Çukurova Üniversitesi Mühendislik Fakültesi Dergisi
Çukurova University Journal of the Faculty of Engineering
ISSN: 2757-9255 CİLT/VOLUME: 39 SAYI/ISSUE: 4 ARALIK/DECEMBER 2024

Performance Comparison of Deep Learning Lane Detection Models for Autonomous Vehicles

Muhammed Said ATAŞ [1,a], Yahya DOĞAN [1,b], Cüneyt ÖZDEMİR [1,c]

[1]*Siirt University, Engineering Faculty, Department of Computer Engineering, Siirt, Türkiye*

[a]***ORCID**: 0009-0007-6572-9010;* [b]***ORCID**: 0000-0003-1529-6118;* [c]***ORCID**: 0000-0002-9252-5888*

Article Info
Received : 19.02.2024
Accepted : 23.12.2024
DOI: 10.21605/cukurovaumfd.1605865

Corresponding Author
Muhammed Said ATAŞ
msaid.atas@siirt.edu.tr

Keywords
Deep learning
Lane detection
TuSimple
U-Net
SCNN

***How to cite:** ATAŞ, M.S., DOĞAN, Y., ÖZDEMİR, C., (2024). Performance Comparison of Deep Learning Lane Detection Models for Autonomous Vehicles. Cukurova University, Journal of the Faculty of Engineering, 39(4), 861-871.*

ABSTRACT

Recent advancements in the field of deep learning have significantly improved the driving capabilities of autonomous vehicles. This study focuses on the lane detection abilities of autonomous vehicles and examines the use of deep learning-based approaches in this context. The research compares the lane detection performance of various deep learning models, including U-Net, SCNN, ENet, and ENet-SAD, utilizing the TuSimple dataset. The models were evaluated using various quantitative metrics such as accuracy, precision, sensitivity, F1 score, and IoU. Extensive experiments have determined that the U-Net model exhibited the highest performance with an accuracy rate of 98.3%. The SCNN model, on the other hand, stood out in terms of precision, sensitivity, F1 score, and IoU metrics. In terms of inference time, the U-Net model was identified as the fastest lane detection model with a time of 20.12 ms. These results indicate that the U-Net model is particularly suitable for real-time systems requiring low computational power. Additionally, a qualitative assessment of lane detection success revealed that the SCNN and U-Net models more accurately detected pixels where lanes are present, whereas the ENet and ENet-SAD models were more prone to false-negative errors.

Otonom Araçlar İçin Derin Öğrenme Şerit Tespiti Modellerinin Performans Karşılaştırması

Makale Bilgileri
Geliş : 19.02.2024
Kabul : 23.12.2024
DOI: 10.21605/cukurovaumfd.1605865

Sorumlu Yazar
Muhammed Said ATAŞ
msaid.atas@siirt.edu.tr

Anahtar Kelimeler
Derin öğrenme
Şerit takibi
TuSimple
U-Net
SCNN

***Atıf şekli:** ATAŞ, M.S., DOĞAN, Y., ÖZDEMİR, C., (2024). Performance Comparison of Deep Learning Lane Detection Models for Autonomous Vehicles. Cukurova University, Journal of the Faculty of Engineering, 39(4), 861-871.*

ÖZ

Son yıllarda derin öğrenme alanlarında meydana gelen ilerlemeler, otonom araçların sürüş yeteneklerini önemli ölçüde geliştirmiştir. Bu çalışma, otonom araçların şerit tespit yeteneklerine odaklanmaktadır ve derin öğrenme tabanlı yaklaşımların bu bağlamdaki kullanımını incelemektedir. Araştırma kapsamında, TuSimple veri seti kullanılarak U-Net, SCNN, ENet ve ENet-SAD gibi çeşitli derin öğrenme modellerinin şerit tespiti performansları karşılaştırılmıştır. Modeller, doğruluk, hassasiyet, duyarlılık, F1 skoru ve IoU gibi çeşitli nicel metrikler kullanılarak değerlendirilmiştir. Yapılan kapsamlı deneyler sonucunda, U-Net modelinin %98.3 doğruluk oranı ile en yüksek performansı sergilediği tespit edilmiştir. SCNN modeli ise hassasiyet, duyarlılık, F1 skoru ve IoU metrikleri açısından öne çıkmıştır. Çıkarım süresi açısından değerlendirildiğinde, 20.12 milisaniye ile U-Net modelinin en hızlı şerit tespitini gerçekleştiren model olduğu belirlenmiştir. Bu sonuçlar, özellikle gerçek zamanlı ve düşük işlem gücü gerektiren sistemler için U-Net modelinin tercih edilebileceğini göstermektedir. Ayrıca, şerit tespit başarısının nitel değerlendirilmesi sonucunda, SCNN ve U-Net modellerinin şeritlerin bulunduğu pikselleri daha doğru bir şekilde tespit ettiği, buna karşın ENet ve ENet-SAD modellerinin false-negative (yanlış negatif) hata yapmaya daha meyilli olduğu gözlemlenmiştir.

1. INTRODUCTION

Recently, research on autonomous vehicles has been advancing rapidly, and it is anticipated that these vehicles will find widespread use soon, in harmony with the concept of smart cities [1,2]. Autonomous vehicles excel in complex and unpredictable traffic conditions. Their ability to safely and efficiently transport passengers in these situations is one of their most significant advantages. In this context, the ability of an autonomous vehicle to perceive its surroundings, as shown in Figure 1, is of vital importance. The perception system continuously collects, processes, and analyzes environmental data obtained through radar, lidar, cameras, and other advanced sensors. This enables the vehicle to instantly monitor its position, the objects around it, the movements of other vehicles, and the condition of the road and lanes. The processing of multi-modal data facilitates the vehicle's ability to make instantaneous decisions, adjust its route, and adapt to changing traffic conditions. The perception system plays a critical role in ensuring the vehicle makes optimal decisions regarding safety, efficiency, and comfort, thereby transporting passengers to their destinations via the safest and most suitable routes. Among these sensors, camera systems, which are also found in leading autonomous vehicles like Tesla and Google Waymo, are particularly effective for environmental perception [3,4]. Cameras process high-resolution images to identify environmental objects, read traffic signs, and assess road conditions, playing a significant role in these areas. Through cameras, lanes can be identified, and this lane tracking enables the driverless vehicle to proceed safely. This capability allows the vehicle to stay in its lane and, when necessary, automatically change lanes [5,6].

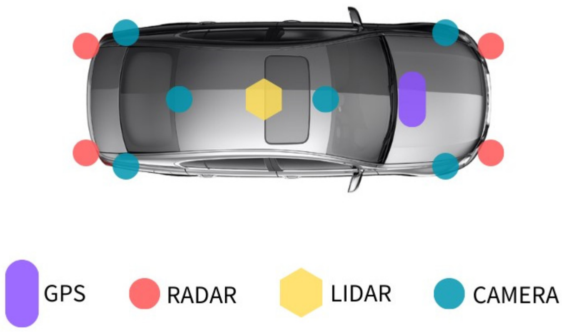

Figure 1. Overview of Environmental Sensing Technologies Used in Autonomous Vehicles for Road and Traffic Perception

Autonomous vehicles and Advanced Driver Assistance Systems (ADAS) aim to enhance safety levels, reduce fuel and energy consumption, and lower road traffic emissions. ADAS offers drivers numerous support features such as advanced collision warnings and safe lane-changing aids [7,8]. According to the World Health Organization, approximately 1.3 million people lose their lives in traffic accidents each year [9]. Research indicates that most accidents are due to driver errors [10,11]. The widespread adoption of ADAS could significantly reduce the number of such traffic accidents by taking necessary measures in the event of a potential accident to prevent it [12]. Lane Departure Warning (LDW), a fundamental feature of ADAS, similarly relies on lane detection and tracking algorithms. LDW alerts the driver when the vehicle unintentionally crosses lane markings and then redirects the vehicle to the desired safe route. Studies have reported a series of challenges and issues associated with the implementation of ADAS and LDW systems, among which distortions in road and lane images due to varying lighting conditions play a significant role [13]. The existing literature generally presents three main approaches for lane detection: feature-based, model-based, and learning-based approaches. The feature-based approach processes attributes such as edges, colors, brightness, and textures. These attributes are usually unaffected by road shapes but are more sensitive to the amount of ambient light [12,13]. Model-based approaches use global road models to accommodate low-level features (pixel values, edges, colors, etc.). These models are resistant to lighting effects but are more sensitive to changes in road shapes. Geometric parameters in the model-based approach are employed in lane detection [14,15]. On the other hand, the learning-based approach consists of two stages: training and inference. During the training phase, a model is created using previously determined features (statistical and image processing-based features extracted from images in classical machine learning or automatically extracted features from images in deep learning). In the classification phase, road images are fed into the model as inputs, and the trained model is expected to detect lanes [16,17].

2. MATERIALS AND METHODS

This section provides details regarding the dataset that forms the basis of this study, the model architecture, and the training process.

2.1. Dataset

In this study, the TuSimple dataset [18], frequently preferred in the literature for comparing the performances of deep learning models, was utilized. It consists of 1-second long video clips recorded on U.S. highways under various weather conditions, at different times, and in different traffic conditions. The video clips were divided into images using specialized video processing software, producing 20 images per second. This process ensured high-quality output and precise frame rate control, enabling consistent and reliable preparation of our dataset. It is commonly seen in the literature that the last frame of the video sequence is used. The image resolution is 1280×720 pixels. The dataset, approximately 25 GB in size, employs a standard splitting method consisting of training (3,626), validation (358), and testing (2,782) images to ensure the general validity and comparability of the results. Upon examining the images in the dataset, it is observed that the images are taken from highways with 2, 3, 4, and 5 lanes. When preparing the masks, labeling was done considering that there could be up to 4 lanes. In the preparation of the masks, all backgrounds other than the lanes were assigned a “0” value. A gray pixel value of “2” for the first lane, “3” for the second lane, “4” for the third lane, and “5” for the fourth lane was assigned. Figure 2 presents an example frame from the dataset and its corresponding mask.

Figure 2. The 20th frame of a video clip from the TuSimple training dataset and its corresponding mask image where lane values are adjusted to 255 for clarity.

2.2. Models

The models utilized in this study were selected based on their widespread use in lane detection, the comparability of their diverse architectures, and their potential for real-time applications. The selection of these models was influenced by several factors: U-Net's success in medical image segmentation, SCNN's effectiveness in understanding traffic scenes, and the lightweight structures and fast operational characteristics of ENet and ENet-SAD. These considerations ensured a comprehensive evaluation of different approaches to lane detection, balancing performance with computational efficiency.

In this study, ResNet-based models were deliberately excluded from the scope. The primary reason for this decision was the study's focus on comparing lightweight and fast models specifically suitable for real-time applications. ResNet-based models typically require higher computational power and memory due to their deeper and more complex structures [19]. While these models often offer high accuracy, their computational demands can be prohibitive for real-time processing on devices with limited resources. However, the inclusion of ResNet-based models in future studies would allow for a more comprehensive evaluation of lane detection performance across a broader spectrum of model architectures, potentially offering insights into the trade-offs between model complexity and detection accuracy in various application scenarios. In this section, details of the most commonly used deep learning models in lane detection problems, i.e. U-Net, SCNN, ENet, and ENet-SAD, are discussed.

U-Net: U-Net is a deep learning model designed for use in applications such as medical imaging and image segmentation. It derives its name from the U-shaped structure between the input and output, as can be seen in Figure 3. In the U-Net model, while the input and output dimensions are preserved, there is a bottleneck structure in the lower layers. U-Net includes parallel connections for each vertical layer between the input and output. These connections allow for the restoration of localization information and details lost during the encoder phase due to pooling in the decoder phase. Skip connections enhance the model's ability to preserve spatial localization and details. In a traditional convolutional deep learning model, as the number of filters in the image increases towards the lower layers, spatial information gradually weakens due to the reduction in image size resulting from pooling. The skip connections in the U-Net architecture play a key role in addressing this issue. U-Net typically consists of two main sections: the encoder and the decoder. The encoder section uses convolutional layers to extract feature maps from the input image. The decoder uses transposed convolutional layers and skip connections to convert these feature maps into a higher-resolution output [20].

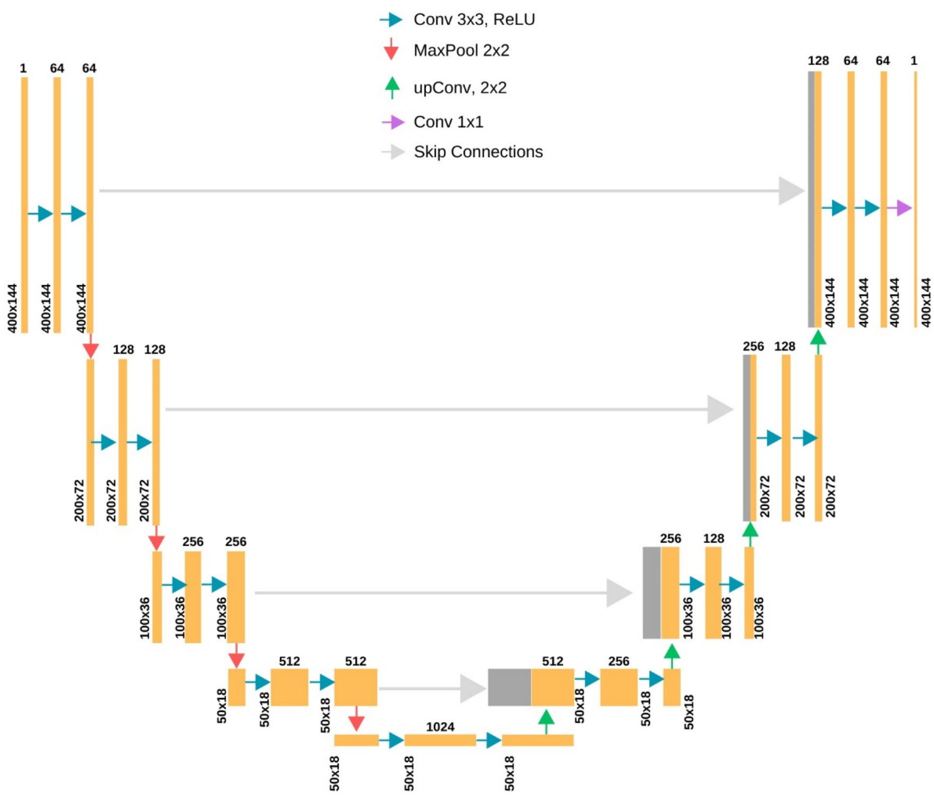

Figure 3. U-Net architecture [20]

SCNN: The Spatial Convolutional Neural Network (SCNN) is effectively utilized in scenarios where traditional Deep Neural Networks (DNNs) fall short, for instance, in highly complex traffic scenes, featuring multiple objects of different shapes and sizes [21]. Capable of delivering successful performance even in high-resolution images, the SCNN model also possesses the ability to accurately identify complex road structures. Fundamentally, DNNs use regional features to extract the meaning of visual content. However, these features can fail in objects that are common in traffic scenes, which may be strong in terms of shape but lack shape consistency. For example, traffic lane objects may not always be visible despite their distinct geometric structure, or their continuity and consistency of shape may diminish over time due to various degradations. In such cases, the prediction performance of DNNs tends to be low. To overcome this issue, the SCNN model has been developed to analyze the entire image globally and spatially [21]. SCNN employs a local spatial filter that includes a region around each pixel, representing the relationship of each pixel with its neighbors. SCNN then uses these spatial relationships to classify objects. Figure 4 illustrates the architecture of the SCNN model.

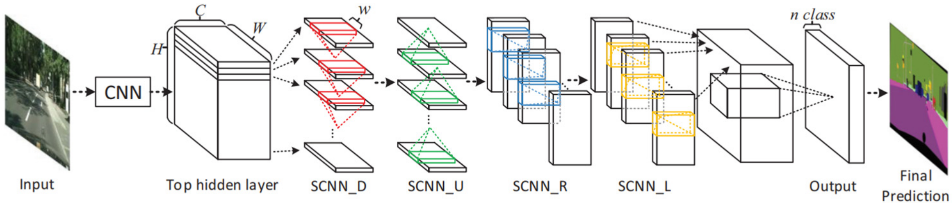

Figure 4. SCNN model architecture [22]

ENet: ENet is a CNN architecture originally designed for real-time semantic segmentation. This model is particularly known for being lightweight and fast, which enhances its usability in real-time applications. ENet employs a unique sequential block design along with asymmetric connections in the bottleneck area, thereby achieving an effective model with lower computational cost and fewer parameters. The model utilizes a starting layer before processing the input data, which transforms the input data into smaller-sized feature maps, accelerating the convolutional processes. ENet uses a structure called bottleneck blocks, as shown in Figure 5. These blocks perform deeper operations on smaller-sized feature maps. This approach reduces computational cost while increasing feature strength. It has been reported that the total number of trainable parameters is lower compared to SCNN [22]. ENet is designed to be lightweight and fast, primarily for real-time applications such as autonomous vehicles, drones, and similar tasks, necessitating the adaptation of the model according to the specific application or dataset [23].

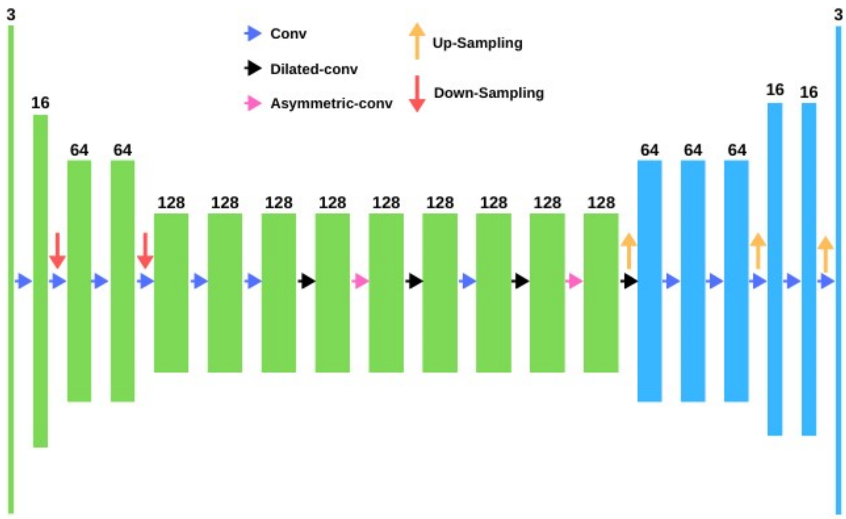

Figure 5. ENet block architecture [24]

ENet-SAD: ENet-SAD introduces a new approach that enhances the self-deep learning capabilities of lane detection networks. Hou et al. [25] developed a method for information distillation across different networks. They also expanded the function of information distillation to attention distillation in their works [26,27]. Zagoruyko et al. [27] proposed an activation and gradient-based attention distillation mechanism. The most notable difference of ENet-SAD from the method proposed by [27] is the transmission of attention information across layers through top-down distillation without the need for a teacher network. Therefore, the mentioned SAD method follows a different path from previous visual attention approaches that focused on feature weights. This method contributes to accelerating the inference time of the base model without external labeling or supervision. SAD allows the use of attention maps derived from the network's layers as distillation targets. This attention distillation mechanism is typically used to complement segmentation-based supervised learning. The strength of the SAD model stems from its ability to refine the contextual information of attention maps coming from different layers of the lane detection network. Without SAD, attention maps from different layers might only capture the scene's location and outlines, but with the addition of SAD, attention maps from lower layers are refined, achieving better representational power for deeper layers. The efficacy of SAD has been successfully demonstrated on popular lightweight models.

Training Details: Within the scope of this study, the image sizes were resized to 800x288 for the training of the SCNN, ENet, ENet-SAD, and U-Net models. The models were trained over a total of 10 epochs with a batch size of 8. The training duration of 10 epochs was chosen to mitigate the risk of overfitting that could occur with higher epoch values. For the loss function, a combined loss function consisting of cross-entropy, binary cross-entropy, and IOU (Intersection Over Union) loss was utilized for the SCNN, ENet, and ENet-SAD models. Conversely, the BinaryFocalCrossEntropy loss function was selected for the U-Net model. As for the optimization algorithm, Stochastic Gradient Descent (SGD) was used for the SCNN, ENet, and ENet-SAD models, while the Adam (adaptive moment estimation) algorithm was chosen for the U-Net model. The learning rate for the models during the training phase was fixed at 0.01. All deep learning models were trained and tested on a laptop with an RTX 3060 graphics card, 16 GB RAM, and an AMD Ryzen 5 5800H processor, running the Windows 10 operating system.

Metrics: In this section, the metrics used to evaluate the performance of the models in lane detection tasks are discussed. In this context, fundamental metrics such as accuracy, precision, recall, F1 score, and IoU have been utilized. These metrics will provide a comprehensive assessment and comparative analysis of each model's reliability and effectiveness in lane detection tasks. The formulas for these performance metrics involve metrics like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Understanding what these metrics represent in the context of lane detection is crucial because, as can be inferred from Figure 7, the lane detection problem is more complex than classic classification problems.

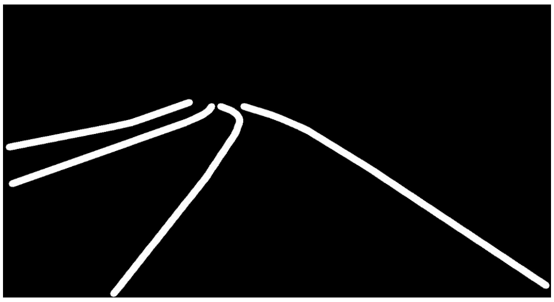

Figure 6. A ground-truth lane mask taken from the TuSimple dataset

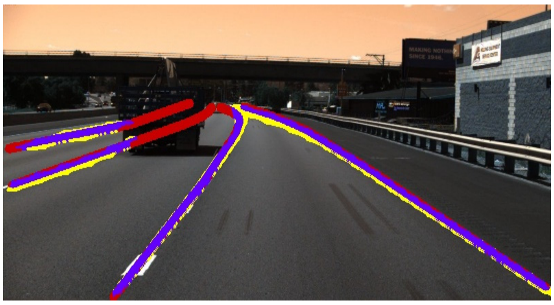

Figure 7. Visualization of SCNN Model Output on a Test Image from the TuSimple Dataset, Highlighting True Positives, False Positives, and False Negatives

In Figure 6, the lane masks shown in white are the ground-truth lane masks that we used in the model's training and expect the model to detect. The red lines in Figure 7 indicate lanes that the model was supposed to find but failed to detect. In this case, the pixels in the red lines are interpreted as false-negative. The yellow lane lines represent the lines that the model detected as lanes but do not have a counterpart in the basic truth. That is, the deep learning model has generated a false alarm, indicating a false-positive situation. The purple lines are where the model's predictions overlap or intersect with the basic truth, meaning the pixels in purple are considered true-positive. Theoretically, the success of a model is directly related to the density of the purple pixels. The remaining original pixels in the image, not artificially colored, are considered true-negative. This situation represents pixels that overlap with the black areas in Figure 6.

Accuracy: A metric that shows how accurately (TP+TN) a model classifies the total number of samples (TP+TN+FP+FN). It is also used to measure the overall performance of the model. However, if the dataset is imbalanced, accuracy may not properly reflect the model's performance. Equation (1) demonstrates how to calculate the accuracy formula. Essentially, this metric calculates the extent to which the pixels predicted by the model as lanes resemble/overlap with the ground-truth-masked pixels. This involves dividing the total number of similar/overlapping pixels by the total number of pixels in the image matrix.

$$A = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{1}$$

Precision: A measure of the model's level of generating false alarms. For instance, in the lane detection problem, having fewer yellow areas as indicated in Figure 7 is considered a measure of the model's higher precision. Thus, the precision value indicates how many of the pixels predicted as positive by the model are actually positive.

$$P = \frac{TP}{TP + FP} \tag{2}$$

Recall: An indicator of the model's rate of making fatal errors, namely false-negative predictions. In the lane detection problem, the goal is to minimize the density of the red pixels seen in Figure 7. In this case, the model's recall is considered to be increased. Typically, a high recall value is expected for the model in situations where the cost of false negatives is high.

$$R = \frac{TP}{TP + FN} \tag{3}$$

F1 Score: The F1 score is used as a metric that combines the precision and recall performance of a classification model. This score helps evaluate the overall classification performance of the model by balancing these two metrics.

$$F1 = \frac{2\ x\ (\text{Precision}\ x\ \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

Intersection over Union(IoU): As illustrated in Figure 8, IoU represents the ratio of the intersection to the union in set theory. Similarly, the IoU metric calculates the ratio of the area overlapped/intersected by the pixels predicted by the model with the ground-truth pixels (purple pixels) to the total area shown in all colors (purple + yellow + red) in Figure 7. Accordingly, a higher TP value indicates that the model has predicted lanes with higher accuracy.

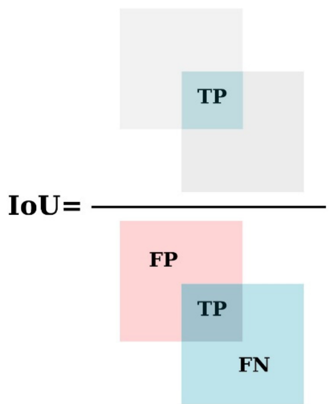

Figure 8. Depicts the IoU metric as a set

$$IoU = \frac{TP}{TP + FP + FN} \quad (5)$$

3. EXPERIMENTAL RESULTS

This study involves training and comparing four distinct lane detection models using the TuSimple dataset to evaluate their performance in detecting lanes. The evaluation of these models encompasses both objective measures, including standard success metrics and real-time inference speeds, and subjective assessments as depicted in Figure 10. Table 1 presents a comparative analysis of the lane detection capabilities of the U-Net, SCNN, ENet, and ENet-SAD models, utilizing metrics such as accuracy, precision, recall, F1-score, and IoU for performance evaluation. The models, after being trained over 10 epochs, were subjected to tests using a total of 2,782 images from the test directory of the TuSimple dataset. The findings are systematically displayed in Table 1, with the highest achieving scores in each model's context being prominently marked in bold across the appropriate rows and columns.

Table 1. Comparative analysis of lane detection model performances on the TuSimple dataset using accuracy, precision, recall, F1-score, and IoU metrics

Models	Accuracy	Precision	Recall	F1-Score	IoU
U-Net	**0.983**	0.805	0.734	0.767	0.623
SCNN	0.938	**0.946**	**0.900**	**0.919**	**0.879**
ENet	0.782	0.426	0.479	0.445	0.321
ENet-SAD	0.756	0.473	0.465	0.466	0.330

According to Table 1, the U-Net model achieves the highest accuracy score, followed by the SCNN model with an accuracy value of 0.938. In terms of precision, SCNN leads with a value of 0.946, followed by U-Net at 0.805 precision. ENet and ENet-SAD models exhibit weaker precision performance, with ENet-SAD being slightly more precise than the ENet model. Regarding the recall criterion, SCNN and U-Net secure the top two positions with values of 0.900 and 0.734, respectively. ENet-SAD falls behind ENet in terms of recall. In the context of F1-Score, SCNN again takes the lead with a value of 0.919, and U-Net follows with a score of 0.767. ENet-SAD outperforms the ENet model in this metric as well. For the Intersection over Union (IoU) criterion, SCNN ranks first with a score of 0.879, with U-Net in second place at 0.623. ENet-SAD and ENet models occupy the third and fourth positions, respectively. In summary, SCNN, with the highest scores, emerges as the best-performing model according to Table 1. The inclusion of the VGG-16 [28] model contributes to SCNN outperforming the other models. However, it's essential to note that the VGG-16 model's substantial size could potentially increase the inference time. Figure 9 provides insight into the average inference speed per image in milliseconds for the models. Notably, despite its high accuracy performance, SCNN exhibits a slower inference speed at 61.11 ms. ENet-SAD and ENet models follow with average inference times of 50.32 ms and 48.52 ms, respectively. U-Net stands out as the fastest model with an inference speed of 20.12 ms.

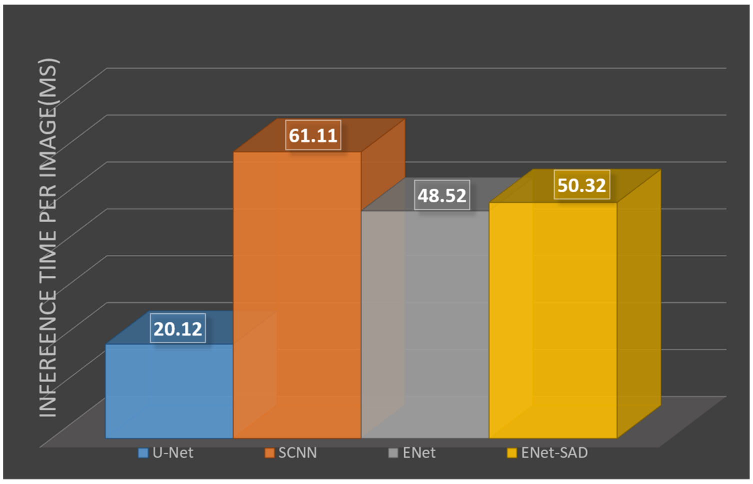

Figure 8. Average inference time comparison of lane detection models (U-Net, SCNN, ENet, ENet-SAD) on the TuSimple dataset

Figure 10 illustrates the lane detection performances of the SCNN, U-Net, ENet, and ENet-SAD models in the context of ground truth for four randomly selected test images from the TuSimple dataset. Purple-colored pixels indicate the success of lane detection as measured by IoU. As evident from Figure 10, the purple lanes are more prominent, especially in the SCNN and U-Net models. In the ENet and ENet-SAD models, red lanes predominate, indicating relatively weaker lane detection capabilities for these models. It is observed that these findings align with the objective results presented in Table 1.

Qualitative analysis reveals significant insights into each model's performance. SCNN and U-Net demonstrated superior lane pixel detection accuracy, while ENet and ENet-SAD showed higher false-negative rates. This difference stems from the models' architectural characteristics. SCNN's spatial design and VGG-16 backbone enable effective capture of long-range spatial relationships, crucial for continuous lane detection. U-Net's skip connections preserve fine-grained spatial information, contributing to its high accuracy. In contrast, ENet and ENet-SAD, designed for efficiency, struggle with false-negatives, likely due to their compact architectures limiting full context capture. These results highlight the trade-off between model size, detection accuracy, and real-time performance in lane detection applications.

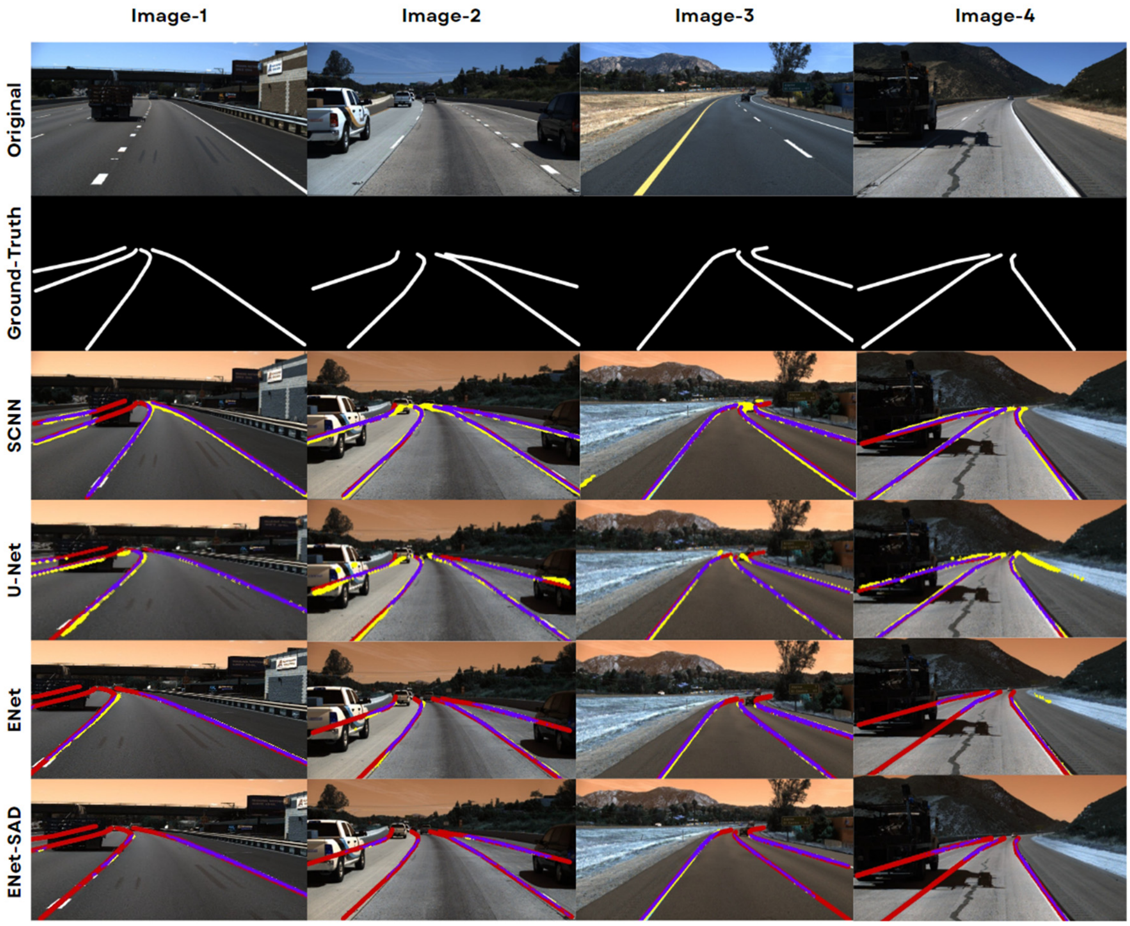

Figure 10. Visual comparison of lane detection results for U-Net, SCNN, ENet, and ENet-SAD models on sample images from the TuSimple dataset

4. CONCLUSIONS

In this study, four different lane detection models (U-Net, SCNN, ENet, and ENet-SAD) were trained on the TuSimple dataset, and their lane detection performances were compared. The models' performances were evaluated using success metrics such as accuracy, precision, recall, F1-score, IoU, and real-time inference time. As a result, it can be concluded that the SCNN model is the best-performing model, particularly due to its highest accuracy and strong performance in other metrics. However, considering the significance of inference time, especially for systems with limited processing power, the U-Net model may be preferred for real-time inference. Furthermore, as shown in Figure 10, both the SCNN and U-Net models demonstrate more significant lane detection than the ENet and ENet-SAD models, which is consistent with the objective results reported in Table 1. Our next work will include evaluating the models on a variety of datasets, building novel and faster lane recognition models, and implementing real-world lane detection applications.

5. REFERENCES

1. Sevim, M.A., Kircova, İ., Çuhadar, E., 2019. Yerel yönetimlerde akıllı şehir vizyonu: şehir yönetim araçları ve trendleri. Strategic Public Management Journal, 5(9), 109-126.
2. Richter, M.A., Hagenmaier, M., Bandte, O., Parida, V., Wincent, J., 2022. Smart cities, urban mobility and autonomous vehicles: How different cities needs different sustainable investment strategies. Technological Forecasting and Social Change, 184, 121857.
3. De-Las-Heras, G., Sanchez-Soriano, J., Puertas, E., 2021. Advanced driver assistance systems (ADAS) based on machine learning techniques for the detection and transcription of variable message signs on roads. Sensors, 21(17), 5866.
4. Marti, E., De Miguel, M.A., Garcia, F., Perez, J., 2019. A review of sensor technologies for perception in automated driving. IEEE Intelligent Transportation Systems Magazine, 11(4), 94-108.
5. Zakaria, N.J., Shapiai, M.I., Ghani, R.A., Yasin, M.N.M., Ibrahim, M.Z., Wahid, N., 2023. Lane detection in autonomous vehicles: A systematic review. IEEE Access.
6. Dillmann, J., den Hartigh, R.J.R., Kurpiers, C.M., Pelzer, J., Raisch, F.K., Cox, R.F.A., de Waard, D., 2021. Keeping the driver in the loop through semi-automated or manual lane changes in conditionally automated driving. Accident Analysis & Prevention, 162, 106397.
7. Haas, R.E., Bhattacharjee, S., Möller, D.P., 2020. Advanced driver assistance systems. Smart Technologies: Scope and Applications, 345-371.
8. Kukkala, V.K., Tunnell, J., Pasricha, S., Bradley, T., 2018. Advanced driver-assistance systems: A path toward autonomous vehicles. IEEE Consumer Electronics Magazine, 7(5), 18-25.
9. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, Erişim tarihi: 08.11.2023.
10. Kuyumcu, Z.Ç., Aslan, H., Yose, M.A., Ahadi, S., 2020. Türkiye'de trafik kazaları ve sürücülerin kazalardaki payı. Academic Perspective Procedia, 3(1), 694-702.
11. Novikov, A., Shevtsova, A., Vasilieva, V., 2020. Development of approach to reduce number of accidents caused by drivers. Transportation Research Procedia, 50, 491-498.
12. Mukhopadhyay, A., Murthy, L.R.D., Mukherjee, I., Biswas, P., 2022. A hybrid lane detection model for wild road conditions. IEEE Transactions on Artificial Intelligence.
13. Ying, Z., Li, G., Zang, X., Wang, R., Wang, W., 2016. A novel shadow-free feature extractor for real-time road detection. In Proceedings of the 24th ACM international conference on Multimedia, 611-615.
14. Khan, H.U., Ali, A.R., Hassan, A., Ali, A., Kazmi, W., Zaheer, A., 2020. Lane detection using lane boundary marker network with road geometry constraints. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 1834-1843.
15. Li, C., Shi, J., Wang, Y., Cheng, G., 2022. Reconstruct from top view: A 3d lane detection approach based on geometry structure prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4370-4379.
16. Giesemann, F., Payá-Vayá, G., Blume, H., Limmer, M., Ritter, W.R., 2017. Deep learning for advanced driver assistance systems. In Towards a Common Software/Hardware Methodology for Future Advanced Driver Assistance Systems, 105. River Publishers.
17. De-Las-Heras, G., Sanchez-Soriano, J., Puertas, E., 2021. Advanced driver assistance systems (ADAS) based on machine learning techniques for the detection and transcription of variable message signs on roads. Sensors, 21(17), 5866.
18. TuSimple, 2024. Tusimple dataset. https://github.com/TuSimple/tusimple-benchmark/wiki/
19. Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition, 6848-6856.
20. Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, 234-241. Springer International Publishing.
21. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X., 2018. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, 32, 1.

22. Hou, Y., Ma, Z., Liu, C., Loy, C.C. 2019. Learning lightweight lane detection cnns by self attention distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 1013-1021.

23. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., 2016. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv Preprint arXiv: 1606.02147.

24. Khoshdeli, M., Winkelmaier, G., Parvin, B., 2018. Fusion of encoder-decoder deep networks improves delineation of multiple nuclear phenotypes. BMC Bioinformatics, 19(1), 1-11.

25. Hou, Y., Ma, Z., Liu, C., Loy, C.C., 2019. Learning to steer by mimicking features from heterogeneous auxiliary networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 8433-8440.

26. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Tang, X., 2017. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3156-3164.

27. Zagoruyko, S., Komodakis, N., 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv Preprint arXiv: 1612.03928.

28. Yang, W.J., Cheng, Y.T., Chung, P.C., 2019. Improved lane detection with multilevel features in branch convolutional neural networks. IEEE Access, 7, 173148-173156.