*Adv Syst Sci Appl* 2024; **01**; 129-141
Published online at https://ijassa.ipu.ru.

# Vision Based Classification of Nocturnal Road Traffic Using a Custom Deep Convolution Neural Network

Sofiane Abdelkrim Khalladi[1*], Asmâa Ouessai[1,2], Mokhtar Keche[1]

[1)] *Signals and images laboratory, Faculty of Electrical Engineering, Department of Electronics, University of Sciences and Technology of Oran Mohamed Boudiaf USTO-MB, B.P. 1505, El Mnaouar-Bir el Djir-Oran, Algeria*

[2)] *Faculty of Technology, Department of Telecommunications, Dr. Tahar Moulay University, Saida, Algeria*

**Abstract**: Intelligent Transport Systems (ITS) have emerged as an efficient solution for enhancing road utilization efficiency, ensuring convenient and safe transportation, and reducing energy consumption. This research addresses the challenging problem of accurately estimating road traffic congestion from videos recorded in low visibility and adverse weather conditions such as rainy, overcast, and sunny weather. To solve this problem, we propose a method based on the use a custom Deep Convolutional Neural Network (DCNN) that we have designed and trained to classify nighttime road traffic videos into three distinct categories. To train and test this DCNN, we used nighttime videos from the UCSD (University of California San Diego) public dataset. For comparison, we have also tested a second method, which uses an existing CNN that we have trained on the same dataset to classify road traffic into the same three categories. The performances of the two methods were assessed and compared using the UCSD nighttime dataset. The first method achieves an impressive Correct Classification Rate (CCR) of 98.91%, which is higher than the 96.18% CCR achieved by the second method. These rates surpass the 89.47% state-of-the-art CCR obtained with this dataset. This exceptional precision in estimating road traffic congestion in challenging low visibility conditions is achieved thanks to the integration of advanced deep learning techniques and CNNs. The proposed method can be integrated as part of a traffic monitoring system, thus contributing to the advancement of Intelligent Transport Systems and their potential to create cleaner, safer, and more efficient transportation networks.

***Keywords***: CNN, Congestion, ITS, Traffic classification, Night traffic, Macroscopic approach, UCSD.

## 1. INTRODUCTION

Road traffic accidents are a worldwide calamity. In fact, over a million people die in road accidents worldwide each year, and an even greater number end up with some form of disability. Low visibility is one of the causes of road accidents; indeed, the risk of a fatal accident during the night is seven times higher than during the day [1]. Road traffic congestion is a significant cause of accidents and has social, economic, health, and environmental impacts, as well. Several measures have been taken to combat this epidemic, which can occur during the day or night. Among these measures, the employment of recent technology occupies a place of prominence. Indeed, it enables the supervision of road traffic using sensors installed on the road. The data provided by these sensors is processed to extract relevant information for road users and road traffic management and control services, in order to help plan departures (vacation periods), avoid traffic bottlenecks, and thereby lessen their misdeeds. Many cameras are now put on urban roadways and highways for security purposes. These cameras can be used to estimate the status of road traffic.

Currently, there is a high demand for the interpretation of nocturnal traffic situations. In this paper, we propose a traffic framework for the classification of traffic congestion from nighttime surveillance videos. Video analysis at night represents a challenge. Night-time vehicle surveillance is even more complex since heavy traffic flows and incidents might occur on city streets or highways at night. The use of night vision sensors such as infrared-thermal cameras to identify automobiles is an efficient method. Unfortunately, due to their high cost, infrared sensors are rarely used for traffic analysis. CCD cameras, unlike thermal cameras, are sensitive to visible light. However, some challenging variables, such as poor sight and loud surroundings, make some weather or illumination circumstances (such as day, night, sun, rain, snow, and occlusions) difficult to deal with. Primarily, these circumstances are difficult because the traditional daytime surveillance framework, which is based on changes detection (such as background subtraction), is unable to work at night. This is due to two factors: first, the camera's generally poor light sensitivity and contrast, second, the numerous foreground/background ambiguities caused by the changing headlamp reflections. To overcome these limitations, we suggest a new macroscopic method for traffic nighttime classification that does not rely on vehicle monitoring (detection and tracking) and extraction of parameters, such as density and velocity. The proposed method is very efficient and outperforms the state-of-the-art methods in terms of accuracy; it can contribute to making transportation safer, cleaner, and more economical.

The rest of the paper is organized as follows. Section II presents a literature review regarding the topic covered by the present paper. Section III provides a brief overview of convolution neural networks. In Section IV the proposed custom DCNN and the CNN architectures, used for road traffic classification, are successively described. In section V, after an overview of the used dataset, the classification results obtained with the methods based on these two models are analyzed and compared with those obtained with previous works. Lastly, section VI presents some conclusions.

## 2. RELATED WORK

There are two major approaches for road traffic congestion classification. The first approach, known as the microscopic [2] approach, is based on the extraction of the movement parameters of individual vehicles; it thus requires the detection and tracking of all the vehicles in the region of interest. The second approach, known as the macroscopic [3] approach, analyses the observed scene as a whole, without the need of extracting the movement parameters of individual vehicles. In this approach, the flow of automobiles is viewed as a liquid traveling through a tube, making it very responsive to changing road conditions, but lacking specific information about individual vehicles motions. This characteristic underpinned our decision to adopt the macroscopic approach in this research. Important efforts have been made in the literature in the field of nighttime surveillance systems and image processing. C. H. Hsia, Y. Kong, Y. K. Lin, and Y. R. Chien [4] led a research team who developed a technology that can track vehicle headlights at night using multiple features. The technology analyses the distribution of light sources in each captured image and separates the headlights based on their length and width ratios. It distinguishes between vehicles to avoid counting the same vehicle twice. Based on their space and color characteristics, an algorithm that uses similarity analysis is used to determine whether two headlights belong to the same vehicle. Pushkar S. and S. B. Dhonde [5] devised a method for detecting and selecting an appropriate vehicle beam suitable for temporary blindness reduction. Researchers used two methods to reduce headlight glare and temporary blindness. The image sequence is then analysed to determine the presence of a vehicle. A light intensity sensor is another option for dealing with temporary blindness. Another work was presented in [6] by Y. L. Chen, B. F. Wu, H. Y. Huang, and C. J. Fan, who proposed a system for identifying and controlling vehicles at night. To distinguish vehicles from bright objects in image sequences, the system employs automated multilevel histogram thresholding. Furthermore, illuminating objects in vehicle lighting clusters are grouped using a spatial analysis and a clustering method based on their likely motion across successive frames. In another research [7], the proposed stereo configuration incorporates an alignment that displaces the cameras along a vertical baseline to extract useful information pertaining to negative obstacle features for various illustrative terrain settings. The convex framework exploits these properties to evaluate depth jumps in the disparity space image and perform geometrical analysis of potential occlusion regions. The corresponding convex formulations are based on linear matrix inequalities to detect sudden disparity, angle profile, and intensity variations of potential negative obstacles and to estimate the internal depth of the detected obstacles. Experimental results demonstrate that consecutive processes can be structured in a convex framework to efficiently identify negative obstacle attributes at a range of distances, for texture-varying environments, providing a vital extension to real-time vehicular system implementations with superior detection rates. Sayed et al. [8] presented an efficient approach for nighttime contrast enhancement that alters standard histogram equalization to keep the original nighttime image's color information. Each color channel is improved separately by multiplying the increased brightness ratio by the original luminance ratio. Furthermore, Cai et al. [9] merge a daylight image with a midnight image. The low-quality static elements of a nighttime image can be replaced by the high-quality counterparts in the daytime image, using the object extraction approach. Liu and Payeur [10] attempted to eliminate the static false positives introduced by Cucchiara's technique [11] by performing background subtraction before detecting motion. This solution, however, cannot deal with the moving headlight's beam. It is also not possible to apply it to illuminated road scenes. The concept of rear-view monitoring was first proposed by Yoneyama et al. [12] who, in order to improve detection accuracy, installed two cameras that detect both vehicle headlights and taillights, where in the past, most studies focused on daytime traffic monitoring [13]. However, the proposed solutions may not work properly at night. It should be reported that very few papers addressed the problem of nighttime traffic congestion classification. To the best of our knowledge, the last work that dealt with this problem dates from 2013, and since then no other work with improved performance was published. One can cite two existing works that estimate the road traffic status from videos recorded in the daytime and the nighttime. The first one [14] is based on scene modeling by dynamic textures and classification by Support Vectors Machine (SVM). The second one [15] uses motion vectors estimation to extract the mean traffic speed and density, which feed a SVM classifier that categorizes the traffic congestion. Five years later, Kurniawan, J et al. [16] proposed a technique that uses a CNN for image classification on CCTV camera feeds, to create an intelligent traffic congestion detection system. This solution requires less preprocessing for small grayscale images than traditional methods, which need high-quality images and manual feature calculation. A dataset of 1000 evenly distributed CCTV monitoring photos was used to train the CNN model by binary classification. The results revealed that a basic CNN architecture is effective for real-time road traffic situation assessment, with a high average classification accuracy of 89.50%.

In order to address the need for efficient traffic estimation in low light to improve transportation management and safety, this paper presents a framework for the classification of traffic congestion in nighttime films. The suggested approach employs a Deep Convolutional Neural Network (DCNN) model, which was designed to provide accurate and consistent assessment of traffic congestion even in difficult environmental circumstances. The study comprises several contributions, which are listed below:

- We propose a macroscopic technique that, unlike the traditional microscopic technique, does not require vehicle identification and tracking.
- Excellent performance is achieved by our proposed DCNN model, when applied to the Nighttime UCSD dataset.
- Our work focuses on road traffic categorization during night and puts an end to the dearth of works in this area since 2018.
- We use video footage taken in a variety of weather conditions and difficult lighting circumstances, including at night, to systematically classify traffic congestion.

- We tested the same architecture for a recent research paper [16] which used a different CNN model to classify traffic, on the UCSD nighttime dataset. The aim was to compare and demonstrate the effectiveness of the best-performing model.
- Our method outperforms the three state of art methods in terms of classification accuracy.

## 3. CONVOLUTIONAL NEURAL NETWORK: AN OVERVIEW

The success of traditional methods in solving computer vision problems is largely dependent on the feature extraction procedure. Convolutional Neural Networks (CNN) [17], on the other hand, provide an alternative by automatically picking up domain-specific information. This paradigm change has led to a reassessment of several issues in the context of computer vision as a whole. Currently, face recognition, object identification, scene labeling, and image categorization are all accomplished by means of CNNs, with an improved performance, compared to traditional methods.

### *3.1. CNN Architecture*

A Convolutional Neural Network is composed of several types of layers that are described below: convolutional layers, pooling layers and fully connected layers.

*3.1.1. Convolution layer*

To extract information, the convolutional layer [18], leverages local correlations within the image. The process entails inserting a kernel in the image's upper-left corner. The pixel values within the kernel's coverage are multiplied by the corresponding kernel values, and the results are added to the bias. The kernel is then shifted by one pixel, and the process is continued recursively until all feasible locations in the image have been filtered. Figure 1 depicts the convolution operation.

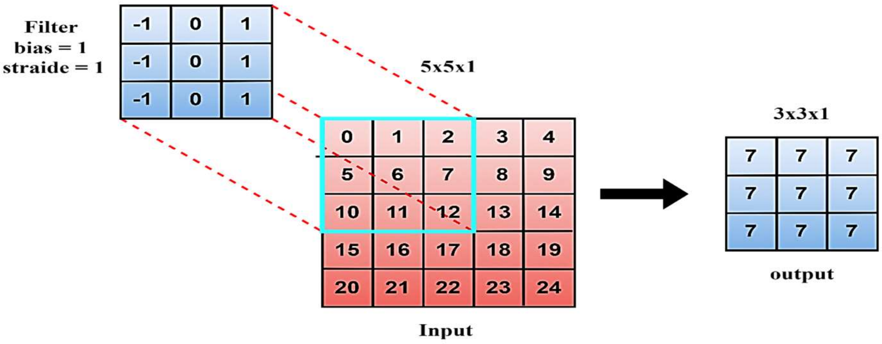

**Fig. 1.** Convolution process operation

*3.1.2. Pooling layer*

The pooling layer [19] provides the model translation, rotation, and scaling invariance while sampling to extract features from the higher layer's feature map. A popular method called maximum pooling (MaxPooling) utilizes the filter size to divide the input picture into rectangular parts and then outputs the maximum value for each zone, as shown in Figure 2. Convolutional and pooling layers often alternate in real-world applications. The integration of multidimensional characteristics is facilitated by connections made between neurons in the fully linked layer and higher neurons. The classifier then translates these features into one-dimensional features that may be used for detection or classification tasks.

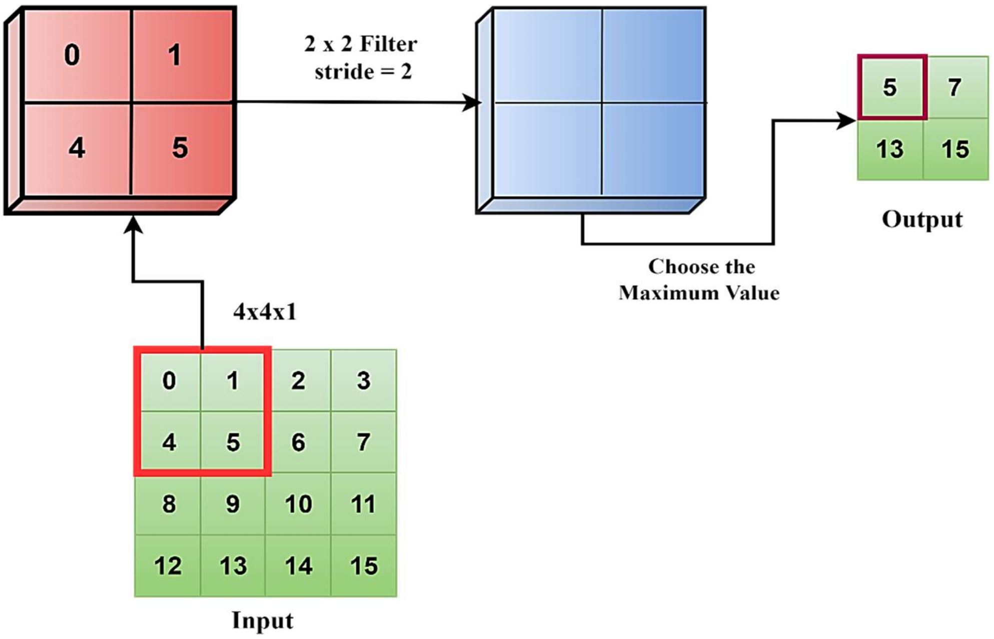

**Fig. 2.** Max Pooling process operation

### *3.1.3. Fully connected layer*

Convolutional Neural Networks (CNNs) conclude by taking the output feature maps generated from the pooling or final convolution layer, and transforming them into a flattened one-dimensional array. Next, this array is joined to one or more dense layers, also referred to as fully connected layer [20], where trainable weights connect each input to every output. The features retrieved by convolution and down sampling are transformed by these fully connected layers into the final outputs of the network, such as class probabilities in classification tasks. The number of classes in the classification task is usually equal to the number of nodes in the final fully linked layer. Each fully connected layer is followed by a nonlinear activation function.

## ***3.2. Activation Functions***

The last fully connected layer in a neural network often employs a different activation function [21] than the preceding layers. The particular task at hand determines which activation function to use. One popular activation function used in multiclass classification is Softmax, defined by the equation:

$$Softmax = \frac{e^{x_i}}{\sum_{i=1}^{k} e^{x_i}}, \quad (1)$$

where $x_i$ represents the *i-th* element of the input vector, which is typically a representation of the neuron activations in the last layer of the neural network before applying the softmax function. Each $x_i$ corresponds to the activation of the neuron associated with the respective output class.

The parameter $k$ in the formula represents the total number of elements in the input vector. In the context of classification, this corresponds to the total number of output classes, to provide a probability estimate for the output classes. The Softmax function takes in a vector of real values as input and produces an output vector of real numbers. The resulting values, which represent estimates of the output classes probabilities, are constrained between 0 and 1, and the entire vector sums up to one.

## 4. THE TESTED METHODS

### *4.1. The Proposed Method*

The proposed method uses a Deep Convolution Neural Network (DCNN) to classify road traffic into three categories, namely, free, medium, and heavy. DCNNs are regarded as the state-of-the-art image classification approach. In this study, a standard ConvNet (Convolutional Network) architecture with convolution layers and pooling layers was used. This architecture is shown in Figure 3, where 'CONV 1' refers to the first convolution layer, 'FC' denotes the fully connected layer, 'S' is the number of stride and 'F' is the number of filters.

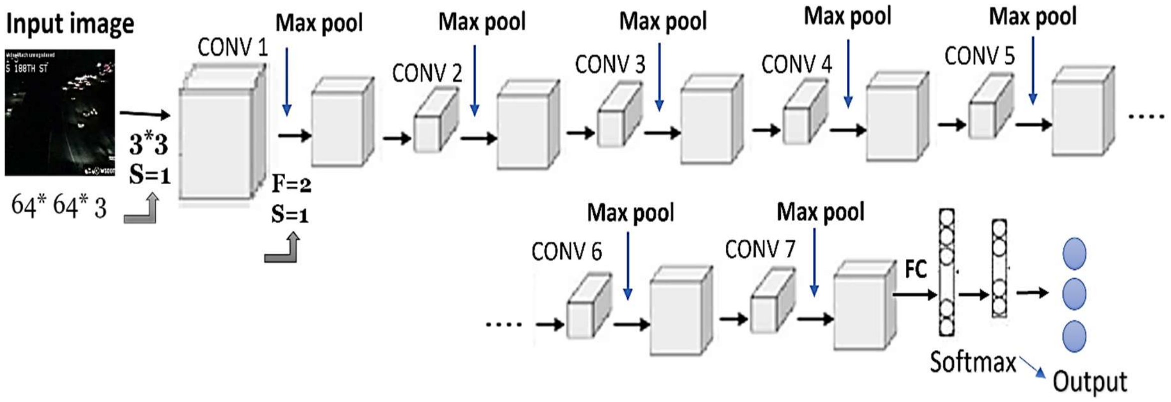

**Fig. 3.** Architecture of the proposed Convolutional Network

The inputs of the network are images extracted from videos provided by the UCSD dataset. Each video contains 40-50 frames at a 320x240 resolution. These frames are cropped and then downsized to get photos of size 64x64 pixels, to avoid memory allocation issues during model training. These photos are fed into the model at the first 64x3x3 convolution layer, which is followed by a max pooling layer of size 2x2. The network is then stretched six times. To prevent overfitting [22], a 0.25 probability dropout layer [23] is included before the final max pooling layer. Softmax [24] is employed as an activation function throughout the model. Table 1 shows the DCNN model architecture.

**Table. 1.** DCNN custom model architecture used

| Layer | Kernel | Stride | Output shape |
|---|---|---|---|
| Input | | | [64, 64, 3] |
| Convolution2D | 3x3 | | [62,62, 64] |
| MaxPooling | 2x2 | 1 | [62,62, 64] |
| Convolution2D | 3x3 | | [60,60, 64] |
| MaxPooling | 2x2 | 1 | [60, 60, 64] |
| Convolution2D | 3x3 | | [58, 58, 64] |
| MaxPooling | 2x2 | 1 | [58, 58, 64] |
| Convolution2D | 3x3 | | [56, 56, 64] |
| MaxPooling | 2x2 | 1 | [56, 56, 64] |
| Convolution2D | 3x3 | | [54, 54, 64] |
| MaxPooling | 2x2 | 1 | [54, 54, 64] |
| Convolution2D | 3x3 | | [52, 52, 64] |
| MaxPooling | 2x2 | 1 | [52, 52, 64] |
| Convolution2D | 3x3 | | [50, 50, 64] |
| Dropout | | | [50, 50, 64] |
| Maxpooling | 2x2 | 1 | [50, 50, 64] |
| Flatten | | | 160000 |
| Dense | | | 3 |

### *4.2 The Method Based on an Existing CNN*

In order to compare our proposed CNN architecture with a state of art method we have tested the CNN architecture proposed in [16] with same UCSD dataset and the same parameters configuration. As shown in Figure 4, this architecture consists of: two convolutional layers, a max pooling layer, and a fully connected layer.

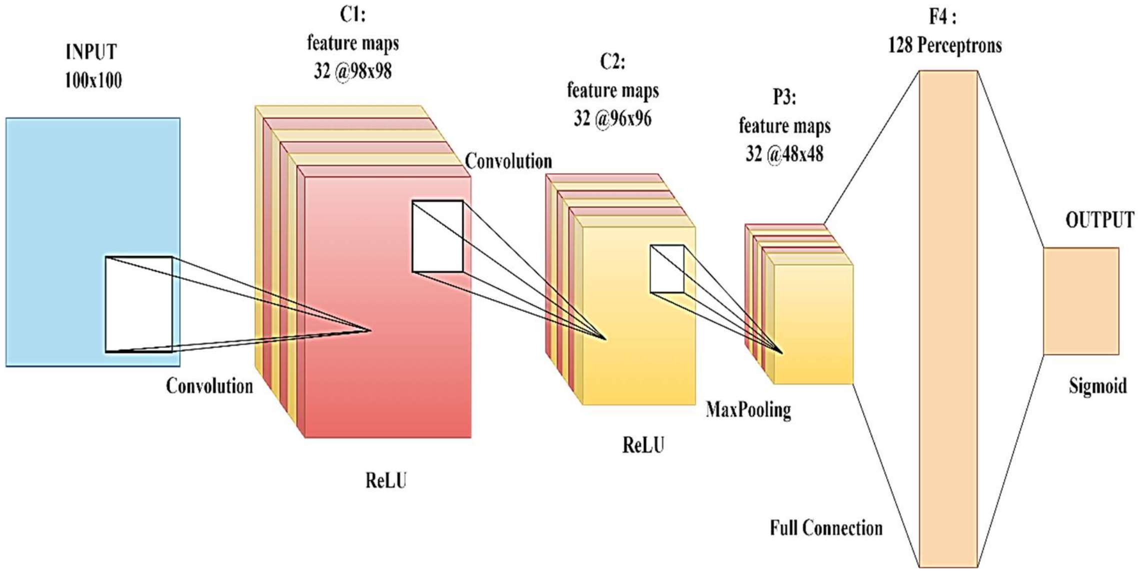

**Fig. 4.** Flowchart of the CNN architecture proposed in [16].

The initial layer, denoted as C1, is a convolutional layer employing 3x3 filters and generating 32 feature maps. Given the 100x100 input size, each feature map has a 98x98 dimension. Subsequently, the second layer, denoted C2, is another convolutional layer utilizing 3x3 filters and producing 32 feature maps, each with a 93x96dimension. The third layer, referred to as P3, is a 2x2 max pooling layer employed for down-sampling; it reduces the size of each feature map to 48x48. The last hidden layer is a fully connected layer comprising 128 perceptron. Each perceptron is fully connected to every unit within the feature maps from P3. The Rectified Linear Units (ReLU) [25] activation function is applied in both the convolutional and fully connected layers. In the output layer, a single perceptron is utilized with a sigmoid activation function.

## 5. EXPERIMENTAL EVALUATION OF THE TWO METHODS

### *5.1. Dataset Preparation*

The UCSD dataset contains 19 Nighttime highway traffic videos recorded by a stationary camera, under different weather conditions: clear, rainy, and overcast. Each video has a duration of about five seconds and contains between 45 and 50 frames. The datasetcomprises diverse traffic categories: light, medium, and heavy. Hand-labeled ground truth was provided; it assigns one of these three categories to each video sequence. In total, the 19 videos produced a set of 1112 frames for all traffic categories, which was used to train and test the network. Some of these frames are shown in Figure 5.

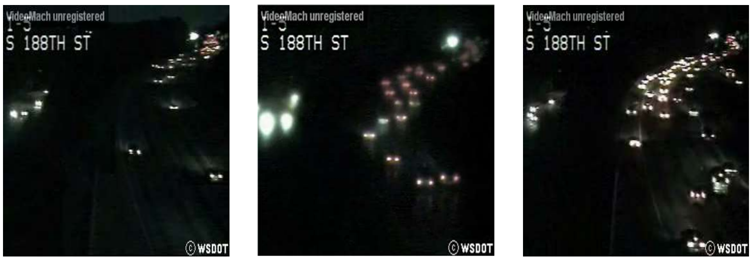

**Fig. 5.** Example frames from the nighttime UCSD dataset

All experimental results were averaged over four trials. In each trial, the dataset was split differently and randomly into 2 sub-sets, containing, respectively, 75% and 25% of the 1112 frames. The first sub-set was used for training and cross-validation [26], and the second one was reserved for testing. To avoid overfitting, dropout regularization was combined with the data augmentation technique [27].

Overfitting is a fundamental problem in supervised machine learning; it limits the ability to perfectly generalize models and fit observed data during training, as well as unseen data in the test set. Overfitting occurs due to the presence of noise, the limited size of the training set, and the complexity of classifiers. A variety of algorithms have been proposed to reduce the effect of overfitting, like dropout and data augmentation where basic image transformations, such as rotation, flipping, and cropping, are applied. Most of these techniques manipulate the images directly and are easy to implement.

To guarantee data augmentation, images were rotated by 30 degrees, flipped horizontally and vertically, with horizontal flipping including flipping the image along the vertical axis (left to right) and vertical flipping involving flipping the image along the horizontal axis (top to bottom). Rotation and flipping are two transformations that let the model generalize to objects in different orientations. Zooming in entails magnifying a section of the image, enlarging the specified location. This can help the model become more resistant to scale changes and increase its ability to distinguish objects of varying sizes. These changes enable for the creation of a more diversified training dataset. This is advantageous because it subjects the model to a broader range of changes that it may meet in real-world circumstances, making it more resilient and less prone to overfitting.

If $M$ distinct augmentation procedures are applied to each of the $N$ original images, each with a unique set of parameter parameters, then the total number of enhanced images can be calculated by the following formula:

$$T = N\ x\ M \tag{2}$$

In our approach for each original image, we generate a total of 7 augmented images including the original image. With 834 original training images, the total number of augmented images would be:

*834 original training images × 7 augmentations per image = 5838 augmented images*

### *5.2. Training process in the first method*

On a Google Colab GPU, the model training took ten minutes. We employed mini-batch gradient descent with Adam [28] optimization with a learning rate of 0.001 to train our CNN model using Python and the Keras library [29], which is built on top of the Theano library [30]. Our implemented CNN model utilizes a batch of size 64 and 100 epochs, for the mini-batch gradient descent. This requires 13 iterations to complete an epoch. For each iteration, a batch of 64 images was submitted to the CNN model, and the weights were then changed through backpropagation. The categorical cross entropy [31] was used as the loss function to adjust the model weights during training. It is defined by:

$$L_{ce} = -\sum t_i \log(P_i)\,, \tag{3}$$

where $t_i$ is the truth label and $P_i$ is the softmax probability of the $i^{th}$ class.

The aim is to minimize the loss, i.e, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0. The loss and accuracy graphs presented in Figure 6 show that the training was successful. Indeed 100% of accuracy was reached during training.

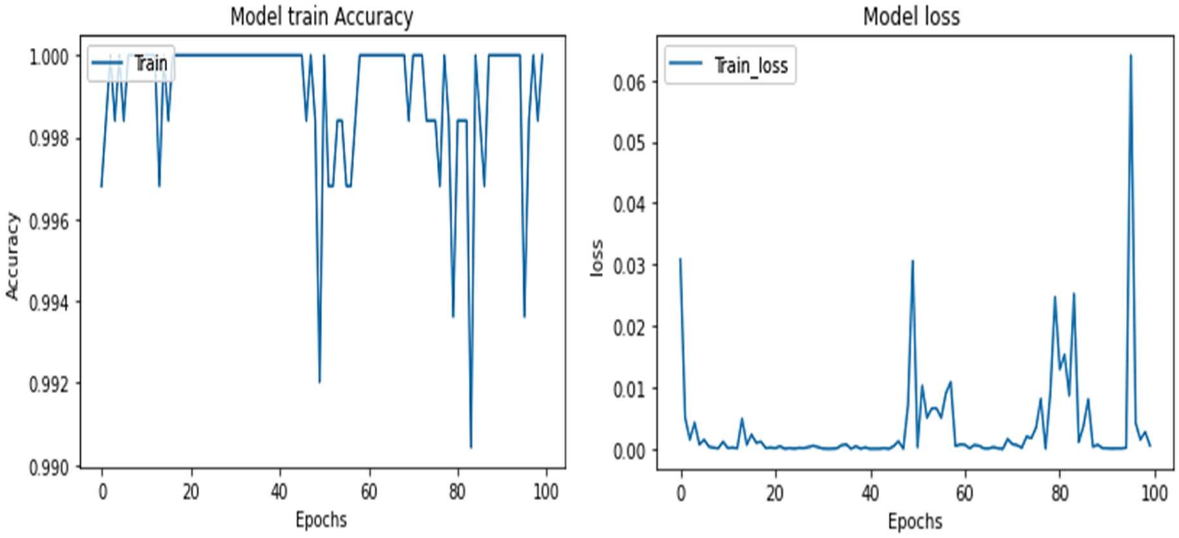

**Fig. 6.** Training model loss and accuracy versus epoch (first method).

### *5.3. Training process in the second method*

The CNN model was trained using the mini-batch gradient descent algorithm with a batch size of 250, and the training process encompassed 100 epochs. As a result, it required 3 iterations to complete one epoch. A batch of 250 images was presented to the CNN model and then the weights were updated by backpropagation. For multiclass classification, categorical cross entropy is used for the objective function without data augmentation. Figure 7 presents the results of the training and validation process.

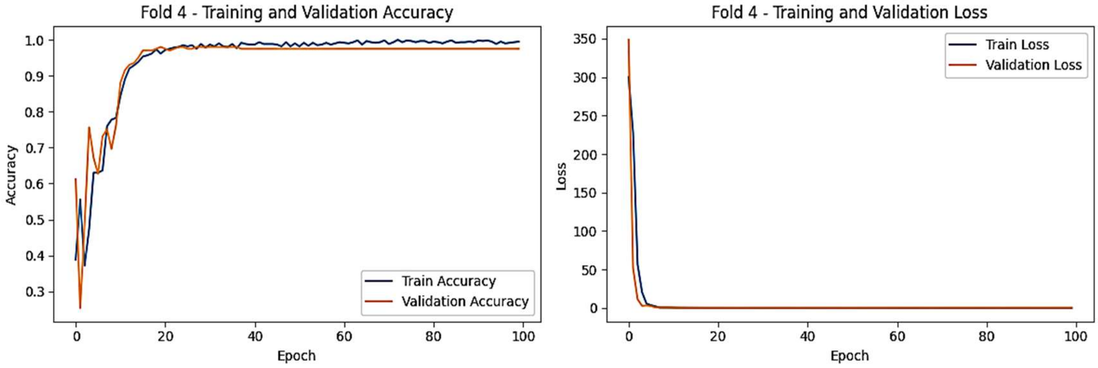

**Fig. 7.** Training model loss and accuracy versus epoch (second method).

### *5.4. Obtained results*

Our proposed custom DCNN model-based method is compared in Table 2 with the method proposed in [15] and the second method based on the CNN used in in [16]. The method [15] is the only published method that was tested with the UCSD nighttime dataset. The method [14] is not included in the comparison, since only accuracy results with the daytime UCSD dataset were presented therein. From Table 2, it can be seen that with an outstanding accuracy of 98.91%, our proposed outperforms the two other methods. In terms of speed, our method is faster than the method in [15] but slower than the method in [16].

**Table. 2.** Comparison between the proposed method and the motion vector method.

| | Motion vectors method [15] | CNN [16] | Proposed method |
|---|---|---|---|
| Accuracy | 89.47% | 96.18% | **98.91%** |
| Speed | 127.23 sec | 53 sec | **95 sec** |

The better accuracy achieved by our proposed custom model, compared to the model proposed in [16] may be explained by the fact that our model is deeper. More convolutional layers and MaxPooling layers result in a complex architecture that allows capturing and extracting more complex hierarchical features in the data. On the other hand, deeper layers with a greater number of kernels enable the network to grasp higher-level abstractions. For comparison, our model uses 64 kernels per each layer instead, which is twice the number of layers used in the model of [16]. Moreover, for a multi-class classification task, softmax activation in the last layer is more appropriate and performs better than sigmoid activation. The latter generates pseudo probabilities that do not necessarily sum up to 1. Conversely, softmax guarantees that the probabilities sum is 1.

For a further comparison between the two methods, we present in Figures 8 and 9 the confusion matrices of the congestion classification obtained with these two methods. As shown from Figure 8, there were only 3 missclassifications by the first method, which occurred between the light and the medium neighboring classes, denoted 0 and 1, respectively. Comparatively, 11 missclassifications occurred by the second method, among which 8 between neighboring classes and 3 between distant classes (light and heavy).

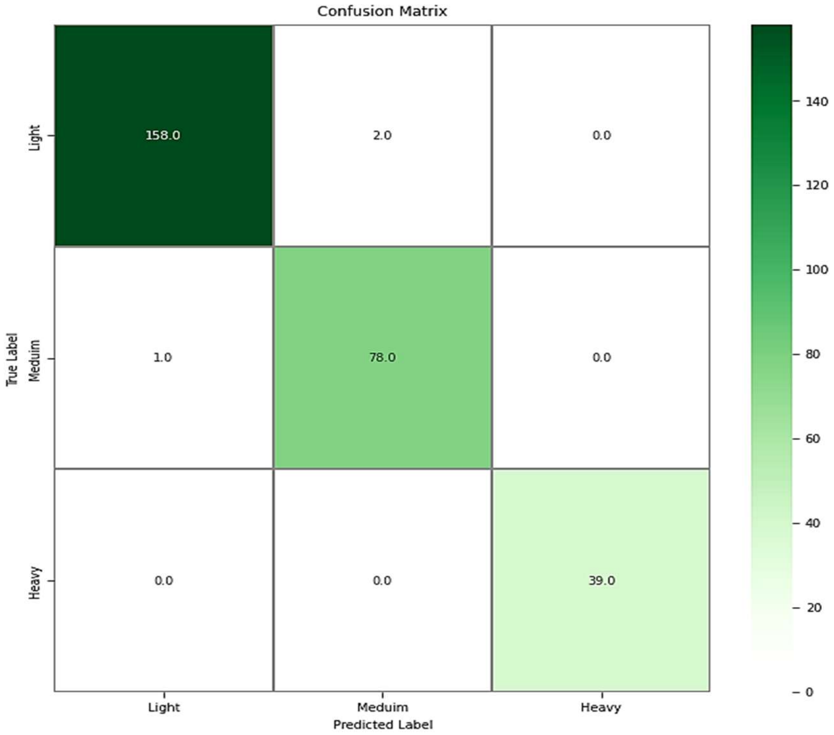

**Fig. 8.** Confusion matrix of congestion classification with the proposed method, using the UCSD dataset.

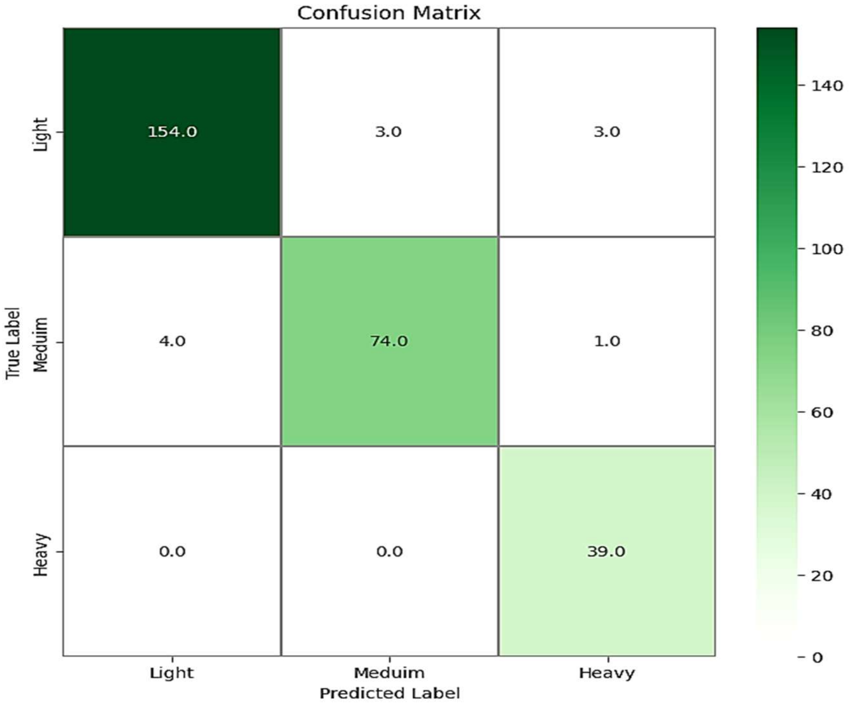

**Fig. 9.** Confusion matrix of congestion classification with the method proposed in [16], using the UCSD dataset.

## 6. CONCLUSION

This research proposes a novel method for effectively classifying traffic congestion in surveillance films captured during nighttime. The proposed method employs a deep-learning neural network, which enables accurate identification of traffic congestion status under various weather conditions. The outcome of the performance assessment of this method using the UCSD nighttime dataset is an outstanding classification accuracy of 98.91%, which surpasses the previously known highest accuracy of 89.47% achieved by the method presented in [15] that also utilized the same dataset. Furthermore, our approach outperforms, in terms of accuracy, the method based on the CNN proposed in [16], trained and tested by the same UCSD nighttime dataset.

Incorporation this method into a traffic monitoring system may help to assist drivers in avoiding traffic congestion and reaping the associated benefits. In our future work, we plan to further enhance our model's classification accuracy by refining the architecture of the convolutional neural network.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Antoni. B. Chan from the City University of Hong Kong for providing the UCSD night-time video traffic dataset.

## REFERENCES

1. Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. & Wang, Y. (2003). Review of road traffic control strategies, *Proceedings of the IEEE*, **91**(12), 2043–2067, https://doi.org/10.1109/JPROC.2003.819610
2. Asmaa, O., Mokhtar, K. & Abdelaziz, O. (2013). Road traffic density estimation using microscopic and macroscopic parameters, *Image and Vision Computing*, **31**(11), 887–894, https://doi.org/10.1016/j.imavis.2013.09.006

3. Asmaa, O., Mokhtar, K. & Abdelaziz, O. (2013, April). Road traffic congestion estimation with macroscopic parameters, *Proc. of 2013 11th International Symposium on Programming and Systems (ISPS)* (USTHB, Algeria), 24–29.
4. Hsia, C. H., Kong, Y., Lin, Y. K. & Chien, Y. R. (2017, June). Real-time vision system for nighttime vehicle detection, *Proc. of 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)* (Taiwan, China), 301–302.
5. Sevekar, P. & Dhonde, S. B. (2017). Night-time vehicle detection for automatic headlight beam control, *International Journal of Computer Applications*, **157**(7), 8–12, https://doi.org/10.5120/ijca2017912737
6. Chen, Y. L., Wu, B. F., Huang, H. Y. & Fan, C. J. (2010). A real-time vision system for nighttime vehicle detection and traffic surveillance, *IEEE Transactions on Industrial Electronics*, **58**(5), 2030–2044, https://doi.org/10.1109/TIE.2010.2055771
7. Dodge, D. & Yilmaz, M. (2022). Convex Vision-based Negative Obstacle Detection Framework for Autonomous Vehicles, *IEEE Transactions on Intelligent Vehicles*, **8**(1), 778–789, https://doi.org/10.1109/TIV.2022.3146877
8. Sayed, M. S. & Delva, J. (2010, November). Low complexity contrast enhancement algorithm for nighttime visual surveillance, *Proc. of 2010 10th International Conference on Intelligent Systems Design and Applications* (Cairo, Egypt), 835–838, https://doi.org/10.1109/ISDA.2010.5687158
9. Cai, Y., Huang, K., Tan, T. & Wang, Y. (2006, August). Context enhancement of nighttime surveillance by image fusion. *Proc. of 18th International Conference on Pattern Recognition (ICPR'06)* (Hong Kong), 980–983.
10. Liu, Y. & Payeur, P. (2003). Robust image-based detection of activity for traffic control, *Canadian Journal of Electrical and Computer Engineering*, **28**(2), 63–67, https://doi.org/10.1109/CJECE.2003.1532510
11. Cucchiara, R., Piccardi, M. & Mello, P. (2000). Image analysis and rule-based reasoning for a traffic monitoring system, *IEEE transactions on intelligent transportation systems*, **1**(2), 119–130, https://doi.org/10.1109/6979.880969
12. Yoneyama, A., Yeh, C. H. & Kuo, C. C. J. (2005). Robust vehicle and traffic information extraction for highway surveillance, *EURASIP Journal on Advances in Signal Processing*, **2005**(14), 1–17, https://doi.org/10.1155/ASP.2005.2305
13. Wang, G., Xiao, D. & Gu, J. (2008, September). Review on vehicle detection based on video for traffic surveillance, *Proc. of 2008 IEEE international conference on automation and logistics* (Chindao, China), 2961–2966.
14. Chan, A. B. & Vasconcelos, N. (2005). Classification and retrieval of traffic video using auto-regressive stochastic processes, *Proc. of IEEE Proceedings. Intelligent Vehicles Symposium, 2005*, (Las Vegas, NV), 771–776.
15. Asmaa, O., Mokhtar, K. & Abdelaziz, O. (2013). Road traffic density estimation using microscopic and macroscopic parameters, *Image and Vision Computing*, **31**(11), 887–894, https://doi.org/10.1016/j.imavis.2013.09.006
16. Kurniawan, J., Syahra, S. G. & Dewa, C. K. (2018). Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network, *Procedia computer science*, **144**, 291–297.
17. Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects, *IEEE transactions on neural networks and learning systems*, **33**(12), 6999–7019.
18. Albawi, S., Mohammed, T. A. & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *Proc. of 2017 international conference on engineering and technology (ICET)* (Antalya, Turkey), 1–6.
19. Gholamalinezhad, H. & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. *arXiv*: 2009.07485, [Online]. Available: https://arxiv.org/abs/2009.07485

20. Basha, S. S., Dubey, S. R., Pulabaigari, V. & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification, *Neurocomputing*, **378**, 112–119.
21. Sharma, S., Sharma, S. & Athaiya, A. (2017). Activation functions in neural networks, *Towards Data Sci*, **6**(12), 310–316.
22. Dietterich, T. (1995). Overfitting and undercomputing in machine learning, *ACM computing surveys (CSUR)*, **27**(3), 326–327.
23. Park, S. & Kwak, N. (2017). Analysis on the dropout effect in convolutional neural networks, *Proc. of Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision* (Taipei, Taiwan), 189–204.
24. Banerjee, K., Gupta, R. R., Vyas, K. & Mishra, B. (2020). Exploring alternatives to softmax function. *arXiv*: 2011.11538, [Online]. Available: https://arxiv.org/abs/2011.11538
25. Dubey, A. K. & Jain, V. (2019). Comparative study of convolution neural network's relu and leaky-relu activation functions, Proc. of MARC 2018 (Chicago, IL), 873–880.
26. Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv*: 1811.12808, [Online]. Available: https://arxiv.org/abs/1811.12808
27. Ahmed, E., Jones, M. & Marks, T. K. (2015). An improved deep learning architecture for person re-identification, *Proc. of the IEEE conference on computer vision and pattern recognition* (Boston, MA), 3908–3916.
28. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. DOI : https://doi.org/10.48550/arxiv.1412.6980
29. François, C. (2017). *Keras. GitHub repository*, [Online]. Available: https://github.com/fchollet/keras.
30. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., et. al. (2010). Theano: A CPU and GPU math compiler in Python, *Proc. of 9th Python in Science Conf* (Austin, TX), 3–10.
31. Zhang, Z. & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels, *arXiv*: 1805.07836, [Online]. Available: https://arxiv.org/abs/1805.07836