# *Pay Attention to Features:* A Hybrid Deep Learning Architecture for Continuous Lane Segmentation

Chaz Allegra*
*Rowan University*
allegr32@students.rowan.edu

Tristan Daniel*
*Troy University*
tristand1220@gmail.com

Loukas Lazos
*University of Arizona*
llazos@arizona.edu

***Abstract*—Lane detection is a fundamental function in autonomous driving and advanced driver assistance systems. State-of-the-art methods apply geometric methods and deep learning (DL) models to identify and predict lane segments in a variety of environments such as highway, urban, and rural roads. DL methods have shown higher accuracy in more challenging environments where lanes are either poorly marked or obscured. However, lanes in many irregular road conditions such as vehicle occlusion, excessive shadowing, and lane marking deterioration remain challenging to detect. In this paper, we study the problem of lane segmentation and object placement on lanes. We propose a hybrid DL architecture that combines a U-Net convolutional neural network (CNN) framework with long-term short-term (LSTM) memory cells and multi-head attention transformers. Our architecture takes advantage of the fact that lane markings are continuous lines and therefore segmentation performance can be improved by using sequences of frames in a continuous driving scene.Extensive experiments on a benchmark dataset (TvtLane) demonstrate that, the proposed method outperforms the current state-of-the-art methods in lane detection with the U-LSTM demonstrating an improvement by 8% in F1 Score on the U-Net, and the U-Former showing an improvement of 11%. We paired our segmentation model with object placement model named the Lane Analyzer adds the capability of predicting which lane a given object is in. The Lane Analyzer with object detection functionality through YOLOv8 (You Only Look Once), was also able to achieve an astounding 96.7% testing accuracy on our object placement test set.**

*Keywords: Deep Learning, Lane detection, U-Net, Multi-head attention*

## I. Introduction

Autonomous driving has become a benchmark for what artificial intelligence can accomplish through recent advancements with machine learning algorithms [1]. The interest in advancing such systems have been largely invested in by the realms of both industry and academia alike. A primary objective that autonomy in driving strives for is maintaining a safe environment for all vehicles involved. In order to ensure a safe driving scene, autonomous vehicles should be able perceive their environment, including the position of other vehicles and themselves. The ability for these autonomous systems to better adapt to real-time driving scenes can be attributed to the introduction of more robust lane detection methods. With lane detection being an imperative module of an autonomous driving system, it is critical that such a framework should behave in a robust manner to handle problematic traffic scenarios such as vehicle occlusion, excessive shadowing, lane marking deterioration, etc.

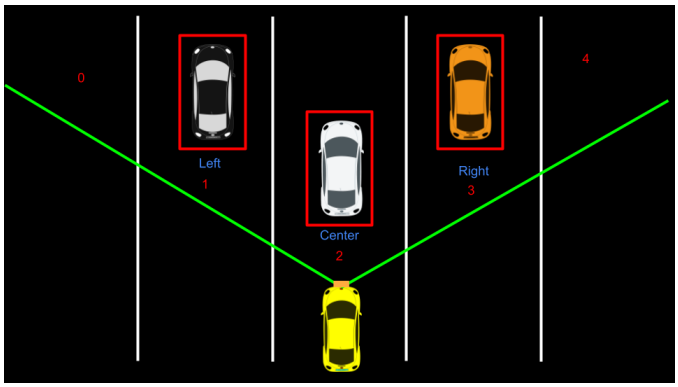



Fig. 1: An overview of lane detection with object detection for vehicle-lane placement

The purpose of lane detection algorithms is to accurately and efficiently identify lane instances within a given driving scene. The task of lane detection can be simplified into five essential markers to measure success.The correct number of lanes needs to be predicted.The positions of each lane marking should be predicted accurately according to the ground truth. The predicted lane lines must be continuous.The lanes located at the boundary of a driving scene should be accurately predicted.The lane detection model should achieve accurate predictions under various driving scenes, more considerably, under challenging situations.

In recent years, a number of lane detection methods have been developed that have produced exceptional results in completing such task, as reported in their respective literature [2]–[4]. Although numerous lane detection methods preform effectively in their task, these methods can fall short in producing accurate results when faced with complex driving scenes. Many of these current models lack components within their neural network architectures that reference important elements deduced from the previous driving scene frames. Another noteworthy inferiority is the use of a baseline neural network approach to lane detection. Although these models can be less computationally expensive, a simple CNN framework may not be robust enough to predict lane markings in extreme road conditions.A hybrid deep learning approach [5] to the

* The authors are undergraduate students who equally contributed to this work during a 2023 summer REU at the University of Arizona. This work was supported by NSF grant CCF-1852199, ARO grant W911NF1910050.

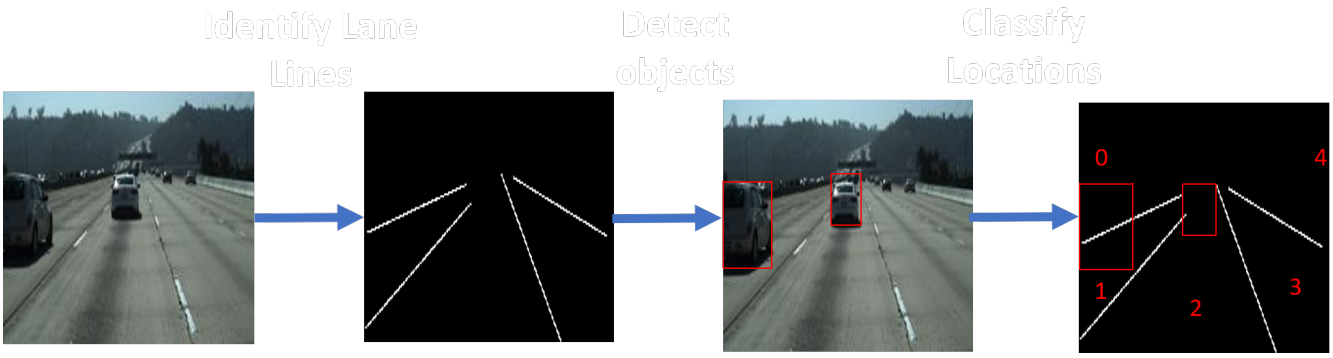



Fig. 2: First the image is taken from the car. Then the lane detection model predicts lanes and YOLOv8 predicts objects. The outputs are combined and sent to the Lane Analyzer for prediction

matter has proven to be more effective in handling such scenarios. In this paper, we focus on implementing a hybrid deep architecture method for lane detection, and propose a U-Net based architecture with multi-head attention convolution blocks as an approach for a robust lane instance detection. Our initial proposal architecture integrates two modules of two distinguishing neural networks with complementary merits, that is, a long short-term memory (LSTM) neural network, under a U-Net structure, to conduct lane detection in challenging driving scenes. We furthered improved this model by experimenting with self attention blocks to better analyze whether all components together, or as separate models would yield better results. These experiments resulted in a more refined model of U-Net structure with transformer self attention convolution blocks [6]. Accompanying our lane detection model is a lane analysis component through the functionality of YOLOv8's object detection mode [7] to enable our model the ability to identify where detected vehicles, with respect to lane instances,are within a given driving scene. The contributions of our research in the domain of lane detection are as follows:

- We achieve robust lane instance detection, particularly in challenging driving scenes.
- Our leading proposal introduces a U-Net based architecture with multi-head attention convolution blocks. This innovation enhances the lane detection performance by effectively capturing spatial dependencies and context information within the driving scene, which is shown to outperform the state-of-the-art approaches available.
- By investigating the integration of various modules and the use of separate models, we gain insights into the most effective configurations for achieving superior lane detection results.
- With the addition of a lane analyzer mechanism, we provide a more comprehensive understanding of the driving scene by offering spatial context information regarding the occupied lanes by detected vehicles relative to the ego-vehicle's location.

## II. Related Works

Lane segmentation is the act of classifying pixels in an image as a particular lane. Deep Learning has made tremendous progress over the years in lane segmentation. Convolutional Encoder/Decoder, or auto encoder, style networks are commonly found to have great success for this task. The basic convolutional auto encoder for image segmentation learns an image's features with convolution layers and uses its encoder/decoder architecture to use the understanding of features to reconstruct the image to fulfill its segmentation task. It first takes an image through the encoder portion of the network which consists of convolution layers and a form of down-sampling. The convolution layers will usually increase image channels as it goes through the encoder, while reducing the length and width dimensions. The encoded image will then go through bottleneck convolution layers before entering in the decoder. The decoder like the encoder consists of convolution layers, but with a form of up-sampling as opposed to down-sampling. The convolution layers will decrease in channels as it goes through and increase the length and width dimensions.

The U-Net was proposed in 2015 for biomedical image segmentation tasks. The U-Net is similar to the convolutional auto encoder, except it adds skip connections from the encoder part to the decoder. This helps the model retain feature information that may have been lost. The U-Net has proven to be an excellent image segmentation architecture, and a step up from the traditional auto encoder. Since then many models have been proposed for lane segmentation such as the E-Net, LaneNet, and SCNN.

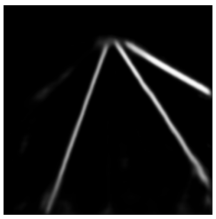
## III. Deep Learning Approach to Lane Segmentation

We wanted to take the U-Net architecture and add our own components to make the model better at continuous lane segmentation. With this we experimented with adding a short term memory component as an LSTM, a attention mechanism for better segmentation, and our final proposal, the U-Former, uses a transformer-like architecture as a different approach from the LSTM and the attention mechanism.

We then added a second model, called the Lane Analyzer, which would take in the input from the segmentation model (the gray scale image of the lanes) and the output bounding boxes from an object detection model (YOLOv8) and predict which lane the object was in. Taking in the lane segmentation model's output makes it much easier for the model to make

(a) Input,

(b) Ground Truth



(c) Model Prediction

Fig. 3: Image and Label Data (Left and Middle) with an example for the Model's Output(Right)

predictions as it removes all the excess noise of which is not needed to make a decision on what lane the object is in.

### A. Models

We based our models around the already effective U-Net model with hopes to improve its segmentation capabilities.This section will show the evolution of our model from a simple U-Net to a U-Net that pays attention to lane features, utilizes short-term memory, or utilizes previous outputs in order to better predict the lanes in a frame.

*1) U-LSTM:* We approached lane segmentation as a continuous task. With this we thought it wouldn't make sense to use a traditional U-Net as it is only capable of looking at single image inputs instead of learning sequences of inputs. Lanes gradually change and evolve as you drive. With this we decided to start by implementing an LSTM into our model to give it a short term memory component so that it could better evolve lanes as it sees them. Major problems in lane segmentation are curvatures and shadows. The hope was that the short term memory component would make the model more robust in dealing with those two issues since it would have memory of the lane from previous frames, and it could be more capable of evolving the lane.

We put the LSTM cells at the bottleneck of the U-Net. Our thought process was that once the image was encoded it would utilize short term memory in reconstructing the image to create its prediction. We felt it would be unnecessary to put an LSTM cell throughout the U-Net, but it remains untested.When Implementing the LSTM we decided to convert the traditional LSTM Cell to a convolutional LSTM Cell that kept all inputs and outputs in the image space. This was because we found when flattening and sending it the image through as a vector at the bottleneck there would be a drop in performance. Figure 1 below shows the Convolutional-LSTM Cell.

*2) U-Former:* The U-Former draws inspiration from the Transformer model's architecture. The Transformer model was, in many cases, an improvement upon the LSTM for sequential data. With this idea in mind we decided to make the U-Net as a a Transformer as a replacement to the LSTM in the bottleneck. Our U-Former essential works the same way as the U-Net with attention, but sends each output back in to the bottleneck block. The bottleneck block then takes in the encoded input image and the previous output and decodes from there. This model is also very similar in principle to the U-LSTM approach as the bottleneck portion utilizes the

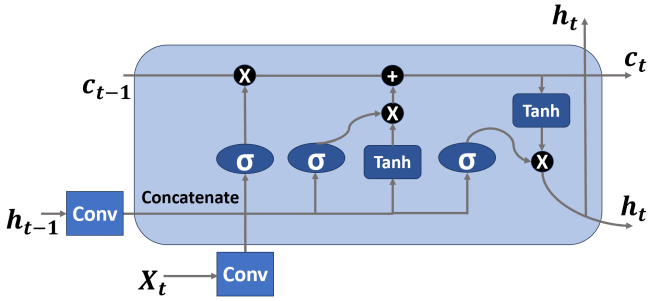



Fig. 4: Convolutional-LSTM Cell

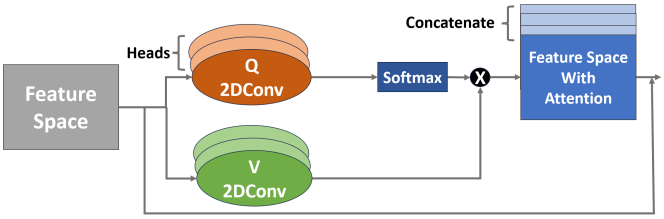



Fig. 5: Convolutional-Attention Block

encoded image and short-term memory so that the model can decode a better output.

In addition, the U-Former also takes inspiration from the Transformer model by making a version of the multi-headed self-attention mechanism, but using convolution layers and in the image-space as seen in figure 2. It utilizes Quarries and Values from the Transformer attention. Each head's Quarries and Values are a 1x1 convolution layers. The Quarry layer then goes through a row-wise softmax activation function to rank the most important features. The Value layer The attention mechanism was meant to help the model pay attention to the relatively small portions of the image that are lanes. The attention mechanism shows the model what features that matter most.

The attention blocks are computationally expensive since there are 2 convolution layers for each head. With this we decided to make each head a 1x1 convolution that is of size input/heads. The heads are then concatenated at the end to form the full feature space with applied attention. To also minimize computation we followed the attention block with a Depth-wise Separable Convolution with Linear bottleneck

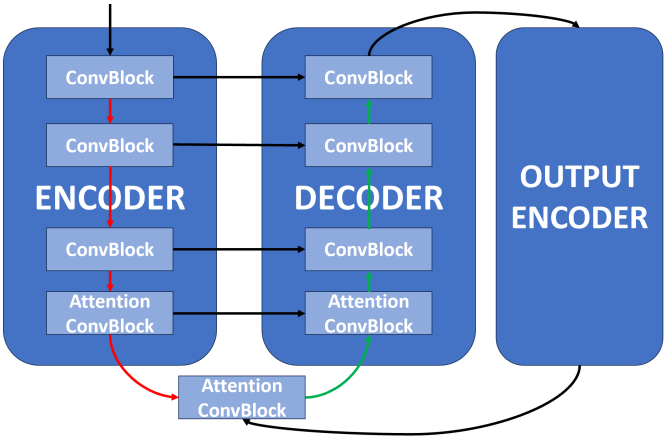



Fig. 6: U-Former Architecture

block, from MobileNet, instead of a normal convolution block. We chose to use this because these blocks proved to be effective with less computational cost than a typical convolution block.

## IV. Lane Analyzer

In addition to our lane detection model, we implemented a module with the purpose to make our lane detection more adaptable than the current state-of-the-art models available.The Lane Analyzer Model is an integral component of our lane detection system, designed to complement the lane detection module by providing spatial context and lane occupancy information for detected objects. Through the integration of YOLOv8 (You Only Look Once) for object detection [7], the Lane Analyzer Model offers real-time analysis of the driving scene.The Lane Analyzer takes YOLO's object detection to the next level. It allows the vehicle to not just gain an understanding of what objects are in the scene, but also where they are in reference to the road. It utilizes the Lane Segmentation model's output

We decided to take a deep-learning approach to this problem instead of a hard coded approach such as making whatever lane the bounding box is mostly in be the output because as lanes curve and vehicles drift side to side, their bounding box is at times mostly in a lane they are not actually in. To counteract this we made an intelligent model that could tell the where the model truly was in such scenarios.

We have made two approaches to the deep-learning Lane Analyzer. One approach draws the bottom-segment of the bounding box directly onto the lane segmentation output, and the other gives the model the lane segmentation output and the midpoint of the bottom-segment's coordinates. The first approach is fully an image classification approach, whereas the second approach uses vision and numerical analysis through fully connected layers.

### A. Bottom-Line Bounding Box Approach

YOLOv8's object detection provides bounding box placement around detected objects within a given scene. Instead of relying on the entirety of the bounding box, which encompasses a vehicle's entire area, only the bottom line segment of the bounding box is considered for analysis. The idea here is that large vehicles/objects can cause bounding boxes to appear to be in other lanes. With this we take the bottom line of the object's bounding box and paste it onto the lane segmentation's output. The lane segmentation's output removes all the excess noise so the Lane Analyzer can much more accurately determine what lane the object is in.

From here it is a simple 5 class image classification problem for the Lane Analyzer. Our model in this approach utilizes ResNet-50, a well known model in image classification, with an augmented output layer for our data. The augmented ResNet-50 would then take in the lane image with the bottom bounding box drawn in and outputs a class.

### B. Pixel-wise Midpoint Approach

Similar to our Bottom-Line Bounding Box Approach, the focus still relies on using the bottom line of a detected vehicle's bounding box. However, instead of employing the entire bottom line of the bounding box, the x and y pixel coordinates are attained and then concatenated as a single midpoint value.Similar to the approach above, the model takes in the lane segmentation model's output and send the image through the CNN portion of the Lane Analyzer. After the image has gone through the convolution layers and has been flattened, the midpoint coordinates are concatenated onto the flattened vector and then sent through fully connected layers to then be classified.

### C. Geometric Approach

A third geometric approach was hypothesized to support the previously identified lanes from the lane detection algorithm and uses various geometric properties to infer the lane in which the vehicle is positioned.Each lane can be represented as a set of lane boundary points or equations representing the lanes' geometries. For example, a lane could be represented as a pair of parallel lines (left and right lane boundaries) with the standard form for a linear equation:

$$Ax + By + C = 0 \tag{1}$$

The above equation represents a line on a 2D plane. Coefficients $A$ and $B$ determine the slope, and the constant $C$ represents the y-intercept. Variables $x$ and $y$ represent the x-axis and y-axis coordinates of a given point, respectively.For each detected vehicle, we can calculate its centroid within its bounding box. The centroid will represent the center point of the detected vehicle. With context to the above equation, we can train our proposed lane analyzer to perform geometric calculations to map detected vehicle centroids to their corresponding lanes and accurately determine which lane each vehicle lies in. For each detected vehicle's centroid, our model estimates the closest lane boundary points or lines by

calculating distance between the vehicle centroid and each lane boundary using the point-line distance formula:

$$Distance = \frac{|A_{x1} + B_{y1} + C|}{\sqrt{A^2 + B^2}}. \quad (2)$$

Lane assignment is then determined by which lane has the minimum distance value from the detected vehicle's centroid.

Making use of these purposed methods, we trained our Lane Analyzer model using ResNet50 [8], a state-of-the-art image classification neural network, to identify which lane a detected object is in, experimenting with each approach's method of obtaining input data to feed our proposed model. For testing accuracy, we implemented a cross entropy [9] loss function accompanied with an Adam [10] optimizer. The Bottom-Line Bounding box approach produced results of a 97.6 % accuracy in vehicle-lane placement. In reference for the remaining two proposals,the Pixel-wise Midpoint and Geometric approach, further configurations are being made to gather viable results.

## V. Results and Discussion

We tested a traditional convolutional auto-encoder, the U-Net, our U-LSTM, and our U-Former on the test set of our dataset. We used the F1 Score metric as a means to compare their effectiveness as accuracy alone is not a great metric since only a small portion of the output is lanes. When comparing the results of all the models we found that the U-LSTM was the most effective. The U-Former performed better than the auto-encoder and the U-Net alone, but it was not able to surpass the U-LSTM. The U-Former is still in its beginning stages as there are a lot of changes we are planning to make. The breakdown of results can be seen in figure 6.

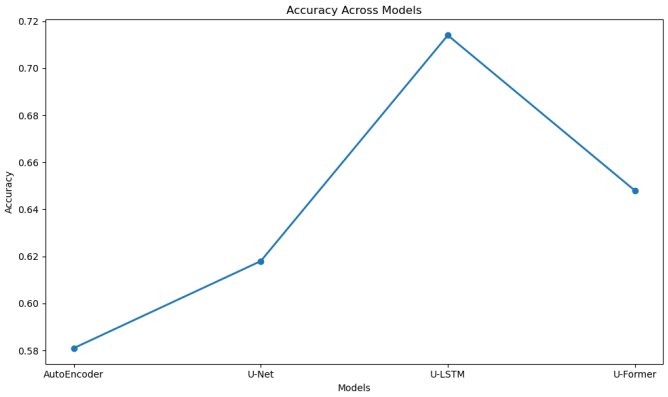



Fig. 7: F1 Score by Model

### A. Datasets

For our experimentation we used the tvtLane dataset [11] which is an alteration of the TU-Simple dataset. The data consists of 19,383 sequences, of 20 frames each, of front camera driving footage. The input images are 128x256 rgb images. There is a labeled ground truth for every 13th and 20th frame in each sequence. The ground truths of the data are grayscale images with pixels of the lane class being a value of 1 and nonlane class being 0. Our models would be trained to take in the input image (or frame) and output just the lanes (at a pixel level). TvtLane also uses data augmentation by doing 3-degree rotations, flipping, and the combination of both to quadruple the amount of data. For this experiment we only used the flipped version of the data due to technology constraints, but we plan to in the future and hope to see our models benefit from doubling the size of our data.

For the Lane Analyzer's data we went through TU-Simple's dataset and had YOLOv8 make predictions. We took the bounding box of each prediction and drew it on the label that corresponded with that prediction. When drawing the bounding boxes we ensured the pixel values for the bounding box differed from that of the lanes. We then labelled the correct lanes of each bounding box as the labels. The classes were: far left, left, center, right, far right. Anything beyond the left or right lane was considered far left or far right.

### B. Loss Functions

The two loss functions we used for our models are L2 distance and a weighted Binary Cross-Entropy. The reason a weighted BCE function is used is because the pixels that make up the lanes are far fewer than the pixels that make up the background. Both of these loss functions are widely used in segmentation tasks so we decided to use both of them and find which was the best. Below the loss functions can be seen with L2 distance being $f_1$, and weighted BCE being $f_2$. In the equations $y_p$ is the models prediction, y is the ground truth, and $w_{pos}$ is the weight added to the positive class (the lanes).

$$f_1 = \sqrt{\sum (y - y_p)^2} \quad (3)$$

$$f_2 = -(w_{pos} * y * log(y_p) + (1 - y) * log(1 - y_p)) \quad (4)$$

### C. Performance Metrics

This subsection analyzes the proposed models' performance through quantitative evaluations.To make quantitative comparisons for each model, we utilized a commonly used metric of the F1 Score [2], [5], [12], [13], denoted by equation:

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

In the equation above, precision refers to the correctly predicted positive samples from all the samples that the model classified as positive (both true positives and false positives). It is the ratio of true positives (*TP*) to the sum of true positives and false positives (*FP*), as denoted in the following equation:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Recall refers to the measure of all positive samples correctly predicted from the total number of positive samples present in the dataset. It is the ratio of true positives to the sum of true positives(*TP*) and false negatives(*FN*), illustrated in the equation:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

By combining precision and recall under a single metric, the F1 score, we can give equal weight to both measures. It provides a balanced evaluation of a model's performance by considering both false positives and false negatives. Our F1 score ranges from 0 to 1, with 1 being the more desirable score.

## VI. Summary and Next Steps

The results showed us that utilizing information from previous frames aided in improving the accuracy of the models. The U-LSTM performed the best over the other models, but there is still a lot of work to be done with the U-Former's architecture. We plan to implement and test changes to the U-Former. We would also like to increase the size of our dataset by utilizing the remaining data augmentations that tvtLane offers and add the more challenging testing conditions that tvtLane has. This would show us just how well our short-term memory would help in the most challenging of conditions and which types of conditions it benefits the model most. We would also like to train and test the models on CULane.

## References


[1] N. B. Chetan, J. Gong, H. Zhou, D. Bi, J. Lan, and L. Qie, "An overview of recent progress of lane detection for autonomous driving," in *2019 6th International conference on dependable systems and their applications (DSA)*. IEEE, 2020, pp. 341–346.

[2] Z. Chen, Q. Liu, and C. Lian, "Pointlanenet: Efficient end-to-end cnns for accurate real-time lane detection," in *2019 IEEE intelligent vehicles symposium (IV)*. IEEE, 2019, pp. 2563–2568.

[3] L. Liu, X. Chen, S. Zhu, and P. Tan, "Condlanenet: a top-to-down lane detection framework based on conditional convolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3773–3782.

[4] M. Philip, M. A. Kurian, R. Joseph, R. Sruthi, and S. Thomas, "A Computer Vision Approach for Lane Detection and Tracking," in *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*. IEEE, 2021, pp. 188–192.

[5] Y. Dong, S. Patil, B. van Arem, and H. Farah, "A hybrid spatial–temporal deep learning architecture for lane detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 1, pp. 67–86, 2023, publisher: Wiley Online Library.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[8] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia computer science*, vol. 132, pp. 377–384, 2018, publisher: Elsevier.

[9] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, pp. 310–318, 2005, publisher: Springer.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.

[12] J. Zhang, T. Deng, F. Yan, and W. Liu, "Lane detection model based on spatio-temporal network with double convolutional gated recurrent units," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6666–6678, 2022.

[13] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.