Raf. J. of Comp. & Math's., Vol. 15, No. 2, 2021
DOI:

Embedded Descriptor Generation in Faster R-CNN for Multi-Object Tracking

Younis A. Younis
younis.bayati@uomosul.edu.iq
Department of Computer Science
College of Education for Pure Science,
University of Mosul, Mosul, Iraq

Khalil I. Alsaif
khalil_alsaif@uomosul.edu.iq
Department of Computer Science
College of Computer Science and Mathematics
University of Mosul, Mosul, Iraq

Received on: 07/01//2021 **Accepted on: 08/03/2021**

ABSTRACT

With the rapid growth of computer usage to extract the required knowledge from a huge amount of information, such as a video file, significant attention has been brought towards multi-object detection and tracking. Artificial Neural Networks (ANNs) have shown outstanding performance in multi-object detection, especially the Faster R-CNN network. In this study, a new method is proposed for multi-object tracking based on descriptors generated by a neural network that is embedded in the Faster R-CNN. This embedding allows the proposed method to directly output a descriptor for each object detected by the Faster R-CNN, based on the features detected by the Faster R-CNN to detect the object. The use of these features allows the proposed method to output accurate values rapidly, as these features are already computed for the detection and have been able to provide outstanding performance in the detection stage. The descriptors that are collected from the proposed method are then clustered into a number of clusters equal to the number of objects detected in the first frame of the video. Then, for further frames, the number of clusters is increased until the distance between the centroid of the newly created cluster and the nearest centroid is less than the average distance among the centroids. Newly added clusters are considered for new objects, whereas older ones are kept in case the object reappears in the video. The proposed method is evaluated using the UA-DETRAC (University at Albany Detection and Tracking) dataset and has been able to achieve 64.8% MOTA and 83.6% MOTP, with a processing speed of 127.3 frames per second.

Keywords: Convolutional Neural Networks; Multi-Object Detection; Multi-Object Tracking.

Introduction

With the rapidly growing use of computers to automate different types of applications, significant attention has been brought to object detection and tracking techniques, according to the enormous numbers of digital videos being captured daily and the huge amount of information they contain. These techniques rely on analyzing the visual features in the images and compare them against the known objects that are predefined or used in the training of the detection and tracking method. However, according to the huge variations that may exist in the shape of a certain object, e.g., cars and humans, it is important to neglect these variations, so that, the objects can be recognized accurately [1, 2].

According to their ability to recognize inter- and intra-class variations, Artificial Neural Networks (ANNs) are being widely used to detect different objects in images and videos. The good performance of these networks in such an application is according

to their ability to emphasize inter-class features and neglect intra-class ones during the training, by evaluating and updating the influence of each detected feature on the loss value. Moreover, according to the hierarchy of artificial neural networks, shown in Figure 1, the outputs of the neurons at a certain layer depend on the outputs of the neurons in the previous ones. Thus, to produce consistent and accurate outputs, the values outputted by the neurons get more similar to each other for similar outputs, especially in deeper layers [3, 4]. However, the primitive features that are learned by the neural network at the layers closer to the input layer are of significant importance, according to the slow updates that occur to these layers as the updates backpropagate from the output layer towards the input layer. The importance of these features is illustrated by the use of transfer learning, in which a pre-trained neural network is fine-tuned and used in another application to accelerate the training by exploiting the features that the neural network has already learned [5, 6].

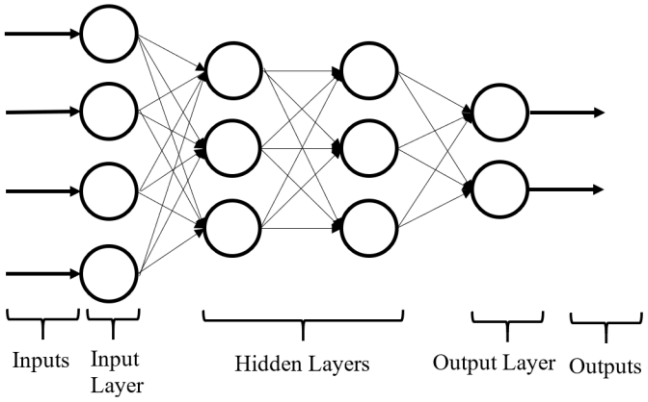

Figure 1: Architecture of an artificial neural network.

Motivated by the importance of the features that are learned by a neural network when trained for similar applications and the need for descriptors that can describe the objects detected by these networks, a new method is proposed in this study to provide descriptors for different objects that are detected by the neural network. The proposed method joins the features detected at the layers closer to the input layer with deeper features, so that, the produced descriptor can be used to distinguish the different objects of the same type. Then, these descriptors can be used to track the movement of the detected object, by predicting whether this object has been seen in a previous frame or not. Accordingly, the computational overhead imposed by the proposed method to generate these descriptors is significantly low, as the features that are used to generate the descriptors are already detected by the Faster R-CNN. Moreover, according to the high performance achieved by the Faster R-CNN, based on these features, the high-quality of these features is also exploited in the proposed method, so that, the descriptors that are generated rapidly are also accurate. The remainder of the paper is organized as: Section Two summarizes the method used in the proposed method and how they are employed to achieve the required methodology. Section Three describes the experimental setup and results acquired from the experiments conducted to evaluate the proposed method and compare it to the existing state-of-the-art methods. Section Four summarizes the conclusions of the study and suggestions for future work.

Materials and Method

Object Detection Using Faster R-CNN

The Faster R-CNN object detection method has been able to significantly reduce the time required by the original R-CNN and fast R-CNN to detect the object while maintaining the same accuracy. As shown in Figure 2, this neural network relies on sharing the detected features between two separate networks, each is designated for a certain task. The first set of layers is the Region Proposal Network (RPN), whereas the second set is the Fast R-CNN. Both of these networks rely on the same features outputted at the "feature map". The regions proposed by the RPN are appended to the feature map and forwarded to the Fast R-CNN, which outputs the probability of finding an object at each pixel, as well as the bounding box of that object [7, 8].

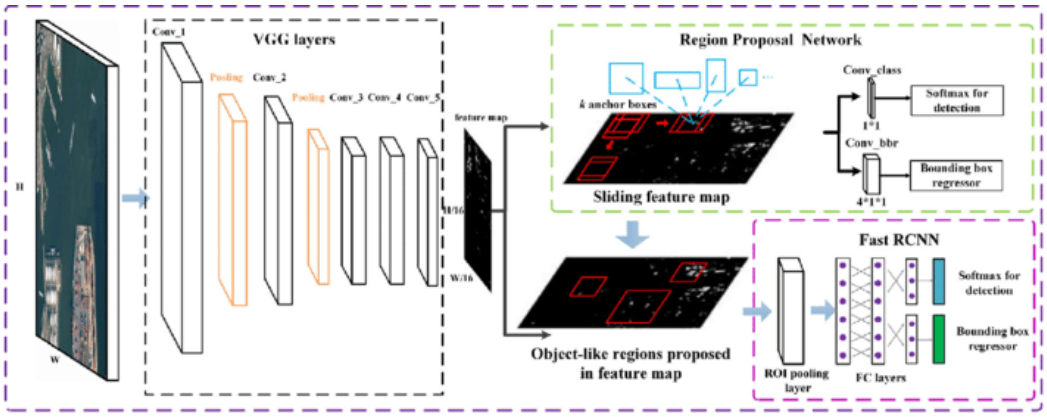

Figure 2: Structure of the Faster R-CNN [8].

Siamese Neural Networks

Extracting features from an image to produce a descriptor of that image has been widely used in ANNs, especially in Siamese Neural Networks (SNNs). For instance, SNNs are used for face recognition, so that, the addition or removal of users from the database does not require retraining the neural network. To achieve such an approach, a single neural network is trained to classify the face images of the users in the training dataset. Accordingly, if the training dataset contains face images of 100 individuals, the output layer contains 100 neurons. After the training of the neural network is completed, it gains the ability to handle the intra- and inter-class variations, so that, the same output is achieved for different images of the same user, whereas different outputs are produced for different users [9, 10].

As the neurons in the output layer rely on the values outputted from the previous layer, the values in the last hidden layer are expected to be very similar for face images of the same user and different for different users. Hence, the output layer is removed and the neural network is employed to extract the important features in the face image and generate a descriptor that can be used to match faces. After training the neural network and removing the output layer, the Euclidean distance can be calculated between any two descriptors generated by the network. Hence, two identical neural networks are used to produce these descriptors, which is the reason behind denoting this approach as Siamese [11, 12].

Clustering

Clustering is an unsupervised machine learning approach that distributes the instances in a dataset into groups, in which an instance is more similar to the other

instances in the same cluster than any other cluster in other groups [13, 14]. Several clustering techniques are proposed and have different performances in different applications. However, the K-means clustering technique has shown significantly better performance than other techniques in different applications. This method distributes the inputs in a *d*-dimensional space, where *d* is the number of features that characterize each input. Then, a number of centroids equal to the number of the required clusters are distributed randomly in the space. Each data instance is assigned to the cluster of the nearest centroid. This clustering method then optimizes the positions of the centroid, so that, the distances between the data instances and the centroid are minimized, while the distances among the centroids are maximized [14, 15].

Methodology

The proposed method embeds a neural network in the Faster R-CNN architecture in order to generate descriptors that represent the detected objects. Then, these objects are clustered into groups, in order to predict the object that is being detected, whether to be an object that is detected in a previous frame or not. Hence, the proposed method allows tracking these objects, as the objects clustered into the same group, from different frames, are considered to be the same object. An overview of the proposed method is presented in Figure 3. The descriptors outputted by the embedded neural network are filtered based on the detected objects, so that, the descriptors corresponding to positions that have no detected objects are neglected. The remaining descriptors are assigned to the corresponding objects that are detected in the same position by the Faster R-CNN neural network. These descriptors are then clustered in order to track the objects in the different frames of the video.

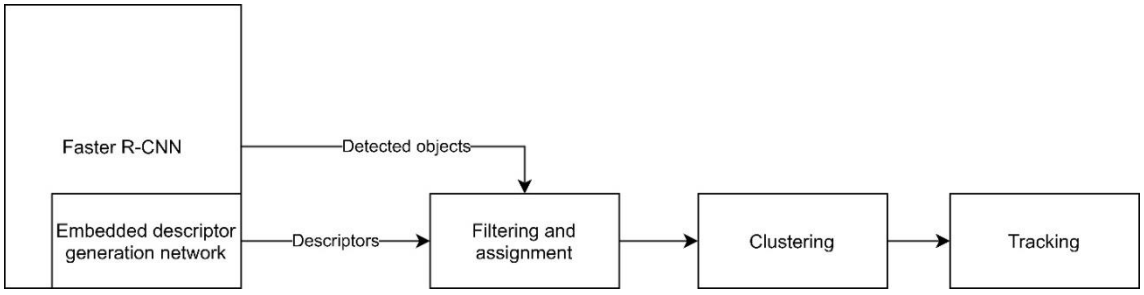

Figure 3: Overview of the proposed method.

To describe the objects detected in a certain region for object tracking, it is important to include some of the intra-class features, from the detection point of view, so that, objects of the same class are distinguished. For instance, if a faster R-CNN neural network has detected a car in the input, it is most probably that the values in the output layer and all the layers close to it to be similar to those who are outputted for a car that is detected in that image, as the values in both cases lead to the same output. The proposed method aims to generate a descriptor that can be used to distinguish these cars from each other if they are different. Hence, an additional set of layers is added to the Faster R-CNN to detect these features based on the features map and RPN's output, so that, the proposed neural network exploits the knowledge of the RPN and the low-level features in the features map to produce the required descriptors. Hence, the proposed descriptor generation neural network collects its inputs from the same position that Fast R-CNN does, which includes the features detected by the VGG16 layers and the regions proposed by the RPN, as well as the features detected in earlier stage, i.e.,

the feature map, which contain shallower features, compared to the ones detected at the output of the Fast R-CNN, as shown in Figure 4.

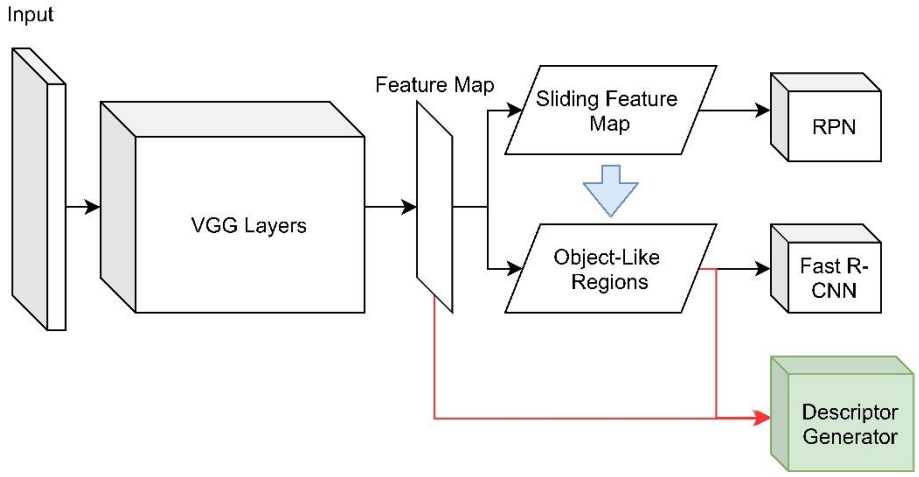

Figure 4: Positioning of the proposed descriptor generation network.

The descriptor generation neural network, marked in green in Figure 4, consists of three convolutional layers with a filter size of (2×2). The inputs are collected from the feature map and Fast R-CNN outputs, which are concatenated into a single array and are padded with zeros, so that, the output of each convolutional layer has identical dimensions to its inputs. These layers contain 256, 256 and 128 neurons, sequentially, so that, the output of the last layer has the same dimensions of the faster R-CNN but each pixel is represented by 128 values.

Training the Descriptor Generation Neural Network

The main challenge in training the proposed method is the need for similar descriptors, regardless of their values, to be outputted from the descriptor generation neural network when the same object is detected in the region. Such training can be handled using the Siamese neural networks approach by providing positive and negative samples with each anchor input. However, to avoid affecting the performance of the remaining neural networks, i.e. the RPN and Fast R-CNN, the weights and biases of all the layers in the neural network are frozen, i.e. set to be non-updateable. Moreover, to train the proposed neural network to recognize and emphasize the inter-class, as well as the intra-class, features, each anchor input is paired with two positive and two negative samples. One of the negative samples is an input for an object of the same class, whereas the other one is for an object from a different class. The positive samples are of the exact same object from different images or frames, where two samples are used to maintain balanced training with a fifty-fifty ratio of positive and negative samples.

To collect the inputs that are required to train the proposed neural network, the concatenation of the sliding feature map and the object-like features, i.e. the inputs of the Fast R-CNN, are collected for each sample image in the training dataset. These inputs are paired with each other and the objects in these inputs are labeled with zero, when the same object exists, and one, when the objects are different. However, to calculate the loss, it is important to neglect any loss caused by pixels that do not contain any objects. Hence, the Euclidian distance is calculated between each pixel in the output of each of the two neural networks, in the Siamese approach, and the output at a similar position in the other one. Hence, for a neural network that outputs *m* by *n* array, the loss is calculated as shown in Equation 1.

$$loss = \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{\sum_{k=1}^{128}\left(P_{1,i,j,k} - P_{2,i,j,k}\right)^2} \times O_{i,j} \quad 1)$$

where,
P_1 and P_2 are the outputs of the Siamese neural network and *O* represent the probability of an object being at each pixel.

However, as Equation 1 shows, the position of the objects must be exactly at the same pixel in order to be able to calculate the loss. As such positioning cannot be found in different images, the detected objects are extracted from the second image and relocated according to its location defined by the labels of the dataset. The extraction is executed based on the boundaries outputted by the Faster R-CNN, as the proposed neural network relies on the same features, with an additional 20% at each dimension, whenever possible.

The Proposed Object Tracking Method

Although the proposed neural network is expected to output similar values for similar objects, which are different from those of a different object, even if the other object is of the same category, the values cannot be identical. However, tracking the objects requires a decisive decision of whether the detected object in this frame is one of the objects detected in the previous ones or not. The proposed method relies on the K-Means clustering techniques to provide such a decision, based on the descriptors outputted by the proposed neural network. The clustering is initialized with the first frame of the video, in which the objects are being tracked, where the number of clusters is set to be equal to the number of objects detected in that frame. The median distance among the centroids of the clusters, which is the same as the instance in the first frame as each cluster contains a single instance, is calculated. This distance is used as a threshold value to detect new objects in the next frame, as it is impossible to have the same object twice in the same video frame. The use of the median value eliminates the influence of the extreme

By the arrival of the next frame, the detected objects are also clustered alongside the ones from a previous frame, or frames. The number of clusters is set to the number of objects detected in this frame initially. Then, the number of clusters is increased by one and the average distances among the centroids are measured. If the minimum distance is smaller than the average distance calculated in the previous frame, the number of clusters is reduced by one, i.e. return to the previous clustering results, and the process is terminated. The objects in the new frame are considered to be the same as the ones that they belong to in the previous frame, unless the object initiates a new cluster, wherein such case is considered a new object and its appearance in the current frame is the initial appearance in the video. Finally, the average distance among the clusters is calculated and the process is repeated until the end of the video. Figure 5 summarizes the proposed tracking method, based on the embedded descriptor generation neural network.

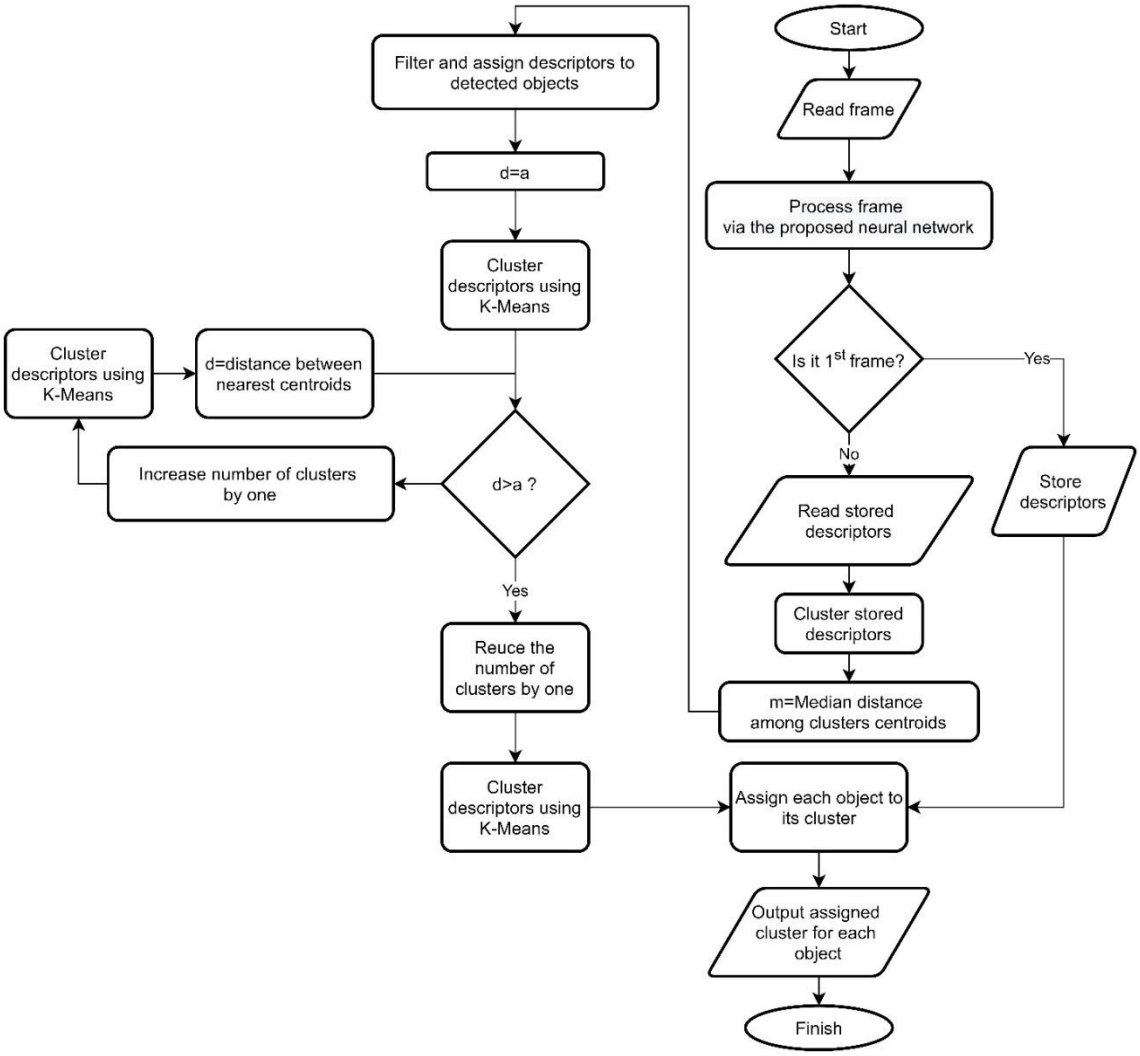

Figure 5: Flow chart of the proposed multi-object tracking method.

Performance Evaluation

The proposed method is implemented using Python programming language, with Tensorflow library to implement, train and evaluate the neural network, and Sci-Kit Learn library for K-means clustering. The performance of the proposed method is evaluated using the UA-DETRAC (University at Albany Detection and Tracking) dataset [16], which contains 100 real-life traffic videos, with a total of more than 140,000 frames in these videos. The videos are distributed into training and testing sets, with 40 videos in the testing set, and each video is manually annotated, where each car is assigned with a unique identifier, so that, it can be tracked among different frames. Although the Faster R-CNN has been able to detect other objects in the frames, as shown in Figure 6, only annotated cars are taken into consideration in the evaluation. The performance is evaluated by measuring the Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP), where the results of the proposed method using the 40 videos in the testing dataset, after training the neural network using the 60 videos in the training dataset, as summarized in Table 1.

Figure 6: Objects in a sample frame from the UA-DETRAC dataset. Top: Detected by the Faster R-CNN. Bottom: Annotated in the dataset.

Table 1: MOTA and MOTP comparison of the proposed method and existing state-of-the-art methods.

Detection Method	Tracking Method	MOTA (%)	MOTP (%)
Deformable Part-Based Model (DPB) [17]	Globally-Optimal Greedy (GOG) [18]	26.2	76.2
ACF [19]	Globally-Optimal Greedy (GOG) [18]	35.7	80.3
R-CNN [20]	Discrete Continuous Tracking (DCT) [21]	38.4	80.6
Faster R-CNN [7]	Multiple Hypothesis Tracking (MHT)	58.2	78.4
Evolving Boxes (EB) [22]	KIOU [23]	62.1	81.9
Faster R-CNN [7]	Proposed	64.8	83.6

In addition to the improvement produced by the proposed method to both the MOTA and MOTP, compared to the existing techniques, significant improvement has been shown compared to the use of the same neural network but with the MHT method for object tracking. This improvement is a result of the use of the same features that are detected by the Faster R-CNN, to detect the objects in the image, in the descriptor generation. Moreover, the embedding of the processing with the Faster R-CNN layers and the use of the low-level features in the object description stage, instead of the use of

a separate neural network with the need to learn the features from scratch, the proposed method has been able to provide faster performance, as shown in Table 2. This hypothesis is proved by the significantly lower number of frames that the method proposed in [24] has achieved, despite the use of CNN in the Siamese approach. However, as the CNN in that method process the images extracted from the frame, in a separate stage, the method proposed in this study has been able to process a significantly larger number of frames per second.

Table 2: Comparison of the frames processing speed using different tracking methods.

Method	Frames per Second
Siamese CNN [24]	2.1
JPDA [25]	35.6
LSTM-DRL [26]	108
Faster R-CNN (This study)	127.3

The results of this study also illustrate the importance of transfer learning, as the features learned by a neural network are being used in similar, yet not identical, applications. Additionally, these results illustrate the importance of integrating the different tasks required from a certain input in the same neural network. Such integration can improve the performance of the neural network by achieving more tasks with minimal overhead. This concept has been the backbone of the Faster R-CNN, which has improved the performance of the Fast R-CNN by integrating the RPN in the same neural network and by processing the same inputs collected by the Fast R-CNN [7].

Conclusion

With the rapidly growing use of computers to capture and analyze videos in different applications, detection and tracking objects that appear in these videos have become the interest of many researchers in recent years. However, the use of separate stages to generate descriptors that can be used to define these objects has increased the overhead in the required computations to track these objects. Hence, a new method is proposed in this paper, in which a neural network is embedded in the Faster R-CNN architecture, so that, this neural network exploits the features that are already detected by the neural network in the Faster R-CNN, which significantly reduces the overhead in computations. The descriptors collected from the proposed neural network for the first frame are considered the centroids of clusters, i.e., the number of clusters is equal to the number of detected objects. Then, with every new frame, the number of clusters is increased by one until the distance between the centroid of the newly generated cluster and the nearest centroid is less than the median distance among all the clusters in the previous frame.

The proposed method has been evaluated using the UA-DETRAC traffic dataset, which contains videos collected from certain streets, where each car is assigned with a unique identifier that is maintained throughout the video. The proposed method has been able to achieve 64.8% MOTA and 83.6% MOTP, which is a slightly better performance, compared to existing state-of-the-art methods. However, the embedding of the descriptor generation neural network, i.e., the reduction of the computations overhead, has significantly increased the number of frames that the proposed method can process per each second, as it has been able to process 127.3 frames per second.

Hence, according to this reduction in complexity by embedding the descriptor generation stage, the proposed method can be implemented with any systems that use the Faster R-CNN for objects detection, without the need for additional computer resources.

In future work, the ability to use similar approach with the You Only Look Once (YOLO) neural network is going to be investigated. Despite the better performance of the YOLO, compared to the Faster R-CNN in several occasions, the use of different scaling levels at the layers of the neural network in YOLO, instead of the anchors in the Faster R-CNN, imposes challenges towards training such a neural network. However, if such a training approach is proves successful, significant improvement to the performance of multi-object detection and tracking can be achieved.

REFERENCES

[1] M. Elhoseny, "Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems," *Circuits, Systems, and Signal Processing,* vol. 39, pp. 611-630, 2020.

[2] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, *et al.*, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942-7951.

[3] M. Van Gerven and S. Bohte, "Artificial neural networks as models of neural information processing," *Frontiers in Computational Neuroscience,* vol. 11, p. 114, 2017.

[4] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon,* vol. 4, p. e00938, 2018.

[5] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, 2018, pp. 270-279.

[6] D. Kim and T. MacKinnon, "Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks," *Clinical radiology,* vol. 73, pp. 439-445, 2018.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, pp. 1137-1149, 2016.

[8] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS journal of photogrammetry and remote sensing,* vol. 145, pp. 3-22, 2018.

[9] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834-4843.

[10] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention siamese network," *IEEE transactions on cybernetics,* vol. 50, pp. 3068-3080, 2019.

[11] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, 2015.

[12] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, *et al.*, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 7, pp. 669-688, 1993.

[13] T. S. Madhulatha, "An overview on clustering methods," *arXiv preprint arXiv:1205.1117,* 2012.

[14] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, *et al.*, "A review of clustering techniques and developments," *Neurocomputing,* vol. 267, pp. 664-681, 2017.

[15] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, pp. 451-461, 2003.

[16] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, *et al.*, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[17] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2497-2504.

[18] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR 2011*, 2011, pp. 1201-1208.

[19] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, pp. 743-761, 2011.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[21] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1926-1933.

[22] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," in *2017 IEEE international conference on multimedia and Expo (ICME)*, 2017, pp. 1135-1140.

[23] S. Lyu, M.-C. Chang, D. Du, W. Li, Y. Wei, M. Del Coco, *et al.*, "UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1-6.

[24] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 33-40.

[25] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3047-3055.

[26] M.-x. Jiang, C. Deng, Z.-g. Pan, L.-f. Wang, and X. Sun, "Multiobject Tracking in Videos Based on LSTM and Deep Reinforcement Learning," *Complexity*, vol. 2018, 2018.