Road segmentation made simple: a practical comparison of segmentation models and post-processing techniques

Júlia Alves Porto[1*], Apostolos Ziakopoulos[1], George Yannis[1]

1. National Technical University of Athens, Greece
*julia_porto@mail.ntua.gr

Abstract

Segmentation models are pixel-wise classifications of an image, dividing it into pre-defined classes. Road segmentation, i.e., a binary segmentation of road and background pixels, is a powerful technique, especially useful for mapping areas with scarce data sources or as a backbone for road safety modelling, providing road data for further analysis. The unique road network characteristics, such as format, color and infrastructure constraints, allow for tailored model development. This study evaluates four state-of-the-art models: LinkNet, U-Net++, GCB-Net, and DiResSeg, alongside post-processing techniques including clustering, thinning, smoothing, and grouping. As expected, LinkNet delivers high accuracy with relatively fast training, even on large datasets. In contrast, the complex architecture of U-Net++ results in significantly longer training times. Among models specifically designed for road segmentation, the convolutional kernel size appears to impact computational demand more than it does predictive performance. As for the post-process techniques, even the simplest ones are valuable to reduce noise, although the high level of confidence from the original models makes it difficult to differentiate noise from actual disconnected road segments. This work offers a practical comparison of accessible techniques to aid researchers in building efficient segmentation pipelines for road safety applications.
Keywords: *Road segmentation, Post-processing, Aerial imagery, LinkNet, U-Net, GCB-Net, DiResSeg, Graph models.*

1. Introduction

Automated road extraction has been a topic of research for many years. What first started as a complex segmentation process based on universal properties of road sections, such as very gradual changes of width and direction, long extensions and almost constant gray level (Barzohar & Cooper, 1996), has reached advanced levels with the proliferation of deep learning techniques and the publication of large publicly available labeled datasets (Chen et al., 2022; Lian et al., 2020).

Road extraction can be subdivided as a two-step process: road area and road centerline extraction. Segmentation of a road area has been successfully applied using different deep learning techniques architectures, such as graph convolutional network (Cui et al., 2021), deep neural network (Li et al., 2021), residual network (Mattyus et al., 2017) and convolutional neural networks (Costea & Leordeanu, 2016; Manandhar et al., 2019). After locating potential road areas, the centerline can be extracted using thinning algorithms. Thinning can be done both by traditional machine learning or by deep learning-based methods (Chen et al., 2022).

Among the datasets used for model training, the most popular ones, likely due to open access availability, are the Massachusetts roads dataset (Mnih, 2013), the DeepGlobe Extraction Challenge (Demir et al., 2018) and the SpaceNet road dataset (Etten et al., 2018). All three datasets provide aerial images with high resolution and their corresponding labeled road segments in a binary mask format: road pixels are classified as 1 and background pixels, as 0.

In this research, we compare the performance of four state-of-the-art segmentation models on the Deep Globe Extraction Challenge dataset (Demir et al., 2018): (1) LinkNet (Chaurasia and Culurciello, 2017.), (2) U-Net++ (Z. Zhou et al., 2018), (3) GCB-Net (Zhu et al., 2021) and (4) DiResSeg (Ding & Bruzzone, 2021). The dataset was selected due to its open availability on the Kaggle platform, eliminating the need for further registration. As a pre-processing step, we explore using different image sizes as training inputs. We also compare the models performance in terms of speed of training and convergence, completeness and correctness metrics using different input image and kernel sizes.

After training the segmentation models, different post-processing techniques were applied to the output for performance improvement. The techniques used were chosen based on previous literature findings: (Laptev et al., 2000) cite the use of edge detection followed by edge grouping as a common post-processing approach; (Mokhtarzade et al., 2008) use a fuzzy shell clustering to vectorize a road raster map identified through neural networks; (Mattyus et al., 2017) use graph vectorization to improve road mapping results. The specific methods used during this research are described in the Methods section.

Results indicate that LinkNet is a strong model, achieving high evaluation metrics with fast convergence before performance plateaus. DiResSeg produced comparable results but required more training epochs and longer processing time. In contrast, the remaining two architectures fell short within our training set. Generally, the post-processing techniques tested were effective in reducing noise; however, they occasionally misclassified truly disconnected road segments as noise, leading to a reduction in recall.

Following this Introduction, the remainder of this paper is organized as follows: Section 2 covers the Materials and Methods, highlighting the data and models used; Section 3 covers the Results and Discussion; Section 4, Conclusion, covers the final thoughts and suggestions for future work development.

2. Materials and Methods

2.1 Data Preparation

Road extraction models were trained using the DeepGlobe Extraction Challenge database (Demir et al., 2018). The dataset contains 6226 aerial images with their respective labeled masks, and 2.344 aerial images without masks. Each image has a resolution of 0.5 meters per pixel and pixel-size of 1024 x 1024. The labeled part of the dataset was subdivided into three datasets: train, validation and test datasets; each with respectively 4604, 872 and 750 (approximately 74:14:12) non-overlapping images, randomly selected using a defined seed number.

The dataset was pre-processed using a modified version of the Notebook uploaded on Kaggle by Ashwath (2021). Instead of using the one-hot-encode format, both images and masks were normalized to [0, 1] intervals. Also, two resizing options were added: using built-in OpenCV functions for the direct resize of the images, with the corresponding loss of resolution; and cropping the images for the desired size, thus increasing the number of images in the dataset. Augmentation was also increased relative to the referenced notebook to include: the possibility of 90 degrees flips, blurring from 3 to 7-pixel kernels, and adjustments on the brightness contrast.

All steps of the code-based analyses were conducted in Python, and the main libraries used were OpenCV, PyTorch and Segmentation Models PyTorch (Iakubovskii, 2019). The models were trained on GPU NVIDIA GeForce RTX 2080.

2.2 Model training

LinkNet is based on a residual connection encoder-decoder architecture, where the results of each encoder block is added to the corresponding layer on the decoder-block, allowing to maintain spatial information that otherwise would have been lost in the downsizing initial part (Chaurasia & Culurciello, 2017). Also, they use residual blocks inside their encoder blocks.

U-Net was initially built for biomedical image segmentation (Ronneberger et al., 2015). It is based on symmetrical encoder-decoder paths that allow for precise localization of the learned features. U-Net++ is a nested U-Net architecture that uses convolutional and residual blocks when connecting the outputs of each encoder layer to the corresponding decoder layer, in order to bridge the semantic gap between the feature maps (Z. Zhou et al., 2018).

GCB-Net (Zhu et al., 2021) has a similar structure to LinkNet, using encoder-decoder architecture with residual blocks. The authors added a step at the end of each residual block which is a global context information incorporating block (GCA block). They also used a bottleneck between the encoder and decoder with multi-parallel dilated convolutional layers.

DiResSeg is a segmentation model that combines the U-Net and FCN (Long et al., 2015) architectures, with the goal of increasing completeness of the road topology regardless of losing information about road boundaries (Ding & Bruzzone, 2021). They use a layer-rearranged version of Resnet to compose the encoder and a simple decoder with three serial deconvolutional layers, using an asymmetrical arrangement for their model.

Figure 1 shows a simplified representation of each model's architecture, where the blue layers represent each encoder block; the orange layers represent the decoder blocks; the green later represents the bottleneck block and the light blue represents the nested layers. Readers are invited to refer to the original papers for the full description of the encoder and decoder blocks used for each layer.

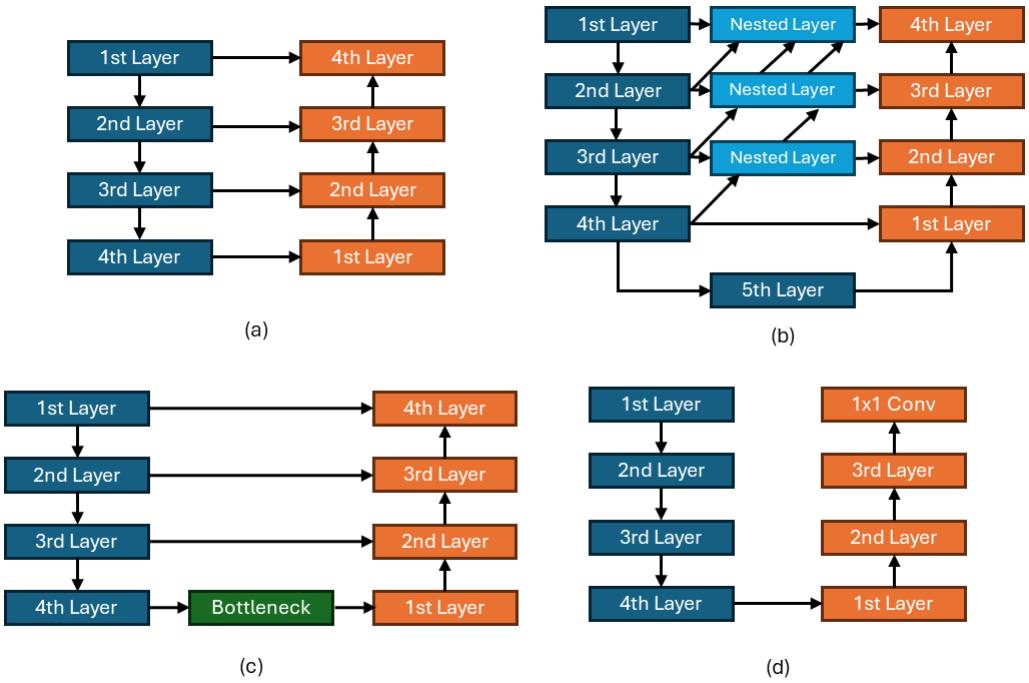


Figure 1: *Simplified representation of each model architecture: a) LinkNet; b) U-Net; c) GCA; and d) DiResSeg.*

The model used as a backbone for the U-Net++ and LinkNet models, ResNet34 (He et al., 2015), consists on a residual network with 34 parameter layers, where a residual (shortcut) connection is made every two layers, increasing the output dimensions when needed.

Finally, optimizer and loss functions were not part of the comparison task pursued during this research. The optimizer used was the PyTorch Adam optimizer with 0.001 learning rate, and the loss function used was the Jaccard Loss, defined by Equation 1, and available through the Segmentation Models PyTorch open-source code (Iakubovskii, 2019). Training and validation logs were stored per epoch. Training was limited to either 24 hours; 100 epochs or 5 consecutive epochs with no IoU improvement on the validation dataset, whatever happens first.

$$JaccardLoss = 1 - \frac{|P \cap T|}{|P \cup T|} = 1 - \frac{\sum(P \cdot T)}{\sum(P + T - P \cdot T) + \varepsilon} \quad \textbf{(1)}$$

Where P corresponds to the predicted pixels, T to the target pixels and ε is a small constant to avoid division by zero.

2.3 Post-processing

The first method employed was region growing using the flood fill algorithm from the Scikit-image library. This technique identifies contiguous regions with similar intensity values starting from seed points, either strictly equal or within a defined similarity threshold. For this study, the algorithm was applied to a Gaussian-smoothed version of the segmentation output (σ = 2). The output image was divided into four quadrants, and up to 2 random pixels classified as road per quadrant, if any were identified within 1000 trials, were selected to be used as seed points. The threshold for region growth was defined as 0.3 considering pixels within the range [0, 1]. These values were selected by trial and error.

The second approach applied graph-based segmentation guided by Canny edge detection. Two implementations were tested: one using NetworkX for explicit graph construction and another using Scikit-image's region-labeling utilities. The method consists of four main steps: (i) a graph is built from the binary segmentation mask, where each predicted road pixel is kept as a node if connected to other road pixels from its 4-connected neighbors (up, down, left, right); (ii) the graph is segmented by identifying connected components, and only components larger than a defined threshold are retained, forming a refined mask that removes small, isolated predictions; (iii) the Canny edge map is used to sharpen boundaries, but only edge pixels that intersect with the retained components are preserved, avoiding noise from background edges; and (iv) morphological closing is applied using a n x n square kernel to fill small gaps and reconnect discontinuities in the segmented road network.

After empirically testing various thresholds, considering output performance and computational time, the Canny edge thresholds were defined as 100 and 200, the component size threshold for NetworkX library was defined at 500 pixels and the kernel-size for morphological closing was defined set to 5x5.

We also implemented an energy-based post-processed technique, inspired by the Conditional Random Fields (CRF) work from Krähenbühl and Koltun (2011). In this case, we used the probability map outcome from the last layer of each model as the prediction input. High-confidence pixels—those with predicted probabilities above 0.9 for the road class or below 0.1 for the background—were selected as markers for segmentation. To guide the propagation of these markers, we constructed a gradient map combining edge information from both the original input image and the model's probability map. Edge detection was performed using the Farid and Simoncelli (2004) derivative filters. From the resulting edge maps, we retained the top 80% of the most prominent edge pixels (by gradient magnitude) to emphasize strong structural boundaries. These components were then passed to the Scikit-image's random walker function, which treats segmentation as a diffusion process: the algorithm assigns labels to ambiguous

pixels by simulating a random walker that is more likely to remain within regions of similar intensity or gradient strength.

Figures 2 and 3 show visual examples of simplified applications for each of the methods described earlier.

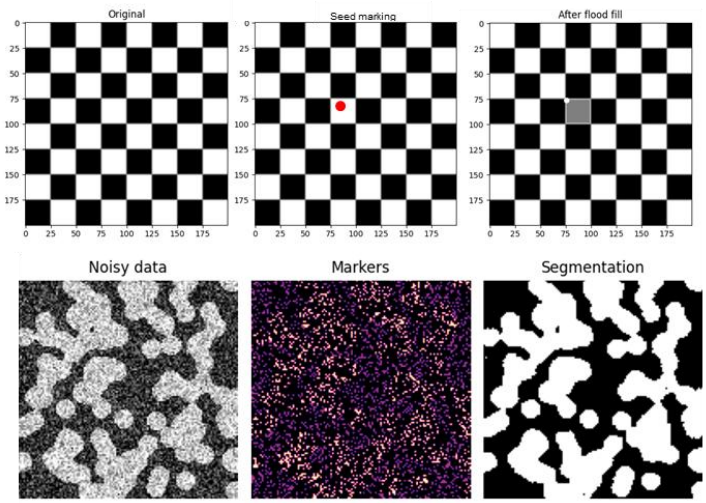


Figure 2: *Visual representation of region growth (top) and energy-based segmentation (bottom). Source: Scikit-image documentation*

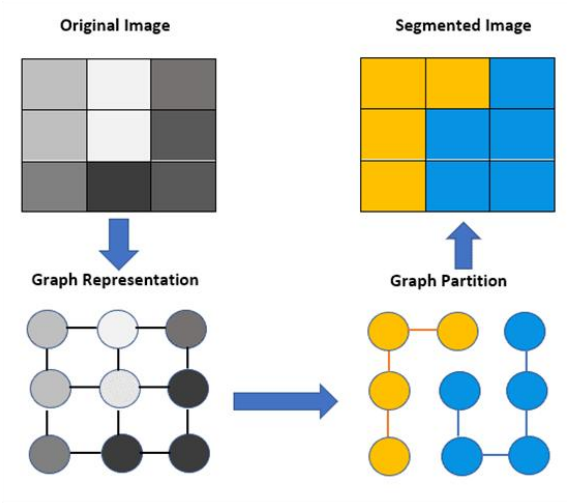


Figure 3: *Visual representation of graph-based segmentation. Source: Baeldung*

2.4 Metrics

The metrics used for comparison were Intersection of Union (IoU), precision (or correctness), recall (or completeness) and F1-Score, which are the most used metrics for road extraction segmentation models (Lian et al., 2020) and defined respectively by Equations 2 to 5:

$$IoU = \frac{P \cap T}{P \cup T} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad \textbf{(2)}$$

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(3)}$$

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(4)}$$

$$F1 = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad \textbf{(5)}$$

Where P corresponds to the predicted pixels and T to the target pixels. Accordingly, TP stands for True Positive, that is, the predicted pixels that correspond to target pixels, FP stands for False Positive and FN stands for False Negative. We also calculated the IoU considering a flexible threshold of a 3x3 matrix, considering that, for the target road mapping, small variabilities on the thresholds of the predicted roads don't compromise the results.

3. Results and Discussion

3.1 Model Training results

Table 1 summarizes the results for each model architecture across the tested combinations of input resolutions and kernel sizes.

A key observation is that combining large models with high-resolution inputs proved impractical, as these configurations demanded significant computational resources without yielding proportional performance gains. This was particularly evident in the case of U-Net++, whose complex architecture led to high memory consumption and extended training times. While U-Net++ showed promising results in preliminary tests, it became impractical to train within the imposed 24-hour limit. In contrast, LinkNet demonstrated exceptional efficiency, achieving high performance while converging quickly. DiResSeg also delivered strong results, though it required longer training periods to reach similar performance levels.

Interestingly, the impact of kernel size varied between models. For GCB-Net, smaller kernel sizes outperformed the larger ones originally proposed by Ding & Bruzzone (2021), despite our overall results not fully matching those reported in their study. DiResSeg, however, benefited from larger kernels, suggesting that optimal kernel configuration is architecture dependent.

Secondly, even though the same early stopping was applied across all models, some architectures plateaued faster than others. This suggests that the early stopping criteria may have been too conservative for this dataset, potentially limiting the full learning capacity of slower-converging models. Due to time constraints, alternative thresholds could not be tested. Models performed consistently better when they ran over more epochs.

Third, the idea of testing a flexible IoU threshold was to provide a more flexible penalization, but it seems to have overcompensated for noisy results. Traditional metrics such as IoU and F1-score proved more reliable for evaluating segmentation accuracy, even if they impose stricter penalties on slight errors, and recall and precision can be used to understand if the model is lacking connectivity or outputting noisy results. Overall, results were considered satisfactory. The original winner of the Deep Globe Challenge was (L. Zhou et al., 2018), who performed a 0.6342 IoU on their benchmark test dataset.

3.2 Post-processing results

To avoid redundancy and focus on maximizing performance, post-processing techniques were applied only to the best-performing model from each architecture, as shown in Table 2. These selected configurations were:

- LinkNet trained with the 512 x 512 cropped images;
- DiResSeg trained with 1024 x 1024 original images and 7 x 7 kernel size;
- GCB-Net trained with 1024 x 1024 original images and a 3 x 3 kernel size; and
- U-Net++ trained with the 512 x 512 cropped images.

Typically, the post-processing techniques identified in the literature are targeted at diminishing noise, but they have the downside of mis-identifying correct predictions as noise and, therefore, decrease overall IoU metrics.

Table 1: *Pre-processing results for each model*

Method	Pixel Size	Model	Kernel Size	Training Dataset Size	Training Time (min)	Total epochs	End of training	IoU	Precision	Recall	F1 Score	Flexible IoU (3 pixels)
None	1024	GCB	7	4604	-	-	24h Time Out	-	-	-	-	-
Resize	512	GCB	7	4604	962	26	No improvement	0.3063	0.3372	0.8001	0.4553	0.8194
Crop	512	GCB	7	6506	1313	27	No improvement	0.3984	0.4163	0.9100	0.5551	0.9154
None	1024	GCB	3	4604	709	36	No improvement	0.4474	0.4631	0.9332	0.6045	0.9395
Resize	512	GCB	3	4604	140	23	No improvement	0.3401	0.3912	0.7484	0.4883	0.7640
Crop	512	GCB	3	15048	1599	10	24h Time Out	0.4175	0.4402	0.9020	0.5734	0.9104
None	1024	DiResSeg	3	4604	206	21	No improvement	0.4585	0.6419	0.6335	0.6129	0.6558
Resize	512	DiResSeg	3	4604	62	11	No improvement	0.2613	0.5346	0.3487	0.3913	0.3769
Crop	512	DiResSeg	3	15093	288	34	No improvement	0.4798	0.6600	0.6519	0.6341	0.6628
None	1024	DiResSeg	7	4604	1114	48	No improvement	0.5736	0.7404	0.7529	0.7160	0.7406
Resize	512	DiResSeg	7	4604	1560	10	24h Time Out	0.3124	0.6541	0.3813	0.4511	0.4087
Crop	512	DiResSeg	7	15085	1626	8	24h Time Out	0.4751	0.5627	0.7709	0.6297	0.7809
None	1024	LinkNet	4	4604	216	14	No improvement	0.4084	0.4557	0.8342	0.5654	0.8323
Resize	512	LinkNet	4	4604	126	23	No improvement	0.4514	0.6667	0.6039	0.6010	0.6149
Crop	512	LinkNet	4	15119	258	28	No improvement	0.5895	0.7563	0.7417	0.7285	0.7792
None	1024	U-Net++	3	4604	-	1	24h Time Out	0.1514	0.7003	0.1727	0.2300	0.5271
Resize	512	U-Net++	3	4604	455	28	No improvement	0.3598	0.7100	0.4297	0.4955	0.4637
Crop	512	U-Net++	3	15057	1917	2	24h Time Out	0.4167	0.5061	0.7369	0.5716	0.7437

Table 2: *Post-processing results for best performing models*

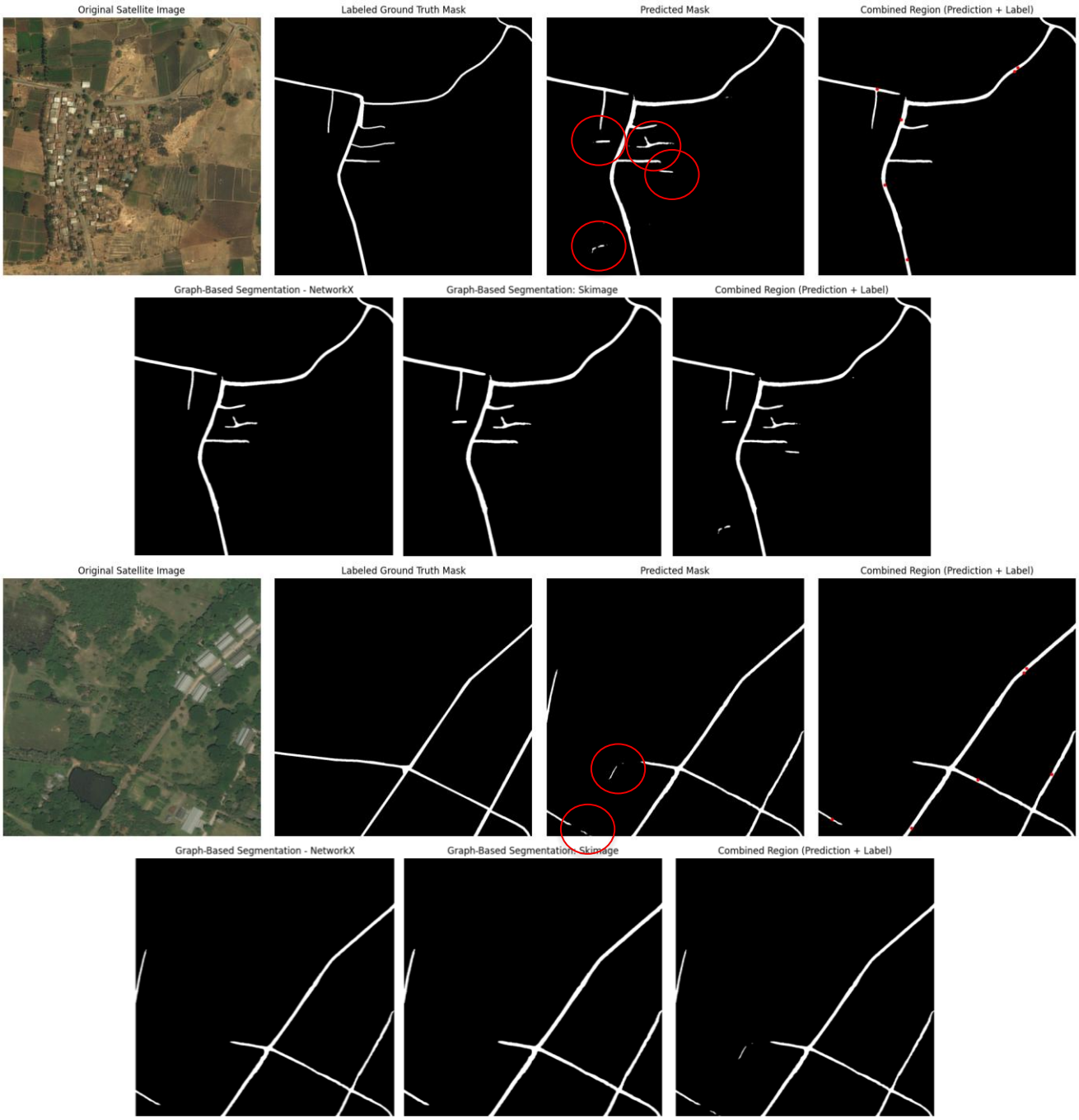
Segmentation Model	Post-Processing Technique	Main Library	IoU	Precision	Recall	F1
LinkNet	-	-	0.5895	0.7563	0.7417	0.7285
LinkNet	Region growth	Skimage	0.5622	0.8086	0.6882	0.6530
LinkNet	Graph-Based Segmentation	Networkx	0.6027	0.8366	0.7108	0.6933
LinkNet	Graph-Based Segmentation	Skimage	0.4049	0.7127	0.5471	0.5071
LinkNet	Energy based	Skimage	0.5895	0.8401	0.6926	0.6811
DiResSeg	-	-	0.5736	0.7404	0.7524	0.7160
DiResSeg	Region growth	Skimage	0.5607	0.7330	0.7524	0.6484
DiResSeg	Graph-Based Segmentation	NetworkX	0.6355	0.8074	0.7706	0.7266
DiResSeg	Graph-Based Segmentation	Skimage	0.4159	0.6804	0.5770	0.5199
DiResSeg	Energy based	Skimage	0.6115	0.8155	0.7316	0.7056
GCB	-	-	0.4474	0.4631	0.9332	0.6045
GCB	Region growth	Skimage	0.4535	0.4917	0.9054	0.5626
GCB	Graph-Based Segmentation	Networkx	0.4479	0.4714	0.9358	0.5605
GCB	Graph-Based Segmentation	Skimage	0.2102	0.2988	0.4841	0.2852
GCB	Conditional Random Fields	Skimage	0.4295	0.4509	0.9374	0.5425
U-Net++	-	-	0.4167	0.5061	0.7369	0.5716
U-Net++	Region growth	Skimage	0.4235	0.5752	0.7208	0.5228
U-Net++	Graph-Based Segmentation	NetworkX	0.4169	0.5106	0.7927	0.5241
U-Net++	Graph-Based Segmentation	Skimage	0.2556	0.4034	0.5480	0.3471
U-Net++	Energy based	Skimage	0.3937	0.4779	0.7982	0.5023

Although region growing shows clear potential for noise reduction, a simple seed-based flood approach is limited in its ability to distinguish between genuine road segments and noise or short disconnected elements, often leading to reduced overall performance. However, the technique holds potential for future algorithm improvement when combining region growth with intelligent seed selection or adaptive growth strategies, potentially guided by artificial intelligence or learning-based heuristics. These specificities in image segmentation are particularly researched within the medical and biology fields (Chen et al., 2024; Rasi & Deepa, 2022).

Graph-based segmentation, as implemented in this study, produced mixed results: one method surpassed the original model outputs in terms of connectivity and precision, while the other underperformed significantly. In our implementation, the NetworkX-based method built a pixel-wise graph using direct connectivity to identify consistent road segments. In contrast, the Scikit-Image approach approximated connectivity using region adjacency graphs based on similarity, which proved less effective in this context. The strong topological structure of road networks supports the use of graph-based techniques, which aligns with indications that Graph Neural Networks have great potential for road extraction, rather than relying on image information by itself (Lian et al., 2022).

The energy-based technique applied in this study presented contradictory results. The imbalance between the confident markers delivered from the output probability maps and the noisy input images used to guide the energy labeling led to a consistent increase in precision, as the algorithm effectively suppressed false positives, but at the cost of a corresponding decrease in recall. As a result, overall segmentation quality measured by IoU and F1-score remained largely unchanged. While this approach conceptually resembles graph-based segmentation — both modeling label propagation across connected structures — the reliance on noisy input gradients limits its standalone effectiveness, although, as with all the other techniques mentioned, it gives ground for further algorithm improvement.

Figure 4 brings a visualization of results for each model and post-processing technique, in the following order: from top to bottom, representing respectively the LinkNet, DiResSeg, GCB and U-Net++; from left to right, there is the original input image, the labeled ground truth mask, the predicted mask without any post-processing step, the results for the region growth technique, the results for the NetworkX graph-based technique, the results for the Scikit-Image graph-based technique and the results for the energy-based technique. On the predicted mask, the regions filtered out by the post-processing techniques are highlighted by the red circles.



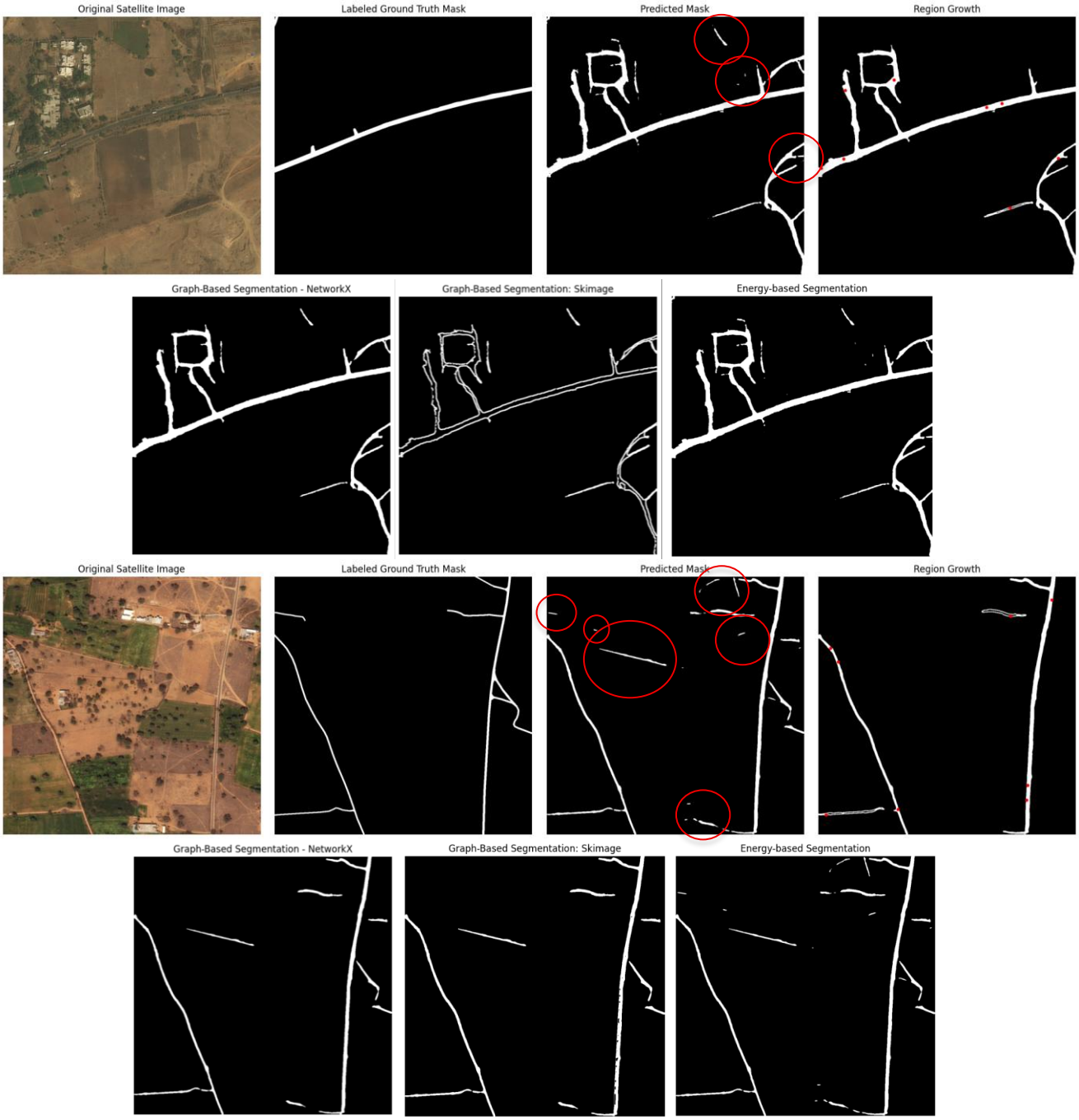


Figure 4: *Full framework model outputs visualization.*

Despite performance differences, all the algorithms tested are capable of extracting meaningful information about road presence and its network. These outputs can be used for different applications, such as detecting unmapped roads, analyzing road network density and complexity, or even converting outputs into vector data for geometric analysis and safety assessment. Automatic quantification of horizontal curvature, for example, an attribute that has been considered as a proxy variable for crash or conflict occurrence in many studies (Ziakopoulos et al., 2021; Saleem & Persaud, 2017; Gooch et al., 2016), could enable faster and more scalable safety assessments. Monitoring pavement condition through aerial footage is also a common approach (Outay et al., 2020), with road segmentation serving as a valuable pre-processing step for targeting specific analysis sites (Ranjbar et al., 2023). Another possible application for the field of road safety is post-disaster analysis based on aerial image (Sebasco & Sevil, 2022).

4. Conclusions

This research conducted a comprehensive comparison of state-of-the-art segmentation models, including those specifically designed for road segmentation and others developed for general-purpose semantic segmentation. All models achieved satisfactory results on a highly challenging dataset, with Intersection over Union (IoU) scores approaching those of the original winner (L. Zhou et al., 2018) of the DeepGlobe Road Extraction Challenge—the benchmark used in this study.

A series of post-processing techniques inspired by approaches in the literature were applied to the segmentation outputs. These techniques tended to improve either completeness (recall) or correctness (precision), but not both simultaneously, although graph-based segmentation was successfully able to increase overall IoU and F1-Score metrics.
Before applying post-processing, various kernel sizes and input image resolutions were evaluated, along with the integration of a frozen pre-trained backbone to encourage earlier convergence. Among these factors, the number of training epochs had the most pronounced effect on model performance. This finding suggests that lightweight models, which enable faster training and hence more epochs within a fixed time budget, are preferable—particularly in practical scenarios where faster iteration and deployment are desired.

The best Intersection of Union score achieved in our experiments was DiResSeg (Ding & Bruzzone, 2021), trained with a kernel size of 7 and using the original image dimensions, with its output improved by the graph-based segmentation post-processing step. The model reached convergence, during the training phase, in 48 epochs over 1114 minutes, and reached an IoU of 0.6355 and F1-Score of 0.7266 after the post-processing step. However, the best F1-Score was achieved by the LinkNet model (Chaurasia & Culurciello, 2017) trained with the cropped – therefore, larger – dataset, without passing through any post-processing. The model reached convergence in 28 epochs over 258 minutes and achieved an IoU of 0.5895 and F1-Score of 0.7285.

As for the limitations of the research, it should be noted that only one dataset was used for both training and testing of the models. So, even though it is a very complete and challenging dataset, results might present some bias. Also, the models were fully trained – both encoder and decoder weights, for all model architectures. The use of pre-trained weights is well advertised throughout the literature, especially in the pursuit of faster convergence and smaller training periods. Lastly, in real world applications, fine-tuning with the actual targeted data should be done, since groups of images might have peculiar characteristics, and therefore, although the consistency of some specific post-processing techniques is well noticeable, the actual metric values shouldn't be taken as a final value for other studies. In future research, we would also like to experiment with a more lenient early termination criterion. Future research should focus on improving the models even further, by either focusing on Graph Neural Network for road segmentation development or fusing the best performing models into a combined output.

5. Acknowledgements

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101119590.

6. References

Abhishek Chaurasia and Eugenio Culurciello, *Linknet: Exploiting encoder representations for efficient semantic segmentation*, 2017

Baeldung. (n.d.). *Image Processing: Graph-based Segmentation*. Retrieved May 14, 2025, from https://www.baeldung.com/cs/graph-based-segmentation

Barzohar, M., & Cooper, D. B. (1996). Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(7), 707–721. https://doi.org/10.1109/34.506793

Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. *2017 IEEE Visual Communications and Image Processing (VCIP)*, 1–4. https://doi.org/10.1109/VCIP.2017.8305148

Chen, Z., Deng, L., Luo, Y., Li, D., Marcato Junior, J., Nunes Gonçalves, W., Awal Md Nurunnabi, A., Li, J., Wang, C., & Li, D. (2022). Road extraction in remote sensing data: A survey. *International Journal of Applied Earth Observation and Geoinformation*, *112*, 102833. https://doi.org/10.1016/j.jag.2022.102833

Chen, Z., Yao, L., Liu, Y. *et al.* (2024). Deep learning-aided 3D proxy-bridged region-growing framework for multi-organ segmentation. *Sci Rep* **14**, 9784. https://doi.org/10.1038/s41598-024-60668-5

Costea, D., & Leordeanu, M. (2016). *Aerial image geolocalization from recognition and matching of roads and intersections* (No. arXiv:1605.08323). arXiv. https://doi.org/10.48550/arXiv.1605.08323

Cui, F., Feng, R., Wang, L., & Wei, L. (2021). Joint Superpixel Segmentation and Graph Convolutional Network Road Extration for High-Resolution Remote Sensing Imagery. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2178–2181. https://doi.org/10.1109/IGARSS47720.2021.9554635

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., & Raskar, R. (2018). DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 172–17209. https://doi.org/10.1109/CVPRW.2018.00031

Ding, L., & Bruzzone, L. (2021). DiResNet: Direction-Aware Residual Network for Road Extraction in VHR Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(12), 10243–10254. https://doi.org/10.1109/TGRS.2020.3034011

Farid, H. & Simoncelli, E. (2004). Differentiation of Discrete Multidimensional Signals. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*. 13. 496-508. 10.1109/TIP.2004.823819.

Gooch, J. P., Gayah, V. V., & Donnell, E. T. (2016). Quantifying the safety effects of horizontal curves on two-way, two-lane rural roads. *Accident Analysis & Prevention*, *92*, 71–81. https://doi.org/10.1016/j.aap.2016.03.024

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (No. arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

Iakubovskii, P. (2019). Segmentation Models Pytorch. In *GitHub repository*. GitHub. https://github.com/qubvel/segmentation_models.pytorch

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. arXiv. https://arxiv.org/abs/1210.5644

Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., & Baumgartner, A. (2000). Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, *12*(1), 23–31. https://doi.org/10.1007/s001380050121

Li, P., He, X., Qiao, M., Cheng, X., Li, Z., Luo, H., Song, D., Li, D., Hu, S., Li, R., Han, P., Qiu, F., Guo, H., Shang, J., & Tian, Z. (2021). Robust Deep Neural Networks for Road Extraction From Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(7), 6182–6197. https://doi.org/10.1109/TGRS.2020.3023112

Lian, R., Wang, W., Mustafa, N., & Huang, L. (2020). Road Extraction Methods in High-Resolution Remote Sensing Images: A Comprehensive Review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 5489–5507. https://doi.org/10.1109/JSTARS.2020.3023549

Long, J., Shelhamer E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

Manandhar, P., Marpu, P. R., Aung, Z., & Melgani, F. (2019). Towards Automatic Extraction and Updating of VGI-Based Road Networks Using Deep Learning. *Remote Sensing*, *11*(9), 1012. https://doi.org/10.3390/rs11091012

Mattyus, G., Luo, W., & Urtasun, R. (2017, October). DeepRoadMapper: Extracting Road Topology From Aerial Images. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Mokhtarzade, M., Zoej, M. J. V., & Ebadi, H. (2008). *AUTOMATIC ROAD EXTRACTION FROM HIGH RESOLUTION SATELLITE IMAGES USING NEURAL NETWORKS, TEXTURE ANALYSIS, FUZZY CLUSTERING AND GENETIC ALGORITHMS*.

Outay, F., Mengash, H. A., & Adnan, M. (2020). Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transportation Research Part A: Policy and Practice* **141**, 116-129. https://doi.org/10.1016/j.tra.2020.09.018.

Ranjbar, H., Forsythe, P., Fini, A. A. F., Maghrebi, M., & Waller, T. S. (2023). Addressing practical challenge of using autopilot drone for asphalt surface monitoring: Road detection, segmentation, and following. *Results in Engineering*, Volume 18. https://doi.org/10.1016/j.rineng.2023.101130.

Rasi, D., Deepa, S.N. (2022). Hybrid optimization enabled deep learning model for colour image segmentation and classification. *Neural Comput & Applic* **34**, 21335–21352. https://doi.org/10.1007/s00521-022-07614-6

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, *abs/1505.04597*. http://arxiv.org/abs/1505.04597

Saleem, T., & Persaud, B. (2017). Another look at the safety effects of horizontal curvature on rural two-lane highways. *Accident Analysis & Prevention*, *106*, 149–159. https://doi.org/10.1016/j.aap.2017.04.001

Sebasco, N. P., & Sevil, H. E. (2022). Graph-Based Image Segmentation for Road Extraction from Post-Disaster Aerial Footage. *Drones* **2022**, *6*(11), 315; https://doi.org/10.3390/drones6110315.

Zhou, L., Zhang, C., & Wu, M. (2018). D-LinkNet: LinkNet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 192–1924. https://doi.org/10.1109/CVPRW.2018.00034

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation* (No. arXiv:1807.10165). arXiv. https://doi.org/10.48550/arXiv.1807.10165

Zhu, Q., Zhang, Y., Wang, L., Zhong, Y., Guan, Q., Lu, X., Zhang, L., & Li, D. (2021). A Global Context-aware and Batch-independent Network for road extraction from VHR satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *175*, 353–365. https://doi.org/10.1016/j.isprsjprs.2021.03.016

Ziakopoulos, A. (2021). Spatial analysis of harsh driving behavior events in urban networks using high-resolution smartphone and geometric data. *Accident Analysis & Prevention*, *157*, 106189. https://doi.org/10.1016/j.aap.2021.106189