Road Segmentation System Using Deep Learning with focused Edge Detection - *A U-Net based approach using CamVid Dataset*

Manpreet Singh[1] **and Sandeep Singh Sandha**[2]

[1]**Khalsa College, Amritsar**
[2]**Punjab AI Excellence**

ABSTRACT

With the rise of autonomous vehicles and smart transportation systems, the ability to accurately detect and segment roads from video or images has become a critical requirement. Accurate road segmentation is a foundational task in autonomous driving, yet detecting precise boundaries remains a challenge. While popular deep learning models like U-Net perform well in segmenting roads, we encountered an unexpected challenge that the model consistently struggled at the edges, where clarity and precision matter most. At that point I learned that high accuracy doesn't always mean high reliability.

In this project, I designed a binary road segmentation system using the CamVid dataset and a U-Net architecture, known for its strength in pixel-wise classification via an encoder–decoder framework with skip connections. Initially, I converted multi-class segmentation masks into binary road masks and trained a U-Net model. Despite achieving high training and validation accuracy (97–98%), I noticed that the predictions often had blurry road boundaries, which could be critical in real-world autonomous navigation scenarios. These edge inaccuracies may seem minor on paper, but in real-world deployment such as lane following or obstacle avoidance, such errors can have significant consequences.

To address this limitation, I developed an edge-aware training strategy. By applying Canny edge detection to the ground truth masks, I identified boundary pixels and created a custom weight map—assigning higher weights (value = 5) to edge pixels and normal weights (value = 1) to others. Then I integrated this map into a modified Binary Cross-Entropy loss function, allowing the model to penalize errors near edges more heavily. This led to a substantial performance gain, improving the Intersection over Union (IoU) metric from 87% to over 92%, while maintaining strong accuracy.

The entire project was implemented using Python, TensorFlow, Keras, and OpenCV in Google Colab. The final model demonstrates both high accuracy and precise edge delineation, setting a solid foundation for future work in real-time road segmentation. All code and documentation will be made publicly available via GitHub to support reproducibility and further exploration.

Keywords: Road Segmentation, Deep Learning, U-Net, Semantic Segmentation, CamVid, Computer Vision, Autonomous Driving, Binary Segmentation, Convolutional Neural Networks(CNN)

INTRODUCTION

In recent years, the emergence of autonomous vehicles, smart transportation systems and AI-powered urban planning has pushed the demand for reliable computer vision models into the spotlight. Among the many perception tasks essential for these systems, road segmentation plays a foundational role. It enables self-driving cars to identify drivable surfaces, supports real-time navigation systems, and feeds data into larger frameworks for traffic analysis, infrastructure planning, and smart city development.

However, despite rapid advances in deep learning, achieving accurate and reliable road segmentation remains a persistent challenge, particularly when it comes to detecting precise road boundaries. In the real world, roads often blend into sidewalks, medians, vegetation, or shadowed areas, making the edges ambiguous. Misinterpreting these boundaries, even slightly, can lead to serious consequences in autonomous navigation, such as veering off course or misclassifying adjacent zones. Thus, it is not

enough for a model to perform well on average; it must also be highly sensitive to edge details, where safety and precision are critical.

This project focuses on binary road segmentation using the CamVid dataset, a well-established benchmark for understanding urban scene. I adopted the U-Net architecture, known for its ability to perform dense pixel-wise classification, to distinguish road pixels from the background. Initially, I preprocessed the dataset by converting its multiclass semantic masks into binary format, simplifying the segmentation task to "road vs. not-road".

The U-Net model showed impressive results in terms of overall performance, achieving approximately 97 – 98% accuracy. However, upon closer inspection, I observed a consistent problem: the boundaries of the segmented roads appeared soft and unclear. This issue persisted even when the rest of the segmentation was nearly perfect. The model excelled in detecting road regions but struggled to precisely delineate where the road ends, a critical factor in the deployment of such models in autonomous systems.

This led us to an important realization: high accuracy alone does not guarantee high reliability, especially when deployed in safety-critical applications. I recognized that conventional loss functions, like Binary Cross-Entropy, treat all pixels equally, regardless of whether they belong to large, clearly visible areas or thin, underrepresented edge regions. This creates an imbalance where models learn to favor dominant features while neglecting subtle yet essential ones.

To address this, I developed an edge-aware training strategy. By applying Canny edge detection to the ground truth masks, I extracted road boundary pixels and created a custom weight map. Edge pixels were assigned higher weights (value = 5), while all other pixels were given normal weights (value = 1), encouraging the model to pay more attention to those areas during learning. Then I integrated this weight map into a modified Binary Cross-Entropy loss function, amplifying the penalty for misclassifying edge pixels.

This refinement led to significant improvements. The Intersection Over Union (IoU) score increased from 87% to over 92%, while the model also produced sharper and cleaner edges, a qualitative improvement that goes beyond metrics. The complete project was developed using Python, TensorFlow, Keras, and OpenCV in a Google Colab environment and is fully reproducible through an open source GitHub repository.

In summary, this work contributes a practical solution to a real-world problem: enhancing the reliability of road segmentation models by explicitly guiding them to focus on boundary accuracy. While the approach is straightforward, its impact is meaningful, offering a simple yet effective way to make computer vision systems more dependable in autonomous navigation and intelligent urban systems.

METHODS AND BACKGROUND

To address the complex task of road segmentation in urban driving environments, this section outlines the foundational methods and decisions that shaped the development of the project, from data preprocessing to model architecture, training strategies and evaluation techniques. Each component was carefully chosen to ensure accuracy, efficiency, and practical relevance to real-world applications.

- **Model Architecture:** For this project, I chose the U-Net architecture as the backbone of our road segmentation model. Originally introduced for biomedical image segmentation, U-Net has gained widespread popularity in the field of semantic segmentation due to its ability to learn from relatively small datasets and still produce precise and pixel-level predictions.

  What makes U-Net particularly well-suited for our problem is its encoder-decoder structure with skip connections. The encoder (or contracting path) extracts high-level features by progressively reducing spatial dimensions using convolutional and max-pooling layers. This helps the model understand what is in the image.

  Meanwhile, the decoder (or expansive path) gradually upsamples the compressed feature maps to restore spatial resolution and locate where objects are within the image. The skip connections between corresponding encoder and decoder layers play a crucial role by passing fine-grained spatial information directly across the network. This is especially useful for segmentation tasks like ours, where preserving edge details is critical.

  Our model begins with an encoder that performs hierarchical feature extraction through a series of convolutional blocks. The implementation starts with three encoder stages, where each stage

comprises two convolutional layers followed by a max-pooling operation. The number of filters increases progressively from 16 $\rightarrow$ 32 $\rightarrow$ 64, helping the network capture increasingly abstract representations. To prevent overfitting and to encourage regularization, dropout layers were introduced, especially in the deeper layers where the model learns high-level abstractions.

At the bottleneck, the model employs two convolutional layers with 128 filters, both using ReLU activation and 'same' padding. This section of the model acts as the compressed knowledge base of the network, encapsulating both high-level semantics and spatial details crucial for precise segmentation.

The decoder path symmetrically reverses the encoder, using upsampling layers followed by concatenation with corresponding encoder outputs. These skip connections are fundamental to U-Net, as they allow the decoder to recover fine-grained details lost during downsampling. The decoder steps reduce filters from 64 $\rightarrow$ 32 $\rightarrow$ 16, bringing the feature map closer to the original input resolution. Finally, a 1×1 convolution with a sigmoid activation is used to generate a binary segmentation mask, marking the presence or absence of road pixels.

This architecture was chosen for several reasons:

- It provides excellent performance even with limited training data, due to its use of skip connections and its relatively compact design.
- It enables pixel-wise classification, which is critical for segmenting narrow road boundaries in complex scenes.
- It is computationally efficient, making it ideal for scaling in real-time applications—a direction I plan to pursue in the next phase of this project.

Implementing U-Net from scratch rather than using a pre-built version was an important part of my learning journey. It allowed me to deeply understand each component's role, from the convolutional layers to the upsampling blocks, and to adapt the architecture based on dataset-specific insights discovered during experimentation.

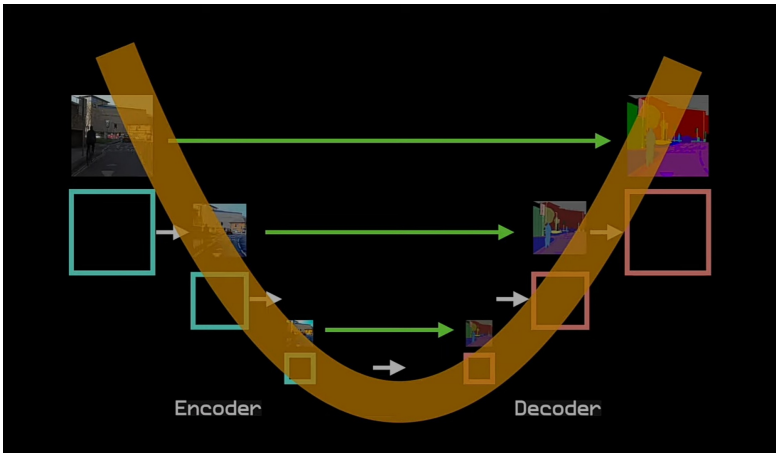

Figure 1. Visual representation of U-Net

- **Training Pipeline:** The training pipeline was carefully structured to facilitate robust binary road segmentation using the CamVid dataset. The dataset was sourced from Google Drive and contained RGB images alongside their pixel-level semantic labels. To streamline the classification task, label masks were converted to binary road masks by isolating the RGB color corresponding to the "road" class (i.e., [128, 64, 128]), and mapping these pixels to 1 (road) and all others to 0 (non-road). All images and corresponding masks were resized to a standard resolution of 256 × 256 pixels, and normalized to the 0–1 range to stabilize training. Preprocessing was implemented using OpenCV

and NumPy, and both images and masks were converted to NumPy arrays for compatibility with the deep learning pipeline. The dataset was then partitioned into training and validation subsets.

For model training, a U-Net architecture was employed due to its effectiveness in semantic segmentation tasks and its ability to capture fine-grained spatial details. The U-Net was trained using the binary cross-entropy loss function, which is well-suited for binary classification problems like road vs. non-road segmentation. The Adam optimizer was chosen for its adaptive learning capabilities and computational efficiency. Training was conducted over multiple epochs with real-time feedback using validation loss and accuracy. The batch size and number of epochs were set to ensure a balance between model convergence and computational feasibility. Training and evaluation were carried out in Google Colab, leveraging its GPU acceleration to expedite convergence and manage high-dimensional tensor computations.

To enhance the model's sensitivity towards road boundaries, where segmentation errors are often most costly, I incorporated a post-processing strategy that emphasizes edge detection. Specifically, I implemented an edge-weighted loss function by generating custom weight maps for each ground truth mask. Using the Canny edge detection algorithm, we assigned higher weights (value of 5.0) to edge pixels, while non-edge pixels retained a default weight of 1.0. This strategy forces the model to learn sharper distinctions between road and non-road regions, particularly along the critical borders where traditional loss functions might be too forgiving. The weight maps were generated through a custom function that converted binary masks into grayscale format, applied Canny edge detection, and created a weight matrix with elevated values for edge pixels. These maps were then passed as sample_weight during training, enabling the binary cross-entropy loss function to penalize misclassified edge pixels more strongly. This targeted supervision significantly improved the model's ability to delineate road boundaries in challenging scenarios.

- **Evaluation Metrics:** To evaluate the performance of the road segmentation model, multiple metrics were used. Accuracy was employed as a basic measure to quantify the proportion of correctly predicted pixels across the entire image. However, in semantic segmentation tasks—particularly those with class imbalance, accuracy can be misleading. To better understand model performance, the Intersection over Union (IoU) metric was calculated, which evaluates the overlap between the predicted road regions and the corresponding ground truth labels. This metric offers a more reliable representation of segmentation quality by penalizing both false positives and false negatives.

  In addition to quantitative metrics, a confusion matrix was used to analyze the distribution of predictions across classes, helping identify common misclassification patterns.

  For qualitative assessment, the model was also tested on self-captured road images, for which ground truth labels were not available. In these cases, performance was evaluated through visual inspection, comparing the predicted segmentation maps with actual road regions to understand how well the model generalized to unseen, real-world scenarios.

  During this visual inspection, it was observed that the model occasionally failed to capture fine road boundaries. This insight led to the introduction of an edge-weighted loss function, designed to assign higher importance to edge pixels during training. The impact of this post-processing enhancement was then evaluated using the same metrics—accuracy, IoU, and visual analysis—allowing for a meaningful comparison of results before and after applying the edge-focused strategy.

Difficulties Faced : During the course of this project, one of the primary challenges was sorting and pairing images with their corresponding masks to ensure accurate visualization and alignment. Even minor mismatches between an image and its mask could lead to misleading interpretations during training and evaluation. Another major difficulty arose while implementing the edge-weighted loss function. Applying the weighting logic correctly across all training and validation masks involved extensive image manipulation using NumPy. Managing the dimensionality of arrays, preserving data formats, and ensuring compatibility with the training pipeline were particularly demanding tasks. Overall, image preprocessing and manipulation using NumPy from resizing and normalizing to generating weight maps, proved to be the most challenging aspect of the project. This phase required repeated experimentation and debugging but it also significantly deepened my understanding of image-level operations and data handling in semantic segmentation tasks.

DATASET AND KNOWLEDGE SOURCES

The success of any deep learning-based segmentation task heavily relies on the quality and relevance of the datasets used. This section outlines the datasets employed in the project, along with their significance and associated preprocessing steps.

1. **CamVid:** The primary dataset used for this project was the CamVid (Cambridge-driving Labeled Video) dataset, which consists of annotated video sequences captured from urban driving environments. It offers pixel-level semantic segmentation labels for each frame, making it particularly suitable for road and background classification tasks and training the model with consistent and diverse driving scene examples.

   Preprocessing involved resizing all images and masks to a uniform resolution, normalizing pixel values, and converting multi-class labels into binary masks (road vs. non-road). This streamlined the data for binary segmentation and reduced computational complexity.

2. **Self-Clicked Images:** A small set of road images captured manually using a mobile phone camera. Although these images lacked annotated labels, they were used to test the model's performance in unseen, real-world conditions. Visual inspection of predictions on these samples helped identify edge segmentation issues, which led to the development of an edge-weighted loss strategy.

   The same preprocessing steps were applied to these images, including resizing, converting multi-class labels into binary masks and normalizing them to align with the input format required by the trained model.

3. **KITTI Dataset (Not used but relevant):** Contains urban and highway driving scenes with segmentation labels. If we integrate in future work, it could contribute to handling different road textures, vehicle types, and camera angles, enhancing the robustness of model.

4. **Cityscapes (Not used but relevant):** Offers high resolution urban street scenes with fine grained pixel-level annotations from multiple European cities. This dataset is well suitable for improving segmentation in complex, dense urban scenarios and can significantly benefit models targeting smart city applications.

EXPECTED BENEFITS AND APPLICATIONS

Highlight the broader value of your project.

- **Autonomous Vehicle Navigation:** Accurate road segmentation aids self-driving cars in detecting drivable areas, lane boundaries, and road edges, ultimately helps in enhancing safety and decision-making.
- **Advanced Driver Assistance Systems (ADAS):** It supports features like lane keeping, automatic braking, and collision avoidance by clearly distinguishing road surfaces, making everyday driving safer for everyone on the road.
- **Useful for City Planning and Road Monitoring:** Detailed road mapping can help urban planners and local authorities monitor road usage and optimize infrastructure development.
- **Real-time Traffic Monitoring:** Intelligent traffic surveillance cameras can identify which parts of the road are being used and help manage congestion or detect incidents quickly.
- **Augmented Reality for Navigation:** Enhances visual navigation applications by overlaying directional guidance on segmented road views in AR-based systems.

EXPERIMENTAL RESULTS

To evaluate the model's performance, we used key metrics: Accuracy, Intersection over Union (IoU), and the Confusion Matrix. These were analyzed before and after introducing an edge-weighted loss function to improve boundary prediction.

1. **Before applying Edge-Weighted Loss**, the model was trained with standard binary cross-entropy loss and it achieved:

Metric	Value
Training Accuracy	0.97
Validation Accuracy	0.96
Test Accuracy	0.96
Mean IoU	**0.87**

Table 1. Model performance before applying edge-weighted loss

Observation: While predictions were generally accurate, boundary regions (road edges) were often imprecise or inconsistent, especially in complex areas.

Figure 2. Example output from the trained model before applying edge-weighted loss.

2. **After applying edge-weighted loss**, we used a modified binary cross-entropy where edge pixels were given weight = 5, others = 1. This focused the model on learning fine road boundaries. Improved Metrics:

Metric	Value
Training Accuracy	0.98
Validation Accuracy	0.97
Test Accuracy	0.97
Mean IoU	**0.92**

Table 1. Model performance after applying edge-weighted loss

Observation: Predictions on both validation and self-captured images showed better edge sharpness and continuity.

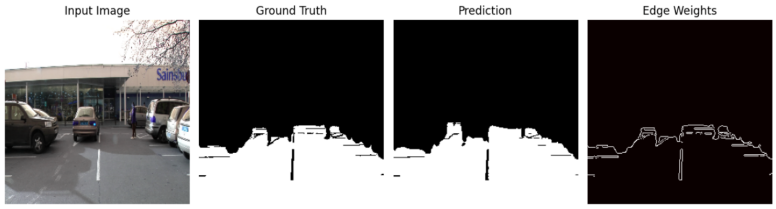

Figure 3. Example output from the trained model after applying edge-weighted loss.

3. **Confusion Matrix** (Post Edge-Weighted Loss):

   True Positives (TP): 1,775,114

   True Negatives (TN): 4,629,807

   False Positives (FP): 73,514

   False Negatives (FN): 75,165

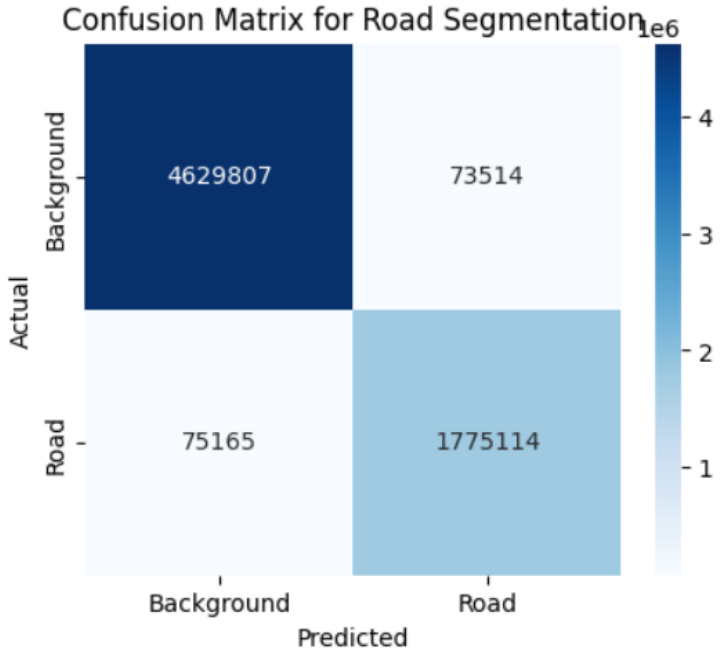

Figure 1. Final Confusion Matrix after end of the project

Conclusion: The low FP and FN rates reflect strong class separation and boundary precision, validating the model's reliability in real-world scenarios.

4. **Model testing on self clicked images:**

Figure 4. Example output from the trained model on self clicked image 1.

Figure 5. Example output from the trained model on self clicked image 2.

RELATED WORK

This section highlight key research efforts and datasets that influenced the design choices and direction of our road segmentation project. These foundational works provided architectural guidance, dataset selection support, and real-world validation of our methods.

- **U-Net for Biomedical Image Segmentation** The original U-Net architecture, introduced by Olaf Ronneberger et al. (2015), was designed for biomedical image segmentation but has since been widely adopted across other domains due to its ability to learn from relatively few images while maintaining spatial context through skip connections. Inspired by U-Net's success in capturing both low-level and high-level features, I adopted this architecture for segmenting roads from urban scenes. The encoder-decoder structure with skip connections helped my model preserve road boundary details, which is crucial for applications like autonomous driving.

- **CamVid Dataset for Urban Scene Understanding** The Cambridge-driving Labeled Video Database (CamVid) is a road scene dataset that provides semantic segmentation labels for over 700 images captured from vehicle-mounted cameras. It is widely used in autonomous driving research due to its real-world urban setting and consistent annotation standards. CamVid Brostow et al. (2009) served as the primary dataset for training and evaluation of my model. I used it to generate binary road masks by isolating the road class and further processed it through resizing and normalization. Its structure aligned well with my model's input expectations and helped assess performance in practical driving scenarios.

- **Computer Vision for autonomous vehicles:** Computer vision has seen remarkable progress in recent years, especially in the context of autonomous vehicles. One insightful work by Janai et al. Janai et al. (2020) offers a clear and detailed overview of the key challenges, benchmark datasets, and evolving techniques in this space. Their study helps make sense of how segmentation and detection technologies are shaping real-world driving scenarios, and it guided me in understanding where my road segmentation project fits within the broader landscape.

CONCLUSION

This project focused on building an effective road segmentation system using a U-Net architecture trained on the CamVid dataset. The model demonstrated strong performance in the identification of road regions, particularly with high precision after incorporating an edge-weighted loss function. The preprocessing pipeline includes resizing, binary mask generation and normalization, which played an essential role in aligning inputs to the model's requirements.

A major learning outcome in this project was the realization that how critical edge accuracy is for segmentation tasks involving real-world navigation. The shift to edge-weighted loss was driven by a careful visual inspection of predictions, especially on unlabeled, self-captured images. This adaptation significantly enhanced the clarity of road boundaries in outputs.

What worked particularly well in this project was the effectiveness of the U-Net architecture, even when trained on a relatively small dataset like CamVid. It consistently produced accurate road segmentation results. Incorporating edge-weighted loss proved to be a valuable decision, as it significantly enhanced the model's ability to detect road borders with greater precision. Additionally, the workflow was robust and adaptable, allowing smooth handling of both labeled datasets and unlabeled real-world images.

However, there are areas where improvements can be made. The current model could be further optimized for better computational efficiency, especially for large-scale or real-time applications. Some inconsistency in the predictions can also be attributed to the noise from the labels present in the original datasets, which sometimes confuses the model, particularly around the edges. Moreover, the performance of model was dropped under poor lighting conditions like rain or night scenes and wants further exploration.

Looking ahead, the goal is to extend this segmentation model for real-time deployment on embedded systems, making it suitable for use in autonomous navigation or mobile robotics. I'm also interested in experimenting it with more advanced architectures like DeepLabv3+ or SegFormer to enhance accuracy and generalization. Additionally, I also want to expand this model for multi-class segmentation, where I'm not just detecting the roads but also lanes, sidewalks and vehicles.

This project not only strengthened my understanding of image segmentation but also opened exciting directions for future research in autonomous navigation and smart city infrastructure.

ACKNOWLEDGMENTS

- I would like to sincerely thank my mentor Dr. Sandeep Singh Sandha (PhD in AI, UCLA) for giving me the opportunity to work with them on this Road Segmentation Project. I an truly thankful for their constant guidance and valuable feedback throughout the course of this project. Their insights played a vital role in helping me refine both the technical and analytical aspects of my work. Punjab AI Excellence is a program started by Dr. Sandeep Singh Sandha. Its main goal is to help students and young people in India learn about Artificial Intelligence (AI) and use it to solve real-life problems. This program is known as one of the best in teaching AI. It focuses on how AI can be used to improve our daily lives, such as better healthcare, helping farmers with smart farming, improving education, and making government services faster and easier.
- I'm also grateful to the faculty of my college for providing foundational knowledge in the field of AI, especially in deep learning and computer vision, which greatly supported my understanding in this project.
- Lastly, I appreciate the open-source communities and online forums that offered practical resources and solutions whenever I hit a roadblock during implementation.

REFERENCES

Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97.

Janai, J., Güney, F., Behl, H., and Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *Foundations and Trends in Computer Graphics and Vision*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.