SVU-International Journal of Engineering Sciences and Applications (2022) 3(1): 91-97
Print ISSN 2785-9967 | Online ISSN 2735-4571
DOI 10.21608/svusrc.2022.133083.1046

Robust Traffic Congestion Recognition in Videos Based on Deep Multi-Stream LSTM

Mohamed A. Abdelwahab
(https://orcid.org/0000-0001-8575-6648)

Abstract Cities with high population density have a serious problem with traffic congestion. Intelligent transportation systems try to overcome these problems by finding smart ways to detect traffic congestion. One of the essential issues in these systems is selecting the appropriate features to detect traffic congestion. Most of the current methods utilize motion or texture features only, which have their limitations. In this paper, a deep neural network (DNN), which has two input paths, is proposed for traffic congestion recognition. It handles the evolution of motion as well as texture through its two inputs simultaneously via Long Short-Term Memory (LSTM) layers. Gaussian noise layers are used to increase the generalization ability of the DNN and to enable training on small datasets without over-fitting. Experimental results applied to the UCSD and NU videos datasets assert the robustness of the proposed method. It achieves an accuracy of 98 % which is high in comparison to the state-of-the-art methods.

Keywords: Traffic congestion; LSTM; Multi-Stream network.

1 Introduction

Congestion on the roads is a major trouble in crowded cities, it has bad environmental and ecological effects, and it causes a lot of time delays. **Figure 1** gives samples of diverse levels of vehicles congestion; light, medium, and heavy, where the latter has the highest level of congestion. The vision-based methods for congestion classification have become more reliable due to the advancement of the artificial intelligence and deep learning approaches. Recently, many methods were introduced as smart ways for traffic congestion classification. According to the used features, these methods fall into the following categories: motion-based [1] and texture-based methods [2], [3]. However, relying on motion features [1] solely could result in misclassifications between heavy classes that have completely stopped vehicles and light classes that have empty roads. On the other hand, based on texture features [2], [3] solely, the congestion classifier may not be able to differentiate between two scenes with the same number of vehicles but different motions. In this research, we propose a robust deep network that has two inputs to handle both texture and motion features simultaneously. The proposed method overcomes the shortcomings of the current methods which depend only on one feature type (motion or texture). In summary, this paper has the following contributions:

- We introduce a deep network that utilizes both motion and texture features for overcoming the limitations of current traffic congestion classification methods
- The proposed network can be used for training a small dataset due to the use of Gaussian noise layers.
- The proposed results outperform the ones of the recent methods with a classification accuracy of 98%.

The rest of this paper is structured as follows. Section 2 discusses the related works. Section 3 gives a short explanation of the LSTM, Section 4 explains the proposed

Received: 13 April 2022/ Accepted: 2 May 2022
Corresponding Author: Mohamed A. Abdelwahab, abdelwahab@aswu.edu.eg
Electrical Department, Faculty of Energy Engineering, Aswan University, Egypt.

(a) Light (b) Medium (c) Heavy

Fig. 1 Examples of various levels of vehicles congestion, images are taken from the NU dataset [4].

method, a summary of the experimental results is presented in Section 5, and Section 6 gives the conclusion.

2 Related works

Classification of traffic congestion has been studied for several decades. We can categorize these studies, according to their applications, into vehicle-based methods and holistic methods. The idea behind vehicle-based methods is to analyze the movements of individual vehicles, while the holistic methods aim to extract global features that represent the whole traffic scene. According to the type of the extracted feature, the holistic methods are divided into motion-based and texture-based methods. We will summarize each of these categories and discuss their advantages and disadvantages.

In the vehicle-based methods, trajectories of individual vehicles are extracted, then they are used to classify the traffic scene into different congestion levels such as (light, medium, or heavy) [5]-[7]. For instance, several adaptive thresholding approaches were employed in [5] to locate moving vehicles based on the differences between frames. Then, a neural network was trained on wavelet coefficients extracted from the detected vehicles, after which vehicles trajectories were obtained. Mo *et al.* [6] used the codebook strategy in detection vehicles, where scale-invariant feature transforms (SIFTs) were employed as a feature for creating the codebook. Huang *et al.* [7] tried to decrease the effect of occlusions by assigning different features for vehicle detection according to the level of occlusion. In low occlusion, they used local features, while they used color features besides local ones in occlusion areas. Other approaches were interested to estimate vehicle counting per unit time as an indicator for traffic congestion. For example, Yang *et al.* [8] separated vehicles from the background using a low-rank decomposition technique and then they employed the Kalman filter to track and count vehicles. To decrease the processing time, Abdelwahab [9] detected and counted vehicles in a small area of the road without the demand for tracking. Recently, because of the efficiency of Deep neural networks (DNNs), many methods had utilized them in vehicle counting [10]-[12], where DNNs were used to efficiently detect vehicles and then KLT tracker was applied in the tracking stage.

However, if vehicle scenes are crowded or weather conditions are poor, the above-mentioned vehicle-based methods may perform poorly for traffic congestion classifications. On the other hand, holistic-based methods tried to overcome these problems by extracting a global feature rather than tracking individual vehicles. These techniques can be divided into texture-based or motion-based methods according to the type of features. In the motion-based methods, only motion features are used for congestion classification. For instance, Asmaa et al. [13] compared between two different motion features; microscopic and macroscopic. The microscopic features depended on tracking individual vehicles, while the macroscopic features were created from tracking small blocks in the traffic video. They found that the macroscopic features gave a higher accuracy than the microscopic features, but they need more processing time. Riaz *et al.* [1] tracked scattered feature points in traffic scenes to extract motion vectors (MVs). From these MVs, four values were estimated representing the average velocity, the number of MVs, velocity's standard deviation, and the average length of MVs. A global motion feature was created by concatenating these four values.

In the texture-based methods, different textures can be extracted from the traffic scenes to classify the congestion level. Luo *et al.* [2] proposed codebook-based and CNN-based approaches to classify traffic scenes. In the codebook-based approach, they employed SIFT features to build codebook descriptors. In the CNN-based method, they exploited the last layers of various CNNs to create texture features. In [3] Luo *et al.* extended their work by introducing different regression techniques to represent the congestion level. The main idea of these techniques is to use pixel or patch segmentation to determine vehicle and road areas in each traffic scene after excluding the

background areas. Then, the ratio of vehicles occupied areas to those of roads is estimated as a measure of the traffic congestion. However, one of the shortcomings of the Luo's approaches [3] is that they have a high computation complexity. In addition, these approaches were mainly designed for classifying traffic congestion in still images without taking into consideration the motion data in traffic videos. In [14], two approaches were introduced to generate a dynamic image for a batch of frames; temporal pooling [15] and optical flow aggregation [16]. Then a deep residual network was utilized for extracting texture features from the created dynamic images.

As discussed above, most of the existing methods utilized motion or texture features for classifying traffic congestion. However, depending solely on one feature has its shortcoming. In our previous work [17], a congestion classification was achieved by aggregating the output from separate motion-based and texture-based classifiers. Motion features were generated by averaging motion trajectories every batch of frames, while the texture features were obtained by generating compact texture vectors using the learning-to-rank (LTR) technique [18]. However, in the proposed method we introduce one deep neural network that can acquire both motion and texture features simultaneously and produce one of the traffic congestion classes at the output (Light, Medium, and Heavy).

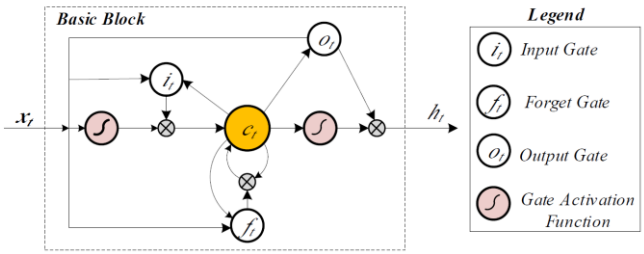


Fig. 2 The block diagram of LSTM

3 Long Short-Term Memory (LSTM)

In the proposed method, Long Short-Term Memory (LSTM) is used to learn the evolution of motion and texture features of the traffic videos. LSTM was proposed by Hochreiter *et al.* [19] as a way of avoiding the long-term dependency issue in recurrent neural networks (RNNs). LSTM cell has three gates; forget, input, and output gates which determine the output and the internal state. **Figure 2** gives the main element components of an LSTM cell. The activation units trigger the gates depending on a time-series input. In the LSTM training phase, the weights of each gate are learned. LSTM can memorize recent steps using the three gates. Assume that at time step t, an LSTM cell has a state c_t. This state is updated using input gate i_t, forget gate f_t, and output gate o_t. Inputs at a moment t are represented by two sources, the present input x_t and the old hidden state h_{t-1}.

4 The Proposed Approach

In this approach, the evolution of both motion and texture features through video frames is considered using dual paths in a deep network. **Figure 3** shows the proposed network, input video frames are processed through the network as follows:

4.1 Computing optical flow

Dense optical flow is estimated from the input video frames employing the Gunnar Farnebäck algorithm [20], where it is estimated based on polynomial expansion. For better representation, magnitudes and angles of the optical flow are used to construct HSV (Hue, Saturation, Value) images [21]. The Hue and Value components are set according to the magnitudes and angles of the optical flow, respectively, and the Saturation dimension is kept at the maximum value.

4.2 Deep feature extraction

Optical flow and raw pixel images are fed into the two input paths of the network as shown in **Fig. 3**. In each path, a deep feature vector, with a length of 2048, is extracted from the input image using a pre-trained ResNet101 network [22]. These feature vectors are extracted from the last layer of the network after removing the classification layer. The extracted vectors from the optical images represent motion features while those extracted from the raw images are used as texture features.

4.3 Learning feature evolution

An LSTM layer is used to learn the motion evolution of the input video through one of the network paths, while in the second path, another LSTM is used to learn the texture evolution. Both LSTM layers have a dimension of 16. The outputs of both LSTM layers are concatenated, then they pass through two fully connected (FC) layers with a length of 16 and 32, respectively. A batch normalization (BN) layer [23] is used to fast the network and make it more stable. It normalizes the layer's inputs by re-scaling them. Finally, a classifier layer with a SoftMax activation having a length of 3 is used to produce the output class label.

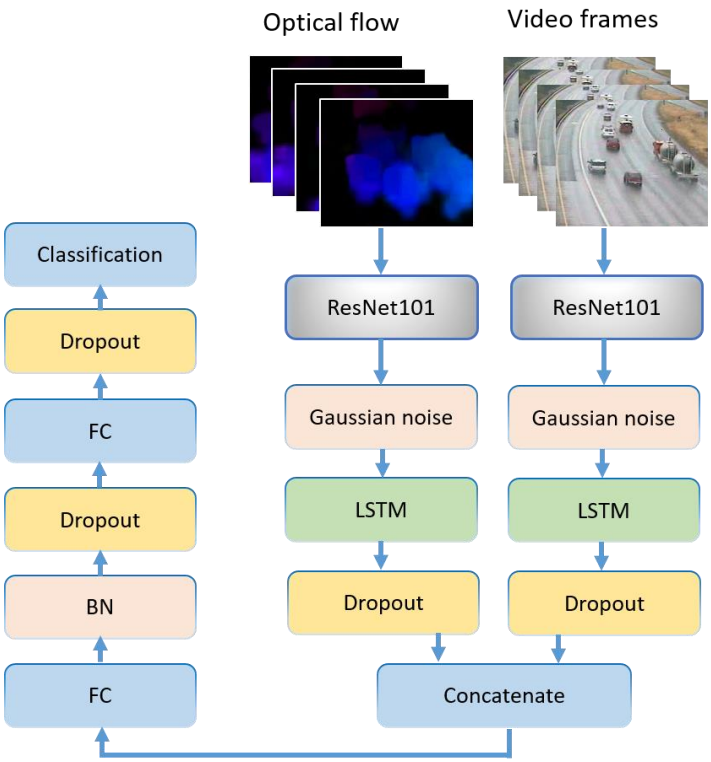


Fig. 3 The proposed network for traffic classification. It has two inputs; one for the input video frames and the other for the optical flow images

4.4 Regularization layers

Regularization of the network using Gaussian noise (GN) is an effective technique to increase its robustness against input variation. The GN layer regularizes CNN by adding adversarial noise to hidden layers. It aids to learn more robust feature representations. Adding attentively noise to the activation of the intermediate layer helps to increase CNN's generalization ability. In the training phase, GN inserts adversarial noise η_t to h_t as in equation 1 and moves $\tilde{h}_t$ to the neighboring layer.

$$\tilde{h}_t = h_t + \eta_t \quad (1)$$

The noise is added in the training phase only. In the detection phase, the noise layers are erased and $\tilde{h}_t$ will become h_t during the inference phase. In the proposed method, a GN layer is used at the beginning of each path. The result is a kind of data augmentation. This gives the proposed network the advantages of training with small dataset videos. Also, the dropout layers are utilized through the network layers to avoid overfitting. They are randomly set some input units to 0. All these regularization layers are used only during the training stage

5 Experimental Results and Analysis

Two experiments were conducted to evaluate the proposed technique. The University of California San Diego (UCSD) [24] and the Nile University (NU) [4] dataset videos were used in the first and the second experiment, respectively. In the two experiments, the network was trained employing the Adam optimizer. The learning rate was initialized to 1×10^{-4} to avoid falling to a local minimum. The batch size and number of epochs were selected to be 64 and 300, respectively. The GN layer added noise with a standard deviation of 0.3 giving the performance of data augmentation. The dropping fraction in the dropout layers was set to 0.3.

5.1 UCSD dataset

In the first experiment, the UCSD dataset [24] was used. It has 254 videos. These videos were taken in different situations such as rainy, overcast, and clear. For accurate comparisons, the same 4-cross validation procedure in [24] was used with the same video indices. The proposed method achieves an accuracy of 98.03 % with only 5 miss-classified videos. **Table 1** gives the confusion matrix.

"cctv..614x00024" Medium | "cctv..617x00071" Medium | "cctv..618x00085" Light

"cctv..615x00034" Heavy | "cctv..615x00036" Heavy | "cctv..615x00044" Heavy

Fig. 4 Examples for true classified videos by the proposed method from the UCSD dataset under different illumination conditions.

Table 1 Confusion matrix of the proposed method for the UCSD dataset

		Predicted		
		Heavy	**Medium**	**Light**
True	**Heavy**	43	1	0
	Medium	3	42	0
	Light	0	1	164

Table 2 Comparison to the state-of-the-art methods using the UCSD dataset

Method	**Accuracy %**
Riaz [1]	95.28
Luo (2015) [2]	96.90
Luo (2018) [3]	97.64
Ribas [25]	96.06
Wang [26]	93.30
Proposed method	98.03

In this matrix, we can observe that all misclassifications occurred between the medium class and the light or heavy classes, and there is no misclassification between the light and heavy classes. **Table 2** compares the proposed accuracy with those of the most recent methods. Our proposed method gives the highest accuracy. **Figure 4** gives examples for successful classified videos under different illumination conditions. Three of miss-classified samples are shown in **Fig. 5**. As seen in this figure, it is hard to classify these videos even by human.

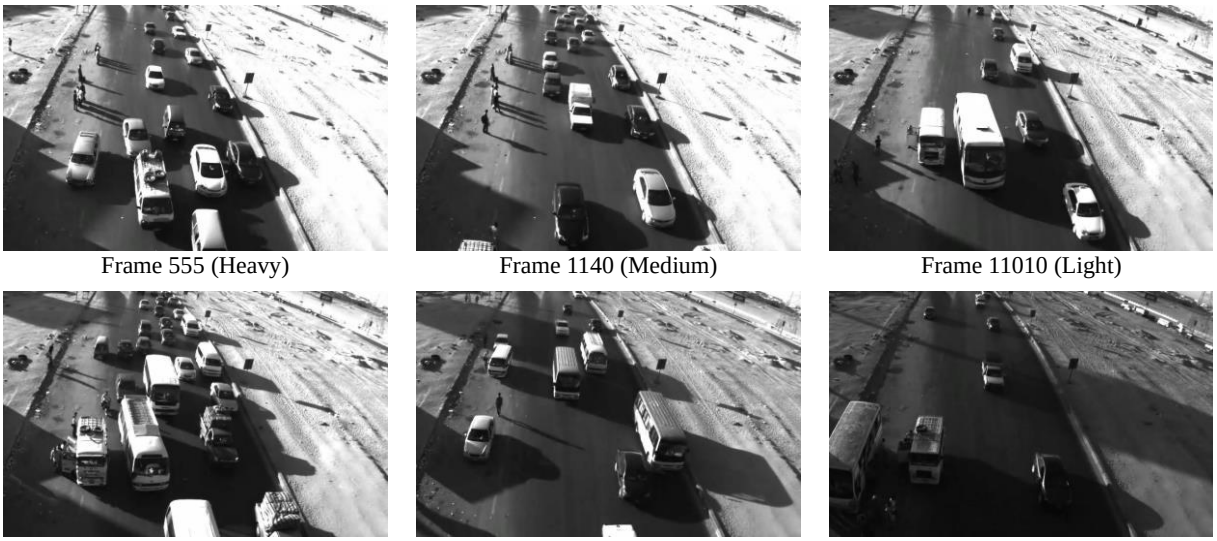
5.2 NU dataset

In the second experiment, we used the NU1 video [4] to simulate a practical classification situation, which was recorded on one of the Egyptian roads. It has a length of 45 minutes, and it contains about 40826 frames. During that video, the traffic congestion is continuously changeable between heavy, medium, and light. One of the challenges in this video is that vehicles do not move in certain road lanes and pedestrians appear many times crossing the road. Due to this chaos, a few numbers of vehicles in a traffic scene move very slowly, however, they may be wrongly classified as a light traffic scene. So, it is not sufficient to classify this video depending only on texture features.

To simulate online traffic monitoring, the classification process was applied every 15 frames. The NU1 video has 2723 batches with 15 frames each. In this experiment, 633 batches were taken and labeled to one of the three classes: heavy (221), medium (208) and light (204) batches. The 633 frame batches were divided randomly into 433 batches for training and 200 batches for testing. **Figure 6** shows result samples for true traffic classification of the NU1 video by the proposed method, where different frames with different classification labels are given. We can see in this figure the shadow variation between the first and end frames, which increases the classification challenges in this video.

"cctv..517x01655" Heavy/Medium | "cctv..616x00052" Medium/Heavy | "cctv..617x00072" Medium/Heavy

Fig. 5 Samples of miss-classified videos. For each video, the real/predicted labels are given.



Frame 555 (Heavy) | Frame 1140 (Medium) | Frame 11010 (Light)

Frame 12900 (Heavy) | Frame 23025 (Medium) | Frame 40470 (Light)

Fig. 6 Samples of the true classification results for the NU1 video. Shadow variation is obvious between the first frame (frame 555) and end frame (frame 40470), which increases the difficulty of classification in this video

Table 3 Classification accuracy comparisons of the NU1 video

Method	Accuracy %
Luo (2015) [2]	93.00
Proposed method	**96.00**

To compare with other methods, the same procedure of the Luo's method [2] was applied to the NU1 video frames with the same training and testing indices. Following the steps mentioned in [2], features were extracted every 5 frames using the pre-trained VGG network, then the classification was performed using an SVM classifier to obtain three decisions for each batch (15 frames). The final decision is taken by voting between the three obtained decisions. A comparison between the results of the proposed method and those of the Luo's method [2] is presented in **Table 3**. The accuracy of the proposed technique is higher than the one of Luo's method by 3 %. This is because both motion and texture features are considered in our technique, while the method in [2] focuses only on texture features.

6 Conclusion

In this research, a robust deep neural network was proposed for traffic congestion classification in videos. It efficiently handles the evolution of both motion and texture features via two separate paths. An LSTM layer was used in each path to learn these evolutions. The outputs of paths are then combined and used for deciding traffic classes. Adding GN layers during the training of the proposed network made it more robust and reduced generalization errors. Compared to the state-of-the-art methods, the proposed method gives higher accuracy. The outperformance of the proposed method becomes clear when it is applied to the challenging NU1 video, where the

classification accuracy is enhanced by 3 % compared to other recent methods. This is because the proposed method handles both motion and texture evolutions while others depend on texture features only.

References

[1] A. Riaz and S. A. Khan, “Traffic congestion classification using motion vector statistical features,” in Sixth International Conference on Machine Vision (ICMV 2013). Dec. 24, 2013, pp. 245-251.

[2] Z. Luo, P.-M. Jodoin, S.-Z. Li, and S.-Z. Su, “Traffic analysis without motion features,” in Image Processing (ICIP), IEEE International Conference on., Sep. 2015, pp. 3290–3294.

[3] Z. Luo, P.-M. Jodoin, S.-Z. Su, S.-Z. Li, and H. Larochelle, “Traffic analytics with low-frame-rate videos,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 4, pp. 878–891, 2018.

[4] S. A. Aly, A. Mamdouh, and M. Abdelwahab, “Vehicles detection and tracking in videos for very crowded scenes.” in The 13th IAPR International Conference on Machine Vision Applications, 2013, pp. 311–314.

[5] J. Kim, C.-W. Lee, K. Lee, T. Yun, and H. Kim, “Wavelet-based vehicle tracking for automatic traffic surveillance,” in Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001 (Cat. No. 01CH37239), vol. 1, 2001, pp. 313–316.

[6] G. Mo and S. Zhang, “Vehicles detection in traffic flow,” in Sixth International Conference on Natural Computation, vol. 2. IEEE, 2010, pp. 751–754.

[7] L. Huang and M. Barth, “Real-time multi-vehicle tracking based on feature detection and color probability model,” in IEEE Intelligent Vehicles Symposium, 2010, pp. 981–986.

[8] H. Yang and S. Qu, “Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low rank decomposition,” IET Intelligent Transport Systems, vol. 12, no. 1, pp. 75–85, 2017.

[9] M. Abdelwahab, “Fast approach for efficient vehicle counting,” Electronics Letters, vol. 55, no. 1, pp. 20–22, 2019.

[10] M. A. Abdelwahab, “Accurate vehicle counting approach based on deep neural networks,” in International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, 2019, pp. 1–5.

[11] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R.-i. Taniguchi, “Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow,” Sensors, vol. 19, no. 20, p. 4588, 2019.

[12] Z. Al-Ariny, M. A. Abdelwahab, M. Fakhry, and E.-S. Hasaneen, “An efficient vehicle counting method using mask r-cnn,” in International Conference on Innovative Trends in Communication and Computer Engineering (ITCE). IEEE, Feb. 2020, pp. 232–237.

[13] O. Asmaa, K. Mokhtar, and O. Abdelaziz, “Road traffic density estimation using microscopic and macroscopic parameters,” Image and Vision Computing, vol. 31, no. 11, pp. 887–894, 2013.

[14] M. A. Abdelwahab, M. Abdel-Nasser, and R.-i. Taniguchi, “Efficient and fast traffic congestion classification based on video dynamics and deep residual network,” in Frontiers of Computer Vision: 26th International Workshop, IW-FCV 2020, Ibusuki, Japan, Revised Selected Papers. Springer, 2020, pp. 3–17.

[15] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, “Rank pooling for action recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 4, pp. 773–787, 2016.

[16] M. Abdelwahab and M. Abdelwahab, “Human action recognition and analysis algorithm for fixed and moving cameras,” Electronics Letters, vol. 51, no. 23, pp. 1869–1871, 2015.

[17] M. A. Abdelwahab, M. Abdel-Nasser, and M. Hori, “Reliable and rapid traffic congestion detection approach based on deep residual learning and motion trajectories,” IEEE Access, vol. 8, pp. 182180–182192, 2020.

[18] B. Fernando, E. Gavves, D. Muselet, and T. Tuytelaars, “Learning to rank based on subsequences,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2785–2793.

[19] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] G. Farneb¨ack, “Two-frame motion estimation based on polynomial expansion,” in Scandinavian conference on Image analysis. Springer, 2003, pp. 363–370.

[21] D. Archana and S. Sanjeevani, “Moving object detection using optical flow and hsv,” in Evolution in Signal Processing and Telecommunication Networks. Springer, 2022, pp. 49–55.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[23] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in international conference on machine learning. PMLR, 2015, pp. 448–456.

[24] A. B. Chan and N. Vasconcelos, “Classification and retrieval of traffic video using auto-regressive stochastic processes,” in Intelligent Vehicles Symposium, Proceedings, IEEE, 2005, pp. 771–776.

[25] L. C. Ribas, W. N. Goncalves, and O. M. Bruno, “Dynamic texture analysis with diffusion in networks,” Digital Signal Processing, vol. 92, pp. 109-126, 2019.

[26] Y. Wang, L. Wang, D. Kong, and B. Yin, “Extrinsic least squares regression with closed-form solution on product grassmann manifold for video-based recognition,” Mathematical Problems in Engineering, vol. 2018, no. 1, pp. 1–7, 2018.