Knowledge-Based Systems 240 (2022) 107941

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep embedded hybrid CNN–LSTM network for lane detection on NVIDIA Jetson Xavier NX

Yassin Kortli [a,b,c,*], Souhir Gabsi [c], Lew F.C. Lew Yan Voon [b], Maher Jridi [a], Mehrez Merzougui [c,d], Mohamed Atri [c,d]

[a] LABISEN, VISION-AD, ISEN Yncréa Ouest, site de Nantes, 33 Quater Chemin du Champ de Manøeuvre 44470 Carquefou, France
[b] Laboratoire ImViA, Université de Bourgogne, France
[c] Electronic and Micro-electronic Laboratory, Faculty of Sciences of Monastir, University of Monastir, Tunisia
[d] College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia

ARTICLE INFO

Article history:
Received 11 August 2021
Received in revised form 13 November 2021
Accepted 10 December 2021
Available online 7 January 2022

Keywords:
Embedded deep LDWS
CNN Encoder–Decoder network
Long Short-Term Memory network
Lane detection

ABSTRACT

In recent years, lane detection has become one of the most important factors in the progress of intelligent vehicles. To deal with the challenging problem of low detection precision and real-time performance of most traditional systems, we proposed a real-time deep lane detection system based on CNN Encoder–Decoder and Long Short-Term Memory (LSTM) networks for dynamic environments and complex road conditions. The CNN Encoder network is used to extract deep features from a dataset and to reduce their dimensionality. A corresponding decoder network is used to map the low resolution encoder feature maps to dense feature maps that correspond to road lane. The LSTM network processes historical data to improve the detection rate through the removal of the influence of false alarm patches on detection results. We propose three network architectures to predict the road lane: CNN Encoder–Decoder network, CNN Encoder–Decoder network with the application of Dropout layers and CNN Encoder–LSTM-Decoder network that are trained and tested on a public dataset comprising 12764 road images under different conditions. Experimental results show that the proposed hybrid CNN Encoder–LSTM-Decoder network that we have integrated into a Lane-Departure-Warning-System (LDWS) achieves high prediction performance namely an average accuracy of 96.36%, a Recall of 97.54%, and a F1-score of 97.42%. A NVIDIA Jetson Xavier NX supercomputer has been used, for its performance and efficiency, to realize an Embedded Deep LDWS.

1. Introduction

In recent years, road accidents have become one of the most common causes of fatal injuries, numbering in the millions. As a result, intelligent vehicle safety has continued to improve year after year, thanks to the advances in the development of many passive safety systems, such as seat belts and airbags, which are now found in virtually every commercial vehicle [1–3]. However, the latest safety systems are designed not only to minimize potential injuries in the case of accidents, but also to prevent accidents from arising. Advanced Driver Assistance Systems (ADAS) provide assistance to the driver on the road, and therefore, enhance the driving experience. The main objective of ADAS is to improve the vehicle's security and to protect other vehicles on the road, as well as the driver and pedestrians or cyclists. The demand for ADAS systems has exploded in recent years due to the general desire to build safer vehicles and roads in order to reduce the number of road deaths [4–7].

ADAS systems must be able to recognize objects, road signs, the road itself, and any other moving vehicles on the road, in order to take real-time decisions to either warn the driver or act directly in his place. Due to advances in the applications of Machine Learning techniques, mobile robotics, self-driving and driver assistance systems are clear examples of areas in which the application of Deep Learning approaches, e.g. Convolutional Neural Network (CNN) [8–12], Recurrent Neural Network (RNN) [4,12,13], Long Short-Term Memory (LSTM) [3,14], *etc.*, can offer new perspectives. More specifically, Deep Learning (DL) techniques are considered as one of the most interesting solutions for computer vision problems [15–19]. DL approaches have been used for object detection and recognition, segmentation, feature extraction, classification and regression tasks [20–27]. One of the areas where DL is opening new perspectives is the development

* Corresponding author at: Laboratoire ImViA, Université de Bourgogne, France.
E-mail addresses: yassin.kortli@isen-ouest.yncrea.fr (Y. Kortli), souhir.gabsi@fsm.rnu.tn (S. Gabsi), lew.lew-yan-voon@u-bourgogne.fr (L.F.C. Lew Yan Voon), maher.jridi@isen-ouest.yncrea.fr (M. Jridi), mhrez@kku.edu.sa (M. Merzougui), mohamed.atri@fsm.rnu.tn (M. Atri).

https://doi.org/10.1016/j.knosys.2021.107941
0950-7051/

of new ADAS [1,4,17,18] such as lane departure warning, forward collision warning, traffic sign recognition, and so on.

The main subject of this paper is the design of a deep-based network that uses vision and Artificial Intelligence (AI) techniques to predict road lane, based on images acquired in real time by a camera installed inside the vehicle. One of the specificity of our method is that the deep learning model is also trained with data acquired directly by the camera. Furthermore, the proposed deep-based network is integrated in a Lane-Departure-Warning-System (LDWS) capable of avoiding potential lane departure situations [28]. Due to its performance and efficiency, a NVIDIA Jetson Xavier NX supercomputer is used to realize the Embedded Deep LDWS for automatic driving movement control. Compared to state-of-the-art systems, this paper highlights the following main contributions:

- A novel Deep Learning architecture based solution to predict road lane for vehicle movement control.
- A combination of three deep networks: CNN Encoder–Decoder network, CNN Encoder–Decoder network with Dropout layer network and CNN Encoder–LSTM-Decoder network, to, automatically and efficiently, predict road lane.
- A false alarm removal solution based on a combined deep CNN Encoder–LSTM-Decoder network. The purpose of the LSTM network is proposed to improve the detection rate through the removal of false alarm patches.
- A post-processing step that enabled us to realize a Deep Embedded Lane Departure Warning System (LDWS) using a NVIDIA Jetson Xavier NX supercomputer.
- A detailed experimental analysis in terms of Loss, Accuracy, Precision, Recall or sensitivity, F1-score and IoU to evaluate the proposed networks and system performance.

The rest of the paper is organized in the following way: A review of recent related work is presented in Section 2. Section 3 describes the proposed networks for road lane prediction, and the dataset collection and preparation. Section 4 presents a series of experimental results and comparative performance to verify the effectiveness of the proposed networks and to select the best one to realize an Embedded Deep LDWS based on a NVIDIA Jetson Xavier NX supercomputer. Section 5 discusses and compares the performance of our system with existing systems. Finally, Section 6 concludes the paper.

2. State-of-the-art

The increase in the number of vehicle ownerships each year has made traffic safety an important factor in the development of a city. Traffic accidents are mainly caused by subjective reasons related to the driver, such as fatigue, drunkenness and driving errors. With intelligent vehicles, these human factors can be eliminated to some extent [1,4,28,29]. Recently, the manufacturing of intelligent vehicles has begun to attract the attention of researchers in related fields around the world. They can intelligently assist humans in driving tasks based on real-time traffic information, highlighting their importance in improving driving safety and freeing humans from tedious driving environments [5,30].

A road lane detection system is an important factor in the development of intelligent vehicles. It has a direct effect on the driving behaviors. Using such a system, it is possible to determine an effective driving direction and provide the accurate position of the car in the road lane. Therefore, it is necessary to conduct a comprehensive study on this topic.

Current lane detection systems use mainly visual sensors that capture the road scenes in front of the vehicle through cameras [4, 28,31]. These sensors have a wide response spectrum that allows them to detect infrared rays, and they can operate continuously with high adaptability and for long periods. However, many challenges complicate the accuracy of lane detection, such as complex backgrounds on both sides, insufficient light, vehicle occlusion problems, and daily natural condition, etc [32–35]. Lane detection techniques are mostly divided into feature-based detection [28,30,36–43] and model-based detection [1,4,5,28,29,31,44–48].

The first techniques separate the lane from the road scene based on the color and edge features. Ghazali, et al. [49] proposed a fast and improved system based on H-MAXIMA transform and improved Hough Transform algorithm to detect unexpected lane changes. First, the region of interest of the input image is defined to reduce the search space, and then the Hough Transform is applied to detect the lane markers after image noise filtering. Zheng et al. [50] applied a system that directly identifies the boundary line in the Hough space, so as to simplify the Hough Transform-based boundary line detection algorithm. The image is subjected to the Hough Transform, and the points corresponding to the parallelism, length and angle, and intercept features of the line are selected in the Hough space. Compared to the traditional algorithm, experimental results showed that the identification is effectively improved on fast lanes and structured roads. Huang et al. [6] used the linear Hough Transform (HT)-based straight lane detection algorithm to evaluate possible perception problems in challenging scenarios, including altered lighting conditions, different weather and hue conditions, and different road types. In fact, the HT-based algorithm was found to have an acceptable detection rate in simple contexts, such as driving on a highway or in conditions with distinct contrast between lane boundaries and their surroundings. In contrast, it failed to detect roadway boundaries under a variety of lighting conditions. Ghanem et al. [3] developed a novel lane detection and tracking method for autonomous vehicle in the IoT-Based Framework (IBF). It consists of three blocks: Vehicle Board (VB), Cloud Module (CM) and Vehicle Remote Control (VRC). In addition, to the detection of lane markers under different light conditions, an illumination invariance operation is introduced. Simulation results present a lane-keeping rates of 95.3% in tunnels and 95.2% on highways and a processing time of 31 ms/frame, which meets the real-time requirements. Dorj et al. [30] presented a cutting edge curve lane detection system based on Kalman filter algorithm for autonomous cars. To estimate the parameters of a curved lane, parabola equation and circle equation models are applied inside the Kalman filter. The developed system has been tested with an autonomous driving vehicle. Simulation results show a high detection success rate in the curved lane. Suder et al. [7] proposed three systems to extract and detect road lane under environmental conditions: (1) horizontal road lane detection based on image segmentation in the HSV color space, (2) optimal path finding using the edge detection-based hyperbolic fitting line detection algorithm, and (3) road lane detection based on edge detection, Scharr mask, and Hough Transform algorithm. Embedded devices such as NVIDIA Jetson Nano and Raspberry Pi 4B were used to develop and test the proposed systems. Yoo and Kim [51] presented a robust method for extracting road marking features using a graph model based-approach. The hat filter with adaptive sizes is applied to extract the local maximum values of the filter response, which are introduced as nodes in a connected graph structure, and the graph edges are constructed using the proposed neighbor search method. The nodes related to the lane markings are then selected, and fitted to the line segments as the proposed features of the lane markings. The experimental results outperform existing methods on the KIST and Caltech datasets. Furthermore, the proposed method requires an average processing time of 3.3 ms, which is fast enough for real-time applications. The accuracy of feature-based lane detection was

significantly reduced when the lane was damaged or visibility was low, these methods are only applicable to real road scenes where the lane edges are clear and under simple road conditions [7,28,30,43,51–56].

Recently, deep learning-based techniques have been used to address the problem of road lane or traffic signs detection, and thus, they have boosted the development of Self-Driving Systems and Advanced Driver Assistance Systems (ADAS). Badrinarayanan et al. [57] presented a novel deep convolutional neural network architecture for segmentation called SegNet. Their architecture consists of an encoder network that is topologically identical to the 13 convolutional layers of the VGG16 network, a corresponding decoder network for mapping low-resolution feature maps from the encoder, and a per-pixel classification layer. The proposed architecture is compared with the well-known FCN, DeepLab-LargeFOV and DeconvNet architectures. SegNet architecture was mainly motivated by scene understanding applications. The experimental results show the good segmentation performance of the proposed network. The SegNet architecture is widely used in object detection [4,57,57,58]. Almeida et al. [4] proposed a new road representation by combining two simultaneous Deep Learning models, based on two adaptations of the ENet model. The results show that the combined solution is capable to cope with the failures or under-performance of each model and produces a more reliable route detection rate than that given by each approach individually. Zhao et al. [45] introduced a model based on deep reinforcement learning for surface lane detection, which is composed of two steps: the bounding box detector and the landmark point localizer. Specifically, a bounding box level convolution neural network is used to locate the road lane, then, a reinforcement-based Deep Q-Learning Localizer (DQLL) is applied to accurately localize the lanes as a group of landmark points for better representation of curved lanes. This proposed model achieves competitive performance in the NWPU Lanes and the TuSimple Lanes datasets. Heo et al. [5] designed a combination of lightweight deep learning models on an embedded GPU platform (eGPU) to identify car movement on the road. Their system analyzes discrete images and creates a continuous trace of the vehicle's movement trajectory. The evaluation results show that the proposed system can well extract horizontal and vertical movements of a vehicle. Model-based lane detection methods (based on deep learning) are suitable for situations where the lane is damaged or visibility is low. But, when the road traffic information is overly complicated or when there are interfering obstacles, the detection is considerably reduced, and false detections occur easily [1,3,4,6,28,31,45]. Using deep learning-based approaches, the accuracy and robustness of lane detection can be significantly improved. At the same time, these approaches have a higher hardware requirement and over-complex structures, which still leads to some limitations [12–14,59–62]. It is therefore necessary to continue to improve lane detection systems and to implement them on embedded platforms for faster execution time so as to meet the Embedded Intelligence (EI) constraints.

In this paper, we propose a solution that addresses the above-mentioned drawbacks of existing approaches. It is a deep-based lane detection system for intelligent vehicles under complex road conditions and dynamic environments. Basically, traffic road images are first pre-processed. Then, the proposed neural networks, CNN Encoder, CNN Decoder and LSTM, are applied to predict the road lane area or Region of Interest (ROI) in order to select the best one. Lastly, a post-processing step is done in which the radius of curvature and the center offset from the road are computed in order to detect any departure of the car from the lane. So as to meet the constraints of embedded intelligence, our selected system is implemented in an embedded NVIDIA Jetson Xavier NX device for fast execution time.

3. Development of lane prediction networks

The architecture of the lane prediction and detection system that we propose is illustrated in Fig. 1. It is a combination of three segmentation networks: a CNN Encoder–Decoder, a CNN Encoder–Decoder with Dropout layer, and a CNN Encoder–LSTM-Decoder. The networks are based on SegNet, a Deep Encoder–Decoder architecture that is composed of two CNN architectures with convolution layers, residual unit, maximum pooling layers, upper sampling layers and batch normalization layers.

Firstly, some preprocessing such as data resizing, shuffling, and normalization are done on the road lane. Secondly, the three-network system that we propose is applied to a single RGB channel to predict lanes and road markings. The functions of the three networks are as follows: the CNN Encoder–Decoder network is used to predict lane markings on the road, the CNN Encoder–Decoder network including Dropout layer is used for regularization and for uncertainty estimation, and the CNN Encoder–LSTM-Decoder network architecture uses the LSTM network to improve the detection rate through the suppression of the influence of false alarm patches on the detection results. Thirdly, the road lane markings are located by selecting the feature. Finally, a post-processing step that includes the following operations is applied: edge detection using the Canny detector, perspective transform to obtain a perpendicular view of the lane, and polynomial curve fitting to determine the quadratic function of the curve of the lane from which the radius of curvature and the offset from the center of the road are computed. The performance of the proposed networks has been measured according to the following metrics: loss, accuracy, precision, Recall, F1-score, and IoU.

3.1. Encoder-decoder network structure

CNN is a special type of deep learning algorithms and an effective technique for feature extraction, which has shown excellent results in many computer vision and signal processing applications such as medical image analysis, object detection, automatic speech recognition, classification and wind speed forecasting. The advantage of CNNs is that the feature extraction and classification processes are combined in a single CNN core, which allows CNNs to optimize feature extraction from the raw data during the training phase. The fundamental concept of a CNN consists in obtaining local features from higher-layers inputs and transferring them to lower-layers to obtain more complex features.

The CNN Encoder–Decoder network proposed in this paper is composed of two main modules: (1) an Encoder network and (2) a Decoder network. The latter is basically a convolutional neural network (CNN). A typical CNN architecture comprises convolutional layers, pooling layers, and fully connected layers.

3.1.1. Encoder network

To identify the pixel-level region which is abnormally similar to the road region, taking into account the importance of spatial information for the localization of the road region, we exploit the convolution layer to design the encoder network, so that the network can identify the similar region based on the shape, appearance, and spatial association between the road lane and non-road lane regions. Some Deep Learning architectures that use the convolution layer for image segmentation are proposed in the literature [13,57]. Based on these work, we have developed an Encoder–Decoder architecture. The encoder network structure is formed by a convolution layer, a residual unit and a pooling layer. In each layer, the input data is a 3-dimensional array with a size of $(80 \times 160 \times 3)$.

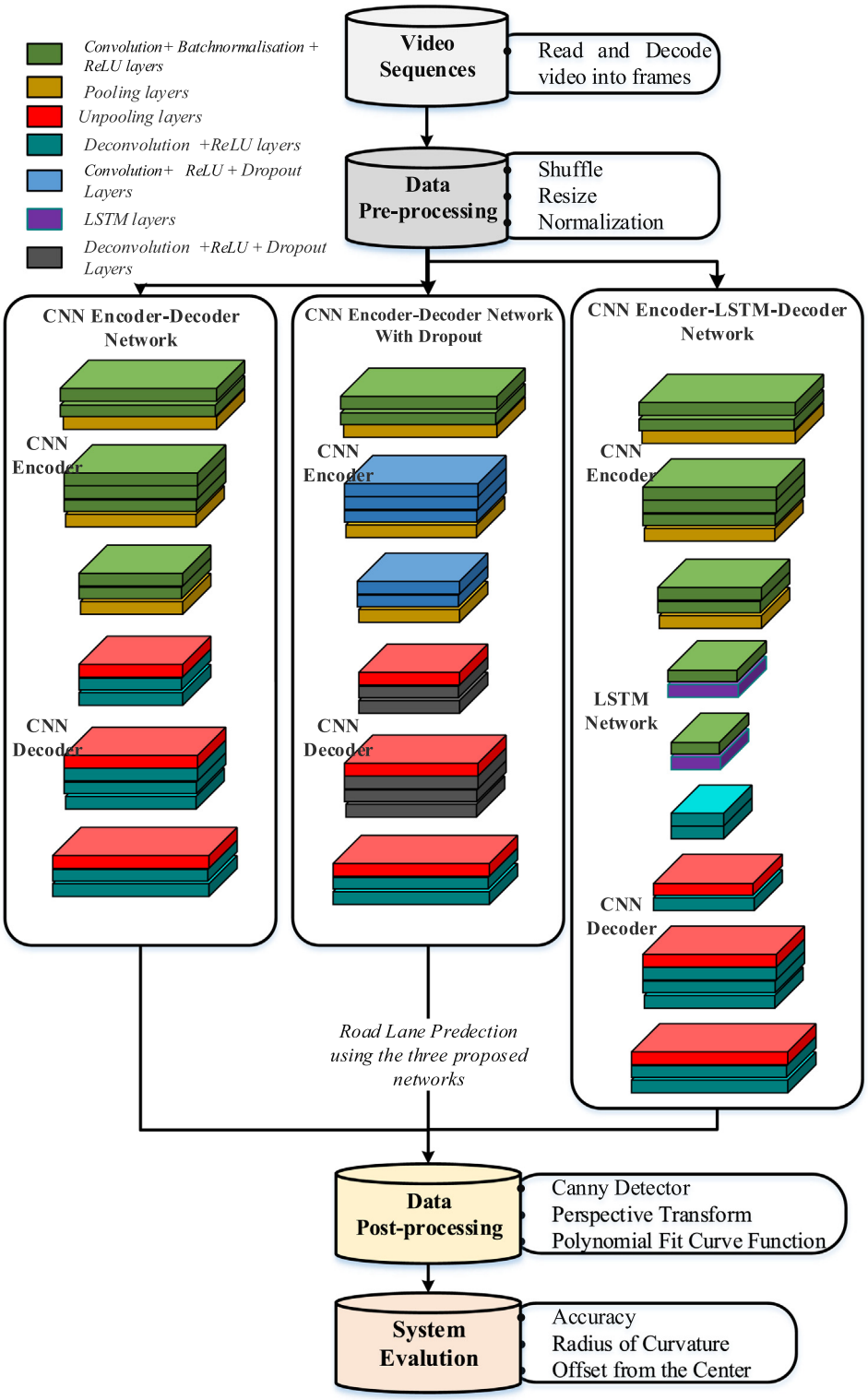

Fig. 1. The overall structure proposed for lane detection based on a three-network architecture.

The convolutional layer is a mathematical operation that applies a set of kernels in order to generate the feature maps. The operation of the convolutional layer between two time-series is a linear operation that is given by:

$$F(i,j) = (I * K)(i,j) = Z(\sum \sum I\,(i+m, j+n)\,K\,(m,n)) \quad (1)$$

where F represents the output of a feature map, I represents the input matrix, K represents a 2D filter of size $m \times n$ and Z is the activation function. The Rectified Linear Unit Layer (ReLU) is employed as activation function to increase the non-linearity of the feature maps. The output of this function is directly the input if it is positive and zero otherwise. This is expressed mathematically

as follows:

$$f(x) = max(0, x) \tag{2}$$

The convolution layer is composed of different self-learning filters with a kernel size of $(3 \times 3 \times d)$, where d is the filter depth. In each layer, these filters create feature maps corresponding to the local regions of the previous layer. A too high filter depth will lead to overfitting, and conversely, a too low filter depth will lead to underfitting. Thus, it is necessary to choose the appropriate filter depth value. The Encoder network proposed in this paper is composed of four layers with 8, 16, 32 and 64 convolution kernels, respectively.

The pooling layer reduces the size of the features and minimizes the number of parameters, Max pooling function is used to obtain the maximum value in the selected input region. This layer is added in the middle of multiple convolution layers. In this paper, each pooling has a pool-size of 2×2 and a stride size of 2.

The Dropout layer is widely used to regularize deep neural networks. It is generally applied to fully connected layers and convolutional layers. In this paper, the Dropout layer is applied to convolutional and pooling layers in the second proposed network. The regularization factor is:

$$C = C_0 + \frac{\lambda}{2} \sum_W W^2 \tag{3}$$

3.1.2. Decoder network

The decoder network architecture is composed of three layers of the CNN network. Each layer is composed of an upper sampling layer and two or three deconvolution or transpose of convolution layers. The upper sampling layer carries out an upper sampling operation on the feature maps trained from the previous convolution layer. The deconvolution layer employs a range of multi-channel filters to convolve with the sparse representation of the upper sampling heat map, and then applies the batch normalization layer to generate a more dense heat map. On the overall, 64, 32 and 16 transpose of convolution kernels of size (3×3) are applied in the first, second and third layers of our proposed decoder network architecture. Finally, the network produces a binary mask that represents the road lane region in the image.

3.2. LSTM network structure

Long Short-Term Memory (LSTM) network is an improvement of the RNNs proposed by Hochreiter and Schmidhuber. The name LSTM means that it can flexibly overcome long or short time lags for the matching tasks [3,12–14,51,59–61]. RNNs have a major drawback known as the gradient vanishing problem, i.e., they have difficulty to learn long-range dependencies. Fortunately, this problem has been solved with the creation of the LSTM network. Indeed, LSTM network is capable of learning long-term dependencies thanks to their structure that has the ability to remember information during long delays, forget unnecessary information and carefully expose information at each time step. The internal structure of RNN–LSTM network for processing sequences is shown in Fig. 2.

It consists of a memory unit called cell and four gates: a forget gate Z^f, an input gate Z^i, an update gate Z^u, and an output gate Z^o. Each gate contains a Fully Connected (FC) layer and an activation function σ. Using these gates, the network can handle the process of adding or removing information from its cells.

The cell state gate remembers the information over time, the forget gate controls the extent of the value kept in the cell, the input gate controls the extent of the value flow in the cell, and the output gate controls the extent of the value in the cell to be used for computing the output. X_t refers to the current input; C_t and C_{t-1} denote the new and previous cell states, respectively; and h_t and h_{t-1} are the current and previous outputs, respectively. The following expressions represent the mathematical formulas that are applied in a RNN–LSTM network:

$$Z^f = \sigma(W_f[X_t, h_{t-1}]) \tag{4}$$

$$Z^i = \sigma(W_i[X_t, h_{t-1}]) \tag{5}$$

$$Z^u = tanh(W_u[X_t, h_{t-1}]) \tag{6}$$

$$Z^o = \sigma(W_o[X_t, h_{t-1}]) \tag{7}$$

$$C_t = Z^f * C_{t-1} + Z^i * Z^u \tag{8}$$

$$h^t = Z^o * \tanh(C_t) \tag{9}$$

In more details, the forget gate is responsible for removing redundant information from the cell state. In this gate and according to Eq. (4) the current cell's input X_t and the previous cell's hidden state h_{t-1} are multiplied by a weight matrix W_f. Next, the output result is processed by a sigmoid function, which produces a vector of values between 1 and 0. A value close to 1 means that the information must be retained, and inversely, a value close to 0 means that the information must be forgotten.

The input gate receives the same two arguments X_t and h_{t-1}. This gate ensures the addition of new information to the cell state. In this gate, X_t and h_{t-1} are multiplied by a new weight matrix W_i, and after that, a new sigmoid function is used to decide which values must be added to the cell state (Eq. (5)). At the update gate, X_t and h_{t-1} are multiplied by another weight matrix, and the result passed through the *tanh* function which gives values between -1 and 1 (Eq. (6)). The results of the forget, input and update gates are used to compute the cell state denoted by C_t (Eq. (8)), and which comprises all the new candidate values that could be added to the cell state. Finally, the output gate of the LSTM decides which states are required for continuation by the two inputs arguments X_t and h_{t-1} according to (Eq. (7)) and (Eq. (9)).

Lastly, it is necessary to compute the time complexity of the CNN Encoder–LSTM-Decoder network. For this purpose, we first calculate the time complexity of the convolutional layers of the CNN Encoder–Decoder network, then that of the LSTM layer and add up both time complexities to obtain the time complexity of our CNN Encoder–LSTM-Decoder network [63]. The time complexity of all convolutional layers is estimated to be $O(\sum_{l=1}^{d} n_{l-1}.s_l^2.n_l.m_l^2)$ where n_{l-1} is the number of input channels of the l_{th} layers, s_l is the spatial size of the filter, n_l is the number of filters in the l_{th} layers, m_l is the spatial size of the output feature map and d is the number of convolutional layers. Regarding the LSTM, the time complexity per weight of an LSTM network is $O(1)$, since this network is local in space and time, which means that the length of the input does not affect the storage requirements of the network for each time step [63]. Therefore, the overall complexity of an LSTM network per time step is equal to $O(w)$, where w is the number of weights. Knowing the time complexities of the two networks, the complexity of the CNN Encoder–LSTM-Decoder network per time step can be computed as the sum of the complexity of the two networks. It is equal to $O\left(\sum_{l=1}^{d}(n_{l-1}.s_l^2.n_l.m_l^2) + w\right)$ and the complexity of the whole training process is equal to $O\left(\left(\sum_{l=1}^{d}(n_{l-1}.s_l^2.n_l.m_l^2) + w\right) i.e\right)$ where i is the input length and e the number of epochs.

3.3. Proposed networks models structure

The three proposed networks: CNN Encoder–Decoder network, CNN Encoder–Decoder network architecture including Dropout

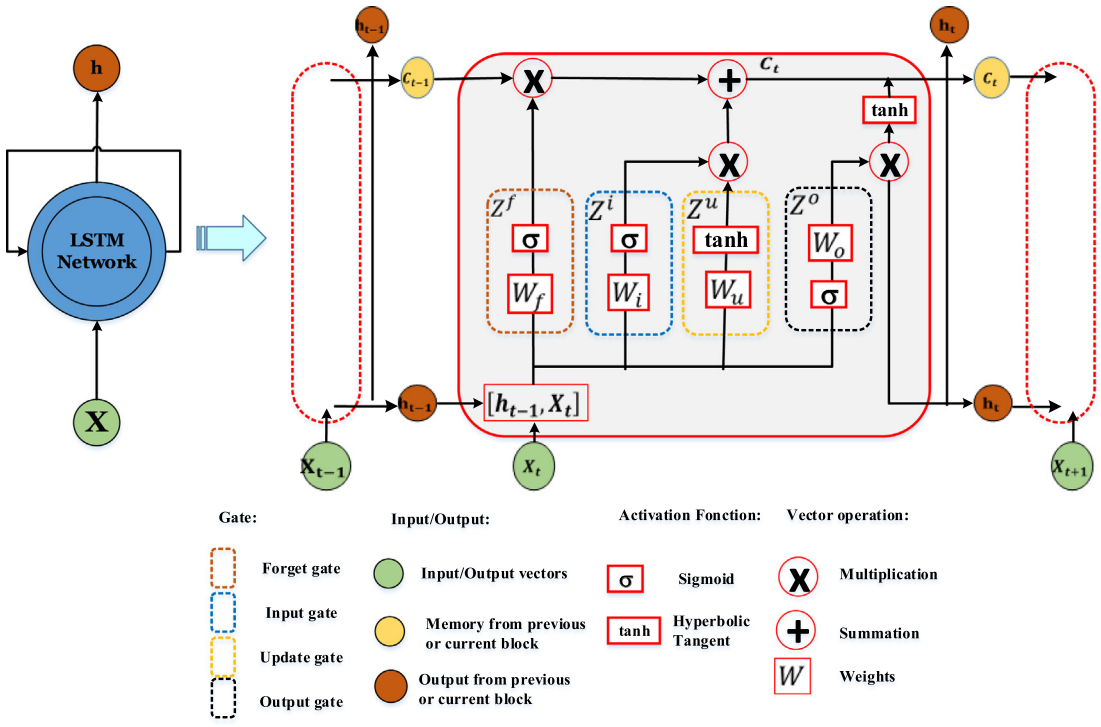

Fig. 2. The internal structure of RNN–LSTM network.

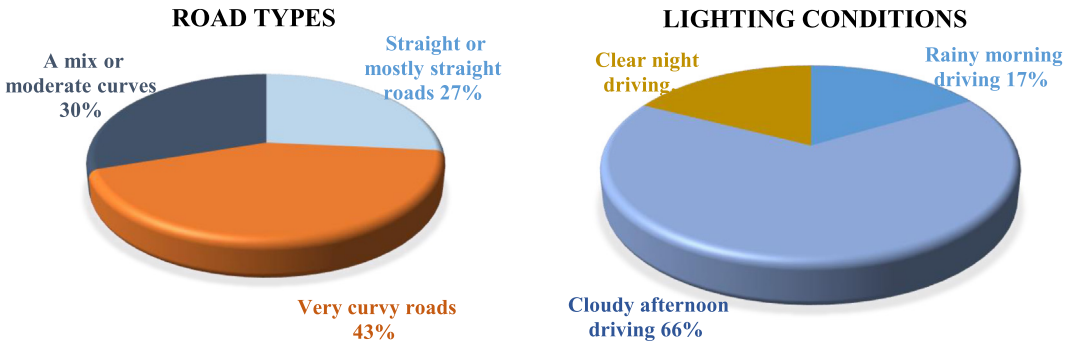

Fig. 3. General information of data collection used for training and testing models.

layer, and the CNN Encoder–LSTM-Decoder network, were evaluated on a public dataset that includes 12764 road images under different conditions such as day, night, low light, weather, traffic, curves and straights road (see Fig. 3) (https://www.dropbox.com/s/rrh8lrdclzlnxzv/full_CNN_train.p?dl=0 and, https://www.dropbox.com/s/ak850zqqfy6ily0/full_CNN_labels.p?dl=0).

The selected database also contains difficult areas such as intersections and construction, very curvy roads, and curved and straight roads. In this paper, we used 12764 (80 × 160 × 3 dimensions) road images to train our proposed networks because of blurring, hidden lines, etc.

There are a number of pre-processing steps that should be performed before using these data in the proposed models such as shuffling and normalization. The purpose of the normalization step is to have the same range of feature values and the purpose of the shuffling step is to reduce the variance and ensure that the models remain general and fit less. Subsequently, we divided the pre-processed dataset into a training set and a test set, and we trained the three proposed network architectures: the CNN Encoder–Decoder network, the CNN Encoder–Decoder network architecture with Dropout layers after convolution pooling layers, and the CNN Encoder–LSTM-Decoder network architecture using the training data (Fig. 4).

The Encoder Neural Network generates low-resolution feature maps of the input data. It is composed of a series of convolution layers followed by a number of max-pooling layers and Dropout layers in the case of the second network. On the other hand, the Decoder Neural Network provides pixel-wise segmentation from the feature maps, it is composed of a series of deconvolution or transpose of convolution layers followed by a number of up-sampling and Dropouts layers in the case of the second network. Concerning the LSTM network architecture, it reduces the false alarm rate through the suppression of the influence of false alarm patches on the detection results. It is composed of two LSTM layers and convolution layer before each layer to maintain matrix dimensions. The Adam Optimizer and Mean Squared operations are used to check the performance of the proposed networks in terms of accuracy and loss. The whole process of the proposed networks for road lane prediction is represented in Fig. 4. A summary of the proposed networks is shown in Tables 1 and 2. For more details see https://github.com/yassinkortl/Deep-Embedded-Hybrid-CNN-LSTM-Network-for-Lane-Detection-on-NVIDIA-Jetson-Xavier-NX.

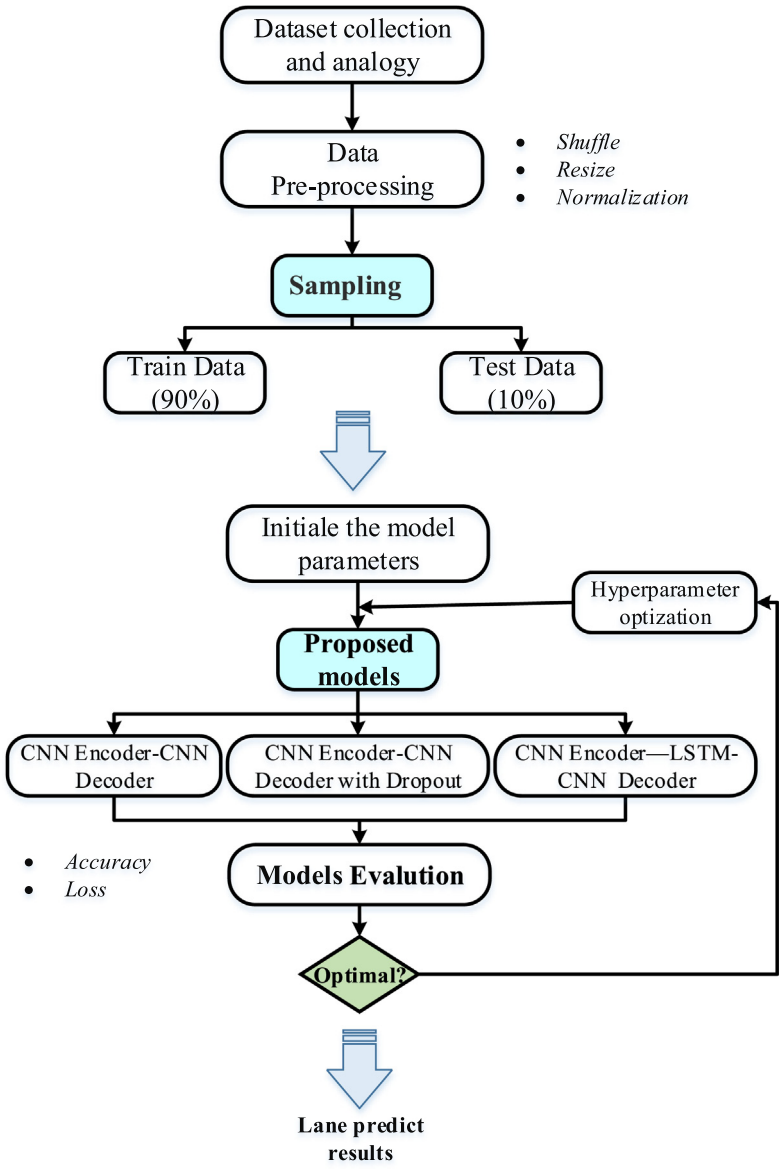

Fig. 4. An illustration of the proposed networks architecture to predict road lane.

The CNN Encoder–Decoder network is composed of 20 layers: 14 convolutional layers (7 layers for the Encoder Network and 7 Layers for the decoder) and 6 pooling layers. The convolution layer is used to extract features with kernels of size 3×3 and activated with the ReLU function. For each convolution block, there are 2 or 3 2D convolution layers and a pooling layer. The max-pooling layer is used to reduce the dimensions with kernels of size 2×2. The CNN Encoder–Decoder with Dropout layer network is composed of 30 layers; 14 convolutional layers (7 layers for the Encoder Network and 7 Layers for the decoder), 6 pooling layers, and 10 Dropout layers characterized by a 20% dropout rate. However, the CNN Encoder–LSTM-Decoder network is composed of 24 layers: 16 convolutional layers (7 layers for the Encoder Network, 7 layers for the decoder network and 2 for the LSTM network), 6 pooling layers and two LSTM layers. After applying the proposed networks, the output shape of the lane is found to be of dimension (80x 160x1) with a single RGB channel.

4. Experimental setup

We propose to independently evaluate the performance. Firstly, we evaluate the proposed networks: the CNN Encoder–Decoder network, the CNN Encoder–Decoder network with Dropout layer network and the CNN Encoder–LSTM-Decoder network, so as to select the best one for predicting road lane. Secondly, we evaluate the proposed system for lane detection using the selected network. Finally, we include a post-processing phase that will enable us to realize a Lane Departure Warning System (LDWS).

4.1. Performance evaluation metrics

For the evaluation, the following metrics are applied to compute the performance of the proposed networks. In this paper, we need two cases of class prediction i.e., true prediction or false prediction. Each network's performance is evaluated using the following scores: Loss, average Accuracy, Precision, Recall,

Table 1
The full summary of CNN Encoder–Decoder and CNN Encoder–Decoder with Dropout layer networks.

Networks	Layers	Kernel Size	Stride	Filter depth	Activation	Input Size	Output Size
Encoder Network	*Conv2D*	3 × 3	1	8	ReLU	***80*** × ***160*** × ***3***	78 × 158 × 8
	Conv2D	3 × 3	1	16	ReLU	78 × 158 × 8	76 × 156 × 16
	MaxPool2D	2 × 2	2	–	–	76 × 156 × 16	38 × 78 × 16
	*Conv2D⊕**Dropout***	3 × 3	1	16	ReLU	38 × 78 × 16	36 × 76 × 16
	*Conv2D⊕**Dropout***	3 × 3	1	32	ReLU	36 × 76 × 16	34 × 74 × 32
	*Conv2D⊕**Dropout***	3 × 3	1	32	ReLU	34 × 74 × 32	32 × 72 × 32
	MaxPool2D	2 × 2	2	–	–	32 × 72 × 32	16 × 36 × 32
	*Conv2D⊕**Dropout***	3 × 3	1	64	ReLU	16 × 36 × 32	14 × 34 × 64
	*Conv2D⊕**Dropout***	3 × 3	1	64	ReLU	14 × 34 × 64	12 × 32 × 64
	MaxPool2D	2 × 2	2	–	–	12 × 32 × 64	6 × 16 × 64
Decoder Network	*UpSampling2D*	2 × 2	2	64	–	6 × 16 × 64	12 × 32 × 64
	*Conv2DTranspose⊕**Dropout***	3 × 3	1	64	ReLU	12 × 32 × 64	14 × 34 × 64
	*Conv2DTranspose⊕**Dropout***	3 × 3	1	64	ReLU	5 × 15 × 128	16 × 36*64
	UpSampling2D	2 × 2	2	64	–	16 × 36 × 64	32 × 72 × 64
	*Conv2DTranspose⊕**Dropout***	3 × 3	1	32	ReLU	32 × 72 × 64	34 × 74 × 32
	*Conv2DTranspose⊕**Dropout***	3 × 3	1	32	ReLU	34 × 74 × 32	36 × 76 × 32
	*Conv2DTranspose⊕**Dropout***	3 × 3	1	16	ReLU	36 × 76 × 32	36 × 76 × 16
	UpSampling2D	2 × 2	2	16	–	36 × 76 × 16	76 × 156 × 16
	Conv2DTranspose	3 × 3	1	16	ReLU	76 × 156 × 16	78 × 158 × 16
	Conv2DTranspose	3 × 3	1	1	ReLU	78 × 158 × 16	***80*** × ***160*** × ***1***

Table 2
The full summary of CNN Encoder–LSTM-Decoder network.

Networks	Type	Kernel Size	Stride	Filter depth	Activation	Input Size	Output Size
Encoder Network	Conv2D	3 × 3	1	8	ReLU	***80*** × ***160*** × ***3***	78 × 158 × 8
	Conv2D	3 × 3	1	16	ReLU	78 × 158 × 8	76 × 156 × 16
	MaxPool2D	2 × 2	2	–	–	76 × 156 × 16	38 × 78 × 16
	Conv2D	3 × 3	1	16	ReLU	38 × 78 × 16	36 × 76 × 16
	Conv2D	3 × 3	1	32	ReLU	36 × 76 × 16	34 × 74 × 32
	Conv2D	3 × 3	1	32	ReLU	34 × 74 × 32	32 × 72 × 32
	MaxPool2D	2 × 2	2	–	–	32 × 72 × 32	16 × 36 × 32
	Conv2D	3 × 3	1	64	ReLU	16 × 36 × 32	14 × 34 × 64
	MaxPool2D	2 × 2	2	–	–	14 × 34 × 64	7 × 17 × 64
LSTM network	Conv2D	3 × 3	1	64	ReLU	7 × 17 × 64	5 × 15 × 64
	LSTM	1×1	1	64	Tanh	5 × 15 × 64	5 × 15 × 64
	Conv2D	3 × 3	1	128	ReLU	5 × 15 × 64	3 × 13 × 128
	LSTM	1×1	1	128	Tanh	3 × 13 × 128	3 × 13 × 128
Decoder Network	Conv2DTranspose	3 × 3	1	128	ReLU	3 × 13 × 128	5 × 15 × 128
	Conv2DTranspose	3 × 3	1	64	ReLU	5 × 15 × 128	7 × 17 × 64
	UpSampling2D	2 × 2	2	–	–	7 × 17 × 64	14 × 34 × 64
	Conv2DTranspose	3 × 3	1	64	ReLU	14 × 34 × 64	16 × 36 × 64
	UpSampling2D	2 × 2	2	–	–	16 × 36 × 64	32 × 72 × 64
	Conv2DTranspose	3 × 3	1	32	ReLU	32 × 72 × 64	34 × 74 × 32
	Conv2DTranspose	3 × 3	1	32	ReLU	34 × 74 × 32	36 × 76 × 16
	Conv2DTranspose	3 × 3	1	16	ReLU	36 × 76 × 16	38 × 78 × 16
	UpSampling2D	2 × 2	2	–	–	38 × 78 × 16	76 × 156 × 16
	Conv2DTranspose	3 × 3	1	16	ReLU	78 × 158 × 16	78 × 158 × 16
	Conv2DTranspose	3 × 3	1	1	ReLU	78 × 158 × 16	***80*** × ***160*** × ***1***

Intersection over Union (IoU) and F1-score. These were determined in terms of True Positives (TP: the truth is positive and the prediction is positive), True Negatives (TN: the truth is negative and the prediction is negative), False Positives (FP: the truth is negative and the prediction is positive), and False Negatives (FN: the truth is positive and the prediction is negative). The IoU metric is used to measure the overlap between the predicted and the ground-truth bounding boxes of the road lane. Accordingly, we have established such measures to evaluate the performance of the proposed models using the following metrics:

$$\text{Accuracy} = (TP + TN)/(TP + FN + TN + FP) * 100 \quad (10)$$

$$\text{Recall} = \text{TPR} = TP/(TP + FN) * 100 \quad (11)$$

$$\text{Precision} = \text{TP}/(TP + FP) * 100 \quad (12)$$

$$\text{F1} - \text{score} = (2 * \text{TP})/(2 * \text{TP} + \text{FP} + \text{FN}) * 100 \quad (13)$$

$$IoU = TP/(TP + FP + FN) * 100 \quad (14)$$

4.2. Training and validation of the proposed networks

4.2.1. Training phase

For a better prediction evaluation, the performance of the three proposed networks is compared. Firstly, the road lane images were first processed through a pre-processing step, which includes data resizing, shuffling, and normalization. Secondly, we divided the pre-processed dataset into a training set (90%) and a test set (10%). Thirdly, we trained our proposed three-network architecture to predict road lane using the training data. Network I integrates the CNN Encoder–Decoder network (Table 1). Network II integrates the CNN Encoder–Decoder network with Dropout layers after convolution pooling layers (Table 1). Network III integrates the CNN Encoder–LSTM-Decoder network architecture, where LSTM network is employed to improve the detection rate through the suppression of the influence of false alarm patches on the detection results (Table 2).

The proposed networks performance was measured according to the following metrics: cross-entropy (loss), accuracy, precision, Recall, F1-score and IoU metrics. In the loss function, the

Table 3
Performance of the proposed networks with accuracy, loss, precision, Recall, F1-score and IoU metrics in the training phase.

Network	Loss	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	IoU (%)	Parameters (m)
CNN Encoder–Decoder	0.0025	96.30	96.98	97.03	97.01	93.96	0.181
CNN Encoder–Decoder + Dropout	0.0045	96.16	96.22	96.12	96.17	92.41	0.181
CNN Encoder–LSTM-Decoder	0.0015	96.38	97.49	97.59	97.54	94.98	16.96

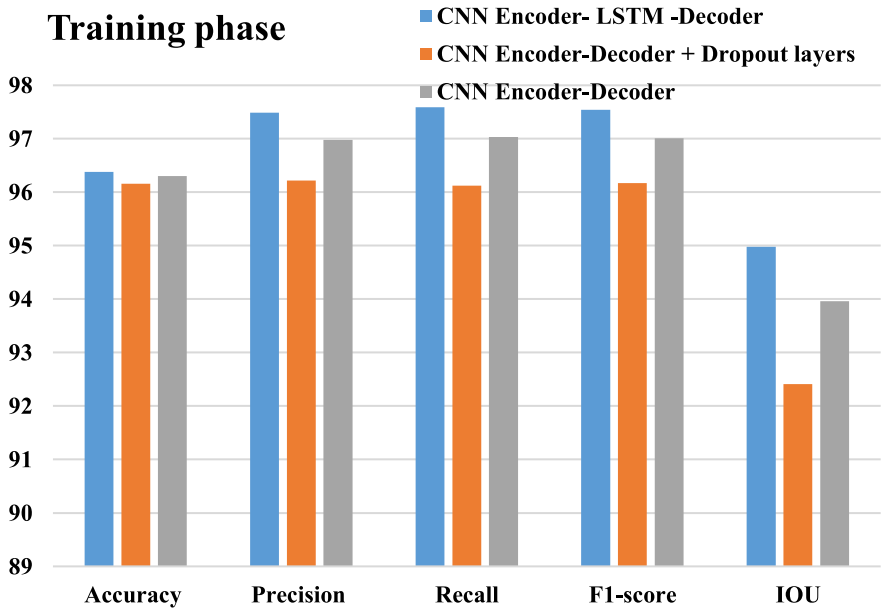

Fig. 5. Visualization of the performance of the proposed networks results in the Training phase.

balancing parameter λ is set to 1 and the Adam algorithm is used to optimize this function. The proposed networks are trained for 30 epochs with the learning rate set to 0.0001. A summary of the loss, overall accuracy, precision, Recall, F1-score, and IoU metrics for each network architecture is presented in Table 3 and illustrated graphically in Fig. 5.

In epoch 30, for the CNN Encoder–Decoder network architecture, the training performance is 96.30% of accuracy, 96.98% of precision, 97.03% of Recall, 97.01% of F1-score, and 93.96 of IoU. Similarly, the obtained training performance is 96.16% of accuracy, 96.22% of precision, 96.12% of Recall, 96.17% of F1-score, and 92.41 of IoU for the CNN Encoder–Decoder network combined with Dropout layers. Furthermore, the obtained training performance is 96.38% of accuracy, 97.49% of precision, 97.59% of Recall, 97.54% of F1-score, and 94.98 of IoU for the CNN Encoder–LSTM-Decoder network architecture. Regarding the loss value, the lowest loss value (0.0015) was found in the third network compared to 0.0045 and 0.0025 in the second and the first networks, respectively. It was observed that the CNN Encoder–LSTM-Decoder network architecture achieves better performance for the training phase than the CNN Encoder–Decoder network architecture and CNN Encoder–Decoder network architecture combined with Dropout layers.

4.2.2. Testing phase

To examine the effectiveness of each proposed network, a set of 1276 road lane images (10%) is used for validation. Table 4 presents the performance results of the proposed networks with accuracy, loss, precision, Recall, F1-score and IoU metrics in the training phase and testing phase for road lane detection on the validation set. In addition, Fig. 6 illustrates graphically the performance metric in the testing phase.

It can be observed that CNN Encoder–LSTM-Decoder network also achieves better performance for test phase compared to both of the other two proposed networks. Specifically, it reaches an accuracy of 96.36%, a precision of 97.30%, a Recall of 97.54%, a F1-score of 97.42%, and an IoU of 94.70%. These results show the effectiveness of the CNN Encoder–LSTM-Decoder network in solving the problem of road lane detection in different environments. The results of the evolution of the accuracy rate and the loss function over time in the training phase and the test phase of the proposed networks are presented in Fig. 7, Fig. 8, and Fig. 9. The performance was done with 30 epochs (an epoch is a complete iteration on the training or testing). We can see from the graphs how the network improves its accuracy on the training and testing phases, while decreasing the cost (loss). One of the main goals of machine learning is that the models created are capable of generalizing their knowledge. In other words, the model must be able to make predictions with acceptable accuracy on data that it has never processed before. It is therefore a good question to consider if a model that is well fitted to the training data will perform well with the evaluation data, since the evaluation data are unknown to the model. A good fit is identified by a training and validation loss that decreases to a stability point with little difference between the two final loss values. The model loss will almost always be smaller on the training phase than on the validation phase (see Table 4).

It implies that we should expect some gap between the training and validation loss learning curves. This gap is called the "generalization gap". A curve plot of learning curves shows a good fit if the training loss plot decreases to a stability point and the validation loss curve decreases to a stability point with little difference from the training loss. We can clearly see that the first and the second networks suffer from overfitting (Fig. 7 and Fig. 8). A good fit is the goal of the learning network. Fig. 9 shows the a case of a good fit learning curves achieved using CNN Encoder–LSTM-Decoder network. The results obtained during the training

Table 4
Performance results of the proposed networks in Training and Testing for road lane detection.

Network		Loss	Accuracy	Precision	Recall	F1-score	IoU
CNN Encoder–Decoder	*Training*	0.0025	96.30	96.89	97.03	97.01	93.96
	Testing	0.0031	96.27	97.37	96.17	96.76	93.94
CNN Encoder–Decoder + Dropout	*Training*	0.0045	96.16	96.22	96.12	96.17	92.41
	Testing	0.0041	96.17	96.56	96.06	96.31	92.64
CNN Encoder–LSTM-Decoder	*Training*	0.0015	96.38	97.49	97.59	97.54	94.98
	Testing	0,0017	96.36	97.30	97.54	97.42	94.70

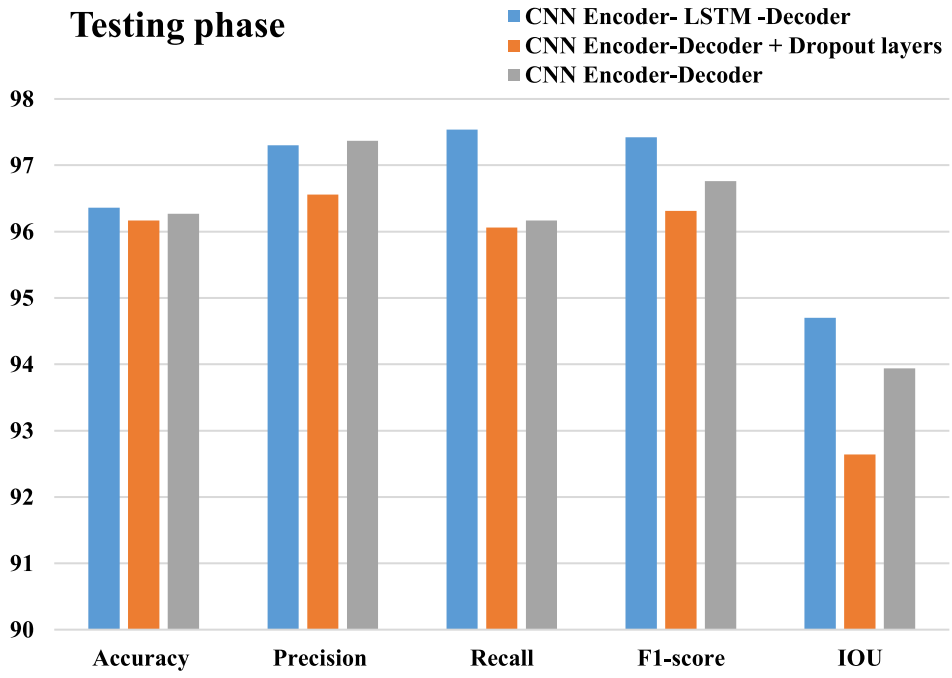

Fig. 6. Visualization of the performance of the proposed networks results in the Testing phase.

phase and the test phase clearly prove the effectiveness of the proposed CNN Encoder–LSTM-Decoder network compared to the other two proposed networks for lane prediction.

In terms of complexity, the proposed CNN Encoder–LSTM-Decoder network requires more parameters than the other models because it has a greater number of layers (See Table 3). Moreover, the CNN Encoder–LSTM-Decoder network requires a slightly longer training and inference time than the other networks. However, it gives better segmentation results. On the overall, the structure of the CNN Encoder–LSTM-Decoder clearly improves the performance of the deep models.

4.3. ADAS application: Deep lane departure warning system (LDWS)

Road lane detection is a common task for all human drivers. It consists in ensuring that their vehicle stays inside its lane, in order to guarantee the smooth flow of traffic and minimize the risk of collision with other cars in neighboring lanes. Similarly, this task is essential for the development of autonomous vehicles. It was found that it is possible to detect lane markings on roads using vision and deep learning techniques [1,3,4,6,7,24–26,46,47]. The purpose of this paper is to design, implement and evaluate lane departure warning system for autonomous driving, using the selected CNN Encoder–LSTM-Decoder network. The selected network combined with pre-processing and pro-processing steps will be used to identify and draw the inside of a road lane, as well as to calculate the curvature of the road lane and also to evaluate the vehicle's position in regards to the center of the road lane. To detect and draw a polygon that takes the shape of the lane in which the car is currently located, the following steps are built into a pipeline (see Fig. 1).

Road lane images were first processed through a pre-processing step, which includes data resizing and normalization. After that, we applied our proposed CNN Encoder–LSTM-Decoder network to predict lanes and road markings with a single RGB channel. Finally, we applied a post-processing step that includes an edge detection operation using the Canny detector to get better lane lines [49,51], a perspective transform to obtain a bird's eye view of the lane, and sliding windows to find hot lane lines pixels [51, 64]. Moreover, we also perform polynomial curve fitting so as to identify the left and right lines and to find out the quadratic function of the lane curve that will allows us to compute the radius of curvature and the offset from the center of the road [30]. Fig. 10 gives an overview of the result obtained at each step.

Fig. 11 presents some examples of detection results obtained by the proposed system based on the proposed CNN Encoder–LSTM-Decoder network. The proposed technique was able to detect and accurately locate road lane markings despite the presence of noise on the road in different lighting conditions. Datasets images were captured with a resolution of 640 $\times$500, 640 $\times$ 480, 640$\times$ 360, and 1280 $\times$ 800 pixels. The experiments were carried out on an Intel Core i7-2630QM CPU, 8 GB RAM and Windows 7 64-bit operating system, using the Python language.

Finally, the LDW driver assistance system should achieve very low false alarm rates. We also supposed that the width between the left and right lanes is 3.7 m, which is the case in the U.S., to estimate the distance across the full scale, and that the camera is mounted approximately in the center of the front window. We therefore can simply take the position of the left and right lanes

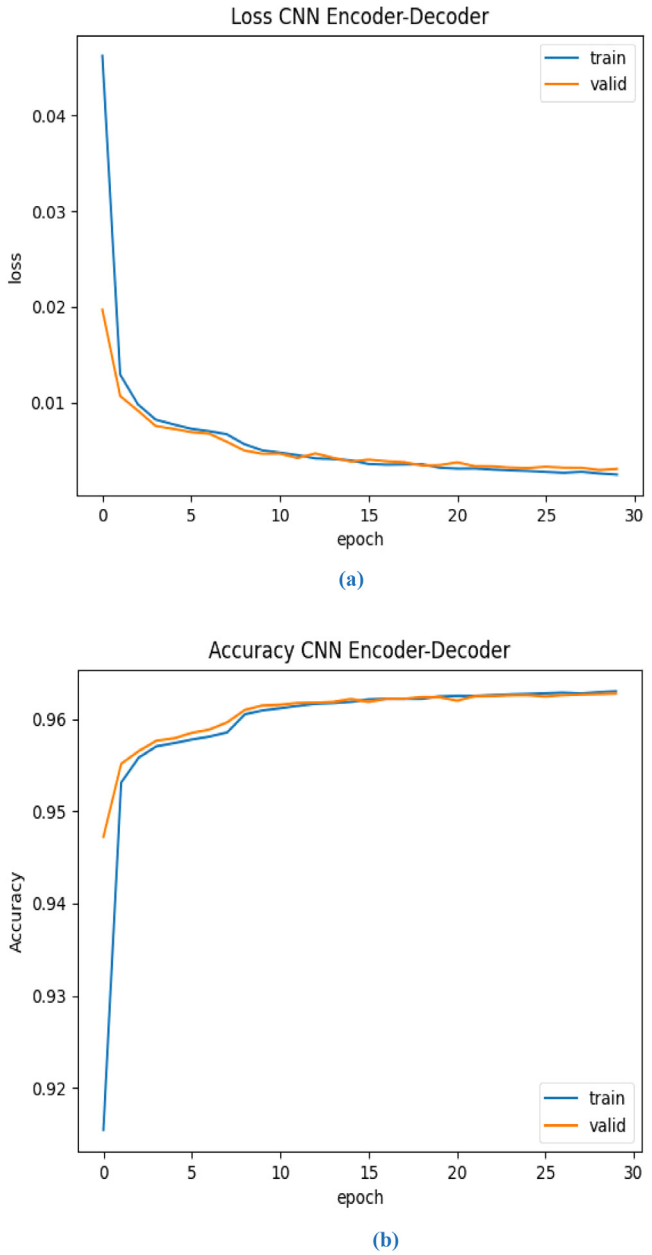

Fig. 7. Evaluation metrics of lane road prediction using CNN Encoder–Decoder network (a) Accuracy (b) Loss.

at the bottom of the image, and compare it with the middle of the image to determine the vehicle's position relative to the center. If the vehicle drives too close to the edges of the lane, an alert is sent to the driver. Fig. 12 displays numerical estimates of the vehicle's position from the center.

4.4. GPU implementation of the proposed deep LDWS on NVIDIA Jetson Xavier NX

Today, many researchers have proposed the use of parallel processors. An attractive solution that uses multi-processors for processing graphics is GPUs, which are used for high performance computing and can be considered as multiple cores with a software layer that enables parallel computing. In contrast to the CPU, the state-of-the-art of the GPU demonstrates improving performance in terms of execution time. The GPU family used

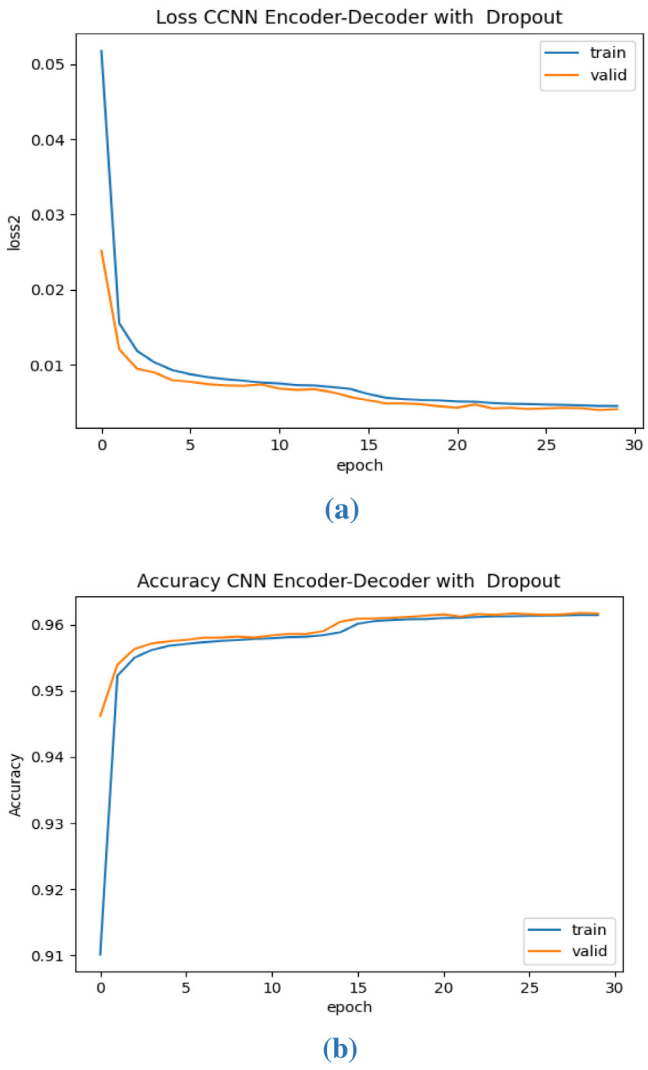

Fig. 8. Evaluation metrics of lane road prediction using CNN Encoder–Decoder network architecture combined with Dropout layers (a) Accuracy (b) Loss.

in this paper is the NVIDIA Jetson Xavier NX (Fig. 13) due to its high adoption in modern computing systems. Given that the system proposed in this paper is a computationally intensive one, graphics processors have been used to take advantage of their performance and efficiency to realize some embedded systems.

NVIDIA Jetson Xavier NX, stands out for being a device that offers great computational performances with very low power consumption. As we can find in the NVIDIA's website: Jetson Xavier NX configuration is based on a 64-Bit hexacore CPU and an NVIDIA Volta GPU with 384 CUDA cores, 48 Tensor Cores and 2 NVDLA (NVIDIA Deep Learning Accelerator), combined with 8 GB of LPDDR4x RAM, with Gigabit Ethernet connectivity and all running on the Ubuntu system. The CPU is based on a 6-core NVIDIA Carmel ARM® v8.2 64-bit CPU 6 MB L2 + 4 MB L3. It weighs only 85 grams and consumes 10 W under normal condition.

This device is perfect for deploying AI and implementing deep learning models. It was decided to use this device since NVIDIA Jetson Xavier NX is designed to accelerate the most DL models in real-time environments. In this paper, we exploit the benefits offered by this platform to activate the possibility of embedding the proposed Deep LDWS. The experiments were performed by setting the NVIDIA Jetson Xavier NX platform to maximum

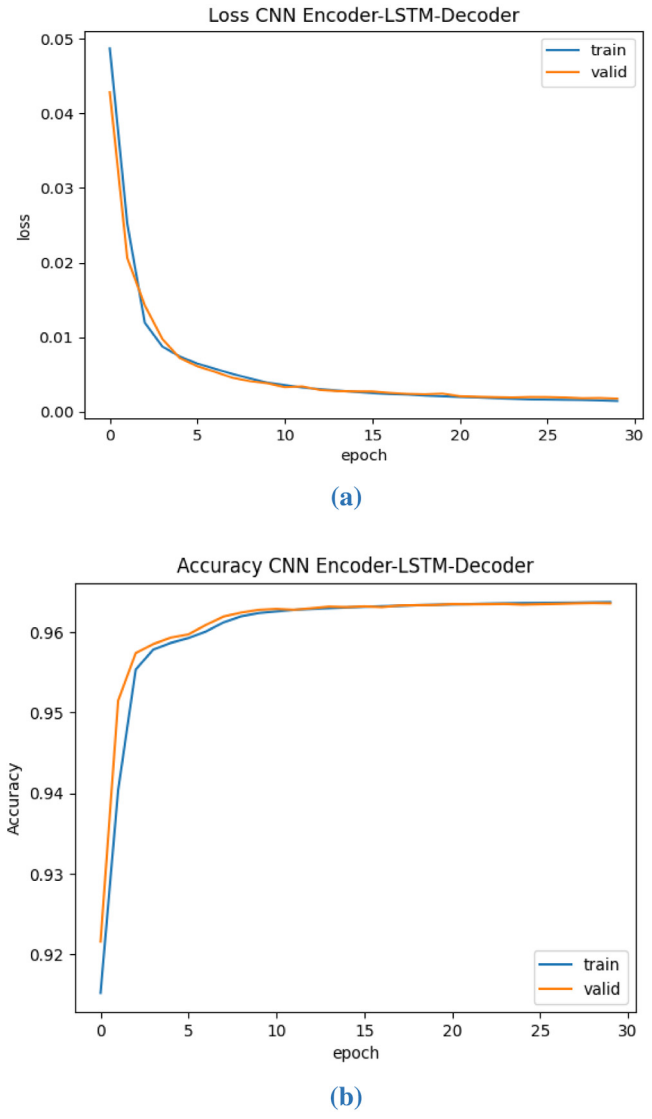

Fig. 9. Evaluation metrics of lane road prediction using CNN Encoder–LSTM-Decoder network architecture (a) Accuracy (b) Loss.

performance mode (MAXN0), i.e., all CPU and GPU cores were activated at full speed. Table 5 shows the performance results of the proposed Deep LDWS implemented on a NVIDIA Jetson Xavier NX and on a PC with a CPU processor Intel® Core™ i7-2630QM 2.00 GHz.

After integrating the proposed Deep LDWS in an embedded platform, the performance on the NVIDIA Jetson Xavier NX has improved significantly. As we know, an ML/DL algorithm for an automatic vision system must meet both the precision and real-time response requirements. For this reason, the proposed Deep LDWS was designed to achieve more than 7.55 fps on the NVIDIA Jetson Xavier NX platform with an input image size of 640 × 360, making it suitable for real-time road scene embedded applications.

5. Discussions

After extensive results analysis, it is found that a combination of CNN Encoder–Decoder network and LSTM network shows significant impact on road lane detection based on automatic feature extraction from road images. Our selected network was able to predict road lane with high accuracy. Table 6 shows a comparison, in terms of accuracy, Recall, F1-score, and processing time, between existing systems and our proposed system. Table 6 shows some moderately high accuracy of 96.2% and a high Recall of 95.2% based on ENet [65], an accuracy of 94.1% and a recall of 80% based on ARFNet [66], an accuracy of 95.2% based on RANSAC algorithm combined with fuzzy controller [3], an accuracy of 94% of the ResNet101 model [1], also, an accuracy of 95.97%, a recall of 87.52% and a F1-score of 91.55% based on U-net model [67] and an accuracy of 95.4% based on LMD-11 network [68]. The slightly lower accuracy between 89% and 93.89% are obtained in [4,45,64,69,70]. A global accuracy of 97.2% was achieved by the system developed in [65], based on SegNet network. The model used is tested on three different datasets such as CamVid and Cityscapes of road scenes, and the SUN RGB-D dataset of indoor scenes. The model is used for the semantic segmentation of the scene, not for road lane prediction. Furthermore, a comparison of existing systems in terms of processing time showed that the developed system in [65] needed 262ms to detect traffic lanes based on the ENet network on a NVIDIA TX1 and 289ms based on the SegNet network on a NVIDIA Titan X with image resolution of 1280×720. The required processing time based on the LMD-11 network is 2470ms in [68] and 113.9ms in [71] based on SCNN network. Moreover, processing times of 21.54 ms, 191ms, and 54 ms were obtained in [64], [72] and [3], respectively, using vision techniques.

The vision techniques do not require significant execution time compared to those based on deep learning, but with low detection accuracy and limited conditions. Our Embedded Deep LDWS system based on the CNN Encoder–LSTM-Decoder network demonstrates a high performance of 96.36% of accuracy, 97.54% of Recall and 97.42% of F1 score, which is comparatively better than other existing systems. In terms of time complexity, our system requires 132 ms of processing time for lane detection and the selected CNN Encoder–LSTM-Decoder network requires 16.96 m of parameters for lane prediction.

Automatic segmentation is, today, essentially dominated by Encoder–Decoder models. Indeed, this type of models is adapted to several image processing tasks such as detection, recognition and tracking. Among the Encoder–Decoder models that have

Table 5
Processing time results obtained for the proposed Deep LDWS based on CPU processor and the NVIDIA Jetson Xavier NX platforms.

	Input Image Size	Metrics	CPU Processor platform	NVIDIA Jetson Xavier NX platform
LDWS System based on CNN Encoder- LSTM-Decoder network	*480*320*	*Time(ms)*	226	129
		FPS	4.42	7.69
	*640*360*	*Time(ms)*	247	132
		FPS	4.04	7.55
	*640*480*	*Time(ms)*	261	138
		FPS	3.83	7.22
	*1280*720*	*Time(ms)*	276	147
		FPS	3.62	6.78

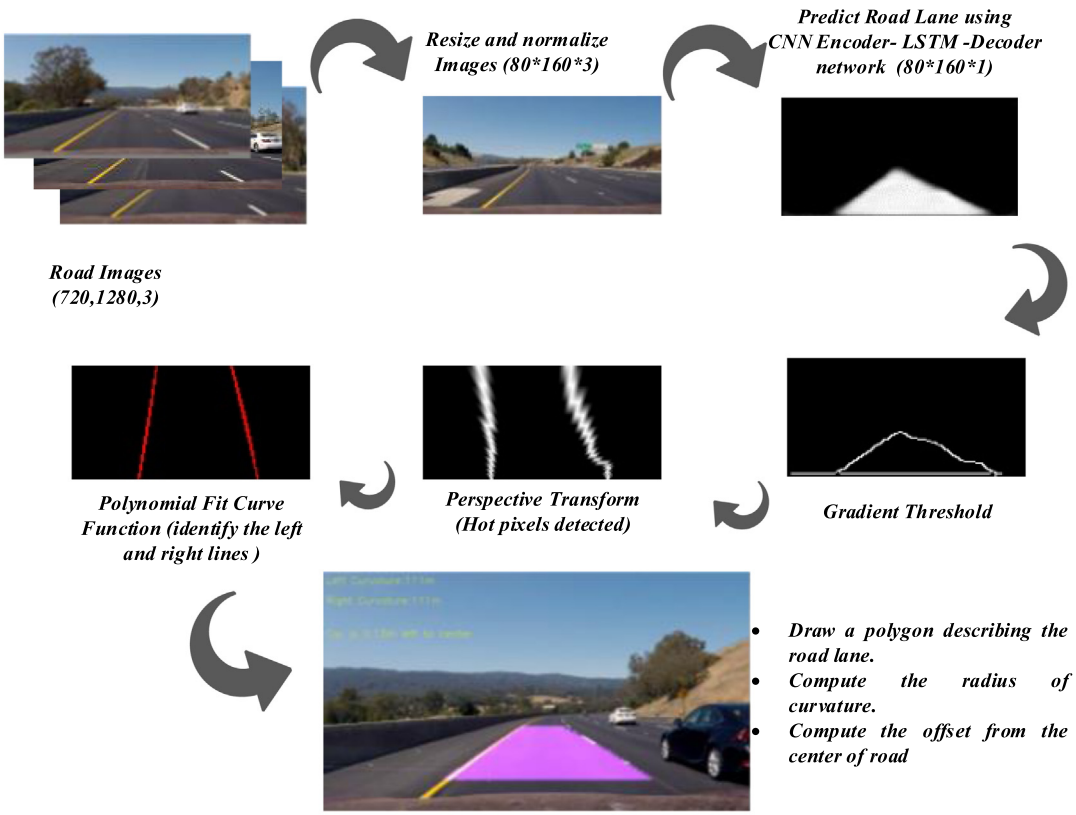

Fig. 10. An overview of the results obtained by the proposed Deep LDWS.

Table 6
Comparison of the proposed system with existing systems in terms of accuracy, recall, F1-score and processing time.

Author	Architecture	Accuracy (%)	Recall (%)	F1-score (%)	Processing Time (ms)
Marzougui et al. [64]	AROI+ PPHT+ Kalman filter	93.82	–	–	21.54
Rastiveis et al. [69]	HT+ Fuzzy theory	–	87	88	–
Ali et al. [72]	HT + RANSAC	96.30	–	–	191
Yoo and Kim [51]	Graph model	93.89		91.6	–
Lee et al. [70]	VPGNet Networks	–	93	88.4	50
Almeida et al. [4]	Two ENet Networks	89	–	–	–
Paszke et al. [65]	ENet	96.2	95.2	–	262
	SegNet	97.2	–	–	289
Zhao et al. [45]	SCNN + DQLL	93.36	–	–	–
Ghanem et al. [3]	RANSAC+ fuzzy controller	95.2	–	–	54
Khairdoost et al. [1]	ResNet101 model	94	–	–	–
Wen et al. [67]	cGAN-based	90.15	82.33	86.06	–
	U-net-based	95.97	87.52	91.55	–
Chen et al. [68]	LMD-11 network	95.4	–	–	2470
Cai et al. [66]	SCNN	93.4	84	–	–
	ARFNet	94.1	80	–	–
	ResNet-34	91.9	78.6	–	–
Xiao et al. [71]	SCNN	93.5	94		113.9
Proposed System	**CNN–LSTM**	**96.36**	**97.54**	**97.42**	**132**

emerged in recent years is the Transformer. The Transformer is a sequence-to-sequence model based on the attention mechanism and not on a recurrent neural network as was the case for the previous models with LSTM (or GRU). The LSTM and GRU models have some limitations since they are relatively slow to train and not very parallelizable. The idea of the Transformer is to preserve the interdependence of a sequence of images by not using a recurrent network but only using the attention mechanism which is at the center of its architecture. The idea behind the attention concept is to measure the extent to which two elements of two sequences are linked. The Transformer was clearly a revolution when it was released because it was both a very powerful translation model and much faster to train than its predecessors. Thus,

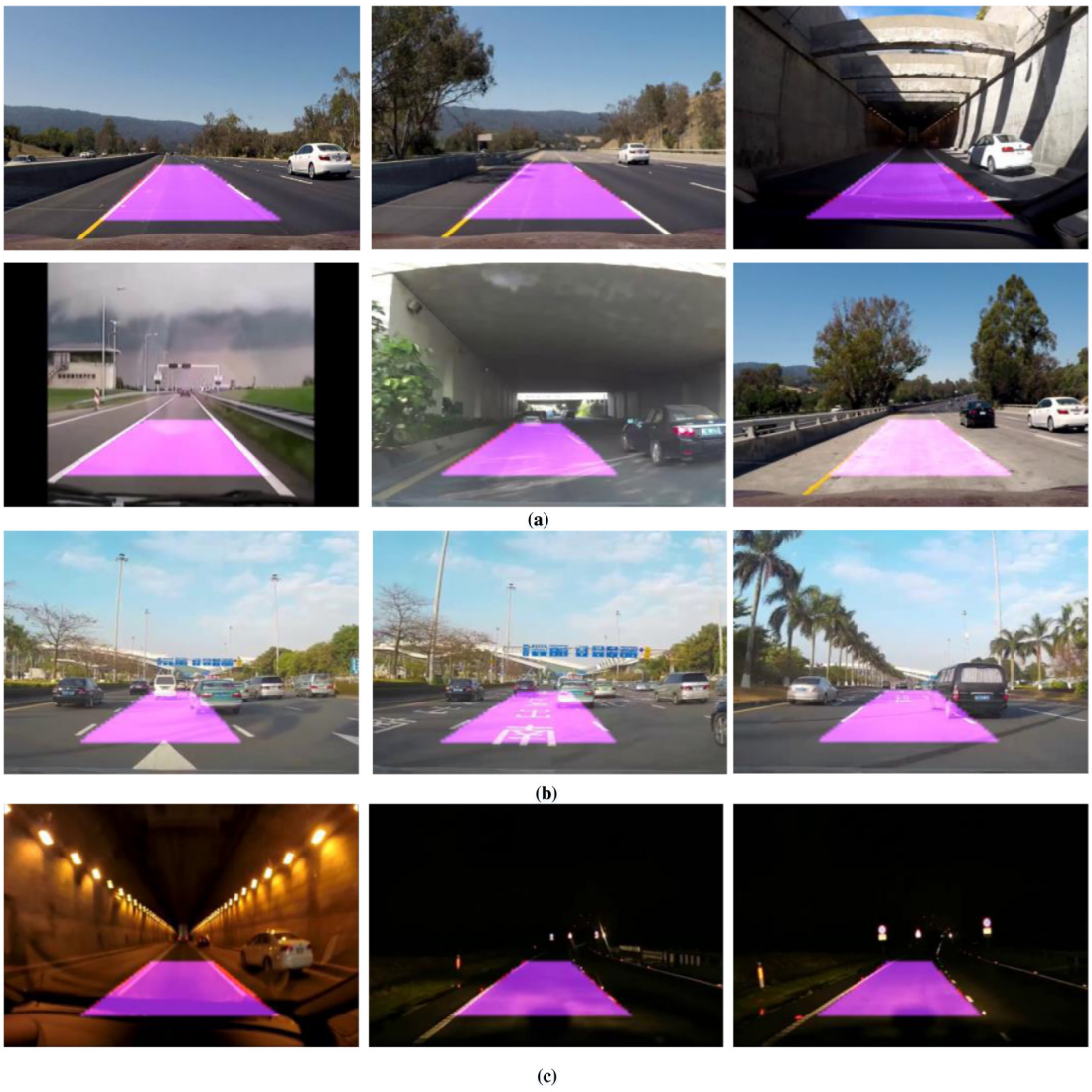

Fig. 11. Lane detection based on the proposed system in different lighting conditions. (a) Urban road in daylight and rainy weather, (b) Highway at high traffic levels, (c) Highway at night, yellow and white lamp tunnel.

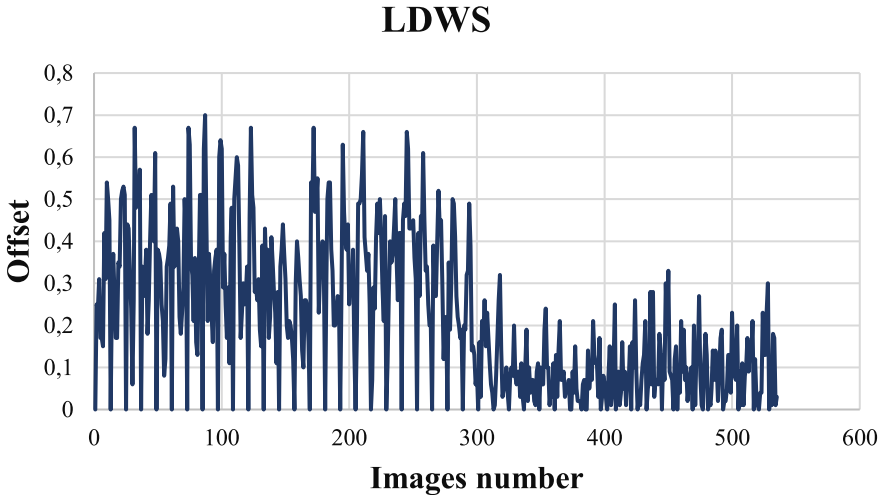

Fig. 12. Numerical estimates of the vehicle's position from the center.

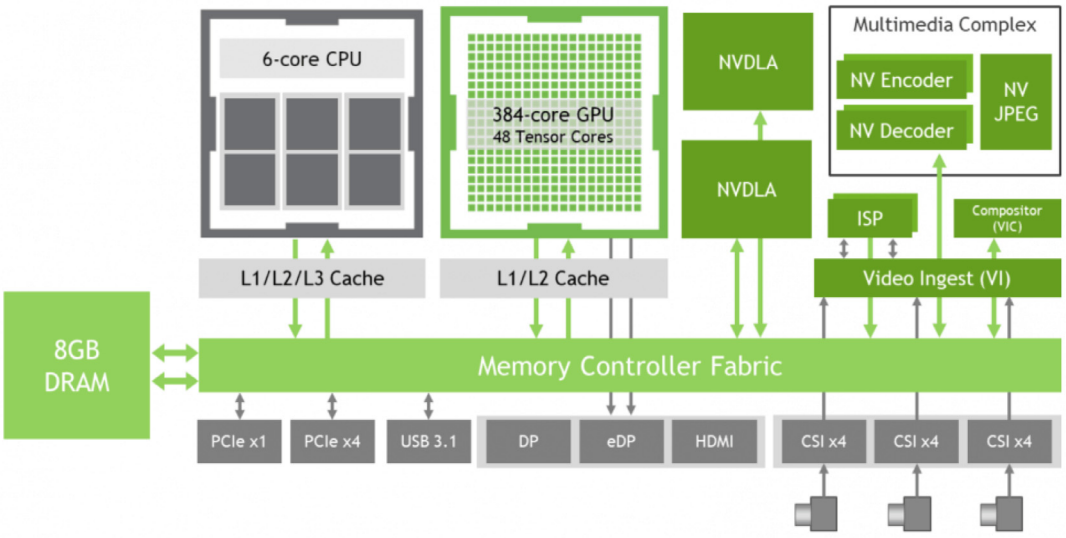

Fig. 13. NVIDIA Jetson Xavier NX.

the Transformer architecture will be studied and implemented in future work.

6. Conclusion

In this paper, we propose an Embedded Deep LDWS based on a hybrid CNN Encoder–LSTM-Decoder network implemented on a NVIDIA Jetson Xavier NX. The CNN Encoder network is employed due to its capacity to extract the most significant features from a dataset, and to reduce their dimensionality at the same time. However, they can be successfully combined with the ability of LSTM networks to detect and store long-term dependencies between extracted data to improve the detection rate through removal the influence of false alarm patches, which allows to design robust models for automatic lane detection. A corresponding decoder network is used to map the low resolution encoder feature maps and to produce dense feature maps that correspond to road lane.

We proposed three network architectures to predict the road lane: CNN Encoder–Decoder network, CNN Encoder–Decoder network with the application of Dropout layers and CNN Encoder–LSTM-Decoder network. The three networks are trained and tested on a public dataset comprising 12764 road images under different conditions. Based on extensive experimental results, we demonstrated that our proposed hybrid Encoder–LSTM-Decoder network outperforms the remaining two architectures and achieves an average accuracy of 96.36%, Recall of 97.54%, and F1-score of 97.42%.

However, despite the principal purpose of using the proposed system in automobiles, we have found that it is also very cost-effective to deploy our Deep LDWS on the high-performance NVIDIA Jetson Xavier NX platform, which allows large-scale computations to be performed much faster and more efficiently. This makes it very suitable for real-time road scene embedded applications, where it is necessary to process a large number of high-resolution images.

CRediT authorship contribution statement

Yassin Kortli: Conceptualization, Methodology, Software, Validation, Writing–original draft. **Souhir Gabsi:** Software, Validation, Writing–original draft. **Lew F.C. Lew Yan Voon:** Visualization, Investigation, Writing – review & editing. **Maher Jridi:** Supervision, Review & editing. **Mehrez Merzougui:** Supervision, Editing. **Mohamed Atri:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The paper is co-financed by LABISEN-VISION-AD, ISEN Nantes, France, by ImViA Laboratory, University of Burgundy, France, and by FSM University of Monastir, Tunisia with collaboration of the Ministry of Higher Education and Scientific Research of Tunisia. The context of the paper is the ATER project of Yassin Kortli.

References

[1] N. Khairdoost, S.S. Beauchemin, M.A. Bauer, Road lane detection and classification in urban and suburban areas based on CNNs, in: VISIGRAPP (5: VISAPP), 2021, pp. 450–457, http://dx.doi.org/10.5220/0010241004500457.
[2] J. Shim, O.H. Kwon, S.H. Park, S. Chung, K. Jang, Evaluation of section speed enforcement system using empirical Bayes approach and turning point analysis, J. Adv. Transp. 2020 (2020) http://dx.doi.org/10.1155/2020/9461483.
[3] S. Ghanem, P. Kanungo, G. Panda, S.C. Satapathy, R. Sharma, Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure, Complex Intell. Syst. (2021) 1–12, http://dx.doi.org/10.1007/s40747-021-00381-2.
[4] T. Almeida, B. Lourenço, V. Santos, Road detection based on simultaneous deep learning approaches, Robot. Auton. Syst. 133 (2020) 103605, http://dx.doi.org/10.1016/j.robot.2020.103605.
[5] T. Heo, W. Nam, J. Paek, J. Ko, Autonomous reckless driving detection using deep learning on embedded GPUs, in: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems, MASS, IEEE, 2020, pp. 464–472, http://dx.doi.org/10.1109/MASS50613.2020.00063.
[6] Q. Huang, J. Liu, Practical limitations of lane detection algorithm based on hough transform in challenging scenarios, Int. J. Adv. Robot. Syst. 18 (2) (2021) 17298814211008752, http://dx.doi.org/10.1177/17298814211008752.

[7] J. Suder, K. Podbucki, T. Marciniak, A. Dąbrowski, Low complexity lane detection methods for light photometry system, Electronics 10 (14) (2021) 1665.
[8] J.H. Koo, S.W. Cho, N.R. Baek, M.C. Kim, K.R. Park, CNN-based multimodal human recognition in surveillance environments, Sensors 18 (9) (2018) 3040, http://dx.doi.org/10.3390/s18093040.
[9] R.Nandhini. Abirami, P.M. Durai Raj Vincent, K. Srinivasan, U. Tariq, C.Y. Chang, Deep CNN and deep GAN in computational visual perception-driven image analysis, Complexity 2021 (2021).
[10] S.N. Shuvo, F. Hasan, M.U. Ahmed, S.A. Hossain, S. Abujar, MathNET: Using CNN bangla handwritten digit, mathematical symbols, and trigonometric function recognition, in: Soft Computing Techniques and Applications, Springer, Singapore, 2021, pp. 515–523, http://dx.doi.org/10.1007/978-981-15-7394-1_47.
[11] G.S.M. Diyasa, A. Fauzi, M. Idhom, A. Setiawan, Multi-face recognition for the detection of prisoners in jail using a modified cascade classifier and CNN, J. Phys. Conf. Ser. 1844 (1) (2021) 012005, http://dx.doi.org/10.1088/1742-6596/1844/1/012005.
[12] B. Gao, X. Huang, J. Shi, Y. Tai, J. Zhang, Hourly forecasting of solar irradiance based on CEEMDAN and multi-strategy CNN-LSTM neural networks, Renew. Energy 162 (2020) 1665–1683, http://dx.doi.org/10.1016/j.renene.2020.09.141.
[13] S. Li, Z. Yan, X. Wu, A. Li, B. Zhou, A method of emotional analysis of movie based on convolution neural network and bi-directional LSTM RNN, in: 2017 IEEE Second International Conference on Data Science in Cyberspace, DSC, IEEE, 2017, pp. 156–161.
[14] M.Z. Islam, M.M. Islam, A. Asraf, A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images, Inform. Med. Unlocked 20 (2020) 100412, http://dx.doi.org/10.1016/j.imu.2020.100412.
[15] C. Hong, J. Yu, J. Zhang, X. Jin, K.H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Ind. Inf. 15 (7) (2018) 3952–3961, http://dx.doi.org/10.1109/TII.2018.2884211.
[16] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670, http://dx.doi.org/10.1109/TIP.2015.2487860.
[17] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2014) 767–779.
[18] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2014) 3742–3751, http://dx.doi.org/10.1109/TIE.2014.2378735.
[19] J. Yu, M. Tan, H. Zhang, D. Tao, Y. Rui, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2019) http://dx.doi.org/10.1109/tpami.2019.2932058.
[20] N. Rahal, M. Tounsi, A. Hussain, A.M. Alimi, Deep sparse auto-encoder features learning for arabic text recognition, IEEE Access 9 (2021) 18569–18584, http://dx.doi.org/10.1109/ACCESS.2021.3053618.
[21] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26, http://dx.doi.org/10.1016/j.neucom.2016.12.038.
[22] W. Yu, N. Lu, X. Qi, P. Gong, R. Xiao, Pick: Processing key information extraction from documents using improved graph learning-convolutional networks, in: 2020 25th International Conference on Pattern Recognition, ICPR, IEEE, 2021, pp. 4363–4370, http://dx.doi.org/10.1109/icpr48806.2021.9412927.
[23] G. Salomon, R. Laroca, D. Menotti, Deep learning for image-based automatic dial meter reading: Dataset and baselines, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–8, http://dx.doi.org/10.1109/IJCNN48605.2020.9207318.
[24] N. Rahal, M. Tounsi, A. Hussain, A.M. Alimi, Deep sparse auto-encoder features learning for arabic text recognition, IEEE Access 9 (2021) 18569–18584, http://dx.doi.org/10.1109/ACCESS.2021.3053618.
[25] M. Mohd, F. Qamar, I. Al-Sheikh, R. Salah, Quranic optical text recognition using deep learning models, IEEE Access 9 (2021) 38318–38330, http://dx.doi.org/10.1109/ACCESS.2021.3064019.
[26] M. Mohd, F. Qamar, I. Al-Sheikh, R. Salah, Quranic optical text recognition using deep learning models, IEEE Access 9 (2021) 38318–38330, http://dx.doi.org/10.1109/ACCESS.2021.3064019.
[27] L. Liu, H. Zuo, X. Qiu, Research on defect pattern recognition of light guide plate based on deep learning semantic segmentation, J. Phys. Conf. Ser. 1865 (2) (2021) 022033, http://dx.doi.org/10.1088/1742-6596/1865/2/022033.
[28] W. Chen, W. Wang, K. Wang, Z. Li, H. Li, S. Liu, Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation-a review, J. Traffic Transp. Eng. (Engl. Ed.) (2020) http://dx.doi.org/10.1016/j.jtte.2020.10.002.
[29] Y.B. Liu, M. Zeng, Q.H. Meng, D-vpnet: A network for real-time dominant vanishing point detection in natural scenes, Neurocomputing 417 (2020) 432–440, http://dx.doi.org/10.1016/j.neucom.2020.08.021.
[30] B. Dorj, S. Hossain, D.J. Lee, Highly curved lane detection algorithms based on Kalman filter, Appl. Sci. 10 (7) (2020) 2372, http://dx.doi.org/10.3390/app10072372.
[31] T. Almeida, B. Lourenço, V. Santos, Road detection based on simultaneous deep learning approaches, Robot. Auton. Syst. 133 (2020) 103605, http://dx.doi.org/10.1016/j.robot.2020.103605.
[32] S.W. VCho, N.R. Baek, M.C. Kim, J.H. Koo, J.H. Kim, K.R. Park, Face detection in nighttime images using visible-light camera sensors with two-step faster region-based convolutional neural network, Sensors 18 (9) (2018) 2995.
[33] R.A. Sagum, Incorporating deblurring techniques in multiple recognition of license plates from video sequences, Turk. J. Comput. Math. Educ. (TURCOMAT) 12 (3) (2021) 5447–5452, http://dx.doi.org/10.17762/turcomat.v12i3.2194.
[34] X. Jin, R. Tang, L. Liu, J. Wu, Vehicle license plate recognition for fog-haze environments, IET Image Process. (2021) http://dx.doi.org/10.1049/ipr2.12103.
[35] Y. Kortli, S. Gabsi, M. Jridi, A. Alfalou, M. Atri, Hw/Sw co-design technique for 2D fast fourier transform algorithm on zynq SoC, Integration 82 (2022) 78–88.
[36] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection using gabor filters and support vector machines, in: 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628), Vol. 2, IEEE, 2002, pp. 1019–1022, http://dx.doi.org/10.1109/ICDSP.2002.1028263.
[37] D.O. Cualain, C. Hughes, M. Glavin, E. Jones, Automotive standards-grade lane departure warning system, IET Intell. Transp. Syst. 6 (1) (2012) 44–57, http://dx.doi.org/10.1049/iet-its.2010.0043.
[38] S.K. Meher, M.N. Murty, Efficient method of moving shadow detection and vehicle classification, AEU-Int. J. Electron. Commun. 67 (8) (2013) 665–670, http://dx.doi.org/10.1016/j.aeue.2013.02.001.
[39] X. An, E. Shang, J. Song, J. Li, H. He, Real-time lane departure warning system based on a single FPGA, EURASIP J. Image Video Process. 2013 (1) (2013) 1–18.
[40] S. Jeon, E. Kwon, I. Jung, Traffic measurement on multiple drive lanes with wireless ultrasonic sensors, Sensors 14 (12) (2014) 22891–22906, http://dx.doi.org/10.3390/s141222891.
[41] M.B. de Paula, C.R. Jung, Automatic detection and classification of road lane markings using onboard vehicular cameras, IEEE Trans. Intell. Transp. Syst. 16 (6) (2015) 3160–3169.
[42] H. Yoo, U. Yang, K. Sohn, Gradient-enhancing conversion for illumination-robust lane detection, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1083–1094.
[43] K. Yassin, J. Maher, M. Mehrez, A. Mohamed, Optical face detection and recognition system on low-end-low-cost Xilinx Zynq SoC, Optik 217 (2020) 164747.
[44] Y. Cai, X. Sun, H. Wang, L. Chen, H. Jiang, Night-time vehicle detection algorithm based on visual saliency and deep learning, J. Sensors 2016 (2016) http://dx.doi.org/10.1155/2016/8046529.
[45] Z. ZZhao, Q. Wang, X. Li, Deep reinforcement learning based lane detection and localization, Neurocomputing 413 (2020) 328–338, http://dx.doi.org/10.1016/j.neucom.2020.06.094.
[46] J. Tang, S. Li, P. Liu, A review of lane detection methods based on deep learning, Pattern Recognit. 111 (2021) 107623, http://dx.doi.org/10.1016/j.patcog.2020.107623.
[47] M. Fang, L. Tang, X. Yang, Y. Chen, C. Li, Q. Li, FTPG: A fine-grained traffic prediction method with graph attention network using big trace data, IEEE Trans. Intell. Transp. Syst. (2021) http://dx.doi.org/10.1109/TITS.2021.3049264.
[48] J. Sun, Y. Fu, S. Li, J. He, C. Xu, L. Tan, Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors, J. Sensors 2018 (2018).
[49] K. Ghazali, R. Xiao, J. Ma, Road lane detection using H-maxima and improved hough transform, in: 2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation, IEEE, 2012, pp. 205–208, http://dx.doi.org/10.1109/CIMSim.2012.31.
[50] F. Zheng, S. Luo, K. Song, C.W. Yan, M.C. Wang, Improved lane line detection algorithm based on hough transform, Pattern Recognit. Image Anal. 28 (2) (2018) 254–260.
[51] J.H. Yoo, D.H. Kim, Graph model-based lane-marking feature extraction for lane detection, Sensors 21 (13) (2021) 4428.
[52] Y. Kortli, M. Jridi, A. Al Falou, M. Atri, A novel face detection approach using local binary pattern histogram and support vector machine, in: 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), IEEE, 2018, pp. 28–33, http://dx.doi.org/10.1109/ASET.2018.8379829.
[53] X.X. Yan, C. Wang, D. Hao, M. Chen, License plate detection using Bayesian method based on edge features, in: 2021 IEEE 5th International Conference on Cryptography, Security and Privacy, CSP, IEEE, 2021, pp. 205–211, http://dx.doi.org/10.1109/CSP51677.2021.9357598.
[54] Y.D. Kim, G.J. Son, H. Kim, C. Song, J.H. Lee, Smart disaster response in vehicular tunnels: Technologies for search and rescue applications, Sustainability 10 (7) (2018) 2509, http://dx.doi.org/10.3390/su10072509.

[55] Y. Ouerhani, A. Alfalou, C. Brosseau, Road mark recognition using HOG-svm and correlation, in: Optics and Photonics for Information Processing XI, Vol. 10395, International Society for Optics and Photonics, 2017, p. 103950Q, http://dx.doi.org/10.1109/ITSC.2014.6957755.

[56] P. Ravindran, A. Costa, R. Soares, A.C. Wiedenhoeft, Classification of CITES-listed and other neotropical meliaceae wood images using convolutional neural networks, Plant Meth. 14 (1) (2018) 1–10.

[57] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder–decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, http://dx.doi.org/10.1109/TPAMI.2016.2644615.

[58] I.E. I Nordeng, A. Hasan, D. Olsen, J. Neubert, DEBC detection with deep learning, in: Scandinavian Conference on Image Analysis, Springer, Cham, 2017, pp. 248–259, http://dx.doi.org/10.1007/978-3-319-59126-1_21.

[59] G.G. Petmezas, K. Haris, L. Stefanopoulos, V. Kilintzis, A. Tzavelis, J.A. Rogers, N.… Maglaveras, Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets, Biomed. Signal Process. Control 63 (2021) 102194, http://dx.doi.org/10.1016/j.bspc.2020.102194.

[60] M. Lu, S. Niu, A detection approach using LSTM-CNN for object removal caused by exemplar-based image inpainting, Electronics 9 (5) (2020) 858, http://dx.doi.org/10.3390/electronics9050858.

[61] T.Y. Kim, S.B. Cho, Predicting residential energy consumption using CNN-LSTM neural networks, Energy 182 (2019) 72–81, http://dx.doi.org/10.1016/j.energy.2019.05.230.

[62] C. Ouchicha, O. Ammor, M. Meknassi, CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images, Chaos Solitons Fractals 140 (2020) 110245, http://dx.doi.org/10.1016/j.chaos.2020.110245.

[63] E. Tsironi, P. Barros, C. Weber, S. Wermter, An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition, Neurocomputing 268 (2017) 76–86, http://dx.doi.org/10.1016/j.neucom.2016.12.088.

[64] M. Marzougui, A. Alasiry, Y. Kortli, J. Baili, A lane tracking method based on progressive probabilistic hough transform, IEEE Access 8 (2020) 84893–84905, http://dx.doi.org/10.1109/ACCESS.2020.2991930.

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

[66] Y. Cai, Y. Zhang, C. Pan, Lane detection based on adaptive network of receptive field, Secur. Commun. Netw. 2021 (2021).

[67] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, A. Habib, A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds, ISPRS J. Photogramm. Remote Sens. 147 (2019) 178–192.

[68] P.R. Chen, S.Y. Lo, H.M. Hang, S.W. Chan, J.J. Lin, Efficient road lane marking detection with deep learning, in: 2018 IEEE 23rd International Conference on Digital Signal Processing, DSP, IEEE, 2018, pp. 1–5.

[69] H. Rastiveis, A. Shams, W.A. Sarasua, J. Li, Automated extraction of lane markings from mobile LiDAR point clouds based on fuzzy inference, ISPRS J. Photogramm. Remote Sens. 160 (2020) 149–166.

[70] S. Lee, J. Kim, J.Shin. Yoon, S. Shin, O. Bailo, N. Kim, I. . So Kweon, Vpgnet: Vanishing point guided network for lane and road marking detection and recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1947–1955.

[71] D. Xiao, X. Yang, J. Li, M. Islam, Attention deep neural network for lane marking detection, Knowl.-Based Syst. 194 (2020) 105584.

[72] M. Aly, Real time detection of lane markers in urban streets, in: 2008 IEEE Intelligent Vehicles Symposium, IEEE, 2008, pp. 7–12.