Road Segmentation: A Comparative Study

Adithyaa Ramesh
MPSYS
Chalmers University of Technology
Gothenburg, Sweden
adithyaa@chalmers.se

Dion Lim
CCDS (NTU)
Chalmers University of Technology
Gothenburg, Sweden
dionl@chalmers.se

***Abstract*—Achieving reliable road segmentation is key for autonomous driving, but it's tough to get consistently high accuracy across all environments. This study presents a two-model approach, utilizing custom versions of U-Net and VGG16, designed to tackle complex road conditions better. Each model was modified with dropout and batch normalization layers to improve stability and handle overfitting, allowing them to play to their strengths. When tested on datasets like Cityscapes, KITTI, and Comma10k, these customized architectures outperformed basic models, showing promising results across urban and rural roads.**

***Index Terms*—Road Segmentation, Semantic Segmentation, Autonomous Driving, ADAS, Visual Geometry Group (VGG), U-Net, Convolutional Neural Networks (CNNs)**

I. Introduction

For autonomous vehicles, precise road segmentation is essential to navigate safely through various environments. Convolutional neural networks (CNNs) like U-Net and VGG have shown great potential in identifying features for these segmentation tasks. However, applying these models in real-world road settings brings unique challenges, including the need to maintain spatial details, adapt to dynamic environments, and ensure stable performance across different lighting and weather conditions.

The U-Net model, with its encoder-decoder layout and skip connections to retain spatial details, is particularly effective for tasks like road segmentation that require dense predictions [1]. On the other hand, VGG, initially developed for classification tasks, brings value as a feature extractor when adapted for segmentation purposes. In this study, we apply tailored versions of U-Net and VGG16 to road segmentation, enhancing each with dropout and batch normalization to improve stability and reduce overfitting across diverse datasets, including Cityscapes, KITTI, and Comma10k [2], [3]. Our results highlight each model's advantages and show that specialized architectures can achieve precise segmentation in both urban and rural settings.

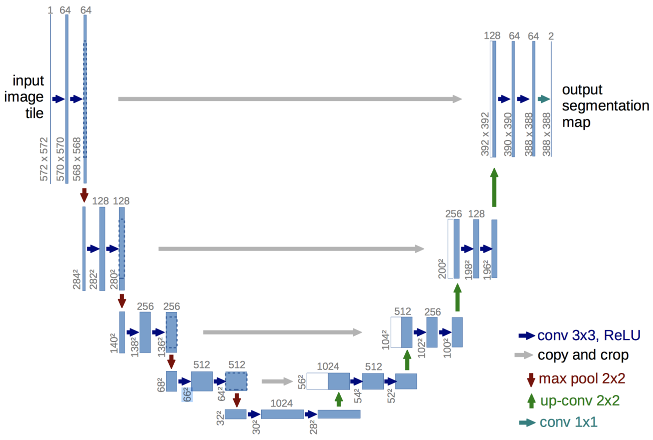

Fig. 1: Standard UNET Architecture for Segmentation

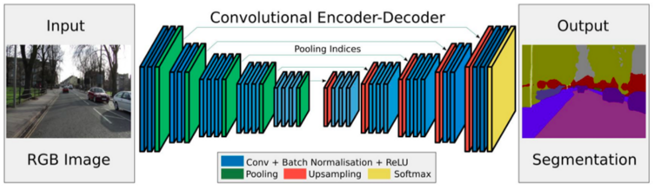

Fig. 2: Standard VGG architecture

II. Background Theory or Related Work

Recent progress in road segmentation has heavily relied on CNNs like U-Net and VGG for their ability to extract features and segment images at the pixel level. Originally developed for biomedical imaging, U-Net's encoder-decoder structure, where each downsampling operation aligns with an upsampling step, has proven effective in road segmentation by preserving spatial details during the process [1]. Over time, modifications like dropout, batch normalization, and even attention mechanisms have made U-Net even more adaptable and accurate, especially for high-resolution tasks [4].

Key techniques such as dropout and batch normalization play a major role in adapting these models for road segmentation. Dropout randomly deactivates neurons during training, which helps to mitigate overfitting by encouraging the model to generalize across a broader set of features—a particularly useful trait when dealing with varied conditions like changes in lighting and weather [5]. Batch normalization, on the other hand, standardizes feature distributions to stabilize training, improving model consistency across different environments, including both urban and rural areas [6].

Although VGG was originally designed for classification tasks, it's been adapted for segmentation because of its deep, sequential layers that capture fine details effectively. Unlike U-Net, VGG lacks skip connections, which means it doesn't retain spatial details as well, but it excels in feature extraction, especially in dense urban scenes with complex structures. Studies comparing the two indicate that while U-Net is excellent at maintaining spatial fidelity, VGG-based models stand out in settings that require detailed feature detection [1], [5].

III. Method / Proposed Solution

Our modified VGG16 and U-Net architectures were tailored for road segmentation tasks on Cityscapes [2], KITTI [7], and Comma10k [3] datasets.

A. *Modified VGG16 Architecture*

The standard VGG16, originally a classification network ending in dense layers, can be effectively adapted for segmentation by transforming it into an encoder-decoder structure. By modifying it with transposed convolutions for upsampling, adding a decoder, and incorporating batch normalization, VGG16 becomes suited for pixel-wise segmentation tasks. In this setup, each convolutional layer in the encoder follows a consistent form, allowing for detailed feature extraction tailored to segmentation needs. They follow the general form:

$$\mathbf{F}_{l+1} = \sigma(\mathrm{BN}(\mathbf{W}_l * \mathbf{F}_l) + \mathbf{b}_l) \tag{1}$$

where $\mathbf{F}_l$ denotes the feature map at layer l, $\mathbf{W}_l$ and $\mathbf{b}_l$ are weights and biases, σ is ReLU activation, and BN is batch normalization. Dropout was applied after certain layers to reduce overfitting, formulated as:

$$\mathbf{F}_{\text{dropout}} = \mathbf{F}_{l+1} \cdot \text{Bernoulli}(p) \tag{2}$$

Where $p = 0.5$ represents dropout probability.

The decoder upsamples the encoded feature maps back to input resolution using transposed convolutions, applied as:

$$\mathbf{U}_{l+1} = \sigma(\mathrm{BN}(\mathbf{W}_l^\top * \mathbf{U}_l + \mathbf{b}_l)) \tag{3}$$

where $\mathbf{W}_l^\top$ is the weight matrix for transposed convolution. Each layer's output is further normalized with batch normalization for stabilization. We used a binary cross-entropy with logits loss for this binary segmentation task:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\sigma(z_i)) + (1 - y_i)\log(1 - \sigma(z_i))\right) \tag{4}$$

where y_i is the ground truth, z_i is the predicted logit, and σ is the sigmoid function.

B. *U-Net Architecture*

Our streamlined U-Net architecture maintains the core encoder-decoder structure with essential skip connections, yet incorporates batch normalization and dropout layers to effectively manage overfitting on smaller datasets. While the original U-Net utilizes more filters and deeper layers, our version simplifies these elements, keeping convolutional layers and max pooling within each encoder block for efficiency without sacrificing performance.

$$\mathbf{F}_{l+1} = \text{MaxPool}(\sigma(\mathrm{BN}(\mathbf{W}_l * \mathbf{F}_l + \mathbf{b}_l))) \tag{5}$$

In the decoder, transposed convolutions upsample feature maps, which are then concatenated with the corresponding encoder feature map via skip connections:

$$\mathbf{U}_{l+1} = \sigma(\mathrm{BN}(\mathbf{W}_l^\top * \mathbf{U}_l + \mathbf{b}_l)) \oplus \mathbf{F}_{\text{skip}} \tag{6}$$

where $\oplus$ denotes concatenation. The final segmentation mask is produced by a 1×1 convolution:

$$\mathbf{O} = \sigma(\mathbf{W}_{\text{out}} * \mathbf{U}_{\text{final}} + \mathbf{b}_{\text{out}}) \tag{7}$$

Where $\mathbf{O}$ is the binary output mask, providing pixel-wise predictions.

IV. Results and Discussion

A. ***VGG16 Trained on Comma10k***

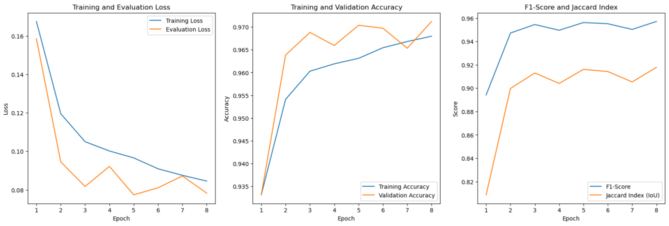

Fig. 3: VGG16 Trained on Comma10k

TABLE I: Performance Metrics for VGG16 Trained on Comma10k Across Datasets

Dataset	Test Loss	Test Accuracy	F1-Score	Jaccard Index (IoU)
Comma10k	0.5516	0.7952	0.5084	0.3409
KITTI	0.2435	0.9205	0.7640	0.6182
Cityscapes	0.1342	0.9525	0.9262	0.8626

TABLE II: Confusion Matrices for Different Datasets

Dataset	Confusion Matrix	
Comma10k	8,513,249	1,465,230
	1,191,553	1,373,968
KITTI	11,483,294	451,344
	700,816	1,865,410
Cityscapes	8,203,481	363,227
	233,044	3,744,248

B. *VGG16 Trained on Cityscapes*

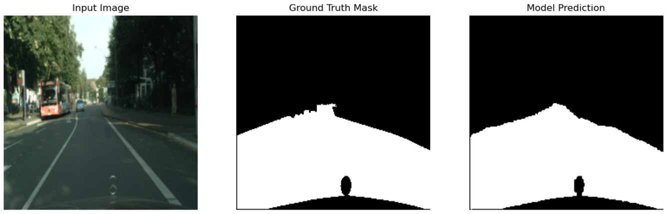

Fig. 4: Comparison between Ground Truth and Prediction

TABLE III: Performance Metrics for VGG16 Trained on Cityscapes Across Datasets

Dataset	Test Loss	Test Accuracy	F1-Score	Jaccard Index (IoU)
Cityscapes	0.1201	0.9521	0.9240	0.8587
KITTI	0.3412	0.8813	0.5409	0.3707
Comma10k	1.1522	0.8102	0.0410	0.0209

TABLE IV: Confusion Matrices for Different Datasets

Dataset	Confusion Matrix	
Cityscapes	8,292,509	274,199
	326,368	3,650,924
KITTI	11,765,283	169,355
	1,552,150	1,014,076
Comma10k	9,877,569	100,910
	2,509,732	55,789

Interestingly, VGG16 performs better on KITTI when trained on Cityscapes than when it's trained on Comma10k. This suggests that Cityscapes and KITTI share more similarities, such as road lighting conditions, than Comma10k. However, this performance does not generalize when VGG16 is tested on the diverse and less controlled scenes in Comma10k, as shown by the lower F1-score and Jaccard Index on this dataset VII.

From observation of the input images, Comma10k is the only dataset that contains day/night images where roads were poorly lit. Hence, it was the initial speculation for VGG-16's poor test performance on other datasets when trained on the Cityscapes dataset. However, when switching the training dataset to Comma10k, we observe the same issue.

This suggests that while VGG16 excels at feature extraction, VGG16's deep feature extraction layers may have overfit to the specific visual features and lighting conditions in Comma10k and Cityscapes, making it distinct from the other two datasets. This overfitting causes the model to struggle when facing new environments, where lighting and road conditions differ from the training set.

C. *UNET Trained on Comma10k*

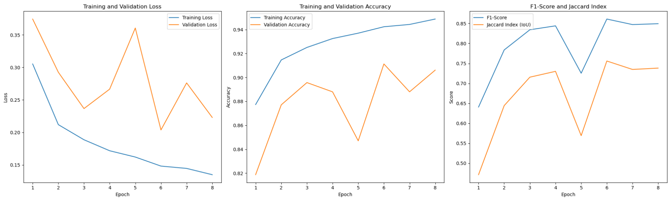

Fig. 5: UNET Trained on Comma10k (Model-3)

TABLE V: Performance Metrics for UNET Trained on Comma10k Across Datasets

Training Dataset	Test Loss	Test Accuracy	F1-Score	Jaccard Index (IoU)
Comma10k	0.1990	0.9155	0.8666	0.7646
KITTI	0.3547	0.8749	0.6146	0.4436
CityScapes	0.1781	0.9246	0.8836	0.7914

TABLE VI: Confusion Matrices for Different Datasets

Dataset	Confusion Matrix	
Comma10k	49,050,361	2,466,547
	4,000,279	21,001,213
KITTI	11,240,946	693,692
	1,119,970	1,446,256
Cityscapes	9,607,416	496,966
	638,693	4,309,725

Conversely, U-Net generalizes better with a more balanced performance across all datasets, with only slight performance decreases when tested on different datasets than it was trained on. This is likely due to U-Net's encoder-decoder architecture with skip connections, which preserves spatial information and may be more robust to varying road and environmental conditions.

The results illustrate the need for robust data diversity in training to achieve consistent model performance. While VGG16's high capacity enables it to achieve peak performance on specific datasets, it sacrifices adaptability across diverse environments, which is better achieved with U-Net's simpler architecture.

V. Conclusions

This study demonstrates the strengths and limitations of VGG16 and U-Net architectures for road segmentation in autonomous driving. VGG16's deep feature extraction capability achieves high accuracy on the dataset it was trained on but shows limited generalization to other environments. U-Net, while slightly less performant on individual datasets, provides more consistent segmentation across various conditions due to its spatial preservation through skip connections and effective regularization.

To improve generalization, a multi-dataset training approach combining Cityscapes, KITTI, and Comma10k, as well as robust data augmentation strategies, could provide the diversity necessary to train models that perform reliably in diverse conditions. Incorporating environmental variations such as

TABLE VII: Dataset Splits and Model Performance across Datasets for VGG16 and U-Net

Model	Dataset Size	Data Split (Images)			Accuracy on Test Datasets (%)		
		Training	**Validation**	**Test**	**Cityscapes**	**KITTI**	**Comma10k**
VGG16 on Comma10k	5000	2780 (55.5%)	695 (14%)	1525 (30.5%)	77.4	84.7	95.8
VGG16 on Cityscapes	4750	2975 (62.5%)	250 (5%)	1525 (32.5%)	95.6	92.0	81.0
U-Net on Comma10k	2975	1160 (39%)	290 (10%)	1525 (51%)	92.5	87.5	91.5

lighting changes, weather effects, and road types into the training process can further enhance robustness.

We could also further explore hybrid architectures that combine VGG16's feature extraction depth with U-Net's spatial retention capabilities. This could involve adding skip connections to VGG-based architectures or using U-Net as a base model and adding more layers selectively to improve feature extraction without overfitting.

In conclusion, balancing high-capacity feature extraction with strong regularization and diverse training data is key to developing road segmentation models suitable for autonomous driving in real-world conditions.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, 2015.

[2] C. Team, "The cityscapes dataset for semantic urban scene understanding," https://www.cityscapes-dataset.com/dataset-overview/, 2016, accessed: October 24, 2024.

[3] C. AI, "Comma10k: A large dataset for autonomous driving," 2020, accessed: October 24, 2024. [Online]. Available: https://github.com/commaai/comma10k

[4] T. Falk, J. Mai, F. Fleck, and T. Kahle, "U-net in the wild: Application to road segmentation for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 209–213, 2019.

[5] A. Ignatov, M. Patel, and P. Ghosh, "Road segmentation for autonomous driving using u-net," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 43–50.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "U-net-based cnn architecture for road crack segmentation," *Sensors*, vol. 20, no. 9, pp. 1–10, 2020.

[7] C. S. R. U. Andreas Geiger, Philip Lenz, "Kitti vision benchmark suite: Road evaluation benchmark," https://www.cvlibs.net/datasets/kitti/eval$_{r}oad.php, 2013, accessed : 2024 - 10 - 26$.