CNN based lane detection with instance segmentation in edge-cloud computing

Abstract—The domain of autonomous vehicle perception and advanced driver-assistance systems heavily relies on robust lane detection algorithms. While a plethora of sophisticated methods have emerged in recent years, their performance often deteriorates significantly in the face of adverse driving conditions. These conditions encompass situations with heavy shadowing, severe degradation of lane markings, and substantial occlusion by other vehicles. It is crucial to recognize that lane markings, by their very design, exhibit the inherent property of continuity on the road surface. This characteristic implies that lane information that remains elusive within a single captured frame holds the potential for retrieval through the incorporation of data from preceding frames. To address this challenge, we present a groundbreaking approach to lane detection that leverages the synergistic power of multiple frames captured within a continuous driving sequence. Our proposed method hinges on a meticulously crafted hybrid deep learning architecture. This architecture strategically combines the feature extraction prowess of a Convolutional Neural Network (CNN) with the adeptness of a Recurrent Neural Network (RNN) in exploiting the temporal dependencies embedded within the sequential frame data. Each frame within the sequence is meticulously processed by a dedicated CNN block, resulting in the generation of highly informative feature representations. These CNN-derived features, constituting a time series, are subsequently fed into the RNN for further feature learning and ultimately, the crucial task of lane prediction. The efficacy of our proposed method has been rigorously validated through extensive experimentation on expansive datasets. The results overwhelmingly demonstrate its superior performance in lane detection, particularly when confronted with challenging driving scenarios.

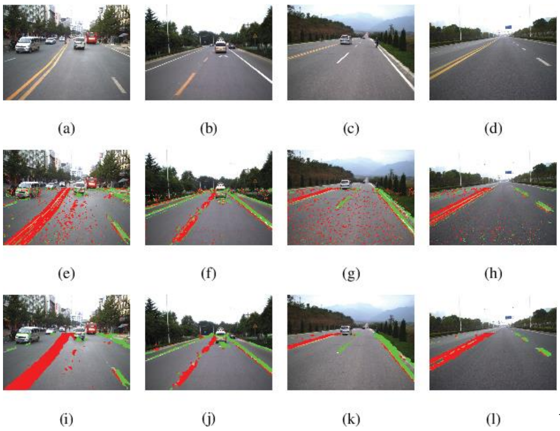
Index~**Terms—Convolutional neural network, LSTM, lane detection, semantic segmentation, autonomous driving. could we use any other terms in plcae of these**

I. INTRODUCTION

With the rapid advancement of high-precision optical sensors and electronic sensors, sophisticated computer vision and machine learning algorithms have made real-time comprehension of driving scenes increasingly attainable. Both academic and industrial research groups have dedicated substantial resources to crafting advanced algorithms for driving scene analysis, with a focus on autonomous vehicles or advanced driver assistance systems (ADAS). Lane detection stands out as a fundamental aspect of this research endeavor. By accurately determining lane positions, vehicles can navigate safely and avoid the peril of straying into adjacent lanes [1].

The National Natural Science Foundation of China provided financial support through grants 61872277, 61773414, and 41571437, while the Hubei Provincial Natural Science Foundation supported this research with grant 2018CFB482. The researchers, Q. Zou, Y. Yue, and Q. Wang, are affiliated with the School of Computer Science, Wuhan University, Wuhan 430072, P.R. China (Emails: {qzou, yhyue, qianwang}@whu.edu.cn). H. Jiang is associated with the Department of Electronic Information, Wuhan University, Wuhan 430072, P.R. China (Email: hwjiang@whu.edu.cn). Q. Dai is part of the Department of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, P.R. China (Email: qiyudai@whu.edu.cn). L. Chen is with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 518001, P.R. China (Email: chenl46@mail.sysu.edu.cn).

In recent years, numerous lane-detection methodologies have been introduced, showcasing remarkable efficacy as documented in scholarly literature. These approaches encompass a spectrum of techniques: some characterize lane structures using geometric models [2], [3], others frame lane detection as energy minimization challenges [4], [5], while some employ supervised learning tactics for lane segmenta



tion [6]–[9], among others. Nevertheless, many of these methods are constrained by their reliance on single-frame lane detection, resulting in subpar performance when confronted with challenging driving scenarios marked by dense shadows, significant road mark deterioration, or severe vehicle occlusion, as depicted in the top three images of Fig. 1. Such circumstances may lead to erroneous lane predictions or even failure to detect lanes altogether. One of the primary causes is the inadequacy of information gleaned from a single frame for precise lane detection or prediction.

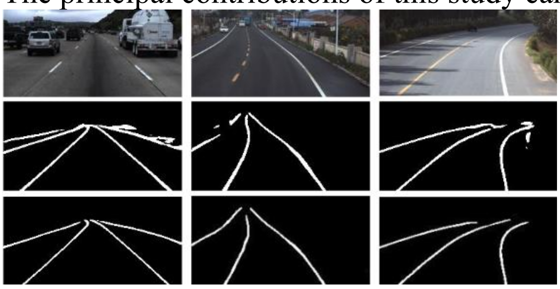
Road lanes typically manifest as continuous line structures on the pavement, appearing either as solid or dashed markings. Given the continuous nature of driving scenes and the substantial overlap between neighboring frames, the positions of lanes in adjacent frames are closely interrelated. Consequently, even in the presence of damage or degradation due to shadows, stains, or occlusion, lane prediction in the current frame can benefit from information extracted from multiple preceding frames. This observation motivates our exploration of lane detection utilizing images from continuous driving scenes.

Meanwhile, the emergence of deep learning has showcased remarkable prowess, often surpassing human capabilities in various computer vision tasks, including object detection [10]–[12], image classification/retrieval [13]–[15], and semantic segmentation [16]–[19]. Deep neural networks primarily comprise two categories: deep convolutional neural networks (DCNNs) and deep recurrent neural networks (DRNNs). DCNNs excel in image and video feature abstraction through multiple convolution stages, while DRNNs specialize in information prediction for time-series signals. Given the sequential nature of continuous driving scene images, ripe for processing with DRNNs, our focus turns to integrating these networks for lane detection.

However, utilizing raw frame images, such as an 800×600 image, as input would yield a vector dimension of 480,000, rendering it unsuitable for DRNNs due to the computational burden imposed by a vast number of weight parameters. To mitigate this challenge while preserving adequate information for lane detection, we employ DCNNs as feature extractors to abstract each image.

Drawing from the preceding discussion, we propose a hybrid deep neural network for lane detection leveraging continuous driving scene images. This hybrid network amalgamates DCNNs and DRNNs. At a macro level, our network operates as a DCNN, ingesting multiple frames to predict the lane of the current frame in a semantic-segmentation manner. Employing a fully convolutional DCNN architecture, comprising an encoder and decoder network, ensures output map alignment with the input image size. At a micro level, features abstracted by the DCNN encoder network undergo further processing by a DRNN, specifically a long short-term memory (LSTM) network. The LSTM incorporates information from continuous input frames, enhancing lane prediction performance within the semantic-segmentation framework.

The principal contributions of this study can be summarized as follows:



First, addressing the challenge of inaccurate lane detection in scenarios characterized by shadows, road mark degradation, and vehicle occlusion, we propose a novel method utilizing continuous driving scene images for lane detection. Leveraging information from multiple continuous images enhances lane prediction accuracy compared to single-image-based methods, particularly in handling challenging situations.

Second, through seamless integration of DRNNs with DCNNs, we present a novel fusion strategy. The DCNN, featuring an encoder and decoder with fully convolutional layers, abstracts each frame into a low-dimensional feature map. The LSTM, acting on each feature map as a full-connection layer in the time line, recursively predicts the lane. This fusion significantly enhances lane detection performance within the semantic-segmentation framework.

Third, we curate two new datasets for performance evaluation: one containing samples from 12 scenarios, each with hundreds of samples, and another focusing on rural roads, comprising thousands of samples. These datasets facilitate quantitative evaluation of various lane-detection methods, promoting research and development in autonomous driving. Additionally, we expand the TuSimple dataset by labeling more frames.

The remainder of this paper unfolds as follows: Section II delves into related work, Section III elucidates the proposed hybrid deep neural network, encompassing deep convolutional neural networks, deep recurrent neural networks, and training strategies. Section IV details experiments and results, while Section V concludes our work, briefly touching upon potential future avenues of exploration.

Drawbacks of 5 Review Papers:

1. "The variability of urban safety performance functions for different road elements: an Italian case study":

- Limited generalizability due to focus on a specific geographic location and dataset.
- Reliance on historical crash data may not reflect current road safety conditions.
- Challenges in accurately predicting crash frequencies for intersections.
- Lack of consideration for non-road-related factors contributing to crashes.
- Dependency on traditional statistical models may overlook complex interactions between variables.

2. "Instance segmentation on distributed deep learning big data cluster":

- High computational resource requirements limit accessibility to smaller organizations.
- Challenges in maintaining synchronization and communication among distributed nodes.
- Scalability issues with extremely large datasets impacting performance.
- Vulnerability to network failures or data inconsistencies during training.
- Limited applicability in resource-constrained environments due to dependency on distributed computing infrastructure.
-

3. "Research on Lane Line Detection Algorithm Based on Instance Segmentation":

- Performance may degrade under challenging lighting or weather conditions.
- Limited generalization to diverse lane marking styles or road geometries.
- Computationally intensive for real-time applications on embedded systems.
- May struggle with detecting lane markings obscured by obstacles or other vehicles.
- Dependency on the quality and diversity of the training dataset.

4."Securing DNN for smart vehicles: an overview of adversarial attacks, defenses, and frameworks":

- Lack of emphasis on practical implementation challenges and real-world deployment.
- Limited coverage of emerging adversarial attack techniques and defense strategies.
- Overemphasis on theoretical aspects without practical case studies or experiments.
- Insufficient discussion on ethical considerations and societal implications of adversarial attacks.
- Dependency on specific neural network architectures may limit applicability to diverse systems.
- "Survey on categorical data for neural networks":
- Narrow focus on categorical data handling techniques within neural networks.
- Limited discussion on broader applications or implications of deep learning algorithms.
- Lack of comparative analysis between different approaches for categorical data handling.
- Reliance on existing literature without significant contributions in methodology or experimentation.
- Limited exploration of challenges and future directions in the field of neural network-based categorical data analysis.

5."Survey on categorical data for neural networks":

- Narrow focus on categorical data handling techniques within neural networks.

- Limited discussion on broader applications or implications of deep learning algorithms.
- Lack of comparative analysis between different approaches for categorical data handling.
- Reliance on existing literature without significant contributions in methodology or experimentation.
- Limited exploration of challenges and future directions in the field of neural network-based categorical data analysis.

Structure

The research paper on CNN-based lane detection techniques using cloud computing unfolds through a structured framework aimed at comprehensive exploration. It commences with an introduction that underscores the criticality of robust algorithms in autonomous vehicle perception and advanced driver-assistance systems. This section sets the context for subsequent discussions. The methodology segment delves into the proposed approach, elucidating the fusion of CNNs for robust feature extraction with RNNs, specifically LSTM networks, to model temporal dependencies in sequential frame data. Detailed techniques such as lane embedding segmentation and the utilization of Convolutional LSTM (ConvLSTM) networks for video analysis are expounded upon. Experimental evaluation forms a pivotal part of the paper, showcasing the superior accuracy and efficiency of the proposed method over traditional approaches. Finally, the applications section explores the relevance of precise lane detection in autonomous vehicles and other pertinent domains, highlighting its pivotal role in ensuring safe and reliable navigation. Throughout the manuscript, discussions on challenges, limitations, and future directions offer insights into the evolving landscape of CNN-based lane detection, with particular emphasis on the integration of cloud computing to bolster scalability and processing capabilities.

II. LITERATURE REVIEW

Lane Detection Methods Over the past two decades, numerous studies have explored lane detection and prediction [20]–[23]. These approaches can be broadly categorized into traditional methods and those based on deep learning. In this section, we provide a brief overview of each category.

Traditional methods. Before the emergence of deep learning technology, lane detection predominantly relied on geometric modeling, specifically line detection or fitting, utilizing basic features such as gradient, color, and texture, often coupled with energy minimization algorithms.

Geometric modeling. Many techniques in this category employ a two-step process, starting with edge detection followed by line fitting [3], [24]–[26]. Various gradient filters are used for edge detection, such as Gaussian, Steerable, and Gabor filters [24]–[26]. Additionally, color and texture were explored for lane detection or segmentation [28], [29]. Hough Transformation (HT) models were commonly used for line fitting, while curve-line fitting was also employed [30], [31]. Stereo vision techniques were also utilized for geometry modeling-based lane detection [32]–[34], enabling estimation of lane distance.

Energy minimization. Conditional random field (CRF) serves as a common energy-minimization model for solving various association tasks, extensively utilized in traffic scene understanding [4]. CRF has been applied to detect multiple lanes by optimizing the association of lane marks in complex scenes [5]. By defining unary and clique potentials, the energy of the graph is computed and refined through energy minimization. This approach can also be integrated into the search for optimal modeling results in lane fitting. An active line model was introduced in [35], constructing an energy function with external and internal components to enhance lane tracking. For continuous lane tracking across frames, the Kalman filter is a prevalent choice, proficient in locating lane markings and estimating lane curvature [36]–[38]. Additionally, the particle filter is widely used for tracking multiple lanes [39]. Combining these filters into a Kalman-Particle filter, as presented in [40], achieves more stable performance in simultaneous lane tracking. In the context of slice-by-slice medical image processing, leveraging information from previous images aids in enhancing results [41], particularly by creating a region of interest in upcoming images based on segmentation of adjacent ones. Deep-learning-based methods have propelled lane detection research to new heights, with various approaches categorized into four groups.

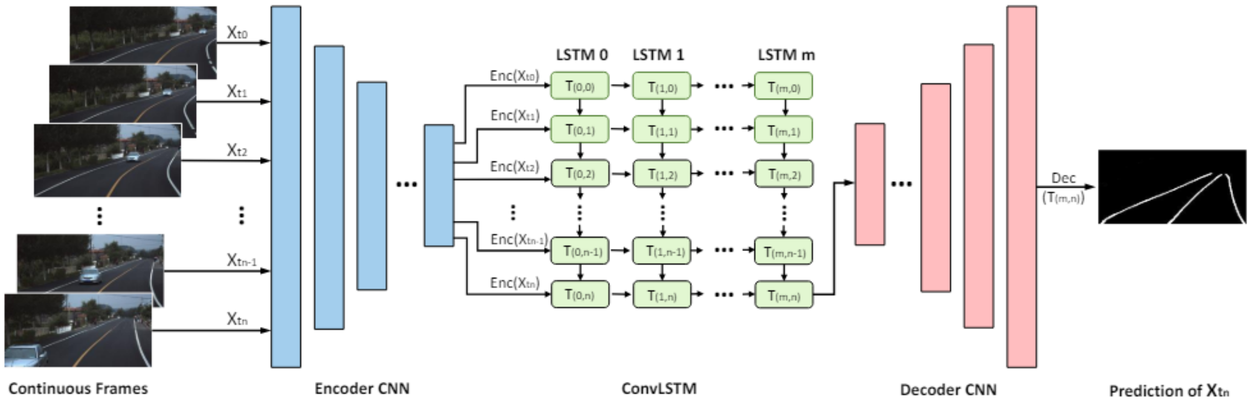


Fig. 2: Architecture of the proposed network.

Encoder-decoder CNN. Typically employed for semantic segmentation [18], lane detection in [42] explores a transfer learning framework. Based on the road scene object segmentation task, the end-to-end encoder-decoder network, trained on ImageNet, was established. LaneNet, introduced in [43], extends SegNet with two decoders: a segmentation branch for lane detection in a binary mask and an embedding branch for road segmentation. The output feature map necessitates clustering and curve-fitting algorithms for final results. Real-time road marking segmentation in [44] addressed the scarcity of labeled data with a weakly-supervised strategy, utilizing additional sensor modalities to generate annotated images, enabling training of a U-Net inspired network for accurate detection.

FCN with optimization algorithms. Lane detection often employs fully-convolutional neural networks (FCN) alongside optimization algorithms. In [45], sliding windows with overfeat features parse driving scenes, detecting lanes as a separate class. [6] utilizes CNN with fully-connective layers as a decoder, employing hat-shape kernels for lane edge inference, refined by RANSAC. An enhanced version in [47] processes data from point cloud registration, road surface segmentation, and orthogonal projection using CNN. Semi-artificial image generation in [48] facilitates lane detection with a CNN having fully-connected layers and softmax classification. VPGNet, proposed in [8], features a shared feature extractor and four similar branch layers, employing optimization algorithms for lane detection, lane-marking identification, and vanishing-point extraction. Spatial CNNs in [50] enhance CNN performance in detecting continuous shape structures by employing slice-by-slice convolutions within feature maps, allowing message passing between pixels across rows and columns. [9] considers spatial and temporal constraints for lane position estimation.

‘CNN+RNN’. Acknowledging the continuity of road lanes, a method amalgamating CNN and RNN was introduced in [7]. Initially, a road image is divided into continuous slices. Then, a convolutional neural network serves as a feature extractor for each slice. Subsequently, a recurrent neural network infers the lane from feature maps obtained on the image slices. This method reportedly yields superior results compared to using CNN alone. However, the RNN in this approach can only model time-series features in a single image, and the RNN and CNN are distinct blocks.

GAN model. The generative adversarial network (GAN) [51], comprising a generator and a discriminator, is also utilized for lane detection [52]. Here, an embedding-loss GAN (EL-GAN) is proposed for semantic segmentation of driving scenes. In this setup, the lanes are predicted by a generator based on the input image and evaluated by a discriminator with shared weights. The advantage of this model lies in the thin and accurate predicted lanes, avoiding the marking of large soft boundaries typically observed with CNNs.

Unlike the aforementioned deep-learning-based methods, the proposed approach conceptualizes lane detection as a time-series problem, detecting lanes across multiple continuous frames rather than just one current frame.

With richer

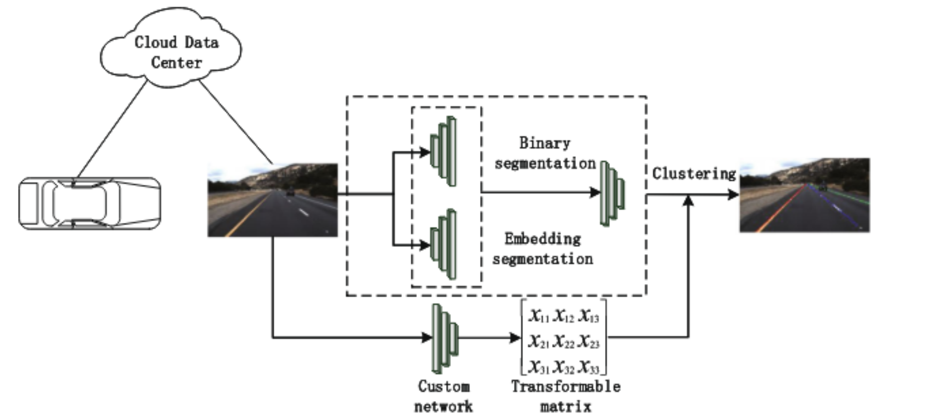


Fig. 1 Model framework

information, the proposed method achieves robust lane detection in challenging conditions. Additionally, it seamlessly integrates CNN and RNN, which

ConvLSTM for Video Analysis LSTM, a fundamental component in deep learning, excels in processing temporal data. Convolutional LSTM (ConvLSTM), equipped with convolution operations, is extensively utilized in video analysis due to its feedback mechanism on temporal dynamics and its ability to abstract image representation [53]. Various approaches leverage ConvLSTM as a fundamental building block.

Specifically for semantic video segmentation, ConvLSTMs are widely employed to efficiently process two-dimensional time-series inputs. In [54], the video segmentation task was restructured into a regression problem, with a ConvLSTM inserted between the convolutional encoder-decoder to forecast each frame. [55] introduced a convolutional GRU to merge the representations from a standard convolutional appearance network and an optical-flow motion network. Another configuration was proposed in [56], positioning a two-stream ConvLSTM after the encoder to generate spatial and temporal region proposals. [57] constructed a spatio-temporal saliency learning module by incorporating a pyramid dilated bidirectional ConvLSTM (PDBConvLSTM) after the spatial saliency learning module. ConvLSTMs are also employed in various other video-analysis tasks such as anomaly detection [58], sequence prediction [59], and passenger-demand prediction [60] [61].

III. **Methodology**

Lane embedding segmentation The lane loss embedding branch function in the dual branch network outputs pixel embeddings for each lane. This ensures that pixel embeddings of the same lane are grouped together while those of different lanes are maximally separated. Two terms achieve this: the variance term (Lvar) pulls embeddings towards the lane's average, and the distance term (Ldist) pushes cluster centers apart. Pull force acts when distance from embedding to center exceeds δv, while thrust force acts when distance is less than δd.

$$\begin{cases} L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_C}\sum_{i=1}^{N_C}[\,\mu_c - \mu_i - \delta_v]_+^2 \\ L_{sum} = \sum_{CA,CB=1}^{C}[\,\delta_d - \mu_{CA} - \mu_{CB}]_+^2 \\ L_{dist} = \frac{1}{C(C-1)}L_{sum} \end{cases}$$

In equation (1), C represents the number of lane lines, NC denotes the number of elements in C, μi signifies the lane pixel embedding, and μc represents the average embedding of C. The term "μc - μi" indicates the average embedding and pixel embedding distance. CA and CB denote two lane lines, while μCA - μCB represents the average embedding distance between the CA lane line and CB lane line. The expression [δd - μCA - μCB - δv]^2+ evaluates to 0, and δd - μCA - μCB yields the largest positive value. Similarly, [μc - μi - δv]^2+ represents the maximum positive value between 0 and μc - μi - δv. Here, δv signifies the threshold of the variance term, and δd denotes the threshold of the distance term. The total loss L = Lvar + Ldist. Once the network converges, the lane pixel embeddings cluster together, ensuring that the distance between each cluster exceeds δd, and the radius of each cluster is less than δv. In Figure 2a and b, lane images are depicted, while (c) and (d) show the images generated by the example embedding segmentation

Clustering:

Clustering involves dividing a large unlabeled dataset based on inherent similarities into distinct categories with inherent differences. Data within the same category exhibit high similarity, while data across different categories show low similarity. To enhance lane line detection's robustness across various environments, clustering analysis

on lane line pixels is necessary, embedding pixels from the same lane line together and distinguishing those from different lanes. Clustering ensures recognition accuracy even in low-light and challenging traffic conditions.
In this study, clustering is achieved iteratively, integrating the clustering algorithm with the loss function. Equation (1) guides the process, where, with 6δv < δd, a random lane with a radius of 2δv and its surrounding threshold are selected to embed all pixels belonging to the same lane. This process repeats until all lanes are embedded and assigned to one lane. To prevent outlier selection, mean shift initially moves toward the cluster center before setting the threshold.

3.1 Pixel fitting:

$$f(y') = \alpha y'^2 + \beta y' + \gamma$$

$$w = \left(Y^T Y\right)^{-1} Y^T x'$$

The two-branch network outputs pixels for each lane line. Fitting a polynomial directly in the input image space isn't optimal, especially for curved lanes requiring higher-order polynomials. Common fitting methods include RANSAC, Bezier curves, spline interpolation, and least squares polynomial fitting. This study first transforms the image using inverse perspective, projecting it into a "bird's eye view" where lanes are parallel, allowing fitting with a second or third-order polynomial. However, using a fixed transformation matrix H calculated once for all images leads to errors with ground-level changes, where the vanishing point moves. To address this, a neural network with a custom loss function is trained to predict the parameters of H. The transformed lane points can then be optimally fitted with polynomials. This prediction, based on the input image, enables the network to adjust projection parameters for changing ground planes, ensuring accurate lane line fitting. In equation (2), H has six degrees of freedom represented by variables a-f. Placing 0 enforces constraints, keeping horizontal lines horizontal during transformation.

$$H = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & f & 1 \end{bmatrix}$$

The network architecture is compact, consisting of a 3×3 convolutional layer block, Batchnorm, ReLU activation functions, and max-pooling to reduce size, followed by 2 fully connected layers. Lane pixels are transformed using a custom function network output transformable matrix before fitting. The road image undergoes an inverse perspective transformation to generate top-view lane pixels, as shown in equation (3): P = HP. Here, P represents transformed lane line pixels, and Pi represents real lane pixels. Xi and Yi denote the horizontal and vertical coordinates of real lanes, while X'i and Y'i represent transformed lane pixel coordinates. To obtain lane position at a given y, real lane pixel Pi=[−, yi, 1]T is transformed to P'i=[−, yi, 1]T after perspective transformation, predicting lane pixel points P*i=(x*i, yi)* at each yi position. A third-order polynomial fits the predicted lane pixel points P*i using least squares, represented by equations (4) and (5).
In equations (4) and (5), α, β, and γ represent the optimal parameters given, while T denotes the transpose of the matrix. The least square method, denoted by w =[α, β, γ] T, represents the letter "x," which comprises the abscissa combination of lane coordinate points, represented as x =[x1, x2, x3, ⋯ xn] T. Similarly, y =[y1, y2, y3, ⋯ yn] T represents the transformable parameter matrix, combining the ordinate points of lane pixel transformation. The matrix Y, given as ⎡⎣ y21 y1 1 ⋯ y2n y1 1 ⎤⎦, is crucial in the least squares formula. By fitting the predicted lane pixels P*i =[x*i, yi, 1]T through perspective transformation to the road lane image, lane pixels P*i are obtained at various lane positions. Equation (6) showcases the perspective transformation formula of the cluster loss function: P*i = H^(-1)P*i. Here, P*i =[x*i, yi, 1]T denotes a lane pixel projected onto the input road lane image, while P*i =[x*i, yi, 1]T represents the predicted lane pixel point, and H^(-1) symbolizes the perspective transformation matrix. To train the output of the custom function network H, aiming for the most suitable transformation matrix H for pixel fitting polynomials, a loss function is constructed. Since lane fitting is accomplished using the least squares method, the loss function is differentiable, depicted as equation (7).

$$Loss = \frac{1}{N} \sum_{i=1}^{N} (x_i^* - x_i)^2$$

3.2 Experimental evaluation

This paper utilizes the Tusimple dataset for both training and testing purposes. The dataset comprises lane line data captured at various times throughout the day, encompassing scenarios with 2, 3, and 4 lanes. To assess the lane segmentation task's final outcome, an evaluation metric is necessary to gauge the test performance. F-Measure is selected as the evaluation index due to its comprehensive consideration of precision and recall in the experiment. All lane line pixels are classified as positive samples (P), while background pixels serve as negative samples (N). The accuracy of pixel prediction is determined by comparing the predicted image result with each pixel of the actual label. Formulas for evaluating image prediction results using the F-Measure index are presented in equations (8) and (9).

Prediction Evaluation Accuracy indicates the ratio of correct predictions to total predictions. Recall measures model coverage by detecting the proportion of existing targets. TP signifies correctly detected positive instances, FP denotes falsely detected positive instances, FN represents falsely detected negative instances, and β, set to 0.8, emphasizes Recall. This mitigates misidentification of lane areas. Experimental Setup This study operates on an Ubuntu 16.04 OS, utilizing an NVIDIA GEFORCE 920M GPU for Tensorflow-based code execution. Training involves regularizing and randomly cropping and rotating input images. Image size is standardized to 256 × 512, with a batch size of 6 for simultaneous training. Network parameters are optimized using gradient descent with a momentum parameter of 0.9, weight decay of 1 × 10−4, and an exponential decay learning rate update strategy.

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}$$

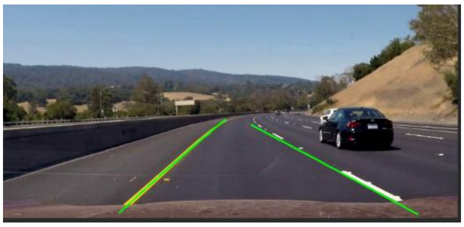
$$F_{\beta} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

3.3 Experimental outcomes

This article conducted experiments in an Ubuntu 16.04 environment, utilizing an NVIDIA GEFORCE 920M GPU for Tensorflow implementation. During training, input images were standardized, randomly cropped, and rotated. Image size was uniformly set to 256 × 512, with a batch size of 6 for simultaneous training. Gradient descent optimized network parameters, with a momentum parameter of 0.9, weight decay of 1 × 10−4, and exponential decay learning rate update strategy (lr = lrinit × (1 − iter/N)power). In this equation, lrinit is the initial learning rate

$$lr = lr_{init} \times \left(1 - \frac{iter}{N}\right)^{power}$$

(0.0005), N is the total training iterations (set to 80010), iter is the current iteration number, and power is the attenuation coefficient (set to 0.9). Testing employed images from the Tusimple dataset, demonstrating superior performance compared to traditional and current deep learning algorithms, especially in challenging scenarios like low lighting, shadows, missing lines, and curves. When deployed in varied road environments, our model showed better adaptability than others, as detailed in Table 1. Performance metrics revealed that each frame took 16.6ms during lane instance segmentation and clustering, achieving 60.2 FPS. Lane fitting required 2.4ms per frame, reaching 416.7 FPS.



Overall, the model processed at 52.6 FPS, demonstrating efficient performance. Figures 3a and b depict lane images in low-light environments, with (c) and (d) showing corresponding detection results, illustrating the model's robustness in such conditions. Figures 4a and b display lane images in degraded road conditions, with (c) and (d) showing corresponding detection results.

Table 1 Comparison of test results of different models

Models	Insufficient light	Shadow occlusion	Missing lane line	Curve lane line	Normal road
VGG-FCN	0.9369	0.9311	0.8837	0.9259	0.9308
SCNN	0.9612	0.9713	0.9412	0.9315	0.9671
U-Net	0.9510	0.9613	0.9126	0.9168	0.9647
Tradition	0.9213	0.9122	0.8635	0.8874	0.9171
Ours	0.9662	0.9785	0.9588	0.9410	0.9757

IV **APPLICATIONS**

4.1 *Autonomous Vehicles*:

Lane Navigation: In autonomous driving systems, CNN-based lane detection is fundamental for understanding road geometry and ensuring safe navigation. By continuously analyzing the road's markings, the vehicle can accurately determine its position within the lane and make decisions accordingly, such as steering adjustments or lane changes.

- *Instance Segmentation*: Instance segmentation takes lane detection a step further by not only identifying lane markings but also categorizing them into different instances, such as distinguishing between solid lines, dashed lines, or curbs. This level of detail allows the vehicle to understand the nuances of the road environment, such as whether it's safe to overtake or if there are lane-specific restrictions.

4.2.*Traffic Monitoring and Management*:

- *Real-time Analysis*: Edge-cloud computing facilitates deploying CNN models at the edge of the network, enabling real-time analysis of traffic conditions. Lane detection plays a crucial role in this process by providing continuous updates on lane occupancy, vehicle speeds, and traffic flow patterns.
- *Lane Violation Detection*: By accurately detecting lane markings and analyzing vehicle trajectories, the system can identify instances of lane violations, such as illegal lane changes or driving on the shoulder. This information can be used to enforce traffic laws and improve overall road safety.
-

Optimized Signal Timing: Using data obtained from lane detection, traffic signal timings can be dynamically adjusted to respond to changing traffic conditions. For example, if lane occupancy is high in one direction, the system can prioritize green lights accordingly to alleviate congestion and improve traffic flow.

4.3. *Driver Assistance Systems*:

- *Lane Departure Warnings*: CNN-based lane detection systems can provide timely alerts to drivers when they unintentionally drift out of their lane, helping prevent accidents caused by distractions, drowsiness, or inattention. These warnings can be visual, auditory, or haptic, depending on the vehicle's interface and the driver's preferences.
- *Assistive Steering*: In situations where a driver's actions pose a risk, such as veering into another lane or off the road, assistive steering systems can intervene to correct the vehicle's trajectory. Lane detection provides the necessary input for these systems to determine the appropriate steering adjustments needed to keep the vehicle safely within its lane.

4.4. *Augmented Reality Navigation*:

- *Intuitive Guidance*: By overlaying lane information onto the driver's view through augmented reality displays, navigation systems can provide clear and intuitive guidance, enhancing situational awareness and reducing cognitive load. For example, lane boundaries, upcoming exits, or lane-specific hazards can be highlighted directly in the driver's line of sight, making navigation more effortless and efficient.

- *Lane-specific Information*: Augmented reality overlays can include additional lane-specific information, such as real-time traffic conditions, lane closures, or lane-specific speed limits. This contextual information helps drivers make informed decisions and navigate complex road environments more confidently.

4.5. *Smart Infrastructure Maintenance*:

- *Road Condition Monitoring*: CNN-based lane detection can be used to continuously monitor the condition of lane markings, detecting signs of wear, fading, or damage. By identifying areas in need of maintenance, authorities can prioritize resources and schedule repairs proactively, minimizing the risk of accidents and ensuring road safety.
- *Scheduled Maintenance*: By analyzing lane detection data over time, maintenance activities can be scheduled based on usage patterns and environmental factors. For example, roads with heavy traffic may require more frequent maintenance to ensure optimal visibility of lane markings, while roads in less populated areas may need less frequent inspections.

4.6. ***Smart Transportation Systems*:**

- *Traffic Prediction*: Historical lane detection data, combined with other traffic-related information, can be used to predict future traffic patterns and congestion hotspots. This predictive capability allows transportation authorities to proactively plan and allocate resources, such as adjusting public transit schedules or deploying additional traffic management measures.
- *Route Optimization*: Real-time lane detection data enables dynamic route optimization, where navigation systems can suggest alternative routes based on current traffic conditions. By avoiding congested areas and selecting the most efficient paths,
drivers can save time and reduce fuel consumption, contributing to overall traffic efficiency and environmental sustainability.

4.7. ***Surveillance and Security*:**

- *Anomaly Detection*: CNN-based lane detection can be integrated into surveillance systems to identify suspicious behaviors or security threats, such as vehicles driving in the wrong direction, sudden lane changes without signaling, or unauthorized vehicles entering restricted areas. These anomalies trigger alerts for security personnel to investigate further and take appropriate action.
- *Integration with Surveillance Systems*: Lane detection data can be combined with other surveillance data, such as video feeds from cameras and sensors, to provide comprehensive monitoring of critical infrastructure areas. By analyzing lane-specific information alongside other contextual data, security systems can detect and respond to security incidents more effectively, enhancing overall security measures.

4.8. ***Environmental Monitoring*:**

- *Pollution Detection*: Edge-cloud computing enables the deployment of CNN models for monitoring pollution sources along roadways, such as vehicle emissions or industrial activities. By analyzing lane detection data in conjunction with environmental sensors, authorities can identify pollution hotspots and implement targeted mitigation measures to improve air quality and public health.
- *Vegetation Health Monitoring*: Lane detection can also be used to monitor the health of vegetation in roadside areas, such as trees and plants. Changes in vegetation patterns, detected through lane detection data, may indicate environmental stressors such as pollution, drought, or disease outbreaks, prompting authorities to take corrective actions such as irrigation or pest control measures.

By harnessing the capabilities of CNN-based lane detection with instance segmentation in edge-cloud computing environments, these applications can achieve higher levels of accuracy, efficiency, and responsiveness, ultimately leading to safer roads, smarter transportation systems, and
enhanced environmental sustainability.

V.LITERATURE

5.1. Name:” The variability of urban safety performance functions for different road elements: an Italian case study**”**

Methods and Information:
The study, conducted within the Pa.S.S.S. National research project, utilized crash data from the City of Bari, Italy, collected during 2012–2016, coupled with traffic data and manual traffic counts from 2018–2019. Crash locations were meticulously analyzed, considering various factors such as crash type and

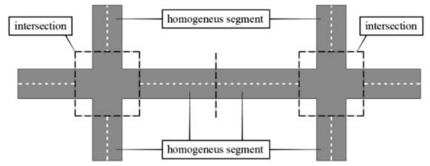


Fig. 2 Division of the road network into sites: intersections and homogeneous road segments

circumstances, road-related variables, and traffic volumes, to ensure accurate segmentation and intersection identification. A comprehensive set of crash predictors was examined, including road attributes and environmental factors. Negative Binomial count data models were employed for crash frequency prediction, with separate models for segments and intersections. Preliminary models were run for the entire dataset, followed by sub-category models based on selected subsets. Model selection was based on the Akaike Information Criterion and Nagelkerke R2. The study's results provide insights into the relationship between crash frequencies and predictors, contributing to enhanced road safety planning and management.

5.2.Name:” Instance segmentation on distributed deep learning big data cluster**”**

Methods and Information:
Distributed deep learning (DDL) is increasingly crucial due to the growing complexity and volume of data in real-world applications, necessitating the training and deployment of large and intricate deep learning models. DDL efficiently handles this challenge by distributing computational tasks across multiple machines, offering advantages over single-machine deep learning. It enhances scalability, fault tolerance, training speed, cost-effectiveness, and resource sharing, particularly beneficial for big data clusters. DDL facilitates training on large datasets and complex models like VGG networks or Inception Resnet networks, significantly reducing training time and improving accuracy. While GPUs are preferred for training, CPUs are suitable for preprocessing and inference. Distributed hyperparameter tuning (DHPT) accelerates model convergence and optimizes large deep-learning models effectively. DDL utilizes both data and model parallelism, distributing data or model segments across multiple machines to handle large datasets and complex models efficiently. Data parallelism speeds up training by distributing data across GPUs/CPUs, while model parallelism splits the model across machines to handle complexity. Careful consideration is required to evenly distribute computational loads in model parallelism. Overall, DDL's ability to handle large datasets and complex models efficiently positions it as a key player in the future of deep learning applications.

5.3.**Name:** “Research on Lane Line Detection Algorithm Based on Instance Segmentation”

Methods and Information:
The paper presents a lane line prediction branch to assess the existence and confidence level of output lane lines, depicted by Figures 7a and 7b. Through convolution and pooling, the network reduces channels to 7 and downsamples to 23 × 40 × 7. It then utilizes two fully connected layers with ReLU and Sigmoid activations, outputting a one-dimensional feature vector representing the probability of 6 pre-selected lane lines' existence. A confidence threshold of 0.5 determines lane line existence. The lane detection model, illustrated in Figure 8, employs ReLU activation and consists of 12 convolutional layers, 3 upsampling layers, and 5 pooling layers. It compresses the input feature map three times and stacks the output feature map via multi-size asymmetric shuffled convolutions. The decoder adaptsively upsamples the output, while the lane line prediction branch assesses confidence through maximum pooling. The asymmetric structure reduces model parameters and computation. Based on the TuSimple dataset, the paper selects training, verification, and testing images, enhancing data diversity via random rotation and horizontal deflection. Experiments employ an 11th Gen Intel Core i5 processor and NVIDIA GeForce RTX3050 GPU, with tensorflow2.4-GPU and CUDA 11.0. A front-view camera captures images at 2592 × 1944 resolution, mounted on a Volkswagen Sagitar vehicle. Hyperparameter settings and training details, including loss functions and iteration limits, are outlined, with a learning rate formula provided.

5.4.**Name**:” Securing DNN for smart vehicles: an overview of adversarial attacks, defenses, and frameworks”

Methods and Information:
The document discusses various types of neural networks and their architectures, focusing on feed-forward neural networks (FNNs) and recurrent neural networks (RNNs). FNNs are characterized by their feed-forward operation, making them suitable for non-sequential applications like image processing, while RNNs retain the values of the last layer neurons, making them ideal for sequential tasks such as audio analysis. Convolutional neural networks (CNNs) are examples of FNNs, combining fully connected neural layers with convolutional layers. The architecture of FNNs includes input, hidden, and output layers, with neurons in each layer connected to every neuron in the next layer. RNNs, on the other hand, allow the output of the last neuron to influence the prediction of the current neuron, making them suitable for sequential data processing tasks. The document also touches on reinforcement learning, where agents learn to improve their behaviour through interaction with the environment, and discusses adversarial attacks on deep neural networks (DNNs). Adversarial vulnerabilities in DNNs, including decision boundary vulnerability and transferability vulnerability, are explored, along with methods for generating adversarial images. Overall, the document provides insights into the architectures of various neural networks and their vulnerabilities to adversarial attacks.

5.5.Name: “Survey on categorical data for neural networks”

Methods and Information:
The paper delves into the understanding and application of deep learning algorithms, particularly focusing on feedforward networks (FNNs), synonymous with neural networks. It defines a deep learning algorithm as one that employs a composition of functions, where each function equates conceptually to a layer of a directed graph. Key terms such as "deep learning algorithm," "neural network," "feedforward network," and "deep neural network" are used interchangeably throughout the paper.
Another critical term discussed is "categorical data," representing variables divided into groups like race or age group. Various techniques for utilizing categorical data in deep learning algorithms are explored, including "entity embeddings," "dense encoding," "distributed representation," and "encoding." The paper analyzes different works and terms related to entity embeddings, highlighting the lack of consensus on terminology within the field.
The search method employed to find relevant literature on entity embeddings involves using synonymous terms like "tabular data" and "dense encoding" combined with phrases like "deep learning" or "neural network." Additionally, searches related to medical research with respect to specific coding systems are conducted to broaden the scope of inquiry. Overall, the paper provides a comprehensive overview of deep learning algorithms and techniques for handling categorical data, shedding light on the terminology and research landscape in the field.

5.6.Name:” Comprehensive study of driver behaviour monitoring systems using computer vision and machine learning techniques**”**

Methods and Information:
The paper delves into driver behavior classification, essential for enhancing driver awareness and road safety. Utilizing advanced technologies like vision systems and machine learning algorithms, it monitors actions such as hand movements, facial expressions, and body posture in real-time. Artifcial neural networks (ANNs) serve as computational models inspired by the human brain, learning from data to make predictions. Convolutional neural networks (CNNs) are vital for image understanding, while recurrent neural networks (RNNs) excel in understanding sequential data. Long short-term memory (LSTM) units further enhance predictions, especially in vehicle safety systems. Research on classifying drivers' behaviors in autonomous vehicles is pivotal, focusing on hand, facial, and body posture classification to recognize distracted or fatigued driving actions. Vision-based systems and sensor networks offer cost-effective solutions, contributing to the evolution of safer autonomous driving systems.
Top of Form

5.7. Name:” Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”

Methods and Information:
Deep learning (DL) finds application in various scenarios where human expertise may be limited or inadequate,

offering solutions to problems such as cases where human experts are unavailable or where explanations for decisions are difficult to articulate. DL's universal learning approach, robustness, generalization capabilities, and scalability make it a preferred choice for tackling diverse challenges. DL techniques are categorized into supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised learning deals with labeled data, while semi-supervised learning utilizes partially labeled datasets, and unsupervised learning operates without labeled data. Reinforcement learning, on the other hand, interacts with the environment to optimize actions based on feedback. Each type of learning has its advantages and limitations, with reinforcement learning being particularly suitable for problems involving complex parameter optimization. Applications of reinforcement learning span various domains, including business strategy planning and robotics, although its speed of learning may be influenced by parameters.

5.8.Name: "Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks"

Methods and Information:
The proposed method introduces a hybrid neural network by combining Deep Convolutional Neural Networks (DCNN) and Deep Recurrent Neural Networks (DRNN) for lane detection tasks. Traditional models for lane detection often struggle under challenging conditions such as heavy shadow, mark degradation, and vehicle occlusion. The hybrid approach aims to address these limitations by leveraging the strengths of both CNN and RNN.
The system overview emphasizes the need for a robust lane detection method that can handle continuous driving scenes. CNN is adept at processing large images, while RNN excels in continuous signal processing and sequential feature extraction. The proposed method combines these capabilities within an encoder-decoder framework, enabling end-to-end training.
The network design details the use of Convolutional LSTM (ConvLSTM) for sequential feature extraction and integration, enhancing the model's ability to handle time-series data efficiently. The architecture incorporates encoder-decoder networks inspired by SegNet and U-Net, with modifications to balance accuracy and efficiency.

5.9 **Name:"** Integration of Vehicular Clouds and Autonomous Driving: Survey and Future Perspectives"
Methods and Information:
This section outlines machine learning techniques applied in autonomous driving systems, emphasizing different sensor inputs. Recent work is categorized into mediated direct perception, behavior, reflex, and LIDAR perception approaches. Mediated perception involves processing scene data from front-facing cameras to recognize driving-relevant objects like lanes, vehicles, and pedestrians. Techniques like Pixel by Pixel Full Scene Labeling (FSL) perform semantic segmentation to label each pixel with its object category, employing methods like Markov Random Fields (MRF) and Convolutional Neural Networks (CNNs) for context understanding. 3D Scene Flow Segmentation aims to understand 3D motion from image sequences, decomposing scenes into independently moving objects and associating pixels with their 3D positions. These machine learning methods enable autonomous vehicles to perceive and understand their surroundings, facilitating informed driving decisions based on recognized objects.

5.10.Name:" CNN based lane detection with instance segmentation in edge-cloud computing"

Methods and Information:
Lane fitting in autonomous driving systems involves converting pixels into parameter descriptions to estimate lane instances. Commonly used fitting algorithms include cubic polynomials, spline curves, or arc curves. To improve fitting quality while maintaining computational efficiency, an inverse perspective transformation is applied to the image, converting it into a "bird's-eye view" for curve fitting. However, changes in the road plane can invalidate the fixed transformation matrix, affecting accuracy. To address this, a custom function network is used to generate a transformable matrix, optimizing it with a customized loss function. This network transforms lane pixels before curve fitting, enabling robust fitting of distant lanes even when the road surface changes. The overall framework includes image collection, cloud data processing, binary and embedded segmentation, clustering, and lane detection. This approach enhances lane detection accuracy and robustness in varying road conditions.

5.11 Name:" Lane Image Detection Based on Convolution Neural Network Multi-Task Learning**"**

Methods and Information:
The process of image preprocessing in autonomous driving systems involves several steps to enhance lane line detection accuracy and computational efficiency. Grayscale conversion simplifies image processing by eliminating color information, reducing computational load, and improving analysis. Target area extraction focuses on the lane line area, reducing redundant information and highlighting key image areas. Image scaling adjusts image size, with bicubic interpolation preferred for smoother results. Inverse perspective transformation corrects distortion caused by perspective, producing an "aerial view" or bird's-eye view of the lane lines. This view enhances detection accuracy by presenting parallel and equally wide lane lines, unaffected by convergence effects. The overall preprocessing framework improves lane detection reliability and efficiency in various driving scenarios.

5.12. Name:" Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation:**"**

Methods and Information:
The passage outlines the emergence and significance of Intelligent Transportation Systems (ITS) and Safety Driving Assistant Systems (SDAS), with a specific focus on Lane Departure Warning Systems (LDWS). Developed nations like Germany, Japan, and the United States have been at the forefront of integrating advanced technologies such as information and communication, electronic sensing, automatic control, and artificial intelligence to address global traffic challenges.
ITS, recognized since the 1990s, aims to improve communication between vehicles, roads, and users, enhancing safety, reducing accidents, lowering transportation costs, and minimizing environmental impact. SDAS, a subset of ITS, provides immediate assistance to drivers during emergencies, ensuring vehicle stability and safety.
LDWS, a critical SDAS component, alerts drivers to lane deviations, leveraging sensors to detect road conditions and analyze driving patterns to issue timely warnings through various means. It significantly reduces accidents caused by unconscious lane departures and driver inattention.
LDWS offers longitudinal and lateral warnings, often using monocular vision technology. Implementation methods include road infrastructure-based and vehicle-based systems, both contributing to improved road safety and reduced accidents, reflecting ongoing efforts to leverage technology for transportation advancements.

5.13. Name: "Real-Time lane detection on Embedded Systems for control of semi-Autonomous Vehicles"

Information:
the evolution of lane detection techniques prior to the emergence of big data and deep learning. Traditional methods, such as the Hough transform and Canny edge detection, were employed for lane detection, particularly effective for straight lanes under ideal conditions. These techniques rely on image processing methods like spatial filtering and convolution to extract features from images. However, they face challenges in handling complex lighting conditions and curved lanes, limiting their applicability to daytime highway driving. Researchers have explored alternative methods, leveraging intensity information and classic machine learning techniques like Decision Trees, to detect and analyze lane markers in real time. These efforts represent early steps towards the development of modern deep learning-based lane detection systems.

5.14. Name:" Hybrid Deep Learning approach for Lane Detection"

Information:
The content delves into three key topics: Convolutional Neural Networks (CNNs), Transformer Networks, and Semantic Segmentation, with a focus on evaluation metrics. CNNs revolutionized image analysis through convolutional operations, reducing network parameters by creating local connections. Transformer Networks, initially developed for sequential data, improved parallelization and captured global dependencies using attention mechanisms. Semantic Segmentation assigns class labels to pixels, with evaluation metrics including Intersection over Union (IoU) and F1 Score. Evaluation in lane detection considers pixel-level accuracy, precision, recall, and F1 Score due to imbalanced datasets. Real-time performance is crucial, measured by frames processed per second (FPS). The introduction of the Vision Transformer (ViT) provides an alternative to CNNs, capturing global image characteristics effectively. However, ViTs require significant computational resources for training, contrasting with CNNs' local filter approach.

5.15. Name:" Lane Detection in Autonomous Vehicles: A Systematic Review"

Methods and Information:

Autonomous vehicles rely heavily on accurate lane detection for safe navigation. Traditional methods depend on manually designed features to identify lane markings, but these struggle with complex road conditions and variations.

Deep learning offers a powerful alternative. Deep learning models can directly learn lane features and location from image data, achieving higher accuracy. They are also more robust to lighting changes, occlusions, and different lane shapes compared to traditional methods.

There are different deep learning approaches for lane detection. Stand-alone deep learning directly learns everything from the data. Deep learning combined with geometric modeling leverages both techniques for better performance. Finally, combining different deep learning architectures can further improve results.

While deep learning shows great promise, challenges remain. Training these models requires vast amounts of data. Additionally, researchers are still working on making deep learning models more interpretable, meaning we need to better understand how they arrive at their decisions.

Overall, deep learning is a significant advancement in lane detection for autonomous vehicles, offering substantial advantages over traditional methods.

5.16.**Name:"** Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation"

Methods and Information:

Recent advancements in lane line detection leverage various deep learning techniques. Long et al. (2015) and Pizzati et al. (2019) divided the process into stages, using the ERFNet for semantic segmentation followed by a classification network. Neven et al. (2018) introduced LaneNet, an encoder-decoder model, while Zou et al. (2020) combined CNN and RNN for temporal information utilization. Kim and Lee (2014) integrated CNN with RANSAC for lane detection in complex scenes. Li (2020) explored DeepLab V3+ for lane segmentation, enhancing it with a density clustering algorithm and Kalman filtering for lane confirmation and tracking. Further advancements include 3D models like 3D-LaneNet (Garnett et al., 2019), PINet (Ko et al., 2020), and the semi-local 3D lane detection method (Efrat et al., 2020), which handle complex scenarios and offer improved accuracy and efficiency through techniques like instance segmentation and uncertainty estimation. These methodologies represent a shift towards automated, precise, and robust lane detection systems, integrating deep learning with traditional algorithms for enhanced performance.

5.17. **Name**:" Lane Marker Detection Based on Multihead Self-Attention"

Methods and Information:

The lane detection system incorporates a multihead self-attention mechanism into a ResNet-34 backbone for robust feature extraction. ResNet-34 generates a local feature map, which is then transformed into a multi-anchor representation. Each anchor represents a point on the feature map, facilitating global contextual understanding. The multihead mechanism projects queries, keys, and values, enhancing feature fusion and expanding the receptive field. Anchors are composed of (x, y) coordinate frames, enabling precise localization. Self-attention computes dot products scaled by the square root of the feature map dimensions and performs a soft-max operation for attention weight computation. Finally, the outputs of parallel attention functions are concatenated to capture diverse spatial information effectively. This approach improves classification and localization accuracy, particularly under challenging conditions like heavy shadow and mark degradation.

5.18.**Name**: Road Lane Detection using Convolutional Neural Network

Methods and Information:

The proposed system architecture for road lane detection involves several key steps. Firstly, in image pre-processing, RGB or BGR images are converted to grayscale to reduce noise using libraries like sklearn and Keras. Next, feature extraction involves resizing images, converting them to grayscale, and splitting them into training and testing datasets, with 20% of features manually provided and 80% extracted by a CNN model. The data is then split using sklearn. Classification employs a convolutional neural network (CNN) with convolutional and pooling layers, followed by flattening and hidden layers to prevent overfitting. Finally, prediction and detection involve testing the model on new images, with user feedback incorporated for model refinement. This iterative process enhances the model's performance based on user input, ensuring accuracy on new data

5.19.**Name**:” Lane Detection Based on Instance Segmentation of BiSeNet V2 Backbone Network”

Methods and Information:
The dataset used for training consists of 100,000 images for training, 9,754 for validation, and 35,680 for testing. Manual labeling includes passing labels through obstacles to enhance the model's generalization. Labelme software is employed for annotation, generating .json files, which are then converted to the TuSimple dataset format. The neural network architecture comprises a multi-task branching network focusing on lane line segmentation and clustering. This approach decomposes lane line detection into two tasks, improving network efficiency and accuracy. Lane line segmentation obtains binary segmentation results, while clustering categorizes lane lines. Instance segmentation accommodates scenarios with varying numbers of lanes, crucial for lane change detection. Cross-entropy loss with weighted pixels addresses class imbalance. Weighted loss functions ensure effective model training despite imbalanced data distribution.

5.19. **Name**: “Lane Line Detection for Autonomous Cars using Python and OpenCV”

Methods and Information:
The proposed lane detection algorithm consists of three key steps: preprocessing, ROI selection, and lane detection. Initially, RGB images are converted to HSV format for better processing. Preprocessing includes white extraction and edge detection, particularly for blurry images. ROI selection focuses on identifying the region of interest where lane detection algorithms will be applied. This step involves deleting unwanted areas from the image. Finally, lane detection is performed using either Hough transformation or lane tracking with an extended Kalman filter. Both methods aim to identify lane lines in the selected ROI, ensuring accurate detection for autonomous driving systems.

5.20.**Name**: Method for Automatic Lane Detection using a Deep Network

Methods and Information:
Generative Adversarial Networks (GANs) consist of a generative and a discriminator part, trained adversarially using game theory principles. Semi-supervised GANs incorporate labeled data into the training process, enhancing classification tasks. The discriminator distinguishes real from fake data, utilizing both labeled and unlabeled datasets. For instance, in lane detection, a dataset with labeled and unlabeled samples is utilized. The discriminator employs a UNet model for segmentation, with supervised and unsupervised training. The generative network generates counterfeit images, affecting overall network performance. A decoder with inverse convolutional layers constructs counterfeit images. Implementation involves dynamic learning coefficients, Adam optimizer, and 500 training iterations. Python 3.2 and Google Colab with GPU support facilitate execution. Overall, the semi-supervised GAN approach demonstrates effectiveness in tasks like image segmentation.

5.21.Name:” Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments”

Methods and Information:
Generative Adversarial Networks (GANs) consist of a generative and a discriminator part, trained adversarially using game theory principles. Semi-supervised GANs incorporate labeled data into the training process, enhancing classification tasks. The discriminator distinguishes real from fake data, utilizing both labeled and unlabeled datasets. For instance, in lane detection, a dataset with labeled and unlabeled samples is utilized. The discriminator employs a UNet model for segmentation, with supervised and unsupervised training. The generative network generates counterfeit images, affecting overall network performance. A decoder with inverse convolutional layers constructs counterfeit images. Implementation involves dynamic learning coefficients, Adam optimizer, and 500 training iterations. Python 3.2 and Google Colab with GPU support facilitate execution. Overall, the semi-supervised GAN approach demonstrates effectiveness in tasks like image segmentation by leveraging both labeled and unlabeled data, enhancing the network's ability to generalize and perform well on unseen data.

5.22 Name:" Lane Line Detection based on Mask R-CNN"

Methods and Information:
The Mask-RCNN network, an extension of Faster-RCNN, integrates a mask layer for segmentation, enhancing target instantiation. By employing a 50-layer deep ResNet and a feature pyramid network (FPN), efficiency and accuracy are ensured. The dataset, TSD-Max, comprises 100,383 samples covering various road conditions like shadows, damage, and regular road scenarios. Training on 4/5 of each category and testing on the remaining 1/5 yielded an accuracy of 97.9% after 100,000 iterations. While slight variations exist across road conditions, overall performance surpasses traditional methods significantly. Despite challenges in learning complex conditions, these results underscore the effectiveness of deep learning in lane detection tasks.

5.23 Name:" Deep Learning in Lane Marking Detection"

Methods and Information:
Deep Learning in Lane Marking Detection:A Survey

Advanced deep architectures for lane marking detection prioritize spatial structure understanding, scene priors utilization, and position regularity exploitation. Techniques like dilated convolution, spatial convolution, graph convolution, and attention mechanisms enhance spatial understanding, while models like U-Net, FCN, and SCNN preserve spatial structures effectively. Incorporating scene priors via methods like vanishing points, pyramid pooling, and optical flow aids in context comprehension. Position regularity is leveraged through LSTM, ConvLSTM, and optical flow, facilitating prediction and continuity inference of lane markings. Moreover, regression-based methods like PointLaneNet enable precise lane marking detection, while LineNet and LaneNet utilize multi-task learning for classification and segmentation tasks. ResNet-inspired architectures such as E-Net and EDANet focus on real-time performance, balancing efficiency and accuracy. Evaluation metrics such as F1 score and mean average precision (mAP) assess the effectiveness of these methods in lane marking detection within Intelligent Transportation Systems.

5.24 Name:"Artificial Intelligence-based Lane Detection Using Satellite Images"

Methods and Information:
Various methods and datasets are employed for road detection from satellite images, showcasing diverse approaches and outcomes. For instance, a study using a dataset from Nigeria achieved 99.4% accuracy employing CNN architectures like Inception ResNetV2 and ResNet50V2. Another study in Mumbai utilized techniques like the Otsu method and Histogram Equalization for excellent results in road detection. Additionally, in China, a dataset facilitated road extraction using Gabor-confined object segmentation and achieved 95% accuracy. Moreover, different datasets such as IKONOS and Space-Net were utilized for emergency road extraction and detection in developing countries, respectively, demonstrating high accuracy with various segmentation and preprocessing techniques. Techniques like image segmentation and model utilization, including VPGNet and FCN-8s, enabled Seoul's road marking detection with 99% accuracy. These diverse methodologies highlight significant advancements in road detection from satellite imagery, addressing various challenges and achieving remarkable accuracies

5.24 Name:" Novel Intelligent Lane Line Detection System using Neural Networks"

Methods and Information:

The lane detection mechanism workflow involves data gathering, CNN model training, real-time video processing, and cooperative driving integration. Data gathering includes collecting diverse road images and annotating them for supervised learning. Preprocessing involves normalizing pixel values and dataset augmentation. CNN model development entails creating layers, splitting datasets, and optimizing weights using backpropagation. Real-time video processing involves capturing frames, preprocessing, running them through the trained CNN model for lane detection, and post-processing for visualization. Integration of cooperative driving includes establishing vehicular communication standards, using onboard sensors for detecting lane changes, employing predictive algorithms, and informing nearby vehicles of intentions for safer driving activities.

5.2.1CondensedResearchPaperSummary(Tabular)

Review Paper Title	Methodology Summary
1. "The variability of urban safety performance functions for different road elements: an Italian case study"	Utilized crash data from Bari, Italy (2012–2016) and traffic data (2018–2019). Analyzed crash locations, considered various factors, and **employed Negative Binomial** count data models for crash frequency prediction.
2. "Instance segmentation on distributed deep learning big data cluster"	Explored distributed **deep learning** (DDL) for handling large datasets and complex models, employing data and model parallelism for scalability and efficiency.
3. "Research on Lane Line Detection Algorithm Based on Instance Segmentation"	Developed a lane line detection model utilizing convolutional layers, pooling layers, and multi-size asymmetric shuffled convolutions, trained on **TuSimple** dataset.
4. "Securing DNN for smart vehicles: an overview of adversarial attacks, defenses, and frameworks"	Discussed neural network architectures, **reinforcement learning**, and adversarial vulnerabilities in **deep neural networks.**
5. "Survey on categorical data for neural networks"	Explored techniques for handling categorical data in **deep learning algorithms**, particularly focusing on **entity embeddings**.
6. "Comprehensive study of driver behaviour monitoring systems using computer vision and machine learning techniques"	Investigated driver behaviour classification using vision systems and machine learning algorithms**, emphasizing neural networks for real-time monitoring.**
7. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions"	Explored various types of deep learning techniques and their applications, categorizing DL techniques into **supervised**, u**nsupervised**, **semi-supervised**, and **reinforcement learning**.
8. "Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks"	Proposed a hybrid neural network combining **CNN** and **RNN** for robust lane detection, addressing challenges like shadow, mark **degradation**, and **vehicle occlusion.**
9. "Integration of Vehicular Clouds and Autonomous Driving: Survey and Future Perspectives"	Explored machine learning techniques in autonomous driving systems, focusing on sensor inputs and techniques like semantic segmentation for scene understanding.
10. "CNN based lane detection with instance segmentation in edge-cloud computing"	Developed a lane fitting approach using custom function networks and **inverse perspective transformation** to improve **lane detection accuracy** and **robustness.**
11. "Lane Image Detection Based on Convolution Neural Network Multi-Task Learning"	Detailed **image preprocessing** steps for lane detection in **autonomous driving systems**, enhancing detection reliability and efficiency.

15. "Lane Detection in Autonomous Vehicles: A Systematic Review"	**Explored deep learning** approaches for lane detection in **autonomous** vehicles, highlighting advantages over traditional methods and challenges such as **data requirements** and interpretability.
16. "Lane Marker Detection Based on Multihead Self-Attention"	Implemented a lane detection system incorporating **multihead self-attention mechanism** for robust feature extraction and localization, particularly effective under challenging conditions.
17. "Road Lane Detection using Convolutional Neural Network"	Proposed a system architecture involving **image pre-processing,** feature extraction using CNN, and classification for road lane detection, aiming for accuracy and efficiency.
18. "Lane Detection Based on Instance Segmentation of BiSeNet V2 Backbone Network"	Utilized a **multi-task branching** network for **lane detection**, decomposing the task into lane line segmentation and clustering to improve efficiency and accuracy.
19. "Lane Line Detection for Autonomous Cars using Python and OpenCV"	Presented a lane **detection** algorithm involving preprocessing, ROI selection, and lane detection using Hough transformation or lane tracking, ensuring accurate detection for autonomous driving systems.
20. "Method for Automatic Lane Detection using a Deep Network"	Utilized **semi-supervised GANs** for lane detection in complex scenarios, leveraging both **labelled and unlabelled data** to improve network performance and generalization.
21.”"Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments"	Utilized semi-supervised GANs for lane detection in complex scenarios, leveraging both labeled and unlabeled data to improve network performance and generalization.
22.”Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments”	Semi-supervised GAN approach leverages labeled and unlabeled data, employing UNet model for segmentation and dynamic learning coefficients for effective lane detection in diverse scenarios

12. "Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation"	Discussed the significance of Lane Departure Warning Systems (LDWS) **in Intelligent Transportation Systems** (ITS) **and Safety Driving Assistant Systems** (SDAS), emphasizing the role of monocular vision technology.
13. "Real-Time lane detection on Embedded Systems for control of semi-Autonomous Vehicles"	Reviewed **traditional lane detection techniques** and their limitations, highlighting early steps towards **modern deep learning-based approaches.**
14. "Hybrid Deep Learning approach for Lane Detection"	Explored CNNs, **Transformer Networks**, and **Semantic Segmentation** for lane detection, focusing on evaluation metrics and real-time performance.

Research Paper Title	Summary
23 Lane Line Detection based on Mask R-CNN	Mask R-CNN integrates mask layer for segmentation, utilizing deep ResNet and feature pyramid network for accuracy in lane detection on TSD-Max dataset, showcasing effectiveness in learning complex road conditions.
24 Deep Learning in Lane Marking Detection	Survey highlights advanced deep architectures prioritizing spatial structure understanding and scene priors utilization, showcasing effectiveness of techniques like U-Net, FCN, and SCNN for precise lane marking detection within Intelligent Transportation Systems.
25 .Artificial Intelligence-based Lane Detection Using Satellite Images	Various methodologies utilizing CNN architectures and segmentation techniques achieve high accuracy in road detection from satellite images, demonstrating significant advancements and addressing diverse challenges in road detection tasks.
26.Novel Intelligent Lane Line Detection System using Neural Networks	Workflow involves data gathering, CNN model training, real-time video processing, and cooperative driving integration for effective lane detection, emphasizing supervised learning, dataset augmentation, and onboard sensor utilization for safer driving activities.

VI.Limitations and Future Scope

Limitations:

- The majority of the research papers are based on specific case studies or datasets, limiting the generalizability of findings to broader contexts.
- Reliance on historical or simulated data may not fully capture real-time or dynamic conditions, affecting the applicability of proposed methodologies.
- Many papers focus on technical aspects without considering broader socio-economic or environmental factors influencing the studied phenomena.
- Computational complexity and resource requirements hinder scalability and practical implementation in real-world scenarios.
- Lack of standardization in methodologies and evaluation metrics makes cross-study comparisons challenging.

- Certain papers may overlook ethical considerations related to data privacy, bias, and societal implications of proposed solutions.

- Dependency on specific software frameworks or hardware configurations may restrict accessibility and adoption.
- Limited interdisciplinary collaboration may result in overlooking alternative perspectives or innovative solutions.
- Few papers address long-term sustainability or maintenance considerations of implemented systems.
- The rapid pace of technological advancements may render some proposed solutions obsolete or outdated over time.
- Challenges related to model interpretability and explainability may hinder trust and acceptance in real-world applications.
- Geographical or cultural biases in dataset selection may introduce unintended biases in algorithmic outcomes.
- Insufficient validation or benchmarking against existing state-of-the-art methods may obscure the true efficacy of proposed approaches.
- Lack of real-world deployment and validation may lead to a gap between research findings and practical utility.
- Limited consideration of potential adversarial attacks or vulnerabilities in proposed systems poses security risks in deployment

Future Scope

- Foster interdisciplinary collaboration to integrate domain knowledge from diverse fields into research methodologies.
- Conduct longitudinal studies to assess the long-term effectiveness and sustainability of implemented solutions.
- Promote the development of standardized datasets and evaluation protocols to facilitate benchmarking and cross-study comparisons.
- Explore novel data sources, such as crowd-sourced or real-time sensor data, to enhance the robustness and real-world applicability of models.
- Address ethical considerations through the adoption of responsible AI principles and frameworks in research and development.
- Invest in research on interpretability and explainability of AI models to enhance trust and transparency in automated systems.
- Develop lightweight and efficient algorithms optimized for deployment on resource-constrained edge devices.
- Investigate strategies for mitigating biases and ensuring fairness in AI models across diverse demographic and cultural contexts.
- Explore adaptive learning techniques to enable AI systems to continuously evolve and adapt to changing environments and requirements.
- Foster collaboration between academia, industry, and policymakers to ensure alignment with societal needs and regulatory standards.
- Advance research on adversarial robustness to enhance the security and resilience of AI systems against potential attacks.
- Invest in user-centered design approaches to ensure the usability and acceptability of AI-driven technologies.
- Promote transparency and reproducibility in research through open-access publication and sharing of code and data.
- Encourage research on AI governance and policy frameworks to address legal and ethical challenges in AI deployment.
- Foster a culture of responsible innovation and inclusivity to ensure that AI benefits society as a whole while minimizing risks and harms.