# Fast-ICA Based Lane Detection Method for Autonomous Vehicles

**Hasibe Busra Dogru[1], Aydin Tarik Zengin[1, *]**
[1]*Department of Computer Engineering, Istanbul Sabahattin Zaim University, Halkali Merkez, Halkali, 34303, Kucukcekmece, Istanbul, Turkey*
*tarik.zengin@izu.edu.tr*

2022 26th International Conference Electronics | 978-1-6654-8321-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/IEEECONF55059.2022.9810405

*Abstract*—**Lane detection is an important process in autonomous vehicle systems. Noise in the image, such as object shadows and terminating lane lines, make lane detection difficult. This study proposes a Convolutional Neural Network architecture with a dimension reduction method that has not been used before in lane detection. The proposed method has been tested with the open-source TuSimple dataset. The results showed that the proposed Fast-Independent Component Analysis based model training improved performance in lane detection and reduced the mean percent error by 42.2%.**

*Index Terms*—**Autonomous vehicles; Deep learning; Independent component analysis; Lane detection.**

## I. Introduction

The main reasons for traffic accidents are driver-related faults and failure to pay attention to traffic rules. Functions in advanced driver assistance systems help prevent driver error of accidents and reduce the impact and cost of the accident. One of the most important functions is the lane tracking system, which allows the vehicle to move forward and change lanes safely. At this point, many traditional computer vision techniques have been used for lane detection, and studies have been carried out to transform data from devices such as sensors and cameras into meaningful results. Studies have provided successful results for lane detection, but there have been problems requiring manual correction to set parameters when the scenario has changed. With the development of computer hardware in recent years, deep learning-based studies have started to be used in lane detection [1]. More successful results were obtained by reducing the need for parameter adjustment.

There are three essential stages in machine vision when lane detection is performed with Convolutional Neural Networks: image preprocessing, feature extraction, and model extraction. Image preprocessing is one of the essential methods of lane detection. Images may contain undesirable factors such as noise, shadows, or inconsistent lighting. With preprocessing, noise is removed and features are enhanced, increasing success in the following steps.

In deep learning, overfitting may occur while training the model. In this case, the model performs well on the training data and poorly on the test data. Predictive models have a variable factor called a feature. As the number of features increases, working on the dataset becomes more challenging. The presence of irrelevant features in the data can reduce the accuracy of the models. For this, basic features are obtained from the dataset using dimension reduction methods [2].

The PCA and ICA methods, which are often used for dimension reduction, appear to be related but have different functions. PCA usually compresses the information. ICA transforms the input into an independent basis to separate the information [3]. In some machine vision studies, the ICA method performed better than PCA and higher recognition rates were obtained [4], [5].

In this paper, we propose a CNN-based model using the dimension reduction method Fast-ICA and the data augmentation method to detect the lane. We used the popular CNN architecture AlexNet to compare their performance in lane detection.

This paper is organized as follows: In the second part, CNN components are given and the CNN architecture used in the study is explained. In the third section, the dataset and experimental work steps are presented. Then the results were compared.

## II. Related Works

While detecting lanes with traditional methods, lanes were modeled geometrically using a gradient, color, and texture features. Selver et al. applied the Gabor filter to the images taken from the videos in the preprocessing stage for lane detection and removed unwanted gaps by applying morphological operations [6]. In another study, Homayounfar et al. performed pattern matching to detect lane candidates, then color clustering to remove lanes using a multilayer detector [7]. Another frequently used method for lane tracking is the Kalman filter [8-10]. Kalman filter helps to predict the position value of the lanes and gives accurate results in predicting the position and curvature of the lanes. One frequently used method for estimating multiple lanes is the particle filter [11]. Loos et al. showed that using the Kalman filter and particular filter together gives better results in cases where there is more than one lane [12].

Due to the success of deep learning in computer vision, many studies on lane detection have been proposed. Kim and Park performed lane detection with the transfer learning method. The authors built an end-to-end encoder-decoder network based on pre-trained ImageNet for road scene object segmentation [13]. Neven et al. performed a segmentation study that detects lanes using the LaneNet network built on the Segnet [14]. In another study, semi-artificial images were produced using CNN architecture and

Manuscript received 24 January, 2022; accepted 11 April, 2022.

978-1-6654-8321-6/22/$31.00 ©2022 IEEE

lane detection was performed [15]. Li et al. using CNN and RNN together, the road image was first divided into a series of continuous slices, then the feature was extracted using a convolutional neural network [16]. The recurrent neural network was used to detect the lane from the obtained features. Using CNN and RNN together, the authors achieved better results than using CNN alone. Ghaforyan et al. studied lane predictions and evaluation from input images for semantic segmentation using the EL-GAN method [17]. The training dataset consists of real data and synthesized data. For lane detection, the CNN model was applied without pre-processing and post-processing [18]. Lee et al. used VPGNet architecture for lane detection. The authors tried to identify the lanes in harsh weather and lighting conditions with the created architecture. In the proposed study, lane and road signs were defined and classified, and in addition, the vanishing point was estimated [19]. Haris and Glowacz used object feature distillation for lane detection. The method was tested with different deep learning models on the CuLane dataset. The approach performed better on the F1 Measure of success relative to the methods compared [20]. Chen and Xiang proposed a new network for lane detection that takes pre-aligned multiple successive frames as input. Their aim in the study is to improve performance in different scenarios such as shadow and lane marking degradation. Experiments show that the proposed method improves lane detection when tested on the TuSimple and ApolloScape dataset [21]. In [22], CNN-Encoder-Decoder Network, CNN Encoder-Decoder network with the application of Dropout layers and CNN Encoder-LSTM-Decoder network were used for lane detection. The results showed that the hybrid Encoder-LSTM-Decoder network performs better than other networks. The open-source TuSimple dataset was trained on the CNN model in another lane detection study. In the study, they determined the lane classes and performed clustering and classification at the same time [23]. Lee et al. proposed a lightweight and fast model. They used the TuSimple dataset to test their model. The results obtained showed that their model adapted successfully to recent technology [24]. PCA, the dimension reduction method, improves the lane detection results [25]-[26].

In this study, the effect of the Fast-ICA method, which is one of the popular ICA methods that have not been used before, on the success rate of lane detection was investigated.

## III. METHOD

### A. *Convolutional Neural Networks (CNN)*

Convolutional Neural Networks (CNN), one of the deep learning methods, is an artificial neural network that gives successful results in analyzing images. CNN's require minimal preprocessing. A separate process is carried out in each layer of the convolutional neural network, which has a multi-layered structure, and data is transferred to the next layer. Traditional CNN architecture consists of five main layers. These are Input Layer, Convolution Layer, Pooling Layer, Fully Connected Layer and Output Layer. The operations performed by the neural network differ according to the type of input data.

AlexNet is one of the popular CNN architectures. A deep learning algorithm was proposed by Alex Krishevsky, Ilya Sutskever, and Geoffrey Hinton [27]. This deep convolutional neural network consists of 25 layers, five convolution layers, three maxpooling layers, two dropout layers, three fully connected layers (dense), seven relu layers, two normalization layers, and a softmax, input and classification (output) layer. The image that will take place in the input layer is 224x224x3. It is a deep learning algorithm with an accuracy rate of 80% in the ImageNet database.

### B. *Fast-Independent Component Analysis (Fast-ICA)*

Independent Component Analysis (ICA) is a statistical method that expresses multivariate data as a linear combination of independent components. Independent component analysis was first introduced by Herault et al. in a study to develop a simplified model of the motion in muscle contraction [28]. Today's independent component analysis has many applications in various disciplines such as image processing, brain tomography, communication, finance, and seismology [29].

Fast-ICA has been named "Fast" ICA as it has fast convergence capability, which means faster estimation and improvement of individual components [30]. Fast-ICA algorithm, which is one of the popular methods of ICA, is simple and gives good results in many applications with its convergence speed [31]-[32]. The symmetric version of some of the features in the FastICA algorithm is also shared [33].

ICA is closely related to multivariate methods such as principal component analysis (PCA), factor analysis (FA), and minimum/maximum autocorrelation factors (MAF) analysis. While PCA, FA and MAF produce unrelated and normally distributed factors, ICA produces independent and at the same time non-normally distributed factors [34]. Independence is a much stronger trait than dissociation. Therefore, while second-order statistics such as covariance and variogram are used to find unrelated factors, higher-order statistics are needed to estimate factors with ICA. Another difference between ICA and other multivariate methods is interpreting the obtained factors. In ICA, components are not ordered by size. In other words, there is no wrong or good component. Secondly, the produced components do not change according to the sign of the source. For example, a white letter on a black background is the same as a black letter on a white background in image processing [35].

## IV. EXPERIMENTS

### A. *Dataset*

The open-source TuSimple dataset was used in the study [36]. The resolution of the images in this dataset is 1280x720. Some images of the dataset are shown in Fig. 1.

High resolution increases the computational cost and makes it difficult to render on the network. That is why we reduced the size of the original images to 128x128. A set of pixel coordinates (x,y) was created as a tag for each image to label the images. While training, the images were separated as 2604 for training, 720 for validation, 300 for tests.

New versions of images were created with the data augmentation method. That helps generalize the model's training data to images rather than appearing in a specific way. Images were randomly rotated, width-height shifted and zoomed during data augmentation. In this way, the number of images in the training dataset increased to 5104. The images after the data augmentation process are shown in Fig. 2.

Fig. 1. Sample images from TuSimple dataset.

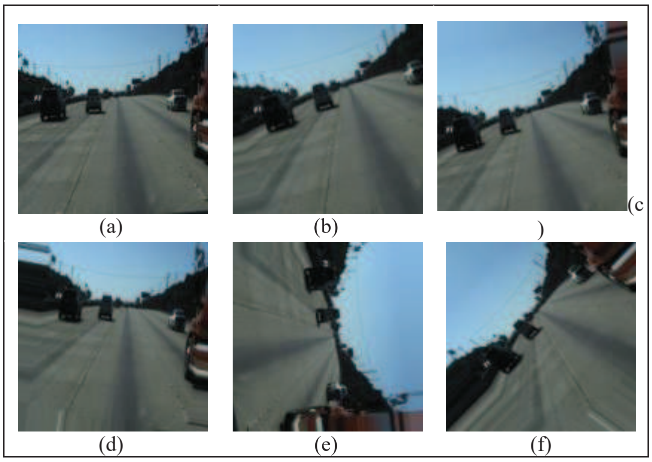


Fig. 2. Some images after the data augmentation process. Original image (a) and images created with data augmentation (b, c, d, e, f).

Fig. 3. Some images after Fast-ICA process.

Feature extraction was performed before the CNN training by applying Fast-ICA with a component value of 50. Some images of the dataset after applying Fast-ICA are shown in Fig. 3.

### B. *Training the Network*

AlexNet, the popular CNN architecture, was used in the study. Adam optimizer and mean square error (MSE) loss function was used in the designed network. The epoch value was set to 100 and the batch size was set to 32. The generated datasets were trained separately. The loss function graphs obtained from the training are shown in Fig. 4.

### C. *Evaluation Metric*

After the training phase, the model was evaluated and measured mean percent error (MPE). MPE is the average percentage of the difference between the model's predicted and actual values.

$$MPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{a_t - f_t}{a_t}, \qquad (1)$$

where $a_t$ is the actual value, $f_t$ is the estimated value, and $n$ is the number of different times the variable is estimated.

## V. EXPERIMENTAL RESULTS

The TuSimple dataset was trained on the network designed based on AlexNet. The performance of this model was calculated with the MPE metric for the test images. First, the 128x128 size dataset (original dataset) was trained. Training the original dataset yielded an MPE of 6.25%. Then, the data augmentation process was applied and the results were compared. Data augmentation did not improve the performance. MPE was increased by 10.5%, which was an unwanted result. Then, the recorded dataset was trained by applying the Fast-ICA method, and an MPE result of 3.61% was obtained, which is a 42.2% improvement. Finally, it was trained by applying the data augmentation method and the Fast-ICA method. Using the two methods together gave 5.8% result. Fast-ICA method alone yielded better results. MPE results are shown in Table I.

TABLE I. COMPARISON OF MPE RESULTS.

| Data | MPE |
|---|---|
| Original Data | 6.247326 |
| Augmented Data | 6.906771 |
| Fast-ICA Data | **3.611743** |
| Augmented + Fast-ICA Data | 5.797020 |

The experimental results of the FICA method, which we applied to the original dataset that gave the best results, are shown in Fig. 5. The green and red points represent the ground truth and the predicted lane points. As it can be seen, they match with the lanes accurately.

TABLE II. COMPARISON OF DETECTION ACCURACY (%) BETWEEN PROPOSED METHOD AND PAPER [1].

| Techniques | Accuracy | References and Year |
|---|---|---|
| Data Augmentation + AlexNet | 85% | Ekşi and Gökmen [1] |
| Fast-ICA + AlexNet | 96% | This Work |

The proposed method was compared with another study using the same dataset and model [1]. In the other study, data augmentation was applied before the training. When the results were compared, it was seen that the proposed method gave better results than the other study. The comparison of the results is shown in Table II.

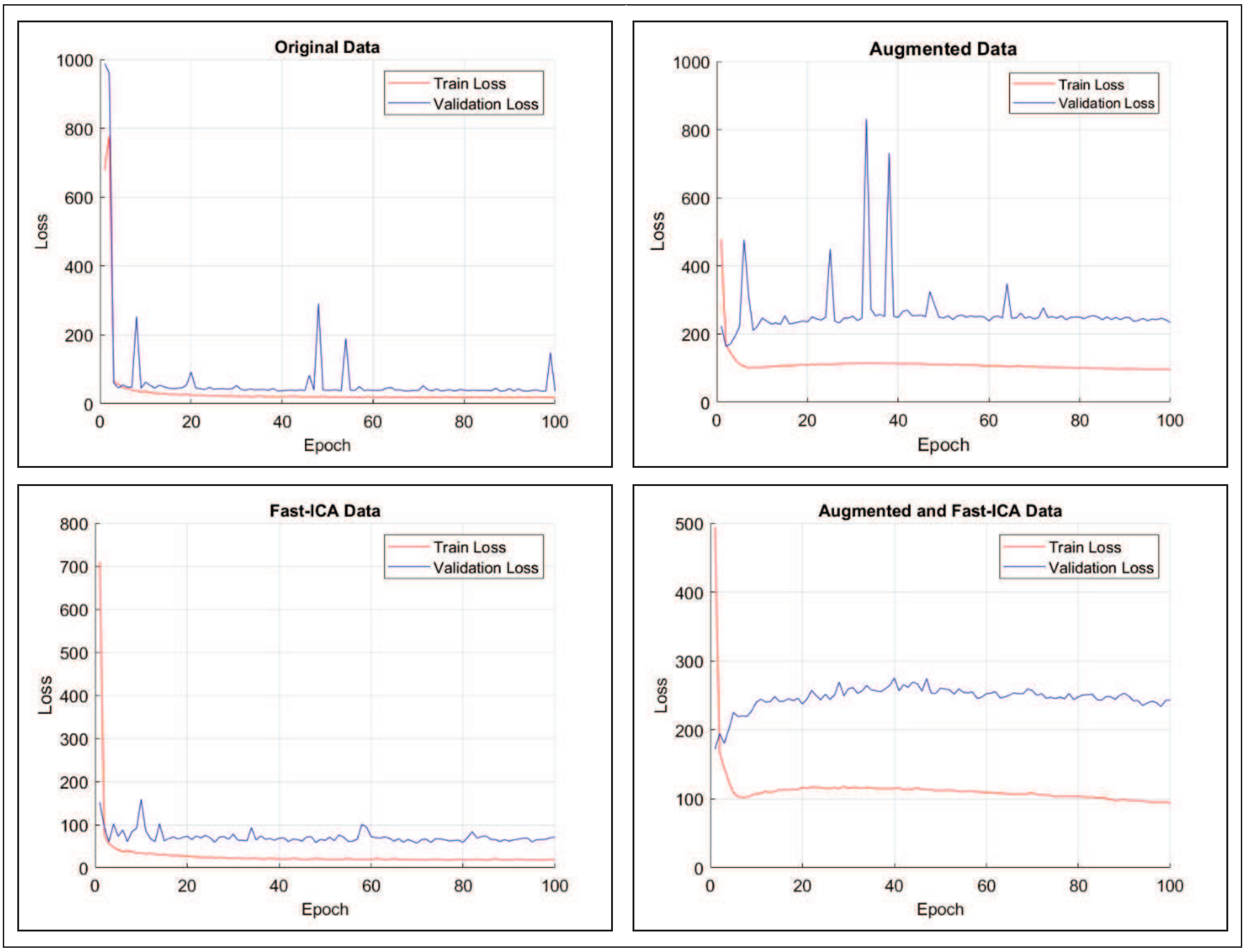


Fig. 4. AlexNet training loss function changes.

Fig. 5. CNN outputs for lane detection by the process of Fast-ICA.

## VI. Discussion

In this study we present and evaluate the performance of Fast-ICA and data augmentation methods on lane detection. The above methods were applied on the TuSimple dataset. The resultant dataset after the application of two methods on the TuSimple dataset were then trained using AlexNet. Experiments on the dataset show that the proposed Fast-ICA method can improve lane detection. [1] obtained 85% using data augmentation whereas we obtained 96% using our proposed method. When the proposed method (Fast-ICA) is tested, lanes are successfully detected even among images with shadows. However, there is room for improvement. For example, images taken in different weather conditions and or recorded at night were not analyzed. Different situations can be used in training by adding various scenarios to the dataset.

## VII. Conclusion

Accurately detecting lanes under some conditions such as presence of extra lighting and shadows is challenging. It is therefore essential to extract information from limited lane visuals. In this study, we improve lane detection using Fast-ICA, which transforms the input into an independent basis for separating information as a dimension reduction method. We evaluated the method's performance using the AlexNet model on the TuSimple dataset. It was observed that the Fast-ICA method gave 42.2% better results than the data augmentation method. Although data augmentation in conjunction Fast-ICA method improved the final result, they did not give better results than those obtained using Fast-ICA alone. As a result, it is seen that the Fast-ICA method gives better results compared to the data augmentation method and improves the performance in lane detection when compared with the original dataset results. Using the proposed approach, 96% accuracy was achieved.

In the future, it is planned to perform real-time lane detection and tracking using a dimension reduction-based CNN architecture on an autonomous vehicle with an embedded computer such as NVIDIA Xavier NX.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References


[1] O. T. Ekşi and G. Gökmen, “Comparing of some convolutional neural network (cnn) architectures for lane detection”, *Balkan Journal of Electrical and Computer Engineering*, *8*(4), pp. 314-319, 2020. DOI: 10.17694/bajece.752177.

[2] M. Verleysen and D. François, The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*. Springer, Berlin, Heidelberg, 2005. p. 758-770. DOI: 10.1007/11494669_93.

[3] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications”, *Neural networks*. 2000, 13.4-5: 411-430. DOI: 10.1016/S0893-6080(00)00026-5.

[4] M. A. Vicente, C. Fernandez, O. Reinoso, and C. Perez, “Robust object detection in complex backgrounds using ICA compression”, 2004.

[5] B. A. Draper, et al. “Recognizing faces with PCA and ICA”, *Computer vision and image understanding*, 2003, 91.1-2: 115-137. DOI: 10.1016/S1077-3142(03)00077-8.

[6] M. A. Selver, E. Er, B. Belenlioglu, and Y. Soyaslan, “Camera based driver support system for rail extraction using 2-d gabor wavelet decompositions and morphological analysis”, in *IEEE Conference on Intelligent Rail Transportation*, 2016, pp. 270–275. DOI: 10.1109/ICIRT.2016.7588744.

[7] N. Homayounfar, W.-.C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun, “Dagmapper: learning to map by discovering lane topology”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2911–2920. DOI: 10.1109/ICCV.2019.00300.

[8] A. Borkar, M. H. Hayes, and M. T. Smith, “Robust lane detection and tracking with ransac and kalman filter”, in *IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 3261–3264. DOI: 10.1109/ICIP.2009.5413980.

[9] T. Suttorp and T. Bucher, “Learning of kalman filter parameters for lane detection”, in *IEEE intelligent vehicles symposium*, 2006, pp. 552–557. DOI: 10.1109/IVS.2006.1689686.

[10] A. Mammeri, A. Boukerche, and G. Lu, “Lane detection and tracking system based on the mser algorithm, hough transform and kalman filter”, in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* 2014. DOI: 10.1145/2641798.2641807.

[11] A. G. Linarth and E. Angelopoulou, “On feature templates for particle filter based lane detection”, in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1721–1726. DOI: 10.1109/ITSC.2011.6083016.

[12] H. Loose, U. Franke, and C. Stiller, “Kalman particle filter for lane recognition on rural roads”, in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 60–65. DOI: 10.1109/IVS.2009.5164253.

[13] J. Kim and C. Park, “End-to-end ego lane estimation based on sequential transfer learning for self-driving cars”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1194–1202. DOI: 10.1109/CVPRW.2017.158.

[14] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, “Towards end-to-end lane detection: an instance segmentation approach”, in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 286-291. DOI: 10.1109/IVS.2018.8500547.

[15] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, “Deeplanes: End-to-end lane position estimation using deep neural networks”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 38–45. DOI: 10.1109/CVPRW.2016.12.

[16] J. Li, X. Mei, D. V. Prokhorov, and D. Tao, “Deep neural network for structural prediction and lane detection in traffic scene”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 690–703, 2017. DOI: 10.1109/TNNLS.2016.2522428.

[17] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, “El-gan: Embedding loss driven generative adversarial networks for lane detection”, in *Proceedings of The European Conference on Computer Vision (ECCV) Workshops*, 2018. DOI: 10.1007/978-3-030-11009-3_15.

[18] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "DeepLanes: End-to-end lane position estimation using deep neural networks", *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, US, pp. 38-45. DOI: 10.1109/CVPRW.2016.12.

[19] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, et al., "VPGNet: Vanishing Point Guided Network for lane and road marking detection and recognition", *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 1965-1973. DOI: 10.1109/ICCV.2017.215.

[20] M. Haris and G. Adam, "Lane line detection based on object feature distillation.", *Electronics*, 2021, 10.9: 1102. DOI: 10.3390/electronics10091102.

[21] Y. Chen, Z. Xiang, “Lane Mark Detection with Pre-Aligned Spatial-Temporal Attention”, *Sensors*, 2022, 22.3: 794. DOI: 10.3390/s22030794.

[22] Y. KORTLI, et al. “Deep embedded hybrid CNN-LSTM network for lane detection on NVIDIA Jetson Xavier NX”, *Knowledge-Based Systems*, 2022, 107941. DOI: 10.1016/j.knosys.2021.107941.

[23] F. Pizzati, M. Allodi, A. Barrera, and F. Garcia, “Lane detection and classification using cascaded CNNs”, in *International Conference on Computer Aided Systems Theory (EUROCAST)*, 2019, pp. 95-103. DOI: 10.1007/978-3-030-45096-0_12.

[24] L. Minhyeok, et al. “Robust lane detection via expanded self attention”, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022. p. 533-542. DOI: 10.1109/WACV51458.2022.00201.

[25] W. Zhang, X. Song, G. Zhang, “Real-time lane departure warning system based on principal component analysis of grayscale distribution and risk evaluation model”, *Journal of Central South University*, 21.4: 1633-1642, 2014. DOI: 10.1007/s11771-014-2105-2.

[26] A. Bosaghzadeh, S. S. Routeh, “A novel PCA perspective mapping for robust lane detection in urban streets”, in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*. IEEE, 2017. p. 145-150. DOI: 10.1109/AISP.2017.8324126.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in *Advances in Neural Information Processing Systems,* 2012, pp. 1097-1105. DOI: 10.1145/3065386.

[28] J. Herault, C. Jutten, B. Ans, “Detection degrandeurs primitives dans un message composite parune architecture de calcul neuromimétique en apprentissage non supervise”, Actes du Xème colloque, GRETSI, pp. 1017- 1022, 1985.

[29] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, Inc., New York, 481 s, 2001. DOI: 10.1002/0471221317.

[30] S. Goel, A. Verma, S. Goel, and K. Juneja, “ICA in image processing: a survey”, in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, pp. 144-149. DOI: 10.1109/CICT.2015.91.

[31] A. Hyvärinen, and E. Oja, “A fast fixed-point algorithm for independent component analysis”, *Neural computation*, vol. 9, no. 7, pp. 1483–1492, 1997. DOI: 10.1162/neco.1997.9.7.1483.

[32] A. Hyvärinen, J. Karhunen, and E. Oja, “Independent Component Analysis”, *Studies in informatics and control, 11(2), 205-207,* 2002. DOI: 10.1002/0471221317.

[33] E. Oja, and Z. Yuan, “The FastICA algorithm revisited: Convergence analysis”, *IEEE transactions on Neural Networks,* vol. 17, no. 6, pp. 1370–1381, 2006. DOI: 10.1109/TNN.2006.880980.

[34] V. J. Stone, *Independent Component Analysis: A Tutorial Introduction*. A Bradford Book, MIT Press, Cambridge, 2004. DOI: 10.7551/mitpress/3717.001.0001.

[35] D. Langlois, S. Chartier, D. Gosselin, “An introduction to independent component analysis: Infomax and FastICA algorithms”, Tutorials in Quantitative Methods for Psychology, vol. 6(1), pp. 31-38, 2010. DOI: 10.20982/tqmp.06.1.p031.

[36] "TuSimple/tusimple-benchmark", GitHub, 2020. [Online]. Available: https://github.com/TuSimple/tusimple-benchmark. Accessed: 17-May- 2020.