResUnet. Image from [2]

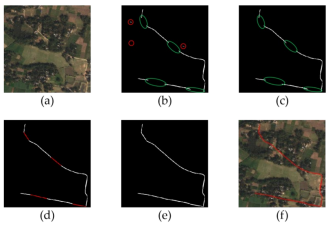

Fig. 3. Sequence of images showing continuation of road segments blocked by occlusion using post-processing techniques

develop their datasets. Wu et al. [4] focused on utilizing crowd-sourced geospatial data from OpenStreetMap and its GPS traces to calculate centerlines to be used for weakly supervised training. Coupled with a novel multi-dilated ResUnet model and a regularized semi-supervised loss function that utilizes permutohedral lattices to achieve linear time complexity, the approach achieved a better performance than the standard full mask supervision method on the segmentation task.

III. Approach

The first portion of the project is generation of a novel dataset consisting of using aerial imagery from NAIP along with the OpenStreetMap road centerlines serving as a basis for road masks. After generating the dataset we verify its accurateness through a visual inspection and summary statistics. The segmentation portion of the project focuses on training and evaluating three models commonly used for segmentation tasks: UNet, Segformer, and ResUNet. The planned strategy for evaluation is using the mean intersection over union (mIoU) and F1 score to quantify how well the models are performing. We then perform error analysis on our models to determine their weaknesses.

An interesting aspect of our evaluation will be to investigate whether augmenting the training data set with images from a different dataset (in other words, diversifying the training data) would have a positive impact on the model test performance. We hypothesize that the model that has been fine-tuned on a greater variety of training data would perform better on the test set.

A stretch goal that will be added if time allows is attempting to create a road network from the extracted road data. One option for implementing this is to require the model to understand intersections or will require a second model for network extraction from the produced roadway extraction models outputs. In this project, we will explore a different approach, which takes the masks, thins them so they are only one pixel wide, and then use the thinned mask to produce a graph that represents the road network. Occlusions will make this task extremely difficult, as the model would need to make an educated guess on whether two roads that enter the same occlusion are separate roads or end up connecting where it is not visible. This problem has been explored before and saw fairly successful results by utilizing some post-processing by filling in broken lines as shown in figure 3 [5].

We then select a model for a proof-of-concept application that demonstrates how a fine-tuned model can be used for real-time road and road damage/obstruction detection.

A. Data Description

1) St. Louis County Aerial Imagery Program: The aerial imagery used in this project was obtained from the St. Louis County Aerial Imagery Program, managed by private contractors. This program provides high-resolution imagery specifically for St. Louis County, Minnesota, and is well-suited for detailed local analyses. The imagery, captured during optimal conditions for visibility, offers a resolution of 1 meter, providing a clear and accurate view of the county's urban and rural landscapes.

The imagery is available in natural color (RGB), making it ideal for visual analysis and interpretation. This dataset's high quality and localized nature ensures it aligns closely with our project requirements for training computer vision models on diverse road networks, from paved highways to unpaved paths in remote areas.

2) OpenStreetMap Road Centerlines: The road centerlines used in this project were sourced from OpenStreetMap (OSM), a community-driven platform providing geospatial data from contributors worldwide. The OSM dataset for St. Louis County, Minnesota, includes detailed information on road geometries and attributes, making it an ideal choice for generating road masks for model training.

Using the OSM data, we constructed road masks by buffering the centerlines with variable widths based on road classifications such as highways, local streets, and unpaved paths. These buffered geometries were then rasterized to match the resolution of the St. Louis County aerial imagery, creating pixel-based representations of roads. This process ensures that the resulting road masks align seamlessly with the aerial imagery, providing accurate labeled training data for the model.

3) Main Dataset: Using the above data sources we created a novel dataset for our road extraction model. Using the OpenStreetMap road centerlines dataset we constructed polygons around the centerlines. This process involved creating variable-width buffers based on the road's classification (e.g., highways, local roads, or unpaved paths). The buffer widths were designed to approximate the real-world width of the roads in the aerial imagery.

Once these buffers were generated, they were converted into polygons representing the road areas. These polygons were then rasterized to match the resolution of the St. Louis County aerial imagery (1 meter), effectively transforming the vector data into pixel-based road masks. These road masks were then

overlaid on the corresponding aerial images to produce labeled training data for the model.

The dataset has been uploaded to HuggingFace at this link. The final dataset split is listed in Table III

Split	Sample count
Train	2814
Validation	60
Test	608

TABLE I
DATASET SPLIT

4) Massachusetts Roads Dataset: This dataset contains aerial images from a variety of urban and rural areas, with road masks generated from OpenStreetMap center lines. This dataset is very similar to our own, and we hoped to enhance the training data for our models in an attempt to improve model performance. The dataset consists of 1500x1500 RGB images and binary masks. We split the original images and masks into 9 sub-tiles of 500x500 each.

B. Models

We have experimented with 3 models: SegFormer [6], U-Net [1] and ResUnet [2]. The following sections summarize the training and evaluation results.

1) SegFormer: SegFormer is a segmentation model unifying Transformers with lightweight multilayer perception (MLP) decoders. We have adapted the HuggingFace implementation of SegFormer [7] for easier training and evaluation. While experimenting with SegFormer we realized that by default, the HuggingFace version is set up for multi-class segmentation. However, our task is binary segmentation of "road" and "background". Thus, we needed to implement a custom loss function, using weighted binary cross-entropy loss. It is important to assign proper weights to the classes, because in our case there is a lot more background in the images than roads. We chose 2 variants of the SegFormer model:

- Segformer-b0. This is a low complexity model with 3.7M parameters, ideal for quick experimentation.
- Segformer-b3. This is a medium complexity model with 45.2M parameters, provides a good trade-off between training efficiency and accuracy.

2) UNet: U-Net is a popular model architecture that is used for a wide range of computer vision tasks. The initial U-Net was proposed in a 2015 paper for the purpose of segmenting biomedical images[1]. The main idea of UNet is to encode the input image into a lower resolution but higher channel feature space. This condensed representation of the image is then brought back to the original dimensions using deconvolution and combining the equivalent resolution encoding stage with the decoding stage as it scales back up. The visual of the architecture is shown in figure 4.

3) ResUNet: ResUNet is a Deep Residual U-Net comprised of three components: encoder, bridge, and decoder. For our implementation, we started with the architecture introduced by Zhang et al. [8] shown in figure 5 and wrote custom metrics and data preparation code.

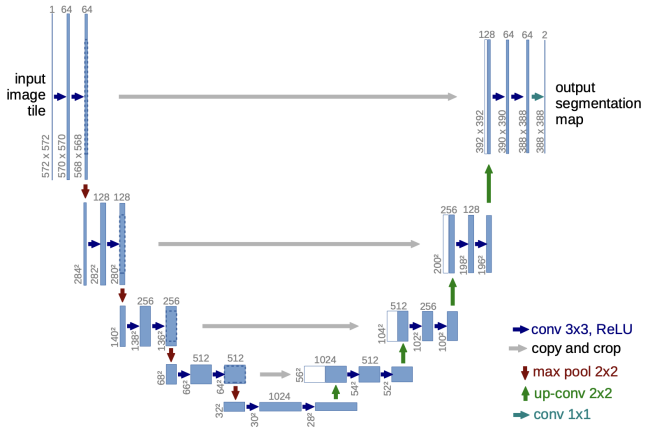

Fig. 4. UNet Model Architecture

[1]

IV. EXPERIMENTS AND RESULTS

A. SegFormer

For our initial experiment, we trained and evaluated SegmFormer-b0 on the St Louis County dataset only. During these experiments, we noticed that the model was not learning, or learning very slowly. Adjusting the learning rate did not help. With some more experimentation, we discovered that the majority of the training images had no roads, being images of countryside, farms, etc, thus having no positive class. We created a variant of our dataset where 90% of the no-road images were filtered out and proceeded to train and evaluate our models on that dataset. This partial dataset The training was done in Google Colab using T4 GPUs.

Parameter	Value
Learning rate	1e-4
Epochs	12
Batch size	16
Training time	36 min

TABLE II
SEGFORMER TRAINING PARAMETERS

We preprocessed the data by resizing the images to 512x512 pixels and randomly rotated and flipped them. For evaluation, we computed mean IOU and F1 scores. The following charts demonstrate the training (6) and evaluation (7) results.

We then repeated the same training on the b3 variant. The following table summarizes the test performance of these models on the test set:

Model Variant	mIOU	F1
b0	0.41	0.56
b3	0.53	0.67

TABLE III
SEGFORMER EVALUATION RESULTS

As expected, the more complex model achieves greater accuracy. However, for a segmentation task, mIOU of 0.53 seems fairly low. To gain further insight into the results, we analyzed some of the lowest scoring test samples:

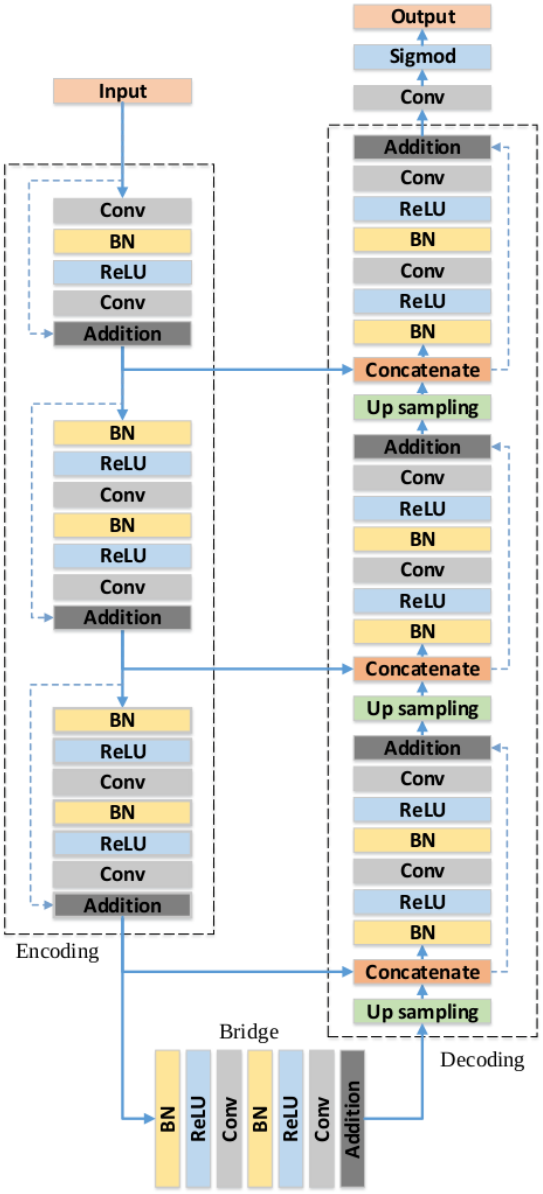

Fig. 5. ResUNet Model Architecture

[8]

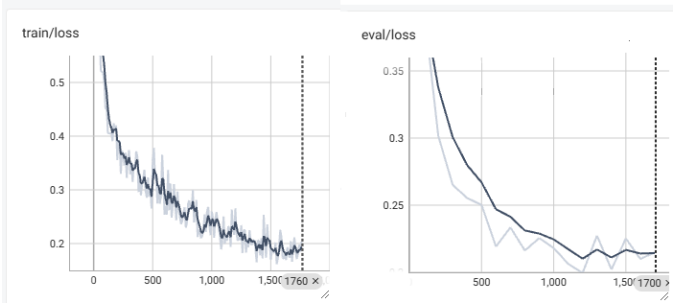

Fig. 6. SegFormer training and validation loss

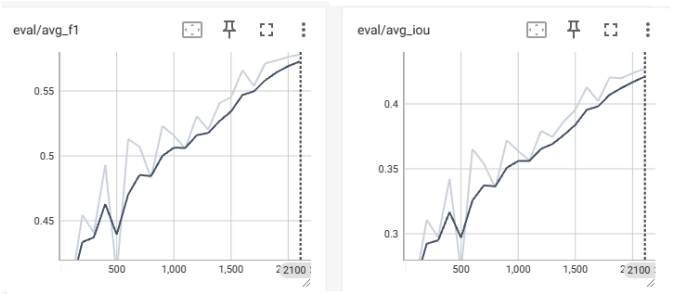

Fig. 7. SegFormer evaluation

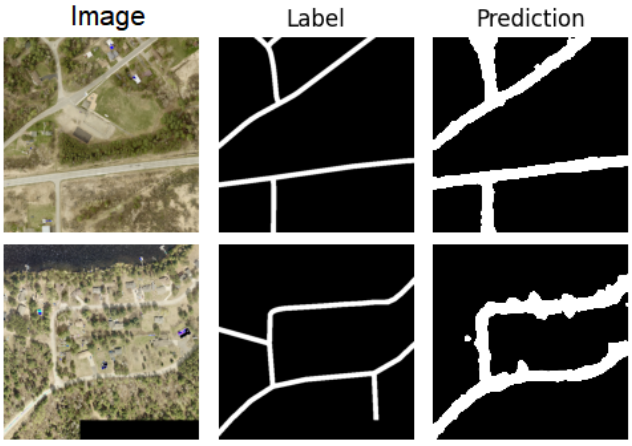

Fig. 8. SegFormer prediction examples

In figure 9 we can see the model struggling with the country roads. In general country roads are far more sparse than the urban roads, it is possible that the model is confusing the two. A potential further experiment would have been to further subclass the roads (country, urban, highways, etc) to make the model more discriminating, and hopefully, more accurate.

In figure 10 we can see the model seemingly identifying roads correctly, but there is no corresponding mask. This could be due to the automated way the labels are generated: if the original data lacks the center lines for a remote road, it would not be present on the mask. We also see that the predicted mask can be thicker than the label, further reducing accuracy. We are uncertain as to the cause of this discrepancy. Perhaps our labeling technique could be improved to more accurately correspond to the road thickness in the images. For example, using the center line data we could identify the polygons that represent the roads in the image using the road color and compute the approximate thickness.

We also attempted to augment our training data with a different dataset (Massachusetts Roads Dataset). We tried to combine the train splits of the two datasets, while keeping the validation and test dataset only from our dataset. Unfortunately, we could not increase the performance of the model this way, in fact the model performance went down to mIOU=0.4 and F1=0.55. Examining the training data we observed that the images and masks from different datasets are perhaps too dissimilar. The color balance is different, and the scale seems to be slightly different. The thickness of the masks is different as well (11).

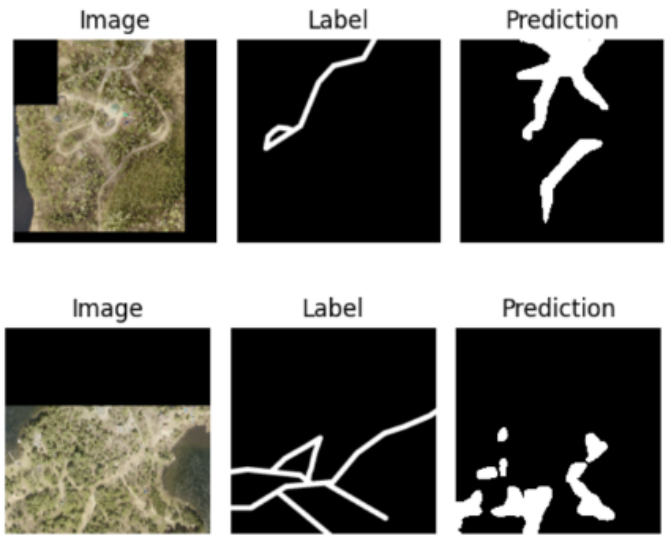

Fig. 9. SegFormer Error Analysis: Country Roads

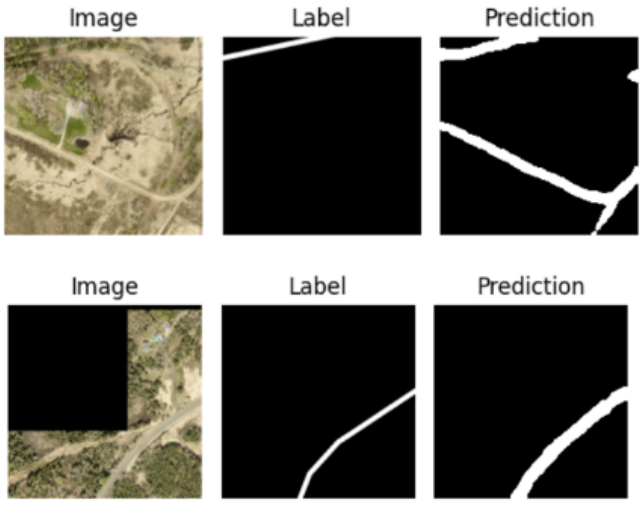

Fig. 10. SegFormer Error Analysis: Potential True Positives and Thickness

A potential future direction would be to apply color balance normalization and rescaling as a pre-processing step.

B. U-Net

The second model that we experimented with was the U-Net model. The U-Net model was trained with the parameters listed in Table IV. All training was done on a personal computer with an NVIDIA GeForce GTX 1660 Super graphics card. For training loss function standard BCELoss was used

Parameter	Value
Learning rate	1e-4
Epochs	10
Batch size	2

TABLE IV
UNET TRAINING PARAMETERS

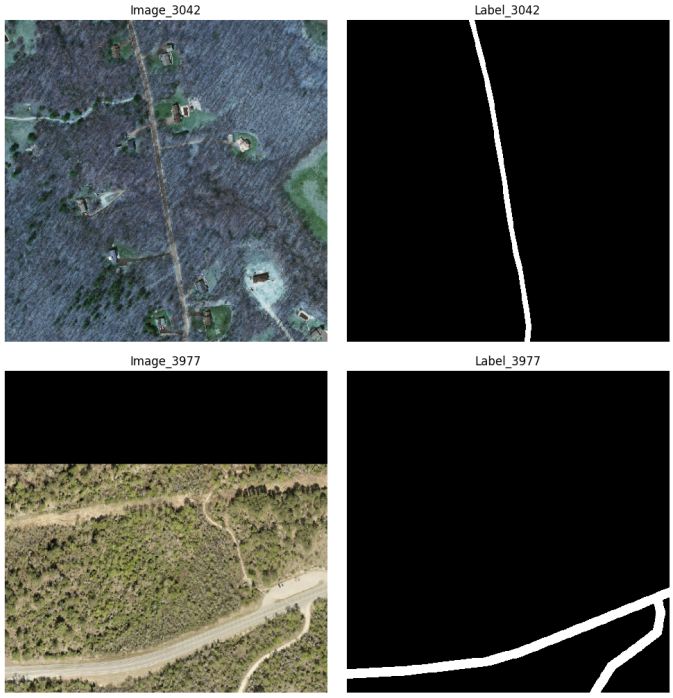

Fig. 11. Comparison of Image/Mask samples from the MA Roads DS (top) and St Louis County DS (bottom)

In figure 12 we can see an example of where the U-Net successfully masked the road, however the label is shifted slightly so the IoU was measured as zero. The figure also illustrates a challenging part of creating a dataset off of centerlines, as the model masks the road up into the driveway/private road, which extends past the label derived off of the public centerline.

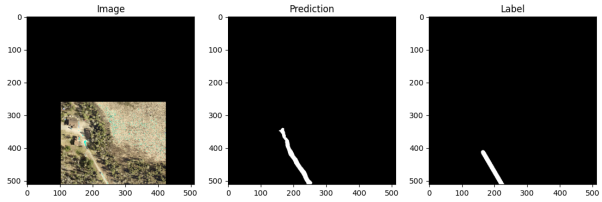

Fig. 12. UNET with misaligned labels

In certain scenarios U-Net did an impressive job in figuring out what types of roads were to be masked and which ones were not. In Figure 13, the model does well not masking the dirt driveways branching off of the curved road in the center of the image. Another example where the model got tripped up and masked some of the sideroad/driveway that was not part of the label is shown in Figure 14.

The model performed very well in the more urban settings of the dataset. When an image showing a town was present in the image the model did an almost perfect job of masking the roads, as can be seen in 15. This is not too surprising, as roads in cities tend to be more organized than rural roads that can wind around in seemingly arbitrary ways. It is also less

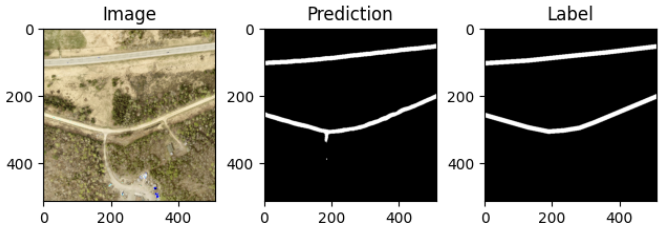

Fig. 13. U-Net successfully not masking dirt roads/paths

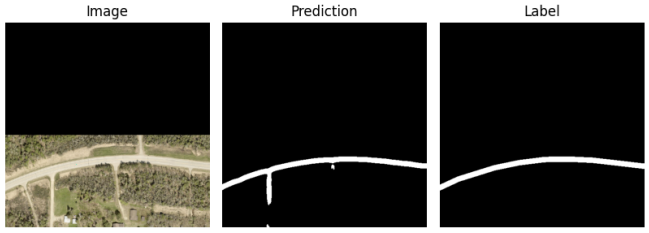

Fig. 14. U-Net prediction example of falsely masking a side road

likely to have other situations that commonly caused the model trouble, such as occlusions from tree cover or driveways/dirt roads that are not classified as road in the labels. The final test metrics for U-Net can be seen in V

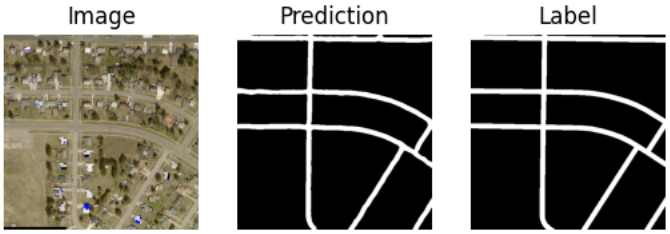

Fig. 15. U-Net prediction example of falsely masking a side road

Metric	Value
IOU	0.553
F1	0.675

TABLE V
U-NET EVALUATION RESULTS

C. ResUNet

Another experiment we conducted was fine-tuning ResUNet on our custom dataset. The standard MLE loss did not deliver the expected results on segmenting the roadway masks similar to those in Zhang et al. [8]. We then experimented with weighted binary cross entropy loss which improved the results but further experimentation with a combined loss using DICE and weighted binary cross entropy with equal weighting achieved the best results. The ResUNet experimentation was conducted in a docker environment and implemented utilizing the Keras library. All trials were run on one Nvidia GeForce RTX 4070Ti with the hyperparameters shown in Table VI.

Parameter	Value
Learning rate	1e-6
Epochs	100
Batch size	8

TABLE VI
RESUNET TRAINING PARAMETERS

The data was pre-processed in the same manner as for SegFormer. Training completed on average in 75 minutes. Both the training and evaluation results are detailed in Figure 16 and the final evaluation metrics are shown in Table VII. Some example prediction masks generated by the model are shown in figure 17.

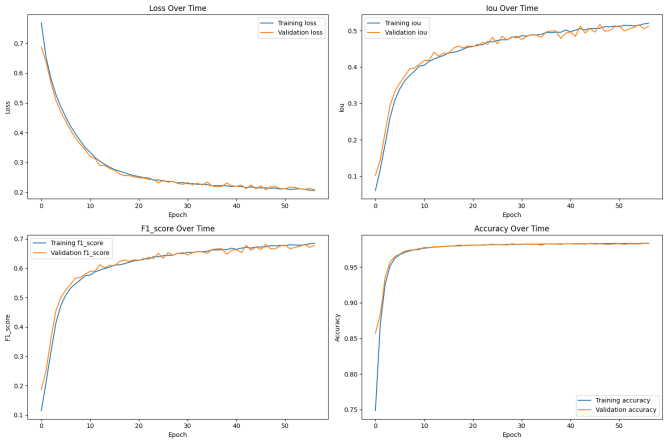

Fig. 16. ResUNet Training and Validation Metrics

Fig. 17. ResUNet Prediction Examples

In the bottom example of figure 17, we observed the model properly predicting additional parts of the roadway mask that are not part of the true mask but are visible within the color image. This case is likely due to the heuristics we used when preprocessing our data causing some smaller roads that did

not meet the expected width requirements to be pruned from the masks.

In figure 18 we can see the model struggles to capture the majority of the irregularly shaped roadway occluded by the overhead trees. Identifying occluded roadways, especially irregularly shaped ones is a more specialized problem that might require additional bands of imagery like IR or a specialized dataset specifically formulated for this problem.

In figure 19 we can see that the model predicts the roadway nearly perfectly but also classifies the driveway of the property as part of this roadway. This particular issue is difficult to solve given both can appear like roadways from overhead images with the only distinctions being the presence of a building or that the road ends.

Metric	Value
IOU	0.499
F1	0.665

TABLE VII
RESUNET EVALUATION RESULTS

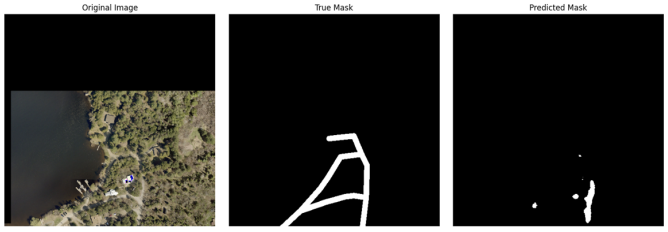

Fig. 18. ResUNet Error Analysis: Occluded Culdesac

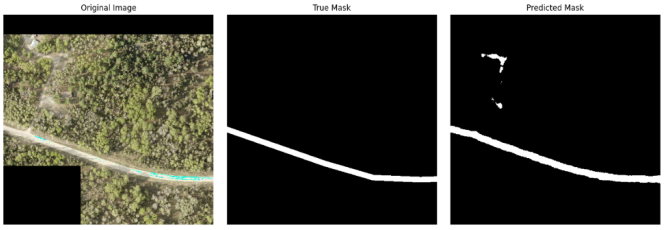

Fig. 19. ResUNet Error Analysis: Driveway False Positive

V. BONUS FEATURES

A. Potential Applications: Real-time Road Damage Detection.

We have developed a proof-of-concept application that demonstrates the usage of a fine-tuned segmentation model for real-time road detection and potential identification of road damage.

The application is written in python and uses Matplotlib to draw the main canvas and process user input. The main screen is split into 3 views 20:

1) Aerial view, meant to simulate the UAV/drone point of view.
2) Known Roads view, which shows the existing road network from the center line lables

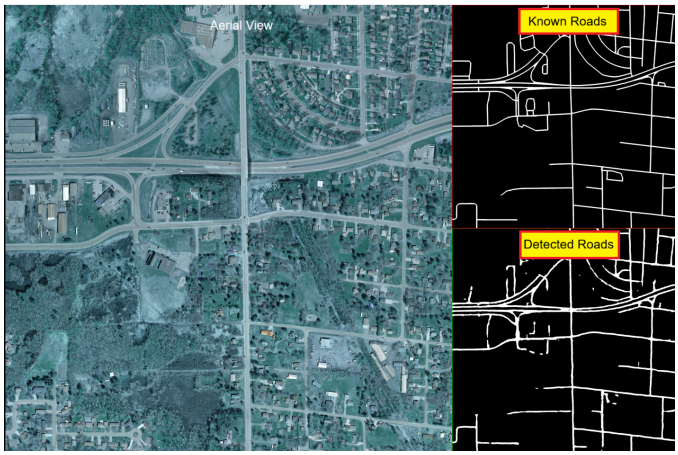

Fig. 20. Road Detection Functionality

3) Detected Roads view, which shows the real-time extracted road network using the fine-tuned model.

The app allows the user to move around the area to simulate the drone surveillance. As the drone moves into the new area using the cursor keys, the app performs real-time inference on the newly visible image tiles and updates the Detected Roads view.

The app also allows the user to simulate road damage (e.g. flooding, blockage, etc) by placing a red mark over the road (mouse left-click) 21. The affected tile is rerun through the model to compute the new road prediction which shows an interruption in the detected road. An overlay mode can be activated (keypress 'o') that places the detected road view on top of known road view and highlights the differences. This highlights the potential areas of investigation for a human operator.

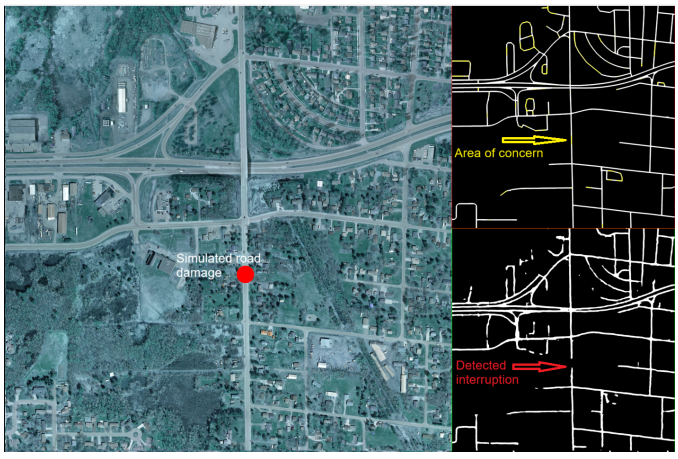

Fig. 21. Road Damage Detection

The app uses a fine-tuned SegFormer-b3 model. In order to get the best quality predictions for the purposes of the POC, we fine-tuned the model on the data used in the app. However, this is not an unrealistic step, since it is conceivable that in the real world, the model could be specific to a particular geographical area. We have benchmarked the inference at 1.15s per 512x512 tile on a modern multicore desktop CPU and 0.15s/tile on a

NVIDIA GeForce GTX1660 GPU with 4GB RAM. Thus we can see that even with a limited computational power available, a UAV could easily compute road predictions in the field, find areas of concern and relay the original aerial image to a human operator who could further evaluate the conditions.

As a potential future direction, the model could be trained to recognize various types of damage and obstruction automatically. This would require a lot of more data, including labeled post-damage data. Alternatively, with some additional preprocessing perhaps the model could learn to classify road damage in an unsupervised way: having found potential areas of concern (e.g. interrupted road) after supervised training, the model could be trained to classify the damaged areas using K-means clustering algorithm. Having learned to classify road damage, a drone operating such a model could then use this information to e.g. automatically re-route emergency vehicles.

B. Road Network Extraction

Another feature that was implemented was producing a road network, represented as a graph with the nodes being intersections and the edges representing the roads. This was implemented by first morphologically thinning the mask produced by one of the models. This thinning produces an output with a similar shape as the original map but the width of the road masks are now 1. An example of this output, referred to as a skeleton, is shown in figure 22

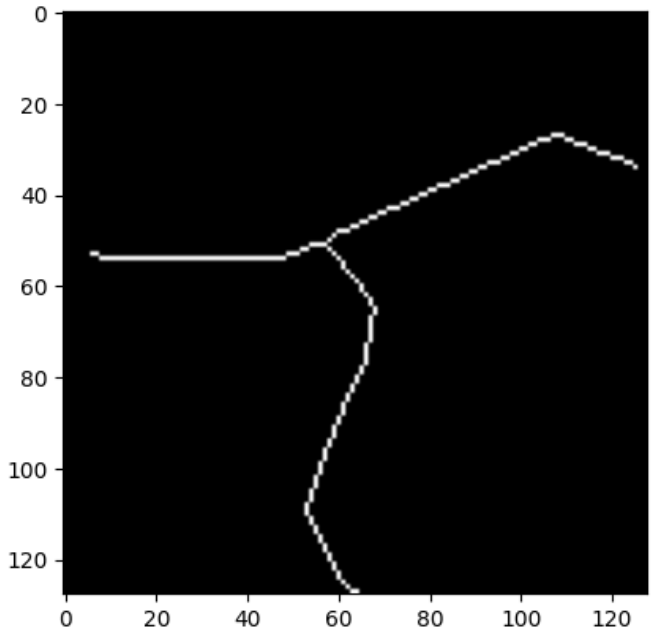

Fig. 22. Road Mask Skeleton Example

The python library ”sknw” was used to turn the resulting skeleton into a graph representing the road network of the image. In the current implementation there are no unique weights assigned to the roads, however this is an addition that would be useful in the case of routing or re-routing emergency vehicles. As a drone is flying overhead, it could increase the weights of roads that have accidents, delivery trucks blocking a lane of traffic, or other obstacles that could hinder the emergency vehicles ability to navigate that road efficiently. An example of the extracted network plotted back on to the original image can be seen in Figures 23 and 24.

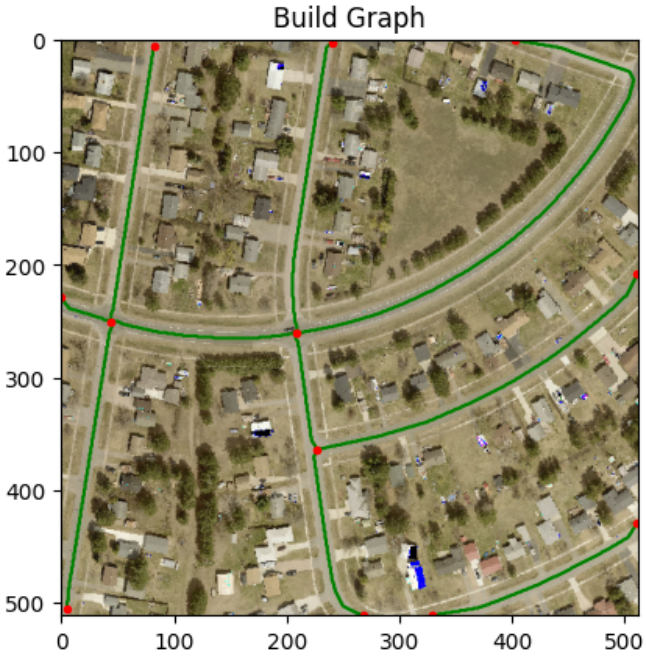

Fig. 23. Road Network Extraction Example on Label Mask

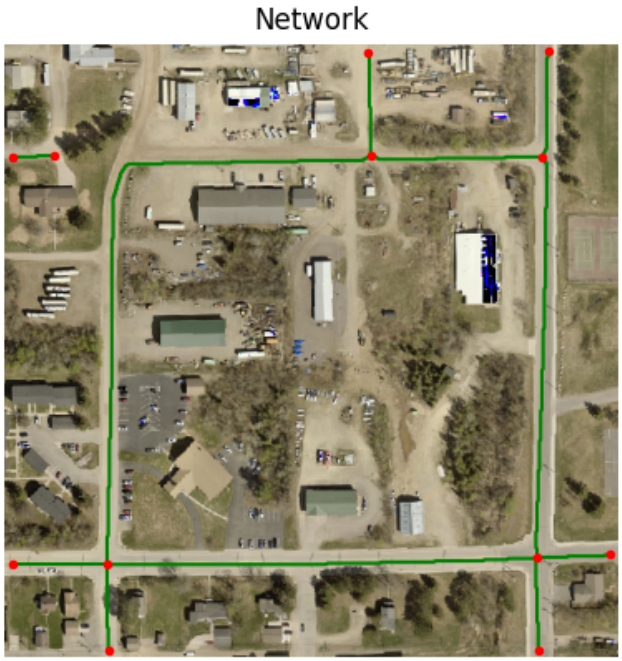

Fig. 24. Road Network Extraction Example on U-Net Mask

VI. Conclusions

We have demonstrated that a variety of models could be used for the task of road extraction from aerial imagery. The models are small and efficient, and their performance is acceptable at least for urban and suburban imagery. The models had more trouble with country roads, and we think that this could be alleviated by changing the task to be a multi-class segmentation with different classes of roads. We have demonstrated how a fine-tuned model could be used for real-time road extraction using a drone. Further, we demonstrated potential ways of how such a system could be used for road obstruction identification and route planning.

As a result of the data augmentation experiment, we realized the importance of normalizing the training images and masks.

VII. Summary of Contributions

All the code created for this project (with the exception of some training notebooks, as referenced below) can be found here

A. Eli Schlossberg

- Gathered and curated data into training dataset using a number of geoprocessing techniques
- Utilized DiffusionSat [9] to generate a novel dataset of disaster data from the existing road network data

B. Andrei Solodin

- Generated HuggingFace dataset using the raw dataset: Dataset
- Created a notebook to train and evaluate SegFormer on variaous datasets, e.g. SegFormer Training/Eval
- Created a demo application to demonstrate the potential usage of a fine-tuned model for a real time road/damage detection: Demo.py

C. Kyle Anthes

- Wrote custom load dataset script to read in custom GeoTIFF dataset, calculate summary statistics, and visualize outlier samples
- Implemented ResUNet notebook with custom evaluation metrics

D. Jack Swanberg

Implemented a U-Net model in PyTorch, then trained it on our dataset and recorded training, validation, and test metrics reported above. This was the first time I had implemented a model in PyTorch so it was a valuable learning experience in getting more familiar with such a well used library. I also implemented road-network extraction, taking the model output road masks and turning it into a graph of the road network shown in the image.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[2] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[4] S. Wu, C. Du, H. Chen, Y. Xu, N. Guo, and N. Jing, "Road extraction from very high resolution images using weakly labeled openstreetmap centerline," *ISPRS International Journal of Geo-Information*, vol. 8, no. 11, 2019. [Online]. Available: https://www.mdpi.com/2220-9964/8/11/478

[5] D. Feng, X. Shen, Y. Xie, Y. Liu, and J. Wang, "Efficient occluded road extraction from high-resolution remote sensing imagery," *Remote Sensing*, vol. 13, no. 24, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/24/4974

[6] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021. [Online]. Available: https://arxiv.org/abs/2105.15203

[7] HuggingFace, "Huggingface: Segformer." [Online]. Available: https://huggingface.co/docs/transformers/model_doc/segformer

[8] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, p. 749–753, May 2018. [Online]. Available: http://dx.doi.org/10.1109/LGRS.2018.2802944

[9] S. Khanna, P. Liu, L. Zhou, C. Meng, R. Rombach, M. Burke, D. B. Lobell, and S. Ermon, "Diffusionsat: A generative foundation model for satellite imagery," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=I5webNFDgQ

[10] M. Karaduman, A. Çınar, and H. Eren, "Uav traffic patrolling via road detection and tracking in anonymous aerial video frames," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 675–690, 2019. [Online]. Available: https://doi.org/10.1007/s10846-018-0954-x

[11] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[12] R. B. A. et. al., "Computer vision-based model for detecting turning lane features on florida's public roadways from aerial images," *Transportation Planning and Technology*, vol. 0, no. 0, pp. 1–32, 2024. [Online]. Available: https://doi.org/10.1080/03081060.2024.2386614

[13] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.

[14] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X98000199

[15] F. S. A. United States Department of Agriculture, "National agriculture imagery program (naip)," https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/, 2024, accessed: 2024-09-29.

[16] S. Choudhury, A. R. Kreidieh, I. Tsogsuren, N. Arora, C. Osorio, and A. Bayen, "Scalable learning of segment-level traffic congestion functions," 2024. [Online]. Available: https://arxiv.org/abs/2405.06080

[17] Y. Zhang, B. Howe, S. Mehta, N.-J. Bolten, and A. Caspi, "Pathwaybench: Assessing routability of pedestrian pathway networks inferred from multi-city imagery," 2024. [Online]. Available: https://arxiv.org/abs/2407.16875

[18] M. Jamal and A. Panov, "Maneuver decision-making with trajectory streams prediction for autonomous vehicles," 2024. [Online]. Available: https://arxiv.org/abs/2409.10165

[19] C. Reed, C. Tatsch, J. N. Gross, and Y. Gu, "Autonomous hiking trail navigation via semantic segmentation and geometric analysis," 2024. [Online]. Available: https://arxiv.org/abs/2409.15671

[20] N. Kumar and M. Raubal, "Applications of deep learning in congestion detection, prediction and alleviation: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 133, p. 103432, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004241

[21] A. Singh, M. Z. U. Rahma, P. Rani, N. k. agrawal, R. Sharma, E. Kariri, and G. Aray, "Smart traffic monitoring through real-time moving vehicle detection using deep learning via aerial images for consumer application," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024.

[22] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude uav datasets: A survey," *Image and Vision Computing*, vol. 104, p. 104046, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885620301785

[23] M. Rahnemoonfar, T. Chowdhury, and R. Murphy, "Rescuenet: A high resolution uav semantic segmentation dataset for natural disaster damage assessment," *Scientific Data*, vol. 10, no. 1, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41597-023-02799-4

[24] J. S. Andrei Solodin, Eli Schlossberg and K. Anthes, "Digital orthoimagery, twin cities, minnesota, spring 2016, 1-ft resolution," 2016. [Online]. Available: https://huggingface.co/datasets/asolodin/orthoimagerytwincitiespos)

[25] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, University of Toronto, 2013.

[26] C. Robinson, I. Corley, A. Ortiz, R. Dodhia, J. M. L. Ferres, and P. Najafirad, "Seeing the roads through the trees: A benchmark for modeling spatial dependencies with aerial imagery," 2024. [Online]. Available: https://arxiv.org/abs/2401.06762