www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 170-194. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.594

Lane Detection Using Computer Vision and Convolutional Neural Networks for Autonomous Vehicles

Sergio Álvarez Silva, Dante Mújica Vargas, Andrés Antonio Arenas Muñiz
Departamento de Ciencias de la Computación, Centro Nacional de Investigación y Desarrollo Tecnológico - CENIDET/TecNM, Cuernavaca Morelos 62594, México.
m23ce001@cenidet.tecnm.mx, dante.mv@cenidet.tecnm.mx, m22ce043@cenidet.tecnm.mx

Abstract. This article presents an analysis of computer vision algorithms for Lane Maintenance Assistants (LMA), comparing traditional methods with Convolutional Neural Networks (CNNs). The objective is to evaluate their effectiveness under diverse driving conditions using recognized databases and testing in both real and simulated environments. A proprietary database containing scenarios from the state of Morelos was also used. Experiments covered adverse conditions, such as rain (light, moderate, heavy), solar glare, road shadows, curves, and night driving with/without artificial lighting. Fog simulations included uniform, heterogeneous, cloudy, and combined types. Results showed traditional methods, such as Sobel + Adaptive Thresholding (SA) and Canny + Hough (CH), perform well in normal conditions, achieving a high Intersection over Union (IoU) of 0.9729 and Precision of 0.9888 in Experiment 1. However, their performance declined in complex scenarios like sharp curves and degraded lane markings, with IoU dropping to 0.4838 and Precision to 0.4925 in Experiment 4. Conversely, CNN-based algorithms, like SCNN and VGG16, demonstrated greater adaptability and accuracy in challenging environments. In Experiment 5, VGG16 achieved a Recall of 0.9796 and a Dice Index of 0.6575, showing superior lane detection in dynamic conditions. Although traditional methods and CNNs showed comparable IoU and Precision, the higher Recall in CNNs indicates better robustness, crucial for real-world applications. This study highlights the advantages of deep learning in road safety, reinforcing CNNs particularly VGG16 as a preferred option for lane detection in autonomous vehicles.

Keywords: ADAS, computer vision, lane detection, lane-keeping assistance.

Article Info
Received December 11, 2024
Accepted February 18, 2025

1 Introduction

Humans make decisions constantly throughout the day, many of which are based on our perceptions of our surroundings. In driving, visual perception is essential to ensure safe and efficient driving. Autonomous vehicles require the ability to detect objects in their environment and react appropriately, which poses several challenges related to computer vision (SAE, 2021). Among these challenges are the identification of traffic signs, traffic lights, pedestrians and, perhaps most importantly, lane lines. The latter play a crucial role in vehicle motion planning, as they help determine the path to follow and avoid unintended deviations (Zakaria et al., 2023). Motion planning is a key technology in autonomous vehicles, but it faces significant challenges, especially when it comes to accurate lane detection. This process begins with lane identification and tracking along the vehicle's route, followed by detection of obstacles, signs, and other road features that can influence driving. This research focuses on evaluating

the most effective methods for lane detection, dividing them into two main categories: traditional computer vision-based approaches and advanced deep learning techniques, such as convolutional neural networks (CNNs). In-vehicle perception systems are central to addressing these challenges. These systems are composed of cameras and sensors strategically placed in the vehicle, whose function is to capture, analyze and process information about the environment. This information is essential for real-time decision making and has a direct impact on driving safety and comfort. Advanced driver assistance systems (ADAS), such as Lane Keeping Assist (LKA) and automatic emergency braking, rely on this capability to interpret road conditions and help the driver avoid accidents. The integration of data from various sensors, such as LiDAR, cameras and radar, enables a more complete view of the environment. This vision is essential not only for the proper operation of autonomous vehicles, but also for the efficient operation of ADAS systems, which perform tasks such as route planning, pedestrian detection, and collision avoidance. Recently, advances in deep learning techniques, especially in image and video processing, have enabled great progress in lane detection, with convolutional neural networks (CNNs) being one of the most effective approaches to address these problems (Zakaria et al., 2024). Real-time lane detection is challenging due to the variability of road shapes, colors, and conditions. Traditional computer vision approaches, such as the Hough Transform, the Canny algorithm for edge detection, and the use of filters such as Kalman and Sobel, have been effective in relatively stable conditions. However, these methods tend to fail in complex scenarios, such as adverse weather conditions or roads with steep curves (Chand Bansal et al., 2021). In contrast, deep learning techniques, particularly those based on CNN, have demonstrated a great ability to improve lane detection accuracy, adapting better to various driving conditions and scenarios. The use of CNN and combined convolutional and recurrent network architectures such as those proposed by (Wang et al., 2022; Zou et al., 2020) has shown great potential to overcome the limitations of traditional methods. Despite their effectiveness, neural networks face challenges such as the need for large amounts of training data and the computational resources required for their real-time training and deployment. However, improvements in architecture and optimization algorithms have made neural networks increasingly applicable to lane detection, even under difficult road conditions. Widely used datasets for the evaluation of lane detection models, such as CULane (Pan et al., 2018) and TUSimple (Yoo et al., 2020), remain essential for training and evaluating robust models. However, these datasets still lack a sufficient variety of scenarios that reflect the diversity of real-world conditions that autonomous vehicles may face. Incorporating more variety in the datasets could improve model generalization and enable lane detection systems to be more accurate and reliable in real-world, adverse driving conditions.

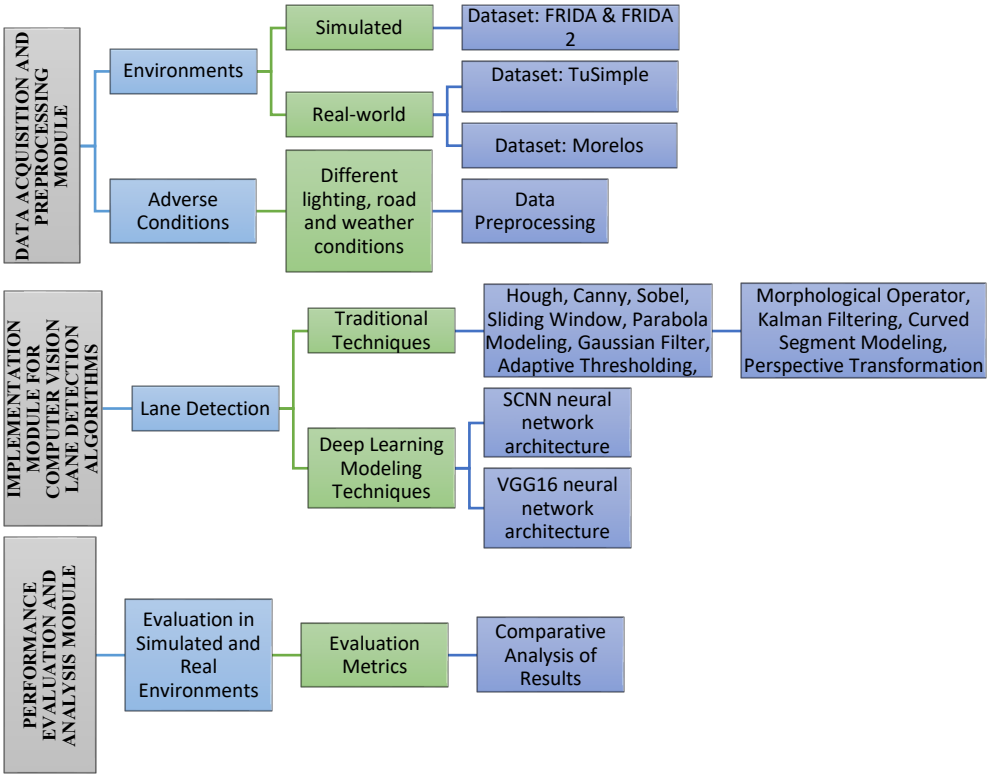

Fig. 1. General diagram of the proposed solution methodology

2 Methodology

The proposed method for analyzing computer vision algorithms applied to lane-keeping assist systems is organized into three main modules: data acquisition and preprocessing, implementation of lane detection algorithms, and performance evaluation. Each module is designed to address a specific aspect of the system, from data collection to comparison of techniques and performance metrics (see Figure 1). This section describes the specifications of the development environment employed for the execution of the experiments, which involved both simulated images and real environments, as well as the characteristics of the computer system (see Table 1), the databases utilized (see Table 2), and the camera used for the creation of the database of the state of Morelos, Mexico. Furthermore, Table 3 outlines the software architecture applied for lane detection in road scenes, including both the software tools and the camera specifications.

Table 1. Computing system

Component	Specification
Processor	13th Gen Intel(R) Core(TM) i7-13650HX 2.60 GHz
RAM memory	16 GB DDR5
GPU	RTX 4060 with 16 GB of VRAM
Storage	1 TB SSD

Table 2. Summary of databases used

Database	Images	Video clips	Description
Frida & Frida 2 [9]	420	N/A	Database of varied simulated road scenes for lane detection with uniform, heterogeneous, and cloudy fog and lighting conditions.
TuSimple [8]	55,640	2,782	Dataset with a variety of road scenes for lane detection and adverse weather conditions such as sharp curves and shadows.
Own database (Morelos) [10]	22,566	41	Database of the state of Morelos, Mexico, with varied road scenes for lane detection, including sharp curves, rain, low illumination, darkness and shadows.

Table 3. Software and Camera Specifications

Element	Specification
Operating System	Linux Ubuntu
Ubuntu Version	20.04 LTS
Programming Framework	Python
Development Environment	Anaconda (virtual environments)
Libraries Used	OpenCV, NumPy, TensorFlow, PyTorch, Scikit-learn Matplotlib, Easydict.
Camera Used	Hero 10 Black
Camera Resolution	5312 x 2988 (Maximum resolution)

Capture Frequency	Up to 240 fps (in 1080p); 60 fps (in 5.3K)
Installed Dependencies	easydict==1.6, matplotlib==2.0.2, glog==0.3.1, opencv-python==3.4.0.12, numpy==1.13.1, scikit-learn==0.19.1, python==3.7,TensorFlow==2.2, Keras==2.3.

Each module is described in detail below, as well as its role in the methodology.

2.1 Acquisition and Preprocessing

The process begins with the acquisition and preprocessing of visual data, utilizing both state-of-the-art databases and newly captured data. Images and videos are collected through vehicle-mounted cameras, capturing a wide range of scenarios, including varying traffic conditions, lighting, and adverse weather such as rain, fog, and shadows. Preprocessing plays a critical role in adapting the visual data to these diverse scenarios by addressing specific challenges. For instance, de-fogging enhances visibility in foggy conditions, while shadow removal improves lane clarity in uneven lighting. Histogram equalization and color correction adjust image contrast and brightness for better feature recognition, particularly in low-light or overexposed conditions. Lane width estimation and marking refinement are essential for accurately identifying lanes on roads with varying geometries or worn markings. Additionally, defining the region of interest (ROI) focuses processing on the relevant areas of the image, minimizing noise and computational overhead. These preprocessing techniques collectively enhance image quality and ensure reliable lane detection across different environments (see Figure 2).

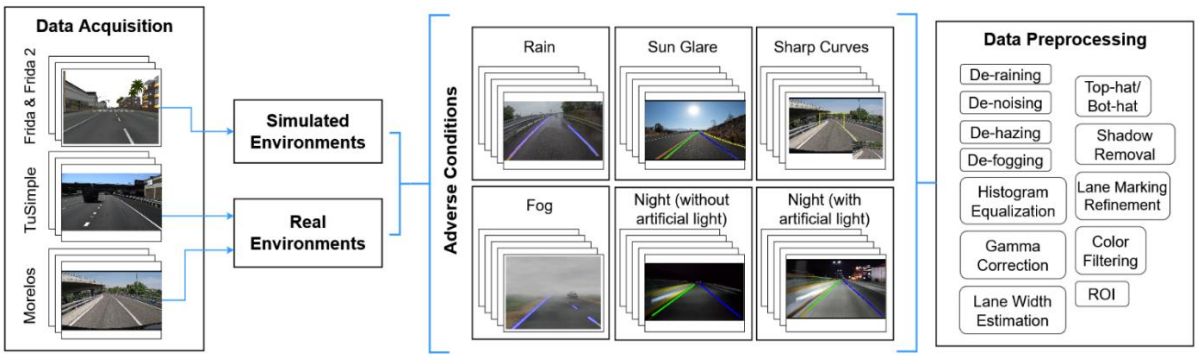

Fig. 2. Acquisition and Preprocessing Module

Block 1: Data Acquisition

This first block focuses on the collection of images and videos using vehicle-mounted cameras, covering various traffic conditions, weather, and road environments. The data is obtained both from state-of-the-art databases and from new scenarios captured in adverse situations, such as rain, fog, or low illumination.

Block 2: Data Preprocessing

Preprocessing is crucial to optimize visual data before applying them to lane detection algorithms. The data preprocessing techniques used are described below:
De-raining: removes raindrop effects in the images to improve visibility. Mathematically, the input image I can be decomposed into two components:

$$I = L + R \tag{1}$$

where L is the clean image and R represents the raindrops (Ren et al., 2020).

De-noising: It reduces the noise generated by low illumination or camera interferences by means of filters such as the Gaussian. This filter smoothes the image by convolving a Gaussian kernel $G(x, y)$ with the original image I :

$$I'(x,y) = \sum_{\{i=-k\}}^{\{k\}} \sum_{\{j=-k\}}^{\{j\}} G(i,j)I(x-i, y-j) \tag{2}$$

The Gaussian filter is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3}$$

where (σ) is the standard deviation that controls the level of smoothing (Goceri, 2023).
Region of Interest (ROI): The Region of Interest (ROI) is a crucial step in lane detection (Haider, 2023) that focuses processing efforts on the areas of the image most likely to contain lanes. This not only reduces computational complexity but also enhances detection accuracy by excluding irrelevant parts of the image, such as the sky or surrounding scenery. The ROI is defined using a binary mask MM that specifies the relevant regions of the image. When applied to the input image II, the ROI is computed using element-wise multiplication as follows:

$$I_{ROI} = I \; x \; M \tag{4}$$

Here, x represents the element-to-element multiplication, effectively zeroing out pixel values in I that fall outside the regions marked by M. The design of the binary mask M is based on prior knowledge of the camera perspective and typical lane positioning, such as limiting the ROI to the lower half of the image where lanes are most likely to appear. This targeted approach optimizes resource allocation during preprocessing and reduces noise from unrelated parts of the frame, significantly improving the overall performance of lane detection algorithms.
De-fogging: Improves visibility in foggy conditions using the atmospheric transmission equation:

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{5}$$

where $I(x)$ is the observed image, $J(x)$ is the fog-free image, $t(x)$ is the transmission map, and A is the ambient light. The transmission map $t(x)$ is estimated to recover the image $J(x)$ (Qu, 2023).
De-hazing: Similar to de-fogging, de-hazing uses the "*Dark Channel Prior*", which estimates the areas less affected by haze based on the darkness of the color channels [15].

$$\mathrm{I(x)} = J(x) \cdot t(x) + \big(1 - t(x)\big) \cdot A \tag{6}$$

In the haze model, I(x) represents the pixel value of the hazy image at a specific point xx, while J(x) corresponds to the pixel value of the clean, haze-free image at the same point. The transmission t(x) at point x refers to the amount of light that has successfully reached that point after passing through the haze. Finally, A denotes the atmospheric color, which is a constant value that represents the contribution of ambient light to the haze.
Histogram Equalization: Increases the distinction between the lanes and the road background. The cumulative histogram $f\ (i)$ is used to adjust the intensity levels of the image:

$$x' = \frac{f(i) - min(f)}{(M \times N) - min(f)} \times (L-1) \tag{7}$$

where M × N is the image size, and L is the number of gray levels (Roy et al., 2024).
Gamma correction: is a technique used to adjust the brightness of an image, compensating for the non-linear behavior of human vision and the way displays reproduce images (Cui et al., 2022). It corrects the image's luminance to make it appear more natural to the human eye by applying a power-law function to the pixel values. In digital imaging, gamma correction is essential because human eyes are more sensitive to changes in darker areas than in lighter ones. Without gamma correction, images could appear

too dark or too bright, and details in shadows or highlights might be lost. By applying gamma correction, the image can be adjusted to better match the non-linear response of the human visual system.

$$I'(x,y) = I(x,y)\gamma \tag{8}$$

where γ controls the correction level. Values $\gamma < 1$ lighten the image, while $\gamma > 1$ darken it.
Color Filtering: is a fundamental technique in computer vision used to identify specific features in an image, such as lane lines in autonomous driving applications. Color filtering leverages the color differences between the features of interest (in this case, lane lines) and the background by applying thresholds to the color channels in an image (Javeed et al., 2023).
Shadow Removal: Identifies and removes shadows that interfere with lane detection (Zhang et al., 2021). The illumination and reflectance components of the image are separated. This decomposition can be expressed by the equation:

$$I(x,y) = L(x,y)R(x,y) \tag{9}$$

where $I(x,y)$ is the observed image, $L(x,y)$ is the illumination component and $R(x,y)$ is the reflectance (the part without shadows). Methods for estimating $L(x,y)$ are usually based on color normalization using the chromaticity ratio, so that shadows are detected as low light intensity regions. Detected as regions of low light intensity:

$$S(x,y) = 1 - \frac{min(IR(x,y), IG(x,y), IB\ (x,y))}{I(x,y)} \tag{10}$$

where IR, IG, IB are the intensities of the color channels.
Lane Width Estimation: Verifies the detection of lanes based on their typical width. Lane width estimation is based on measuring the distance between detected lane lines in pixels and converting that measurement to physical units (such as meters) using a perspective transformation. Given a camera model and its calibration, the transformation of the image to a top view plane can be described by a homography matrix H. If (u,v) are the coordinates of the points in the image and (x,y) are the coordinates in the real plane, the relationship is:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{11}$$

Then, the lane width is estimated by measuring the distance between the lane lines in the real plane (Ghanem et al., 2023).
Top-hat/Bot-hat transforms: They highlight brightness or darkness features through the differences between the original image and its morphological operations (Salvi et al., 2021). Top-hat transformation is defined as:

$$Itophat = I - Io \tag{12}$$

where Io is the morphological aperture. The Bot-hat transform is used to highlight dark areas:

$$Ibothat = Ic - I \tag{13}$$

where Ic is the morphological closure.
Lane Mark Refinement: The position and shape of detected lane lines are adjusted using refinement algorithms such as Canny's, which combines gradients and non-maximum suppression to detect edges.
These preprocessing techniques ensure that the processed images are in optimal condition for accurate and reliable lane detection, regardless of weather or lighting conditions.

2.2 Algorithm Implementation

The second module focuses on the implementation of traditional computer vision and deep learning algorithms. Classical techniques are implemented to identify lane lines and analyze their geometry under standard and complex conditions. In addition, convolutional neural networks are used to improve the accuracy of lane identification.

These models are trained and tested on various datasets, addressing challenges such as adverse weather conditions or roads without visible markings (see Figure 3).

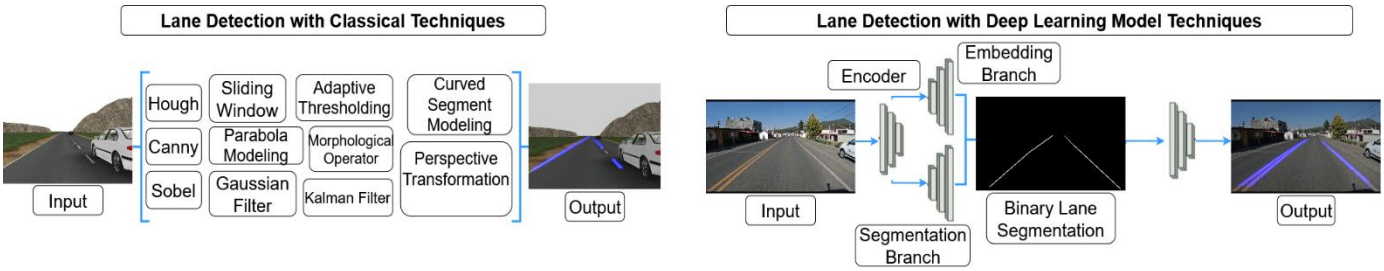

Fig. 3. Algorithm Implementation Module

Block 3: Lane Detection with Classical Algorithms

This block applies traditional computer vision techniques for rail line detection. Classical methods, such as the Hough Transform and the Canny edge detector, are effective under ideal conditions, where lane lines are clearly visible. Although these algorithms are less computationally demanding, they may have limitations in more complex scenarios. Table 4 below reviews in detail the methods and algorithms used, including techniques, improvements, and differentiators:

Table 4. Methods and algorithms of classical techniques

Method/Algorithm	Techniques	Enhancement	Differentiator	Reference
Canny, Hough.	Canny + Hough.	Improved detection of straight lines from edges.	Uses Canny for edge detection and Hough for straight lines.	(Shriwas et al., 2024)
Sobel, Hough.	Sobel + Hough.	Use the Sobel operator to detect edges before applying Hough.	Sobel highlights edges based on gradients gradient-based edges, Hough extracts lines.	(Mushtaq and Bedi, 2024)
Sobel, Adaptive Thresholding.	Sobel + Adaptive Thresholding	Improved adaptability to different lighting conditions.	Adaptive threshold adjusts detection according to the illumination.	(Ghanem et al., 2023)
Canny, Morphological.	Canny + Morphological Operator.	Improves the continuity of the detected edges.	Morphological operators correct discontinuities at the edges.	(Kishor et al., 2024)
Canny, Hough, Perspective Transformation.	Canny + Hough + Perspective Transformation.	Improves the visual representation of detected lanes.	Transformation corrects perspective for better visualization.	(Lee and Liu, 2023

Sobel, Perspective Transformation, Sliding Window.	Sobel + Perspective Transformation + Sliding Window.	Allows for a more detailed analysis of the rails in different sections.	Combination of different approaches and improves detection.	(Farag, 2020)
Canny, Perspective Transformation, Kalman.	Canny + Perspective Transformation + Kalman Filtering	Combines edge detection and motion tracking.	Kalman filter predicts trajectories after edge detection.	(Panev et al., 2019)
Sliding Window, Canny, Hough.	Sliding Window + Canny + Hough.	Adjusts lane detection based on position.	Dynamic analysis improves the detection of moving lanes.	(Panda and Mohanty, 2020)
Canny, Hough, Parabola Modeling.	Canny + Hough + Parabola Modeling.	Improved detection of curved rails.	Parabola modeling adjusts lines to smooth curves.	(Bilal et al., 2019)
Gaussian, Sobel, Curved.	Gaussian Filter + Sobel + Curved Segment Modeling.	Allows accurate detection of curved rails from smoothed edges.	Pre-smoothing improves detection of curved edges.	(Fakhfakh et al., 2020)

Block 4: Lane Detection with Convolutional Neural Networks (CNN)

This block utilizes CNNs to enhance the accuracy and robustness of lane detection under challenging conditions. The networks are trained on specialized datasets, enabling them to recognize distinct lane features, even when the markings are faint or degraded. During the training, network parameters are optimized through loss functions, which drive their improvement throughout the process.

Neural Network Training

Both the SCNN and VGG16 networks were initially trained exclusively on the Culane dataset to ensure proper generalization and avoid overtraining. Culane provides extensive labeled data for urban and highway lane detection, covering diverse conditions. In contrast, TuSimple, focused on highway scenarios, was reserved for testing purposes. This approach minimized overfitting by limiting training to a single dataset while leveraging TuSimple for evaluations and meaningful performance comparisons across different driving environments.

Model implementation

Neural networks such as VGG16 and SCNN (Spatial CNN) are employed due to their proven effectiveness in addressing adverse conditions, including roads with unclear markings or challenging weather. VGG16 is a deep convolutional neural network with 16 layers, comprising 13 convolutional and 3 fully connected layers. It utilizes small 3x3 filters and max pooling to reduce spatial dimensions while capturing hierarchical features, making it highly effective for identifying complex lane structures. SCNN extends traditional CNN architectures by incorporating spatial convolutional operations, which propagate information along horizontal and vertical directions. This unique design allows SCNN to model long-range spatial dependencies, enabling it to detect continuous lanes even in cases of sparse, broken, or occluded markings, making it particularly suited for challenging lane detection tasks.

2.3 Evaluation and Analysis

The third module focuses on the evaluation and analysis of the algorithms in real and simulated environments on simulated synthetic roads, state of the art databases and a proprietary dataset of roads in the state of Morelos.

Block 5: Evaluation in Simulated Environments

This block involves the initial evaluation of the algorithms in controlled simulations. These simulations allow recreating specific driving conditions, facilitating the measurement of performance in a controlled environment before performing real tests, using the Frida and Frida 2 image databases.

Block 6: Evaluation in Real Environments

This block is crucial to validate the algorithms using real-world data. Tests are performed with videos recorded on roads, including the dataset of the state of Morelos, which covers a variety of road scenes and adverse weather conditions such as curves, sharp curves, presence of rain, poor lighting conditions or darkness and presence of shadows on the road, also incorporated the TuSimple database, recognized as one of the most used datasets for lane detection.

Block 7: Evaluation Metrics

In this section, the metrics used to evaluate the algorithms implemented for lane detection are presented. Key quantitative metrics such as *precision, recall, F1 Score, accuracy, IoU, specificity,* and *Dice Index* were employed to precisely measure the effectiveness of the algorithms under various conditions and environments.
Additionally, specific indicators like true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*) were considered, providing a more detailed assessment of algorithm performance. These metrics offer a clear view of how the models correctly identify lanes and manage errors. Both classical computer vision methods and those based on Convolutional Neural Networks (CNNs) were evaluated in simulated scenarios and real-world tests using a proprietary dataset collected from roads in the state of Morelos. This rigorous and comparative evaluation enables the identification of the most effective approaches for implementing lane-keeping assistance systems. The comprehensive analysis of the metrics provides a deep understanding of each algorithm's performance, facilitating informed decisions for their application in real-world situations.

Precision is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

where $\boldsymbol{TP}$ represents true positives and $\boldsymbol{FP}$ represents false positives.
Recall is expressed as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

where $\boldsymbol{FN}$ represents the false negatives.
F1 Score is the harmonic mean of precision and recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{110}$$

where $\boldsymbol{TN}$ are the true negatives.
Intersection over Union *(IoU)* is calculated as:

$$\text{IoU} = \frac{|\text{A} \cap \text{B}|}{|\text{A} \cup \text{B}|} \tag{11}$$

where $\boldsymbol{A}$ is the predicted area and $\boldsymbol{B}$ is the ground truth area.
Specificity it is defined as:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{12}$$

where $\boldsymbol{TN}$ *are the true negatives and* $\boldsymbol{FP}$ *are the false positives.*
Dice Index it is calculated as:

$$\text{Dice} = \frac{2\,|\text{A} \cap \text{B}|}{|\text{A}| + |\text{B}|} \tag{13}$$

where $\boldsymbol{A}$ is the predicted area and $\boldsymbol{B}$ is the actual area.

Block 8: Comparative Analysis of Results

In this block, a comparative analysis of the results of different algorithms, both traditional and neural network-based, is performed.

3 Experimental Procedures

Below, five experiments are presented to evaluate the performance of computer vision algorithms for lane detection under various scenarios, using different databases and techniques. The objective of this series of experiments is to assess the effectiveness of the algorithms in diverse conditions and to gain a deeper understanding of their performance. The experiments conducted encompass various approaches to lane detection, beginning with the application of traditional computer vision techniques for detecting straight lanes in the Frida, Frida 2, TuSimple, and a proprietary database. This is followed by an exploration of extended traditional methods tailored to curved lanes, utilizing multiple datasets with road curves. Finally, the study transitions to advanced experimentation using deep learning models for lane detection. For the experimental phase, three databases were employed. The previously mentioned metrics were utilized to evaluate the performance of the lane detection models on the Frida and Frida 2 image databases, specifically designed for this purpose. The Frida database consists of 90 synthetic images organized into 18 urban road scenes, while Frida 2 includes 330 synthetic images covering 66 varied road scenes. Additionally, the TuSimple database, one of the most widely used datasets for lane detection, was included. The TuSimple dataset contains 55,640 images from 2,782 video clips, with each clip comprising 20 frames. Moreover, a proprietary database of roads from the state of Morelos was incorporated. This dataset includes 22,566 images from 41 video clips, featuring a variety of road scenes and adverse weather conditions, such as curves, sharp turns, rain, low lighting conditions, darkness, and road shadows. Figures 2, 3, and 4 provide a preview of the databases used for lane detection, with Figure 2 detailing the Frida and Frida 2 databases, Figure 3 covering the TuSimple database, and Figure 4 outlining the proprietary Morelos database.

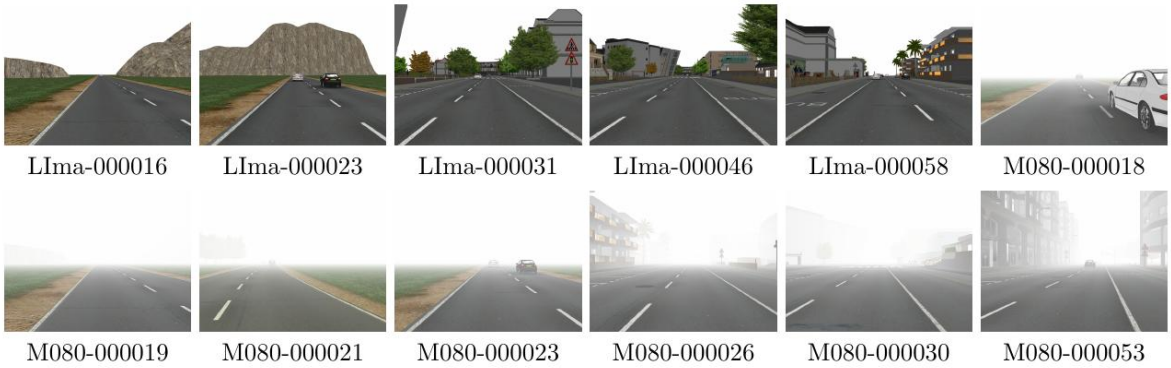

Fig.4. Samples from the Frida and Frida 2 databases,
featuring a variety of road scenes and weather conditions

Fig. 5. Samples from the TuSimple database with a variety of road scenes

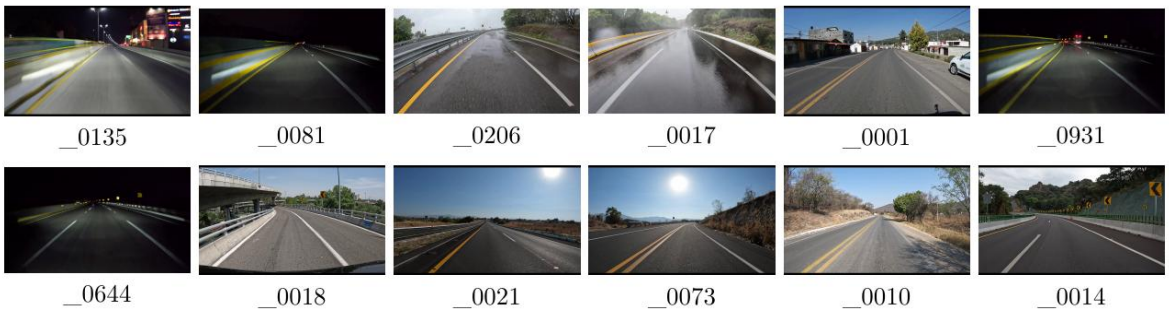

Fig. 6. Samples from the proprietary database Morelos with a variety of road scenes and weather conditions

3.1 Experimental Results and Discussion

This study involves five experiments aimed at evaluating lane detection performance under various challenging conditions. Each experiment utilizes different datasets and detection techniques, either traditional methods or deep learning models. The datasets include Frida, TuSimple, Morelos, and a combination of these, while the techniques range from classical methods to modern deep learning models. The experiments assess performance under conditions such as direct sunlight, reflections, fog, rain, shadows, and curves. The metrics evaluated include Precision, Recall, F1 Score, Accuracy, IoU, Specificity, and Dice Index. See Table 5.

Table 5. Summary for the 5 experiments

Experiment	Technique	Dataset	Adverse Conditions	Metrics Evaluated
Frida and Frida 2 (Straight Lanes)	Traditional	Frida, Frida 2	Direct Sunlight, Sun Glare, Fog (uniform, heterogeneous, cloudy)	
TuSimple (Straight Lanes)	Traditional	TuSimple	Direct Sunlight, Sun Glare, Shadows, Curves (light, moderate, sharp)	
Morelos Database (Straight Lanes)	Traditional	Own Database (Morelos)	Rain (light, moderate, heavy), Sunlight (Sun Glare), Curves, with Artificial Light, Night without Artificial Light	Precision, Recall, F1 Score, Accuracy, IoU, Specificity, Dice Index
Curved Lanes (Multiple Datasets)	Traditional	Frida, TuSimple, Morelos	Direct Sunlight, Sun Glare, Fog, Curves (different severities)	

Deep Learning Models (Multiple Datasets)	Deep Learning	Frida, TuSimple, Morelos	Various conditions (similar to the previous ones)

Traditional techniques on straight rails.

First experiment. In this experiment, lane detection algorithms based on traditional computer vision techniques were evaluated under adverse conditions. These conditions included direct sun, normal curves, and various stages of haze: uniform, heterogeneous, cloudy, and heterogeneous-cloudy haze. Images from the Frida and Frida 2 databases, which simulate driving conditions in reduced visibility scenarios, were used. The graphs of the metrics are presented in Figure 5, where the legend SA refers to Sobel + Adaptive, CM refers to Canny + Morphological, SCH refers to Sliding + Canny + Hough, and CH refers to Canny + Hough. In Figure 7, the results obtained by the different straight-line detectors are presented. Figure 8 presents the results obtained by the different straight-line detectors. For the confusion matrix details, see Table 6.

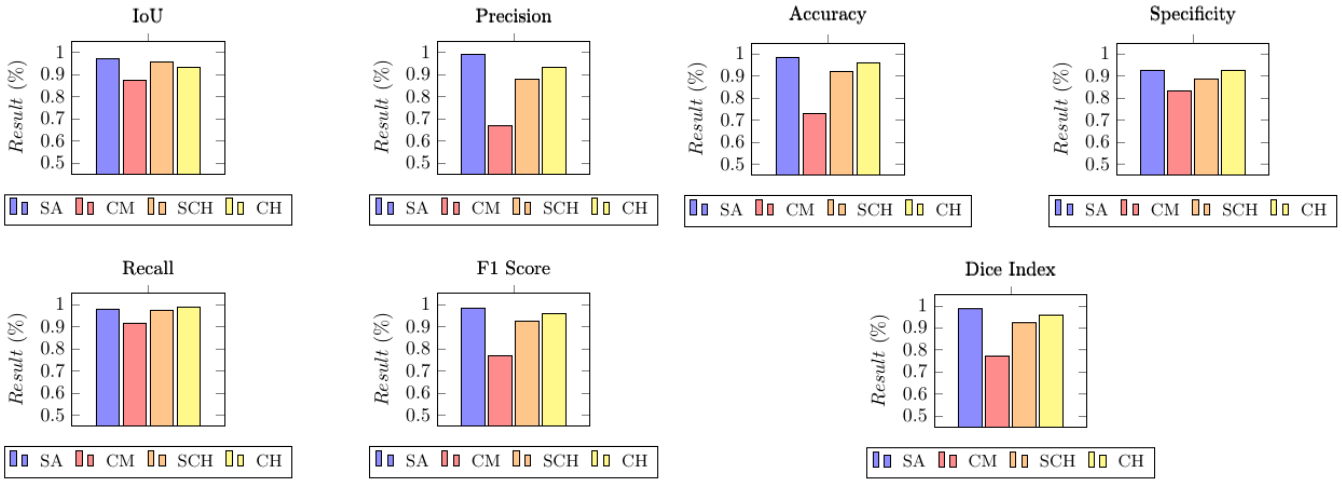

Fig.7. Quantitative results of lane detection on the Frida database

The Sobel + Adaptive approach demonstrated outstanding performance, achieving a high true positive rate and maintaining low false positive and false negative values, particularly excelling in foggy conditions. In contrast, the Canny + Morphological method showed less efficient performance due to a high false positive rate, although it successfully identified most lanes. The Sliding + Canny + Hough technique struck a good balance between true positives and negatives, minimizing false negatives and delivering solid performance overall. Finally, the Canny + Hough algorithm proved to be the most effective, offering high precision and recall even under challenging conditions.

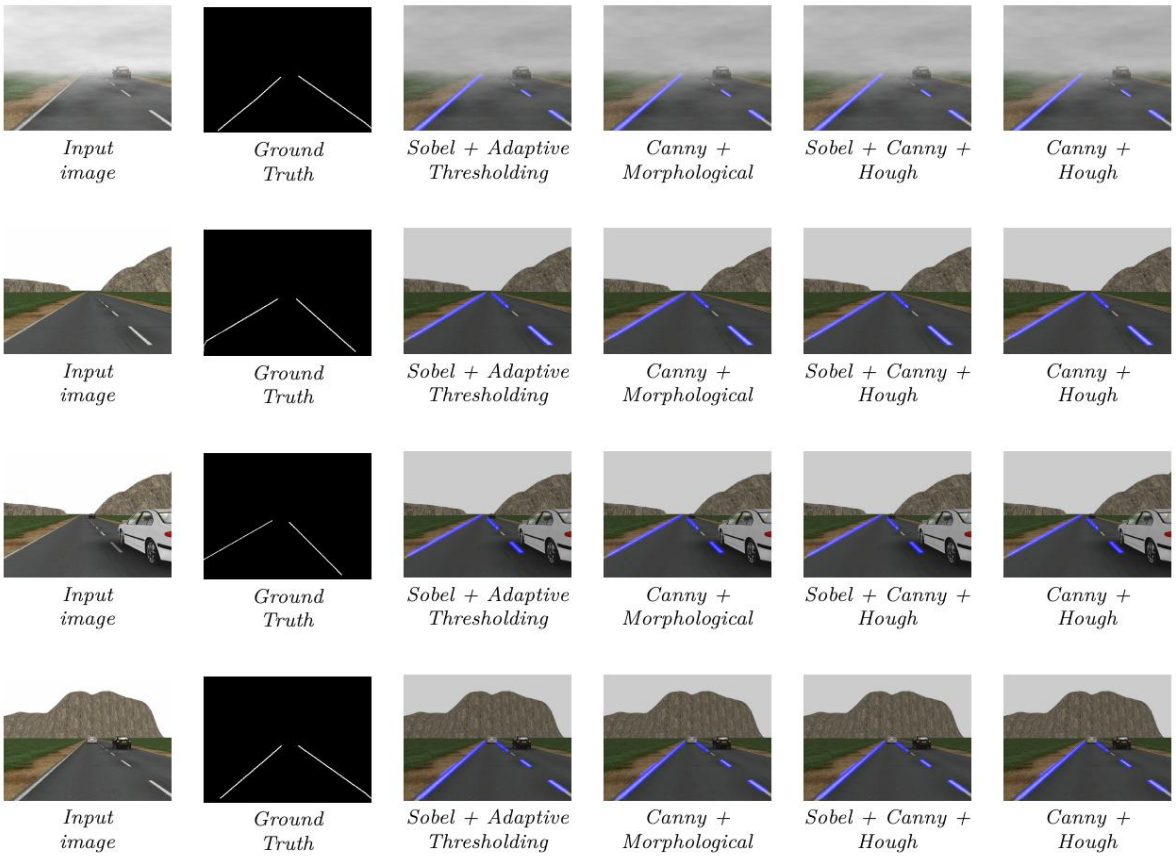

Fig.8. Qualitative results on the Frida database

Table 6. Confusion matrix elements for the experiment

Algorithm	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97995	0.01105	0.01905	0.98795
Canny + Morphological	0.91605	0.45905	0.08495	0.54195
Sliding + Canny + Hough	0.97195	0.13205	0.02695	0.86895
Canny + Hough	0.98895	0.07295	0.01105	0.92795

Second experiment. In this other experiment, lane detection algorithms based on traditional computer vision techniques were evaluated, applied to the TuSimple database. This database is widely used in the state of the art to evaluate lane detection methods in realistic and challenging driving conditions. The conditions evaluated included direct sunlight, sun glare, shadows, and curves of varying complexity: mild, moderate, and sharp. The metric charts are shown in Figure 9. Figure 10 presents the results obtained by the different straight-line detectors. For the confusion matrix details, see Table 7.

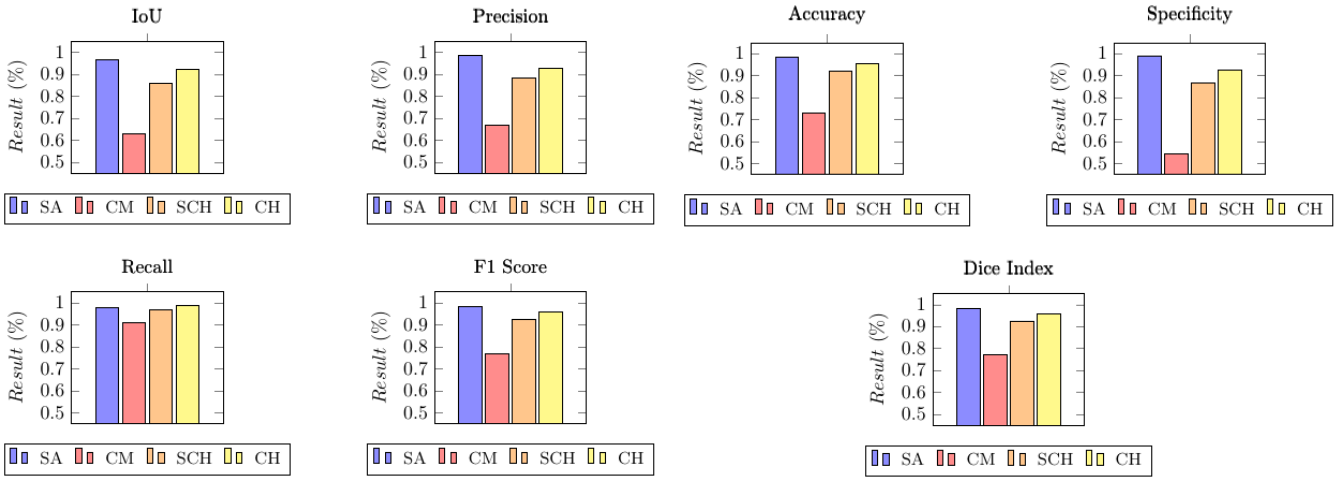

Fig.9. Quantitative results of lane detection in the TuSimple database

The Sobel + Adaptive method demonstrated outstanding performance, achieving a high true positive rate and low false positive values, with results consistent with those of previous experiments. On the other hand, the Canny + Morphological approach correctly detected a significant number of lanes but suffered from a high number of false positives, which reduced its overall effectiveness. The Sliding + Canny + Hough technique struck a good balance between true positives and false positives, delivering solid performance comparable to Sobel + Adaptive. Finally, the Canny + Hough algorithm proved to be the most effective, achieving high precision, recall, and F1 score, and demonstrating an exceptional ability to accurately detect lanes.

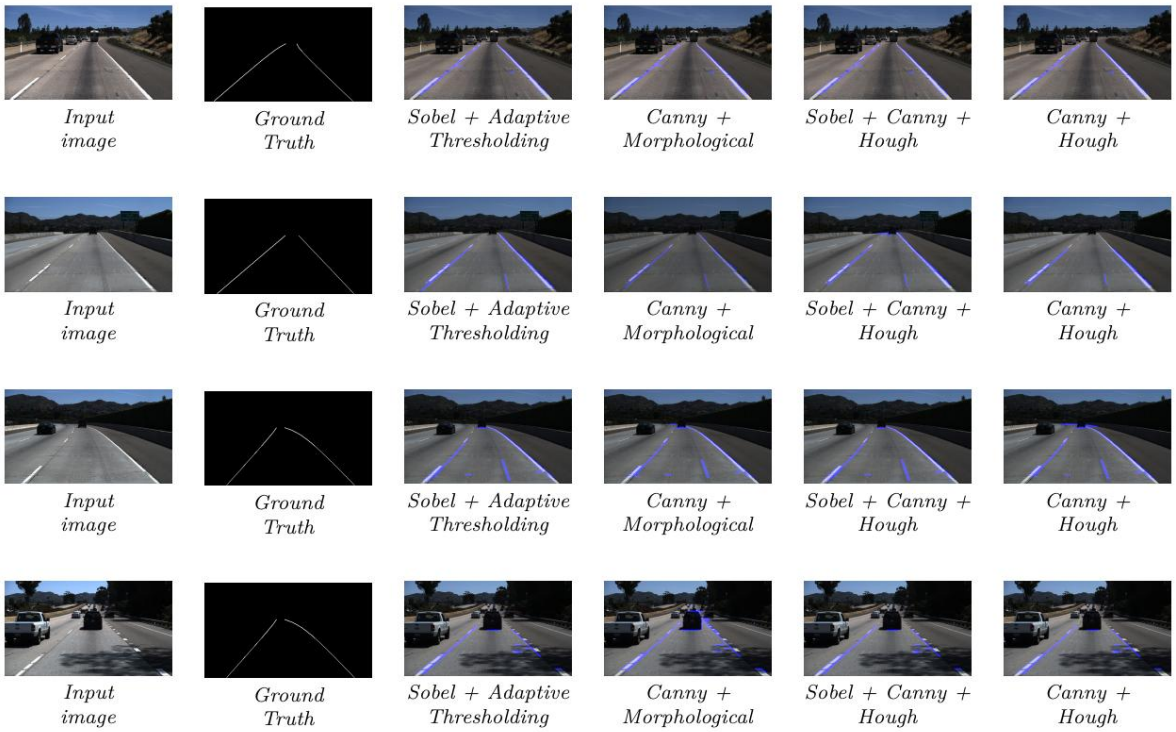

Fig.10. Qualitative results in the TuSimple database

Table 7. Confusion matrix elements for the experiment

Algorithm	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97734	0.01322	0.02156	0.98567
Canny + Morphological	0.91823	0.45734	0.08745	0.54456
Sliding + Canny + Hough	0.97344	0.13122	0.02834	0.87145
Canny + Hough	0.99034	0.07511	0.01089	0.93012

Third experiment. This experiment focused on the evaluation of lane detection algorithms based on traditional computer vision techniques, using a database specifically created to represent the unique conditions of roads in the state of Morelos. These conditions include adverse scenarios such as rain (with light, moderate, and heavy intensity), assessing the impact on road marking clarity; solar glare caused by direct sunlight, complicating edge detection; night driving with either no artificial lighting or partial road lighting; and variations in road surface, including different types of asphalt, wear, and deteriorated road markings. The metric graphs are presented in Figure 11. The results obtained by the different straight-line detectors are presented in Figure 12. For the confusion matrix details, see Table 8.

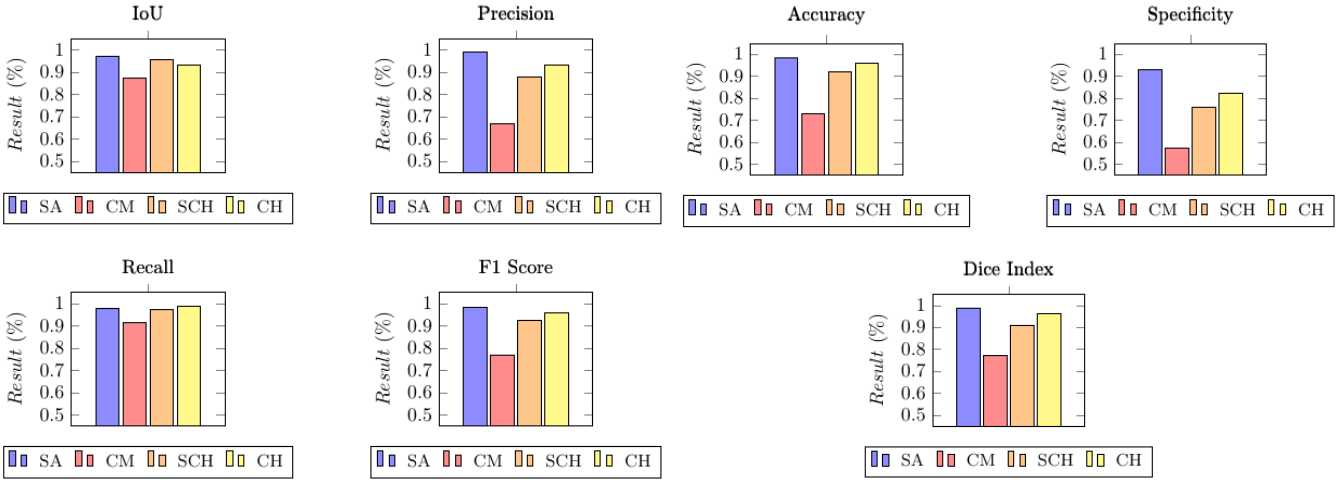

Fig.11. Quantitative results of lane detection on the proprietary database

Sobel+Adaptive: It showed the best overall performance, with an IoU of 0.96801 and an F1 Score of 0.98374. This algorithm stood out for its high accuracy and robustness against false positives. Canny+Morphological: It achieved the worst results, with a high percentage of false positives and low specificity. This suggests difficulties in complex scenarios, such as heavy rain and sun glare. Sliding+Canny+Hough: It demonstrated a balanced performance, excelling in scenarios with lighting variations and worn surfaces, achieving an F1 Score of 0.92269. Canny+Hough: Although it did not reach the values of Sobel + Adaptive, this algorithm was robust, especially in nighttime driving and fog conditions, obtaining an IoU of 0.91843 and an F1 Score of 0.95748.

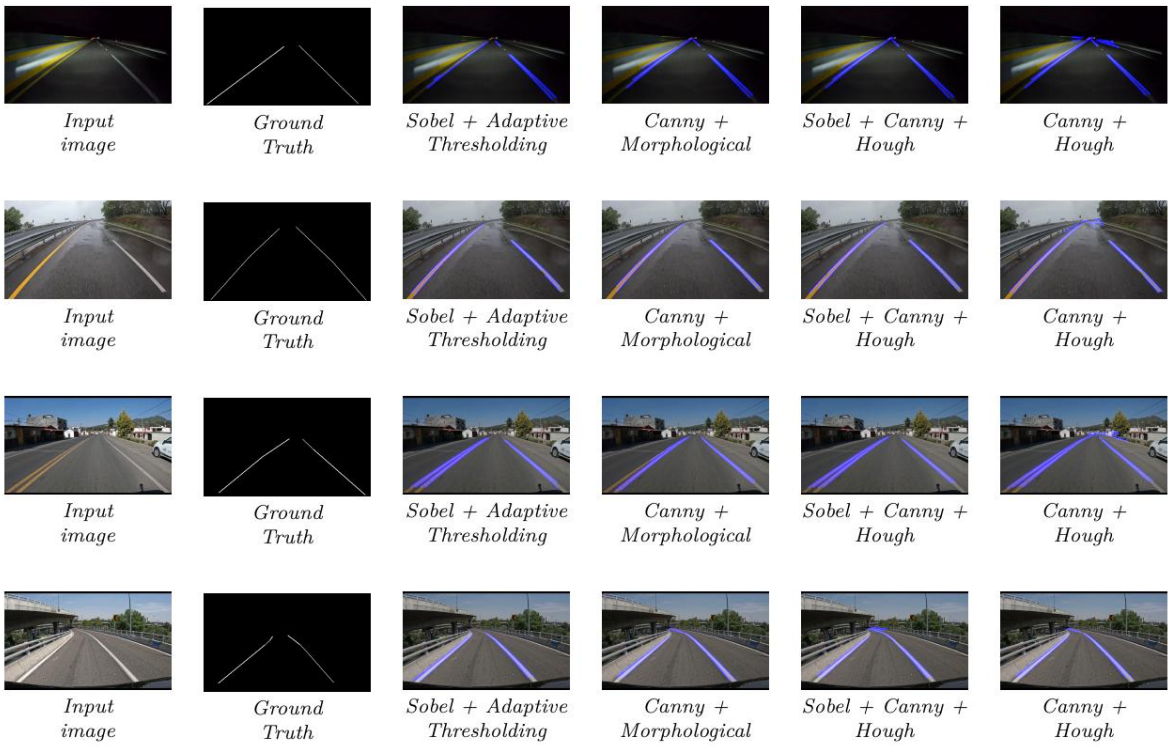

Fig.12. Qualitative results on the proprietary database

Table 8. Confusion matrix elements for the experiment

Algorithm	TP	FP	FN	TN
Sobel + Adaptive Thresholding	0.97888	0.01234	0.02001	0.98677
Canny + Morphological	0.91465	0.46012	0.08678	0.54389
Sliding + Canny + Hough	0.97023	0.13347	0.02911	0.87056
Canny + Hough	0.98712	0.07422	0.01345	0.92678

Extension of traditional techniques to curved lanes.

Fourth experiment. This experiment evaluated different lane detection methods for curved lanes using computer vision algorithms. Unlike straight lanes, curved lanes represent a greater challenge due to how the perception of the curve changes as the vehicle approaches. This variation in perspective can make curves appear sharper or smoother, depending on the distance and angle of view. The metric graphs are presented in Figure 11, where the legend CHP refers to Canny + Hough + Parable, GSC refers to Gaussian + Sobel + Curved, CHP2 refers to Canny + Hough + Perspective and SPS refers to Sobel + Perspective + Sliding. The metric graphs are presented in Figure 13. The results obtained by the different straight-line detectors are presented in Figure 14. For the confusion matrix details, see Table 9.

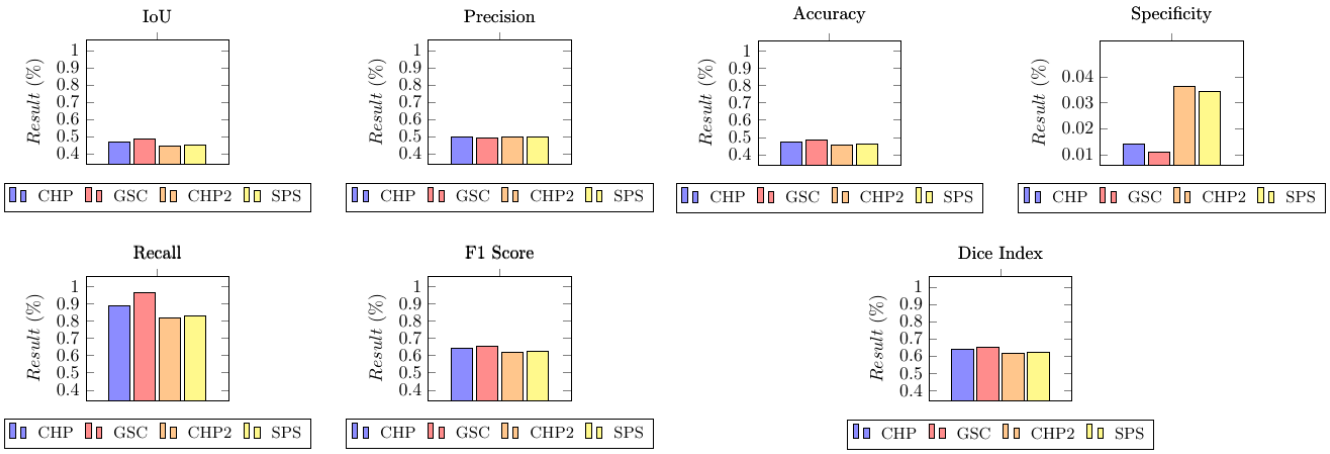

Fig.13. Qualitative results of curved lane detection across various datasets

Canny + Hough + Parabola (CHP) stood out due to its high lane detection rate, reflected in a high number of true positives (TP = 0.974). However, this performance comes at a significant cost, as it also showed a high rate of false positives (FP = 0.975). This suggests that while it is effective at detecting lanes, it tends to confuse other elements, such as lines or edges that are not part of the lane. On the other hand, Gaussian + Sobel + Curved (GSC) was the most balanced and precise algorithm. Its low false negative rate (FN = 0.035) indicates that it rarely misses lanes, and its lower proportion of false positives compared to CHP makes it robust, especially in sharp curves. The Canny + Hough + Perspective (CHP2) and Sobel + Perspective + Sliding (SPS) algorithms offered acceptable performance, but with limitations. Both had higher false negative and false positive rates, suggesting they are more prone to errors, especially in adverse conditions such as shadows, lighting changes, or very tight curve geometries.

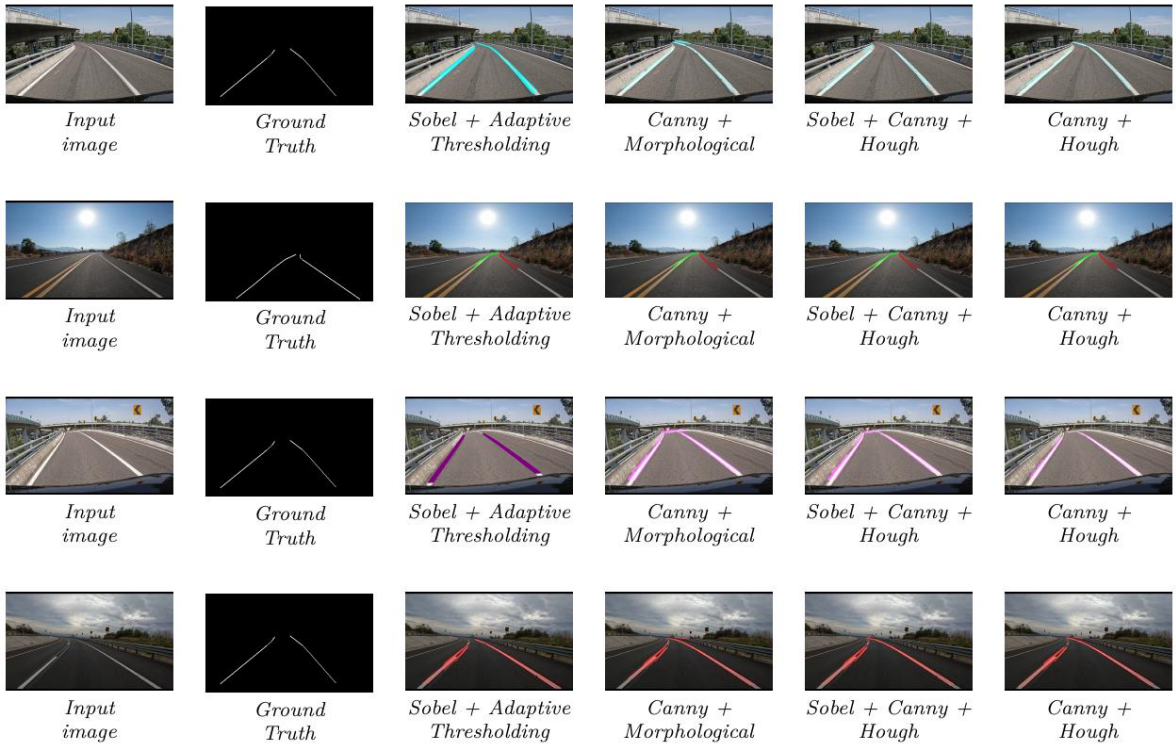

Fig.14. Quantitative results of curved lane detection across various datasets

Table 9. Confusion matrix elements for the experiment

Algorithm	TP	FP	FN	TN
Canny + Hough + Parabola	0.974	0.975	0.119	0.014
Gaussian + Sobel + Curved	0.959	0.988	0.035	0.011
Canny + Hough + Perspective	0.950	0.956	0.212	0.036
Sobel + Perspective + Sliding	0.951	0.957	0.193	0.034

Deep Learning Models Techniques.

Fifth Experiment: In this experiment, deep learning techniques were adopted for lane detection, moving beyond the traditional computer vision methods used in previous studies. Two neural network architectures were evaluated: SCNN (Spatial Convolutional Neural Network) and VGG16. The first, specifically designed to capture the spatial structure of roads, is capable of modeling relationships between different points of the lanes, making it particularly effective in complex scenarios such as sharp curves or multiple lane lines. Meanwhile, VGG16, with its deep convolutional design, allows for hierarchical feature extraction, which is crucial for tackling challenging conditions such as shadows, worn lane lines, or the presence of obstacles.

For this experiment, the networks were evaluated using three datasets: Frida & Frida 2, TuSimple, and a proprietary dataset developed specifically to represent scenarios from the state of Morelos. These datasets included diverse environmental conditions, such as daylight, night, rain, fog, and a variety of roads with sharp curves and worn lane markings. The metric graphs are presented in Figure 15, and the results obtained by the different SCNN & VGG16 architectures for the three datasets are shown in Figures 16-18. For the confusion matrix details, see Table 10.

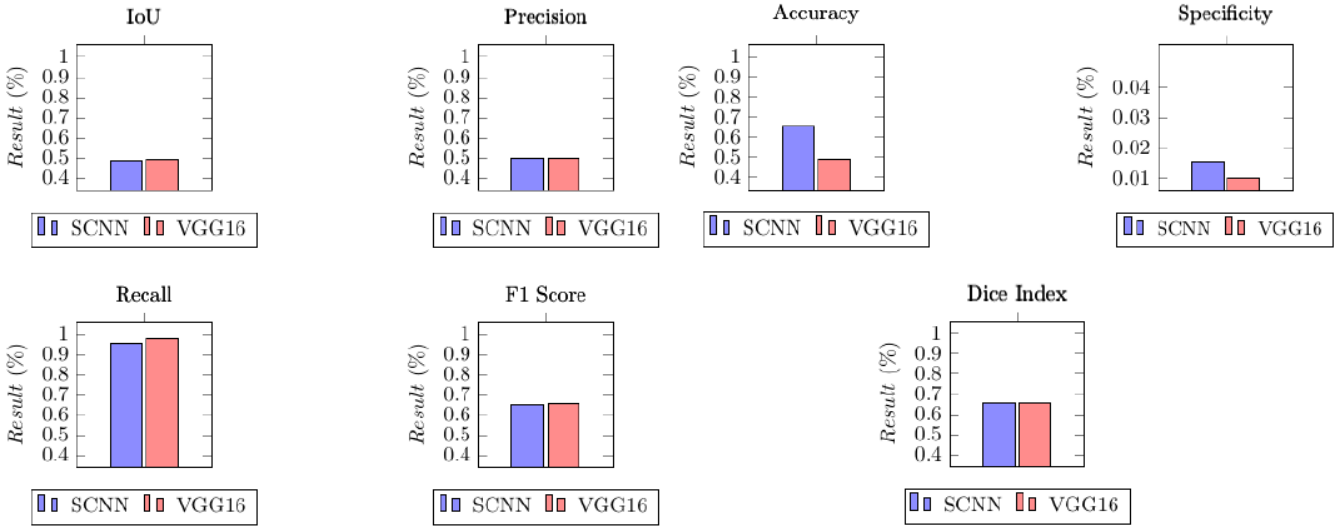

Fig.15. Quantitative results of lane detection with deep learning on the proprietary database

Both architectures showed high recall values, indicating that they are highly effective at detecting most of the present lanes. However, VGG16 slightly outperformed SCNN in balancing precision and recall, reflected in a higher F1 Score (0.65588 vs. 0.65165). Both faced challenges in reducing false positives, a crucial aspect for improving specificity.

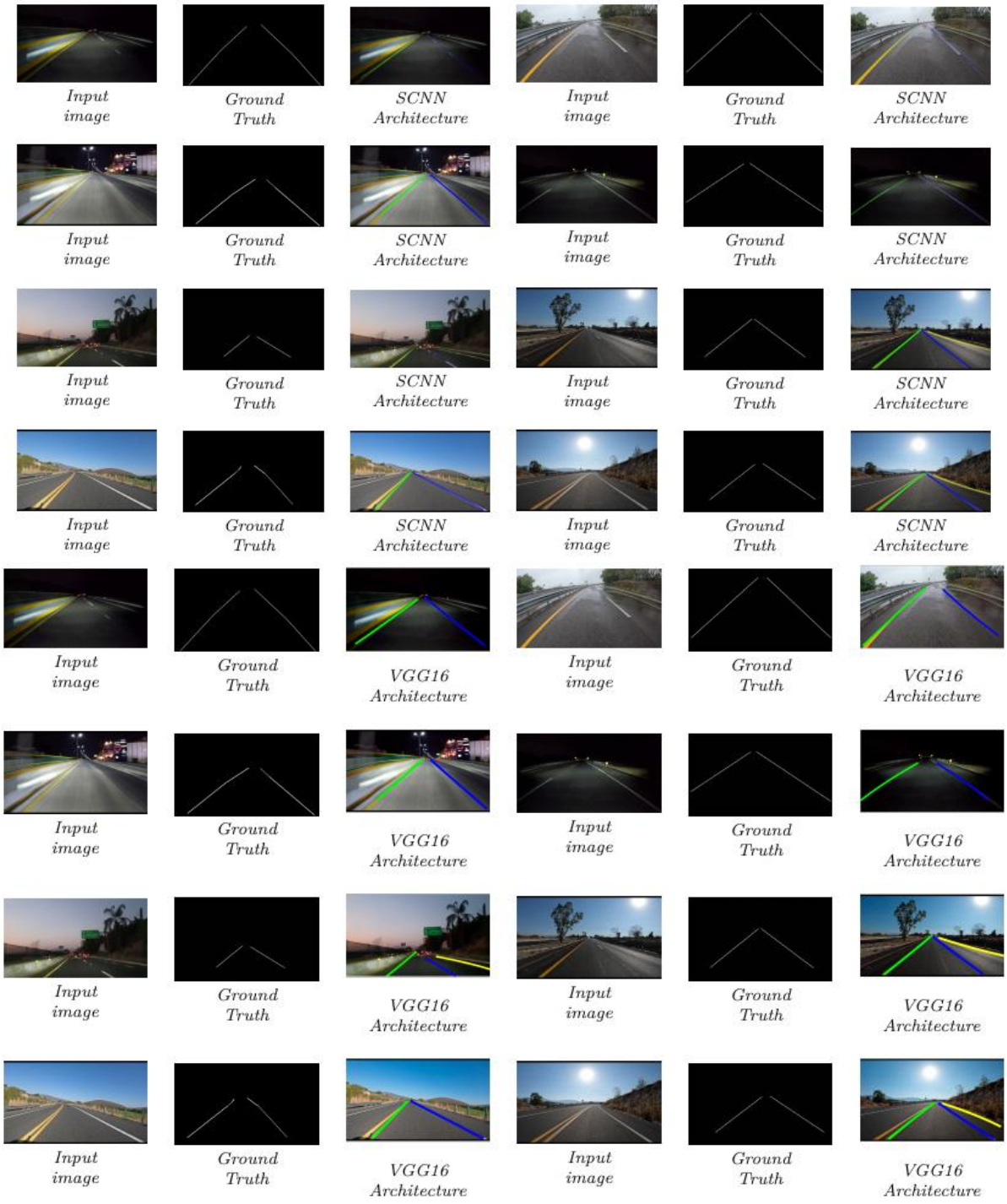

Fig.16. Qualitative results with SCNN & VGG16 architectures on the proprietary database

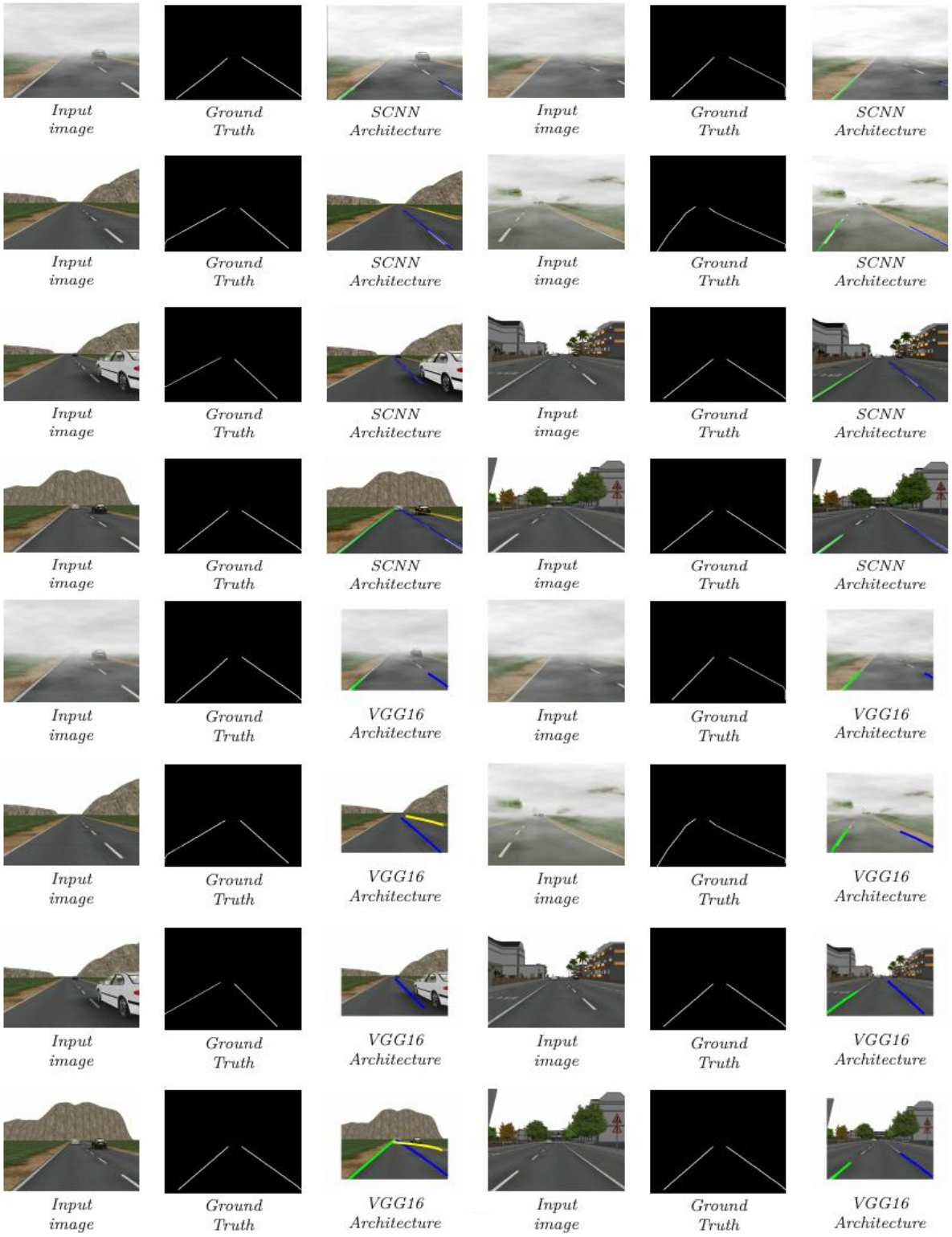

Fig.17. Qualitative results with SCNN & VGG16 architectures on the Frida & Frida 2 database

Fig.18. Qualitative results with SCNN & VGG16 architectures on the TuSimple database

Table 10. Confusion matrix elements for the experiment

Algorithm	TP	FP	FN	TN
SCNN	0.955	0.960	0.045	0.015
VGG16	0.960	0.980	0.020	0.010

As shown in Table 11, the highlighted approaches for each experiment demonstrate the best-performing configurations based on the confusion matrix metrics. The table summarizes the True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) for each selected approach.

Table 11. Highlighted Approaches by Experiment

Experiment	Highlighted Approach	TP	FP	FN	TN
1	Canny + Hough	0.988	0.072	0.011	0.927
2	Canny + Hough	0.990	0.075	0.010	0.930
3	Canny + Hough	0.987	0.074	0.013	0.926
4	Gaussian+Sobel+Curve	0.959	0.988	0.035	0.011
5	VGG16	0.960	0.988	0.020	0.010

The table presenting metrics derived from the confusion matrix including True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) focuses specifically on the model's binary classification performance. These metrics evaluate how many lanes were correctly detected (TP), how many were incorrectly classified as present when they were not (FP), how many were missed (FN), and how many were correctly identified as absent (TN). This assessment provides a more precise understanding of the model's ability to distinguish between lane presence and absence, which is crucial for minimizing classification errors. While confusion matrix-based metrics offer detailed insights into classification accuracy (true and false positives and negatives), broader performance metrics such as Intersection over Union (IoU), Precision, Recall, F1 Score, Accuracy, Specificity, and Dice Index provide a more comprehensive evaluation of the model's segmentation capabilities and overall effectiveness.

Comparison of Algorithms Across Experiments

Table 12 presents a comparative analysis of the best-performing algorithms for each experiment based on key evaluation metrics, including Intersection over Union (IoU), Precision, Recall, F1 Score, Accuracy, Specificity, and Dice Index. The highlighted algorithm in each experiment corresponds to the one that achieved the highest performance across most metrics.

Table 12. Best-Performing Algorithms by Experiment

Experiment	Best Algorithm	IoU	Precision	Recall	F1 Score	Accuracy	Specificity	Dice Index
1	Sobel + Adaptive Thresholding	0.9729	0.9888	0.9809	0.9848	0.9849	0.9889	0.9848
2	Sobel + Adaptive Thresholding	0.9656	0.9866	0.9784	0.9825	0.9825	0.9867	0.9825
3	Sobel + Adaptive Thresholding	0.9680	0.9875	0.9799	0.9837	0.9838	0.9876	0.9837
4	Gaussian + Sobel + Curved	0.4838	0.4925	0.9647	0.6521	0.4867	0.0110	0.6521
5	VGG16	0.4898	0.4948	0.9796	0.6575	0.4924	0.0101	0.6575

These metrics serve to evaluate the overall performance of lane detection algorithms. They include Intersection over Union (IoU), Precision, Recall, F1 Score, Accuracy, Specificity, and the Dice Index. Together, these metrics offer a comprehensive assessment of the model's ability to accurately detect lane markings in images. IoU quantifies the overlap between the model's prediction and the ground truth, while Precision and Recall evaluate the model's capability to minimize false positives and correctly identify true positives, respectively. The F1 Score provides a balanced measure that combines Precision and Recall, and Accuracy reflects the overall proportion of correct predictions.

4 Conclusions

Based on the results obtained from the various experiments, significant conclusions can be drawn regarding the performance of the algorithms under different conditions. In the conducted experiments, the Sobel + Adaptive Thresholding and Canny + Hough algorithms demonstrated outstanding performance, particularly in simple or ideal conditions, where they achieved high values in key metrics such as IoU (up to 0.97295 for Sobel + Adaptive Thresholding in Experiment 1) and Precision (up to 0.98885 for

Sobel + Adaptive Thresholding in the same experiment). These algorithms maintained superior performance in terms of Recall, F1 Score, Accuracy, and Dice Index, suggesting a high capability for correctly detecting lanes in straightforward scenarios. However, as test conditions became more complex, as evidenced in Experiment 4, traditional algorithms such as Canny + Hough + Parabola and Gaussian + Sobel + Curved experienced a significant decline in performance. These methods exhibited low IoU values (around 0.47) and moderate Precision (around 0.49), indicating that they are not as effective in handling challenging situations such as sharp curves or degraded lane markings. This behavior highlights the limitations of classical approaches in managing more dynamic and complex conditions. On the other hand, deep neural network-based models such as SCNN and VGG16 demonstrated superior performance in terms of Recall and F1 Score, particularly in more difficult scenarios. In Experiment 5, VGG16 achieved a Recall of 0.9796 and a Dice Index of 0.6575, indicating a remarkable ability to detect lanes in challenging conditions with a lower false negative rate. Although metrics such as IoU and Precision were comparable between deep networks and traditional algorithms, the higher Recall suggests that neural networks are more sensitive to lane detection, which is crucial in dynamic traffic environments. The results suggest that deep neural network-based algorithms, particularly VGG16, offer a significant advantage over traditional methods when adapting to complex conditions. While Sobel + Adaptive Thresholding and Canny + Hough are effective in simple environments, neural networks exhibit greater generalization capabilities and more robust performance in dynamic settings. This reinforces the importance of considering neural networks as a preferred option for lane detection in real-world scenarios, where conditions can vary significantly.

References

SAE International. (2021, April). *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles* (Technical Report J3016_202104). Warrendale, PA: SAE International. https://doi.org/10.4271/J3016_202104

Zakaria, N. J., Shapiai, M. I., Ghani, R. A., Yassin, M. N. M., Ibrahim, M. Z., & Wahid, N. (2023). Lane detection in autonomous vehicles: A systematic review. *IEEE Access, 11*, 3729–3765. https://doi.org/10.1109/ACCESS.2023.3234442

Zakaria, N. J., Shapiai, M. I., Ghani, R. A., Wahid, N., & Lai, D. T. C. (2024). Fully convolutional network model applied attention mechanism on Kitti lane dataset for lane detection. *Journal of Advanced Research in Applied Sciences and Engineering Technology, 39*(2), 166–180. https://doi.org/10.37934/araset.39.2.166180

Bansal, J. C., et al. (2021). *Applications of advanced computing in systems* (1st ed.). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-33-4862-2

Wang, Q., Han, T., Qin, Z., Gao, J., & Li, X. (2022). Multitask attention network for lane detection and fitting. *IEEE Transactions on Neural Networks and Learning Systems, 33*(3), 1066–1078. https://doi.org/10.1109/TNNLS.2020.3039675

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2020). Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Transactions on Vehicular Technology, 69*(1), 41–54. https://doi.org/10.1109/TVT.2019.2949603

Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018). Spatial as deep: Spatial CNN for traffic scene understanding. *arXiv*. https://doi.org/10.48550/arXiv.1712.06080

Yoo, S., et al. (2020). End-to-end lane marker detection via row-wise classification. *arXiv*. https://doi.org/10.48550/arXiv.2005.08630

Tarel, J.-P. (n.d.). *BDD-Frida: Dataset for lane detection* [Data set]. Laboratoire de Cybernétique de l'École Polytechnique. Retrieved 22 March 2023, from http://perso.lcpc.fr/tarel.jean-philippe/bdd/fri/da.html

Álvarez Silva, S. (n.d.). *Lane Detection Dataset Morelos Sergio* [Data set]. Kaggle. Retrieved 10 December 2024, from https://www.kaggle.com/datasets/sergiolvarezsilva/lane-detection-dataset-morelos-sergio

Ren, D., Shang, W., Zhu, P., Hu, Q., Meng, D., & Zuo, W. (2020). Single image deraining using bilateral recurrent network. *IEEE Transactions on Image Processing, 29*(1), 6852–6863. https://doi.org/10.1109/TIP.2020.2994443

Goceri, E. (2023). Evaluation of denoising techniques to remove speckle and Gaussian noise from dermoscopy images. *Computer Biology and Medicine, 152*, 106474. https://doi.org/10.1016/j.compbiomed.2022.106474

Haider, S. S. M., & Kumar, [initial unknown]. (2023). Road lane line detection based on ROI using Hough transform algorithm. In *Proceedings of the Third International Conference on Computing, Communications, and Cyber-Security* (pp. 567–580). Springer Nature Singapore.

Qu, F. (2023). Image defogging algorithm based on physical prior and contrast learning. In *Proceedings of the 5th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM 2023)* (pp. 208–216). https://doi.org/10.1049/icp.2023.2940

Li, X. (2020). Image defogging algorithm based on dark channel prior of adaptive weight. *Journal of Physics: Conference Series, 1650*(3), 032067. https://doi.org/10.1088/1742-6596/1650/3/032067

Roy, S., Bhalla, K., & Patel, R. (2024). Mathematical analysis of histogram equalization techniques for medical image enhancement: A tutorial from the perspective of data loss. *Multimedia Tools and Applications, 83*(5), 14363–14392. https://doi.org/10.1007/s11042-023-15799-8

Cui, Z., et al. (2022). You only need 90K parameters to adapt light: A lightweight transformer for image enhancement and exposure correction. *arXiv*. https://doi.org/10.48550/arXiv.2205.14871

Javeed, M. A., et al. (2023). Lane line detection and object scene segmentation using Otsu thresholding and the fast Hough transform for intelligent vehicles in complex road conditions. *Electronics, 12*(5), 1079. https://doi.org/10.3390/electronics12051079

Zhang, J., Guo, X., Zhang, C., & Liu, P. (2021). A vehicle detection and shadow elimination method based on greyscale information, edge information, and prior knowledge. *Computers & Electrical Engineering, 94*, 107366. https://doi.org/10.1016/j.compeleceng.2021.107366

Ghanem, S., Kanungo, P., Panda, G., & Parwekar, P. (2023). An improved and low-complexity neural network model for curved lane detection of autonomous driving system. *Soft Computing, 27*(1), 493–504. https://doi.org/10.1007/s00500-021-05815-0

Salvi, M., Acharya, U. R., Molinari, F., & Meiburger, K. M. (2021). The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. *Computer Biology and Medicine, 128*, 104–129. https://doi.org/10.1016/j.compbiomed.2020.104129

Shriwas, R. N., Bodkhe, Y., Mane, A., & Kulkarni, R. (2024). Overview of Canny edge detection and Hough transform for lane detection. In *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0* (pp. 1–5). IEEE. https://doi.org/10.1109/OTCON60325.2024.10688024

Mushtaq, F., & Bedi, H. (2024). A review based on the comparison between Canny edge detection and Sobel algorithm. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4485325

Ghanem, S., Kanungo, P., Panda, G., Satapathy, S. C., & Sharma, R. (2023). Lane detection under artificial coloured light in tunnels and on highways: An IoT-based framework for smart city infrastructure. *Complex & Intelligent Systems, 9*(4), 3601–3612. https://doi.org/10.1007/s40747-021-00381-2

Kishor, S., Nair, R. R., Babu, T., Sindhu, S., & Vilashini, S. V. (2024). Lane detection for autonomous vehicles with Canny edge detection and general filter convolutional neural network. In *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)* (Vol. 1, pp. 1331–1336). https://doi.org/10.23919/INDIACom61295.2024.10499078

Lee, D.-H., & Liu, J.-L. (2023). End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal, Image and Video Processing, 17*(1), 199–205. https://doi.org/10.1007/s11760-022-02222-2

Farag, W. (2020). A comprehensive real-time road-lanes tracking technique for autonomous driving. *International Journal of Computing and Digital Systems, 9*(3), 349–362. https://doi.org/10.12785/IJCDS/090302

Panev, S., Vicente, F., De La Torre, F., & Prinet, V. (2019). Road curb detection and localization with monocular forward-view vehicle camera. *IEEE Transactions on Intelligent Transportation Systems, 20*(9), 3568–3584. https://doi.org/10.1109/TITS.2018.2878652

Panda, L., & Mohanty, B. P. (2020). Recent developments in lane departure warning system: An analysis. *Zibeline International Publishing*, 151–153. https://doi.org/10.26480/etit.02.2020.151.153

Bilal, H., Yin, B., Khan, J., Wang, L., Zhang, J., & Kumar, A. (2019). Real-time lane detection and tracking for advanced driver assistance systems. In *2019 Chinese Control Conference (CCC)* (pp. 6772–6777). IEEE. https://doi.org/10.23919/ChiCC.2019.8866334

Fakhfakh, M., Chaari, L., & Fakhfakh, N. (2020). Bayesian curved lane estimation for autonomous driving. *Journal of Ambient Intelligence and Humanized Computing, 11*(10), 4133–4143. https://doi.org/10.1007/s12652-020-01688-7