Semantic Segmentation Based Lightweight Lane Detection Network (LW Net) for Intelligent Vehicles

Madiha Shabir Shaikh[*1], Sadia Muniza Faraz[1,2], Yawar Rehman[1,3]

[1]Department of Electronic Engineering, NED University of Engineering & Technology, Karachi, 75270, Pakistan

[2]Electronics Design Center, Department of Electronic Engineering, NED University of Engineering and Technology, Karachi 75270, Pakistan

[3]AI4B, Seoul, South Korea

[*]**Corresponding Author:** madiha@neduet.edu.pk

Citation | Shaikh. M. S, Faraz. S. M, Rehman. Y, "Semantic Segmentation Based Lightweight Lane Detection Network (LW Net) for Intelligent Vehicles", IJIST, Special Issue. pp 81-92, Oct 2024

Received| Oct 07, 2024 **Revised|** Oct 11, 2024 **Accepted|** Oct 16, 2024 **Published|** Oct 20, 2024.

A novel lane detection system is proposed for intelligent vehicles. A key feature of this system is its lightweight design, which requires less computational power. Our lightweight network (LW Net) for semantic segmentation comprises convolutional and separable convolutional layers. We designed a total of six lightweight encoder models (LW Net-A, LW Net-B, LW Net-C, LW Net-D, LW Net-E, and LW Net-F), each paired with matching decoders. The first group of three models is based on depth D1, while the remaining models are based on depth D2. In these models, convolutional layers are either fully or partially replaced by separable convolutional layers. The lightweight network LW Net-A achieved an 88% reduction in training parameters, along with a 2.45% increase in test accuracy compared to the benchmark Seg Net model. Meanwhile, LW Net-F attained a 2% increase in test accuracy and a remarkable 94% reduction in training parameters compared to the benchmark Seg Net model. Overall, the proposed models are less computationally demanding than other benchmark networks, without compromising the pixel accuracy of the semantic model.

Keywords: Semantic Segmentation, Encoder-Decoder, Lane Detection, Light Weight.

Introduction:

Over the past two decades, there has been a significant increase in the number of cars and automobiles on the roads. The primary causes of road accidents include driver inattention, excessive speeding, and irresponsible behavior. Intelligent Transportation Systems (ITS) offer a promising solution by providing safe, eco-friendly, and efficient transportation that benefits society as a whole.

ITS-based intelligent vehicle systems include Advanced Driver Assistance Systems (ADAS), which rely on sensors, cameras, or a combination of both to assist drivers. Features integrated into ADAS may include Lane Departure Warning (LDW), Parking Assistance, Adaptive Cruise Control (ACC), traffic sign recognition, and obstacle detection. Among these features, the detection of lanes and obstacles is a primary objective [1],[2]. However, external factors such as fog, rain, lighting variations, and occlusions complicate the design of accurate lane detection systems [3]. Traditionally, lane detection relied on handcrafted feature extraction methods, including Hough Transform [4], Kalman Filter [5], RANSAC [6], Support Vector Machine [7], Linear Vector Quantization (LVQ) [8], UNSCARF [9], SCARF [10], watershed transform [11], and Finite State Machine [12]. These methods depend on human expertise to design features, which often include color histograms, edges, corners, and textures, making feature extraction a tedious task.

With the advent of neural networks, traditional lane detection methods have increasingly been replaced by deep learning technologies, shifting research focus from conventional approaches to deep learning techniques. Contemporary lane detection algorithms are based on deep learning architectures, such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN).

There are three primary types of lane detection architectures: segmentation-based, object detection-based, and classification-based. A significant drawback of classification-based networks is their reliance on prior knowledge to obtain lane positions. Additionally, object detection approaches use regression bounding boxes to identify lane points, a process that can be labor-intensive. In contrast, segmentation techniques classify image pixels into lane and non-lane categories, demonstrating higher accuracy compared to the other two methodologies. Figure 1 presents some popular semantic segmentation-based architectures and their properties.

In this study, we propose a semantic segmentation-based encoder-decoder design for lane detection, comprising preprocessing and feature extraction stages. Our architecture is lightweight in terms of training parameters while maintaining high accuracy. The main contributions of our research are summarized as follows:

- In the preprocessing module, we enhanced the algorithm by incorporating a data augmentation module to artificially increase the dataset during training.
- In the lane detection module, we propose a lightweight encoder-decoder structure that requires fewer parameters during training, thereby reducing training time.
- Additionally, we simplify the lane detection module by combining a Convolutional Neural Network with a separable convolution network, further minimizing training time.

The structure of this paper is as follows: The literature review addresses traditional and neural network models for lane detection in recent years. The architectural description and methodology section details our proposed network, emphasizing its stages and the neural network architectures employed. Finally, the results and discussion section provide a comprehensive explanation of our findings and compare them with other semantic segmentation-based networks.

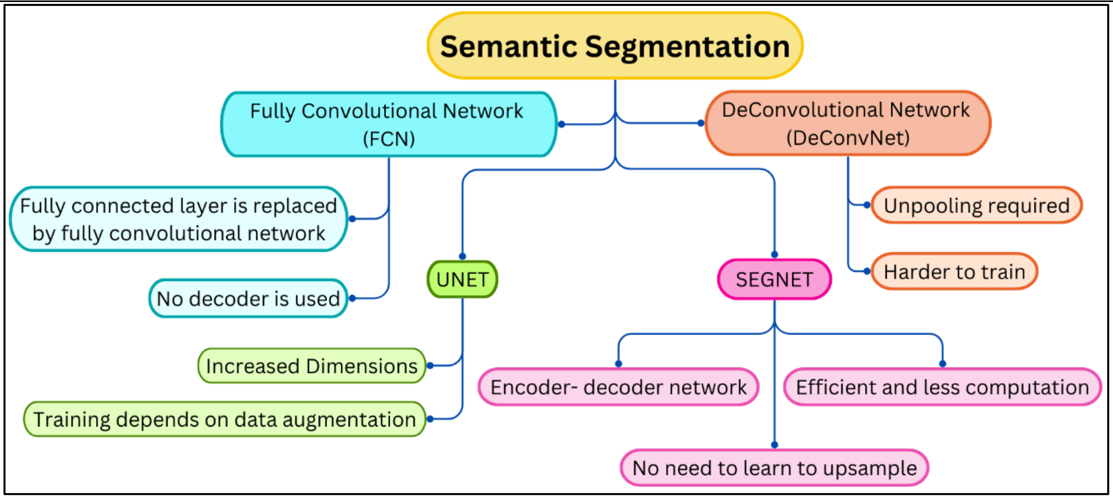

Semantic Segmentation
Fully Convolutional Network (FCN)
DeConvolutional Network (DeConvNet)
Fully connected layer is replaced by fully convolutional network
No decoder is used
UNET
SEGNET
Unpooling required
Harder to train
Increased Dimensions
Training depends on data augmentation
Encoder- decoder network
Efficient and less computation
No need to learn to upsample

Figure 1. Popular semantic segmentation architectures

Literature Review:

Semantic segmentation was first implemented using Fully Convolutional Networks (FCN) [13]. However, FCNs face challenges due to reduced resolution caused by pooling layers. The significant strides in FCNs necessitate additional layers to capture finer details, resulting in increased computational demands and memory usage. To overcome the limitations of FCNs, encoder-decoder architectures were introduced for semantic segmentation. The encoder component utilizes multiple convolutional and pooling layers to extract features, often drawing inspiration from leading image classification networks like Res Net [14] and VGG [15]. The decoder's role is to up sample images back to their original dimensions. Notable convolutional neural networks for semantic segmentation include U-Net [16], Seg Net [17], Deconv Net [18], and Deep Lab [19]. Deconv Net's extensive parameter count makes it challenging to train end-to-end, while U-Net requires higher memory due to its lack of pooling index reuse. In contrast, Seg Net has demonstrated superior performance among encoder-decoder approaches for several reasons: it offers reduced computation time and improved memory efficiency during inference. Additionally, Seg Net leverages pre-trained weights from the VGG16 network, simplifying the initial training process.

Researchers have applied established benchmark architectures in semantic segmentation to tasks such as lane detection. A notable example is Lane Net, a two-stage deep neural network specifically designed for lane detection, comprising a point-wise encoder [20] and an LSTM decoder. Yang et al. proposed an alternative approach to semantic segmentation-based lane detection, focusing on low computational complexity and utilizing a multi-level feature extraction technique inspired by Lane Net. Their method employs the VGG-19 backbone network for extracting multimodal features. By reducing the number of decoder layers and integrating encoder features, they mitigated additional computational burdens [21]. Yu et al. enhanced the Lane Net model with a bilateral network, termed Bi-Lane Net [22], optimizing the architecture by integrating the lightweight ENet [23] with Lane Net.

A similar approach based on the Seg Net architecture was proposed to reduce complexity and enhance simplicity [24]. Chan et al. introduced a Lane Marking Detector (LMD) that utilizes encoder-decoder semantic segmentation. This network is based on the Seg Net and U-Net architectures, incorporating dilated convolution into the encoder to compensate for resolution loss caused by pooling layers. The overall speed of the network is improved by reducing the size of the decoder component [25]. Another approach employed multi-layer perceptrons (MLPs) as an encoder-decoder, as noted in [26]. This method further reduces computational costs by replacing convolutional layers with 1D layers. Yoo et al. developed an end-to-end (E2E) lane detection network that eliminates the need for complex post-processing following semantic segmentation, as described in [27]. This network is based on the U-Net architecture. Additionally, a hybrid network combining CNNs and RNNs was proposed for the continuous detection of lane lines on the road [28]. Another approach suggested the fusion of an encoder-decoder network with dilated convolution [29], computing the weighted average of dilated convolution and encoder-decoder components. A novel loss function was also introduced to minimize classification errors for both pixel and non-pixel predictions.

Objective:

A novel lane detection system is proposed for intelligent vehicles, aiming to reduce human error that can lead to road accidents.

Novelty Statement:

The novelty and key feature of this system is that it is lightweight and requires less computation.

Flow of Study Diagram:

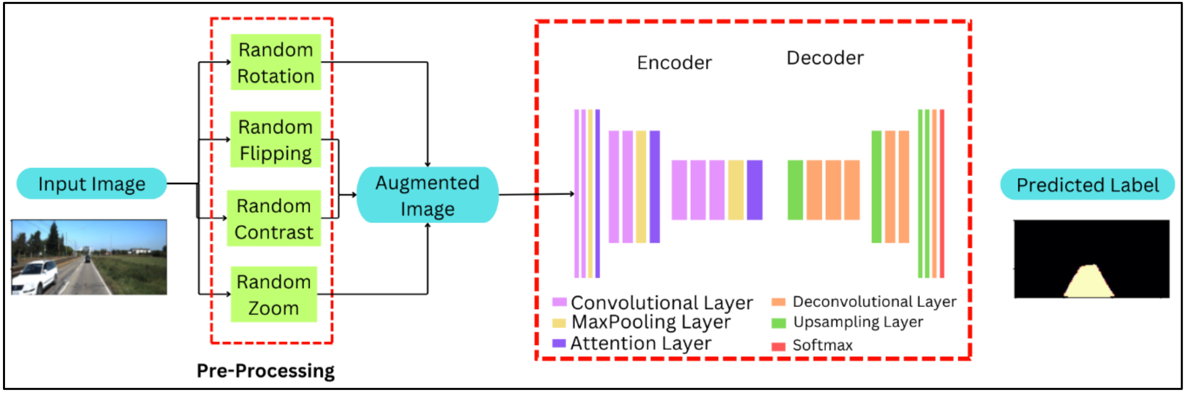


Figure 2. Architectural Description of Neural Network Model

Our proposed lightweight network (LWNet) for semantic segmentation comprises convolutional and separable convolutional layers. We designed a total of six lightweight encoder models (LWNet-A, LWNet-B, LWNet-C, LWNet-D, LWNet-E, and LWNet-F), each with corresponding decoders. The first group of three models is based on depth D1, while the remaining models utilize depth D2. In these models, convolutional layers are either fully or partially replaced with separable convolutional layers. LWNet-A achieves an 88% reduction in training parameters along with a 2.45% increase in test accuracy compared to the benchmark SegNet model. Similarly, LWNet-F shows a 2% increase in test accuracy while achieving a remarkable 94% reduction in training parameters compared to SegNet. Overall, our proposed models are less computationally demanding than other benchmark networks, without compromising the pixel accuracy of the semantic model.

An approach for multi-lane classification employs encoder-decoder networks for lane detection, enhancing segmentation accuracy for weak class objects, such as lane boundaries. In contrast to multi-class segmentation networks, this binary-class network achieves compactness by utilizing fewer convolutional and deconvolutional layers, as discussed in [30]. The Fast-Hybrid Branch network (Fast-HB Net) tackles a related task by leveraging both global and spatial feature extraction. The authors introduced a Hierarchical Feature Learning (HFL) module to enhance the decoder's generalization ability, as noted in [31].

Architecture and Methodology:

In this section, we introduce a novel and efficient lane detection architecture, as illustrated in Figure 2. The architecture consists of three main modules: a preprocessing block, a data augmentation block, and the neural network architecture. The preprocessing block begins by applying a tone-mapping algorithm to the images, followed by resizing and shuffling the tone-mapped images. Next, the pre-processed images move through the data augmentation block, where artificial variations are introduced. This augmentation technique is particularly beneficial when training data is limited. After the augmentation block, the processed images are fed into our proposed neural network architecture, which is a streamlined version inspired by the SegNet encoder-decoder algorithm. Detailed explanations of each of these modules are provided below.

Preprocessing:

Resizing:

Resizing involves adjusting the dimensions of an image, which is essential in image processing for enhancing efficiency and ensuring consistency. Neural networks typically require images to be resized to a standard size for optimal performance. Processing large images can be challenging; therefore, selecting smaller dimensions, such as 80x160 or 160x320, can significantly improve both training and inference speeds. This becomes particularly important for handling large datasets and supporting real-time applications.

Shuffling:

Shuffling refers to the random rearrangement of images within a dataset, a technique that is vital for improving neural network performance. By shuffling the dataset, we mitigate the risk of overfitting the model. This randomization allows the neural network to learn from a diverse array of data patterns instead of merely memorizing specific sequences.

Data Augmentation:

Data augmentation effectively tackles the limitations of small training datasets by artificially expanding them through various techniques. Common methods for image data augmentation include rotation, flipping, zooming, and adjusting contrast, among others. These techniques enhance the robustness and generalization of neural networks by exposing them to a broader spectrum of data variations during training.

Image Rotation:

Image rotation is a technique employed to expand the image dataset, increasing its size and diversity for training purposes. By incorporating rotated images, the model can learn from a wider range of examples, thereby enhancing its generalization capabilities. This method ultimately improves the model's ability to predict and comprehend a broader array of previously unseen images.

Image Flipping:

Flipping images is another data augmentation technique utilized to enhance diversity within the training dataset. This method increases the robustness of deep learning models by exposing them to variations in image orientation. By learning from flipped images, the model becomes more adept at recognizing patterns irrespective of orientation, thereby reducing the likelihood of overfitting.

Image Contrast Adjustment:

Image contrast refers to the difference in brightness between various parts of an image, significantly impacting the visibility and clarity of objects within it. Contrast levels are typically categorized into two types: high contrast, where objects appear sharp and distinct, and low contrast, where objects appear less defined and dull. Randomly adjusting the contrast of training images creates a diverse dataset, enhancing the robustness and generalization capabilities of neural network models. This augmentation technique exposes the model to a broader range of visual variations, enabling it to better adapt to different lighting conditions and ultimately improve overall performance.

Image Zooming:

Zooming is a technique employed to adjust the scaling of images, thereby adding variability to the training dataset. This method modifies the size and position of objects within an image. During data augmentation, images can be zoomed in or out. Zooming in magnifies specific areas of an image, highlighting details that may occupy only a few pixels within the overall composition. This is particularly beneficial for enhancing the model's ability to recognize fine details. Conversely, zooming out offers a broader perspective, aiding the model in understanding spatial relationships and context within larger scenes. Collectively, these techniques enhance the robustness and performance of neural network models.

Neural Network Architecture:

Our neural network architecture integrates encoder and decoder networks. The encoder network is designed to extract features from images, processing them at various scales and resolutions. In its lower layers, the encoder captures intricate details such as boundaries, textures, and local patterns, which are essential for object classification and boundary delineation. The higher layers of the encoder focus on.

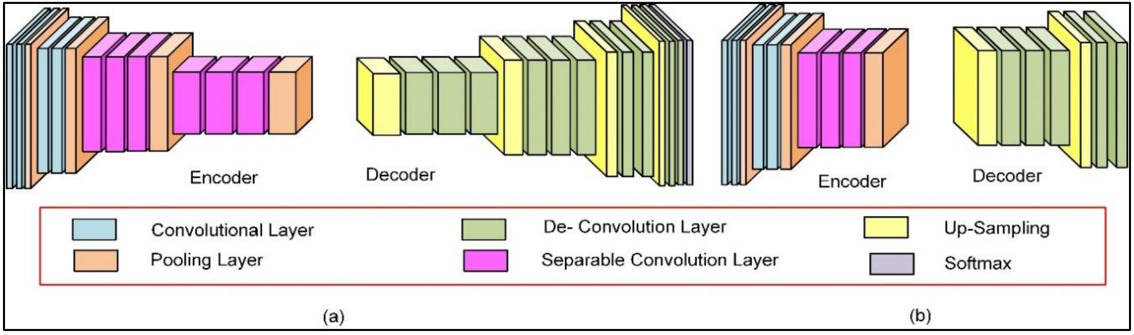


Figure 3. Semantic segmentation architectures consisting of convolutional and separable convolutional layers (a) LW Net-C (b) LW Net-D

Capturing broader context and relationships between objects in the image, the encoder aids in achieving accurate pixel-level predictions. Conversely, the decoder network restores the resolution lost during the encoder's max pooling operations, generating a pixel-wise segmentation map from the high-level features extracted by the encoder. This ensures that the output image aligns with the input size. Our architecture adopts an encoder-decoder semantic segmentation approach, with the encoder network inspired by VGG-16 [15]. We utilize 3x3 convolution filters with a stride of 1 to minimize computational load, effectively reducing both training and inference times. This encoder network produces hierarchical features across various scales and resolutions, enhancing its capacity to manage diverse visual data.

For an input image represented as (W, H, C)—where W denotes width, H represents height, and C indicates the number of channels—the image resolution decreases following each max pooling layer. Figure 3 illustrates our designed neural network architectures. In Figure 3(a), a lightweight network architecture is depicted, incorporating both standard convolutional and separable convolutional layers. The encoder section initiates with convolutional layers in the first four stages, transitioning to separable convolutional layers in subsequent stages to reduce the number of trainable parameters. This approach allows the network to remain shallow while maintaining efficiency. Figure 3(b) illustrates the lightest encoder network configuration, featuring an initial sequence of four convolutional layers followed by three layers of separable convolution. This design is optimized for performance with minimal computational overhead.

Table 1. Kitty Dataset Description

Type	Training Dataset Size	Test Dataset Size
Urban Unmarked (UU)	98	100
Urban Marked (UM) (two-way road)	95	96
Urban Marked (UMM) (multi-lines)	96	94

Dataset & Implementation Details:

In this section, we begin by discussing the lane detection dataset, KITTI [32]. Following this, we outline our implementation strategies. We then evaluate our proposed methods using the selected dataset, concluding with a comparison of our results against state-of-the-art methods. All models were tested on an NVIDIA platform.

We evaluated our model using the Quadro P100 GPU and selected the KITTI dataset, a prominent open-access benchmark for road and lane detection. This dataset consists of 600 labeled images, utilized for both training and testing and encompasses a diverse array of urban road scenes, including both marked and unmarked lane lines. Each image in the dataset measures 375x1242 pixels. The KITTI dataset is divided into three subsets, with approximately 100 training and 100 test images in each, as summarized in Table 1.

Our model employs ReLU activation functions and is trained from scratch without any pretraining. We use a mini-batch size of 5 and the 'Adam' optimizer during training, which spans 200 epochs. To enhance the training process, we converted our dataset into low-resolution images, opting for two resolution sizes: 80x160 pixels and 160x320 pixels.

Results and Discussion:

The evaluation metrics employed in this study include binary accuracy, Area Under the Curve (AUC), Intersection over Union (IoU), and the Dice Coefficient. Binary accuracy, calculated using Equation 1, measures the percentage of correct predictions made by the algorithm relative to the total number of images. It represents the ratio of correct predictions to the total number of images.

Another important metric for assessing segmentation-based detection is Intersection over Union (IoU), which quantifies the overlap between two bounding boxes: one representing the actual ground truth and the other the predicted bounding box. IoU for a single image can be computed using Equation 2.

We also evaluated our algorithm using the Area Under the Curve (AUC), which is commonly used for binary classification tasks involving a single "positive" class. AUC can be derived by plotting the Recall value on the x-axis against the Precision on the y-axis. Additionally, we utilized the Dice Coefficient, which measures the similarity between two sets. This metric is particularly useful for image segmentation tasks.

$$\text{Accuracy (Acc)} = \frac{\text{Total number of correct predictions}}{\text{Total numbe of images}} \qquad 1$$

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Aea of union}} \qquad 2$$

We have developed several lightweight architectures utilizing convolutional and separable convolutional layers, as detailed in Table 2. LW Net (A) consists of 21 layers, including 10 convolutional layers. In contrast, LW Net (B) mirrors LW Net (A) but replaces conventional convolutional layers with separable convolutional layers to reduce computational load. LW Net (C) optimizes this approach by integrating four initial convolutional layers followed by six separable convolutional layers. Additionally, we created shallower networks without compromising accuracy: LW Net (D), LW Net (E), and LW Net (F), each comprising 15 layers with seven layers in both the encoder and decoder sections, along with a SoftMax layer. LW Net (D) employs convolutional layers, LW Net (E) utilizes separable convolutional layers, and LW Net (F) combines both types.

Experiments were conducted using two sets of low-resolution images, with results presented in Table 3 and Table 4 for image resolutions of 80x160 pixels and 160x320 pixels, respectively. As shown in Table 3, LW Net (B) and LW Net (E), although lighter in computational parameters, achieved lower accuracy and performance metrics compared to other models. In contrast, LW Net (C) and LW Net (F), which leverage a mix of convolutional and separable convolutional layers, demonstrated superior performance. Specifically, LW Net (C) and LW Net (F) excelled in lane detection tasks while also reducing computational demands.

Table 2. Lightweight Architectures of LW Net

A	B	C	D	E	F
		Encoder Configuration			
Conv3-32	SeparableConv3-32	conv3-32	conv3-32	SeparableConv3-32	conv3-32
Conv3-32	SeparableConv3-32	conv3-32	conv3-32	SeparableConv3-32	conv3-32
		Max Pooling			
Conv3-64	SeparableConv3-64	conv3-64	conv3-64	SeparableConv3-64	conv3-64
Conv3-64	SeparableConv3-64	conv3-64	conv3-64	SeparableConv3-64	conv3-64
		Max Pooling			
Conv3-128	SeparableConv3-128	SeparableConv3-128	conv3-128	SeparableConv3-128	SeparableConv3-128
Conv3-128	SeparableConv3-128	SeparableConv3-128	conv3-128	SeparableConv3-128	SeparableConv3-128
Conv3-128	SeparableConv3-128	SeparableConv3-128	conv3-128	SeparableConv3-128	SeparableConv3-128
Conv3-256	SeparableConv3-256	SeparableConv3-256			
Conv3-256	SeparableConv3-256	SeparableConv3-256			
Conv3-256	SeparableConv3-256	SeparableConv3-256			

Table 3. Results of different architectures for Image resolution of (80x160px)

Network	Training Acc	Test Accuracy	IoU	Dice Coeff	AUC	No. of Parameters
U net [16]	0.9981	0.96418	0.7689	0.8694	0.919	11.77MB
LW Net-A (ours)	0.9861	0.9624	0.775	0.8737	0.915	16.82MB
LW Net-B (ours)	0.9825	0.9531	0.7312	0.844	0.912	10.39MB
LW Net-C (ours)	0.9870	**0.97068**	**0.8217**	0.902	**0.9565**	10.6MB
LW Net-D (ours)	0.9911	**0.97013**	0.819	0.9008	0.9475	3.88MB
LW Net-E (ours)	0.9882	**0.96622**	0.773	0.8724	0.923	2.42MB
LW Net-F (ours)	0.9907	**0.9716**	**0.834**	**0.9099**	0.9469	**2.64MB**

Table 4. Results of different architectures for Image resolution of (160x320px)

Network	Training Acc.	Test Acc.	IoU	Dice Coeff	AUC	No. of Parameters
Seg Net [17]	0.9224	0.918	-	-	-	68.58MB
LW Net-A (ours)	0.9903	**0.9716**	0.826	0.9048	0.9515	16.82MB
LW Net-B (ours)	09885	0.9629	0.7848	0.8794	0.9304	10.39MB
LW Net-C (ours)	0.9885	0.9672	0.803	0.8907	0.9444	10.6MB
LW Net-D (ours)	0.9926	0.9707	0.8233	0.903	0.9481	3.888MB
LW Net-E (ours)	0.9855	0.965	0.796	0.886	0.9398	2.42MB
LW Net-F (ours)	0.9911	0.9700	0.822	0.9026	0.947	2.64MB

We compared our architectures with the widely used U-Net algorithm [16], observing improvements in test accuracies and the Area Under the Curve (AUC) despite a smaller number of parameters, as indicated in Table 3. Predictions from the various LW Net models with an image resolution of 80x160 pixels are illustrated in Figure 4. Input images and their corresponding labels are shown in Figures 4(a) and 4(b), while predictions are presented in Figures 4(c) to 4(h). Notably, improved predictions are observed for designs C and F, as illustrated in Figures 4(e) and 4(h).

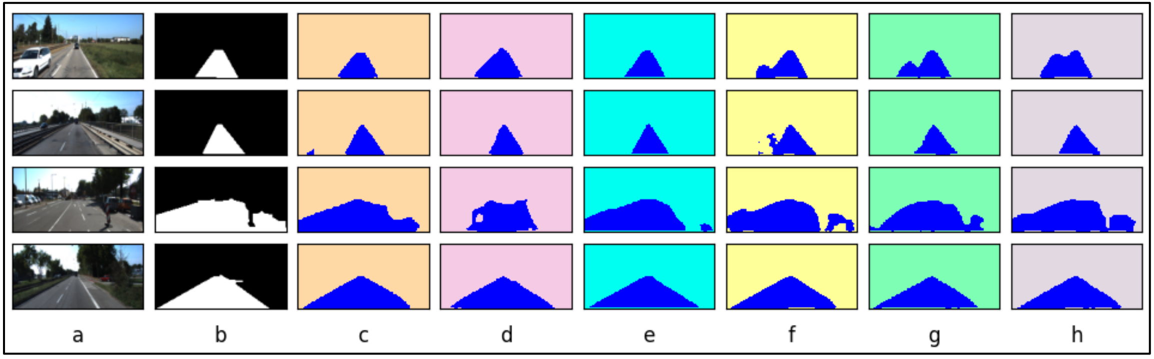


Figure 4. Predictions from different models of LW Net with 80x160 px image resolution (a) Input Picture (b) Labels (c) Design A (d) Design B (e) Design C (f) Design D (g) Design E (h) Design F

Additionally, we benchmarked our models against the state-of-the-art SegNet [17], showcasing the predicted lane area images in Figure 4. These results demonstrate the effectiveness of our approach to lane detection, highlighting the balance between computational efficiency and accuracy. Table 4 presents various metrics for our lane detection model at an image resolution of 160x320 pixels. Notably, the number of parameters remains consistent with that of the 80x160 pixel resolution case. While image resolution affects training and inference times, it does not impact the parameter count. Similar patterns can be observed in Table 4 as in Table 3, with LW Net (C) and LW Net (F) achieving significantly better accuracies with minimal compromise on the number of parameters compared to the LW Net (B) and LW Net (E) models.

Predictions from various LW Net models with a 160x320 pixel image resolution are illustrated in Figure 5. The input images and their corresponding labels are displayed in Figures 5(a) and 5(b), while the predictions are presented in Figures 5(c) through 5(h). Notably, improved predictions are achieved with designs C and F, as shown in Figures 5(e) and 5(h). The analysis indicates that models operating on high-resolution images yield better results.

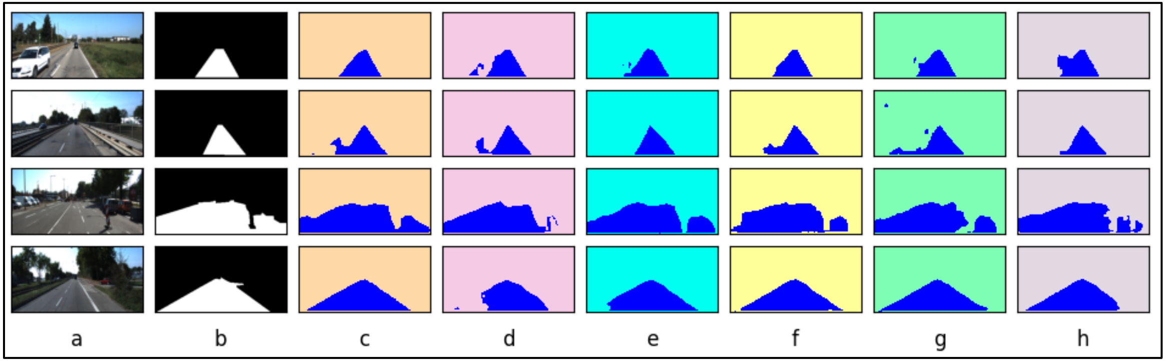
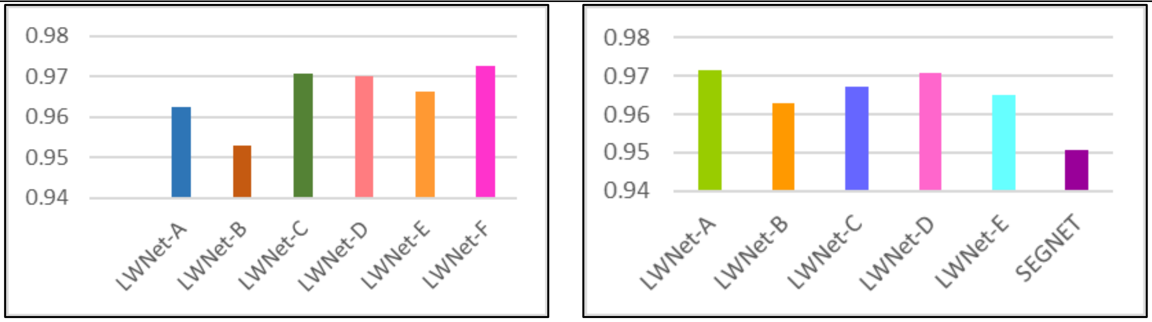


Figure 5. Predictions from different models of LW Net with 160x320 px image resolution (a) Input Picture (b) Labels (c) Design A (d) Design B (e) Design C (f) Design D (g) Design E (h) Design F



(a) **(b)**

Figure 6. Pixel accuracy comparison chart for our different designed models (a) Image resolution (80x160) (b) Image Resolution (160x320)

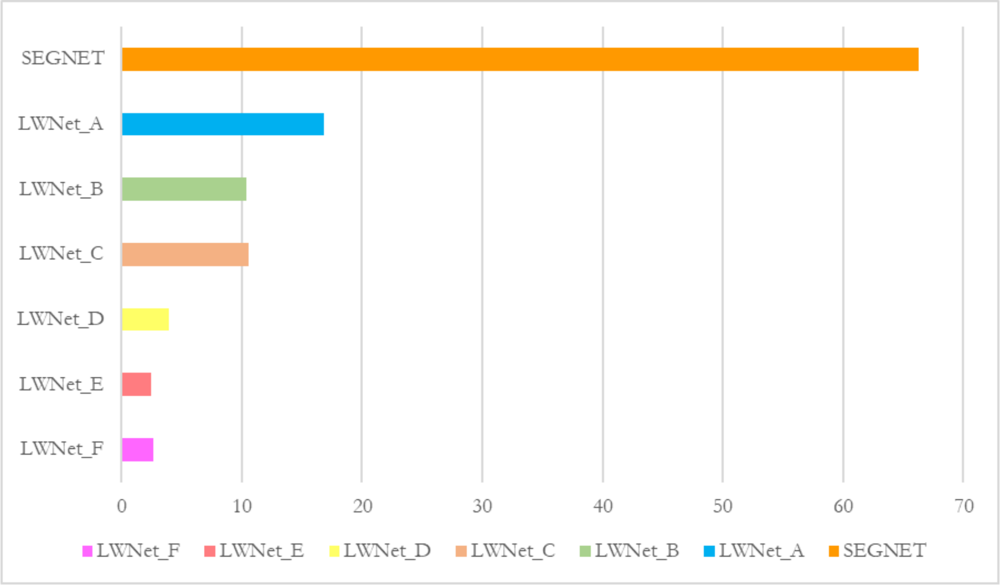


Figure 7. Comparison of a number of parameters in MB required for training for our different architectures.

This underscores the effectiveness of the LWNet-C and LWNet-F architectures in lane detection tasks, emphasizing their potential to reduce computational complexity without compromising performance. Figure 6(a) presents graphs for an image resolution of 80x160 pixels, where the LWNet-F model, known for its lightweight design, achieves the highest pixel accuracy. A comparison of pixel accuracies at an image resolution of 160x320 pixels is illustrated in Figure 6(b), showing that our models outperform the benchmark SegNet model. This demonstrates the superior performance of our architectures in lane detection across various image resolutions. Figure 7 visually represents the significant variation in the number of parameters required for training among the models. This comparison highlights the diverse architectural designs and computational efficiencies of each model, illustrating their impact on training complexity and resource utilization.

Conclusion:

A semantic segmentation-based lightweight network (LWNet) has been developed, consisting of convolutional and separable convolutional layers. Six lightweight encoder models—LWNet-A, LWNet-B, LWNet-C, LWNet-D, LWNet-E, and LWNet-F—have been designed and their performance evaluated. The LWNet-A model, which relies on traditional convolutional layers, achieved an 88% reduction in training parameters along with an approximately 2.45% increase in test accuracy compared to the benchmark SegNet model. In contrast, LWNet-F, which incorporates initial convolutional layers followed by separable convolutional layers, attained about a 2% increase in test accuracy and a remarkable 94%

reduction in training parameters relative to the SegNet benchmark. Our proposed models are significantly less computationally demanding than other benchmark networks, without sacrificing pixel accuracy in the semantic segmentation task. These innovative lane detection networks hold promise for applications in intelligent vehicles and advanced driving assistance systems.

Acknowledgment: Authors are thankful for the facilities of the Sultan Qaboos Oman IT chair office at the NED University of Engineering and Technology, Pakistan.

Author's Contribution: M. Shabir designed the network, performed modeling and simulations, and wrote the original draft**.** S. M. Faraz and Y. Rehman conceptualized the study, performed formal analysis, and assisted in methods and supervision.

Conflict of Interest: Authors declare that there is no conflict of interest.

References:

[1] S. Ishikawa and H. Kuwamoto, "Visual Navigation of an Autonomous Vehicle Using White Line Recognition," 1988.

[2] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," Proceedings of the IEEE, vol. 90, no. 7. Institute of Electrical and Electronics Engineers Inc., pp. 1258–1270, 2002. doi: 10.1109/JPROC.2002.801444.

[3] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," Pattern Recognit, vol. 111, Mar. 2021, doi: 10.1016/j.patcog.2020.107623.

[4] W. S. Wijesoma, K. R. S. Kodagoda, A. P. Balasuriya, and E. K. Teoh, "Road Edge and Lane Boundary Detection using Laser and Vision," 2001.

[5] J. Lu, M. Yang, H. Wang, and B. Zhang, "Vision-based Real-time Road Detection in Urban Traffic." [Online]. Available: http://www.lits.tsinghua.edu.cn/lujianye/;

[6] Q. Chen, "3-13 A Real-time Lane Detection Algorithm Based on a Hyperbola-Pair Model," 2006.

[7] H. Zhang, D. Hou, and Z. Zhou, "A Novel Lane Detection Algorithm Based on Support Vector Machine," 2005.

[8] S. G. Foda and A. K. Dawoud, "Highway Lane Boundary Determination For Autonomous Navigation."

[9] "Crisman_1991_UNSCARF--A color vision system for the detection of unstructured roads".

[10] J. D. Crisman and C. E. Thorpe, "CHRlSMAN AND THORPE: COLOR VISION SYSTEM THAT TRACKS ROADS AND INTERSECTIONS SCARF: A Color Vision System that Tracks Roads and Intersections."

[11] S. Beucher and M. Bilodeau, "Road Segmentation and Obstacle Detection by a fast watershed algorithm," 1994. [Online]. Available: https://www.researchgate.net/publication/240221983

[12] B. F. Wu, C. T. Lin, and Y. L. Chen, "Dynamic calibration and occlusion handling algorithms for lane tracking," IEEE Transactions on Industrial Electronics, vol. 56, no. 5, pp. 1757–1773, 2009, doi: 10.1109/TIE.2008.2011295.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015, [Online]. Available: http://arxiv.org/abs/1505.04597

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Trans Pattern Anal Mach Intell, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[18] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” 2015.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” Jun. 2016, [Online]. Available: http://arxiv.org/abs/1606.00915

[20] Z. Wang, W. Ren, and Q. Qiu, “LaneNet: Real-Time Lane Detection Networks for Autonomous Driving,” Jul. 2018, [Online]. Available: http://arxiv.org/abs/1807.01726

[21] W. J. Yang, Y. T. Cheng, and P. C. Chung, “Improved Lane Detection with Multilevel Features in Branch Convolutional Neural Networks,” IEEE Access, vol. 7, pp. 173148–173156, 2019, doi: 10.1109/ACCESS.2019.2957053.

[22] F. Yu, Y. Wu, Y. Suo, and Y. Su, “Shallow Detail and Semantic Segmentation Combined Bilateral Network Model for Lane Detection,” IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp. 8617–8627, Aug. 2023, doi: 10.1109/TITS.2023.3289165.

[23] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation.”

[24] A. Al Mamun, P. P. Em, and J. Hossen, “Lane marking detection using simple encode decode deep learning technique: SegNet,” International Journal of Electrical and Computer Engineering, vol. 11, no. 4, pp. 3032–3039, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3032-3039.

[25] P.-R. Chen, S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, “Efficient Road Lane Marking Detection with Deep Learning; Efficient Road Lane Marking Detection with Deep Learning,” 2018.

[26] X. Yao, Y. Wang, Y. Wu, G. He, and S. Luo, “MLP-Based Efficient Convolutional Neural Network for Lane Detection,” IEEE Trans Veh Technol, vol. 72, no. 10, pp. 12602–12614, Oct. 2023, doi: 10.1109/TVT.2023.3275571.

[27] S. Yoo et al., “End-to-End Lane Marker Detection via Row-wise Classification.”

[28] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks,” Mar. 2019, doi: 10.1109/TVT.2019.2949603.

[29] A. Mukhopadhyay, L. R. D. Murthy, I. Mukherjee, and P. Biswas, “A Hybrid Lane Detection Model for Wild Road Conditions,” IEEE Transactions on Artificial Intelligence, vol. 4, no. 6, pp. 1592–1601, Dec. 2023, doi: 10.1109/TAI.2022.3212347.

[30] S. Chougule, A. Ismail, A. Soni, N. Kozonek, V. Narayan, and M. Schulze, An efficient encoder-decoder CNN architecture for reliable multilane detection in real time; An efficient encoder-decoder CNN architecture for reliable multilane detection in real time. 2018. doi: 10.0/Linux-x86_64.

[31] G. Pang, B. Zhang, Z. Teng, N. Ma, and J. Fan, “Fast-HBNet: Hybrid Branch Network for Fast Lane Detection,” IEEE Transactions on Intelligent Transportation Systems, 2022, doi: 10.1109/TITS.2022.3145018.

[32] J. Fritsch, T. Kühnl, and A. Geiger, “A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms.” [Online]. Available: http://www.cvlibs.net/datasets/kitti/