Pedestrian Detection for Advanced Driver Assistance Systems using Deep Learning Algorithms

Yahia Fahem Said[1,2] and Mohammad Barr[1]

[1]Electrical Engineering Department, College of Engineering, Northern Border University, Arar, Saudi Arabia
[2]Laboratory of Electronics and Microelectronics (LR99ES30), Faculty of Sciences of Monastir, University of Monastir, TUNISIA

Summary
The notion of pedestrian detection is used in computer vision, which not only detects humans but also counts the numbers of pedestrians and determines their movement in real time. This technique is used in many applications such as surveillance, advanced robotics, intelligent vehicles and Advanced Driver Assistance Systems (ADAS). The pedestrian detection system needs acceleration to enable real-time adaptive processing. Hardware acceleration has the potential to speedup these algorithms, making real-time processing for many image and video processing. In order to meet the real-time requirement, high-speed pedestrian detection architecture must be designed carefully.
This paper presents a pedestrian detection application for Advanced Driver Assistance Systems based on a Deep Learning algorithm. It's about proposing a structure of a Deep Learning model which makes it possible to improve the precisions existing in the state-of-the-art and the processing time by images. This is a very difficult problem because of the complexity of the task and the challenges presented by the detection of humans in general.

Key words:
Deep Learning, Artificial Intelligence, Pedestrian Detection, Advanced Driver Assistance Systems.

1. Introduction

According to the World Health Organization, each year, approximately 1.35 million people are killed as a result of traffic accidents. Between 20 million and 50 million others are affected by non-fatal injuries and many are disabled as a result [1]. Half of road traffic fatalities belong to the category of vulnerable road users, cyclists and pedestrians.
Car manufacturers currently offer a large number of advanced driver assistance systems in the new vehicles they market. At the same time, they are actively working on new generation ADAS applications as well as autonomous driving features. It has also become increasingly important to offer features that are suitable for pedestrian detection and accident prevention with animals. These developments not only improve the safety of pedestrians, but also that of the occupants of the vehicle [2].

Developing algorithms for the detection and classification of obstacles as "pedestrian objects" is complicated regardless of the context (inside or outside) and the sensor used (laser, video or radar) by the large scale variability of the pedestrian, posture and appearance. The problem is therefore to find a representation of a pedestrian being that is both generic enough to encompass all types of situations, and sufficiently discriminating to represent only pedestrians. For this purpose, a machine learning algorithm using an intermediate representation is generally used, based on features computation which are a set of scalar numbers generated to describe an object (or a form). Characteristics of areas containing people are generally used by a supervised learning method to determine a person model. Then, each area of the image is classified as a person or not, from a feature vector calculated on this area, according to this model.
Detection of objects by machine learning algorithms achieves maximum performance and a very slow detection time that prevents real-time use [3]. All these disadvantages require the research of new techniques for objects detection.
Deep Learning techniques are a huge success in the field of computer vision. It allows an algorithm to assimilate and identify the content of an image or to understand natural language. It is a process based on networks of multi-layered artificial neurons made up of thousands of units of computation. These artificial neurons have nothing material: they are mathematical functions with several adjustable parameters
Convolutional Neural Networks (CNN) are one of the most remarkable approaches to Deep Learning, in which multiple layers of neurons are formed in a robust manner. They have demonstrated an impressive ability to generalize large datasets with millions of images. It was inspired by the simulation of the human brain system aims to find a way to solve general learning problems. Mostly known for their classification performance, these techniques can also be used in image enhancement, objects detection, tracking, multi-camera analysis, and 3D reconstruction.
Detecting objects by Deep Learning is a complex task consisting of two parts. The first is the classification of the

Manuscript received September 5, 2019
Manuscript revised September 20, 2019

object and the second is the location of the extent of the object in the image. Deep learning object detection methods fix the problems of machine learning algorithms by improving detection accuracy and reducing test time.

In this paper, we propose a Deep Learning algorithm for pedestrian observation and detection application, covering a wide spectrum of computer vision applications. The remainder of the paper is organized as follows. Related works on pedestrian recognition and classification are presented in Section 2. Section 3 describes the proposed model for pedestrian detection. In Section 4, experiments and results are detailed. Finally, Section 5 concludes the paper.

2. Related Works

Pedestrian detection on embedded platforms is a challenging task since accurate recognition requires extensive computation. To achieve real-time pedestrian detection, numerous implementations of deep learning algorithms on different platforms have been proposed in the literature [14, 15, 16]. We present below, some related works on implementations of pedestrian detection structures and the obtained results.

Kang et al. [4] proposed parallelization of SIFT features detector, on GPU using NVIDIA CUDA framework. In order to accelerate the elementary CUDA-SIFT, they used CUDA streams and instruction level parallelization. The results show that their implementation runs 60 times faster than the embedded CPU implementation.

Haar descriptors combined with AdaBoost algorithm was implemented by Chao et al. [5] using CUDA framework. Compared with CPU implementation, a near real-time performance was achieved with the GPU implementation running 6 times faster.

Said et al. [6], proposed a real-time pedestrian recognition system on FPGA prototyping boards based on an efficient feature extraction scheme. They used sliding covariance matrices densely extracted from a detection window as descriptors and linear SVM as classifier. The proposed framework detects pedestrians more accurately than existing methods with minimum logic area and can be integrated in intelligent transport system.

In [7], a multiple objects detector (human, head, bicycle silhouettes), is presented using the Histogram of Oriented Gradients descriptor and SVM classifier. The design was embedded into an FPGA reconfigurable device. The proposed system reaches a process speed of 60 fps for VGA image resolution.

Automatic learning algorithms based on manually designed features have low accuracy and slow execution time that prevents us from implementing in real time. On the contrary, deep learning allows an automatic extraction of the characteristics and ensures a high precision and a very fast processing time. As a result, all current researches in the field of computer vision are based on deep learning.

Song et al. [8] used Single shot multibox detector (SSD) structure and Mobilenet model for pedestrian detection and tracking from video streams of Seoul National University CCTV dataset. Experimental results show that the proposed system has faster processing time with high accuracy.

A pedestrian detection framework based on deep learning is proposed in [9]. An optimized PVANet is first utilized to generate the feature maps; then, the region proposal network is utilized to generate the pedestrian candidates and the corresponding scores; finally, the enhanced decision trees are utilized to complete the issue of pedestrian detection. Extensive evaluations conducted on Caltech pedestrian detection dataset demonstrate the effectiveness of the proposed method.

Du et al. [10] proposed network fusion architecture for efficient pedestrian detection. Different sizes of pedestrian candidates from single shot detector (SSD), were refined by multiple deep neural networks used in parallel, before generating the final detection results. When tested on Caltech Pedestrian dataset, the proposed system achieved state-of-the-art performance.

Taking pedestrian detection as an example, Li et al. [11] developed a Scale-Aware Fast R-CNN framework. Pedestrians proposals with different scales were adaptively combined from sub-networks to generate the final confidence scores. Experimental results on different datasets showed that the proposed model was the fastest with highest accuracy.

In [12], Lin et al. detected pedestrian with a graininess-aware deep feature learning method. Their system encodes pedestrian attention masks into deep convolutional feature maps. A zoom-in-zoom-out module was introduced to identify occluded pedestrian at small size. The proposed method has better detection accuracy and speed on challenging pedestrian datasets than state-of-the-art systems.

A deep neural network fusion architecture for pedestrian detection is proposed in [13]. Deep Single shot CNN structure generate pedestrian candidates with a high detection rate. Then, a soft-rejection strategy is used to adjust the confidence in the detector candidates by fusion with a classification network. Comprehensive experimental results on Caltech Pedestrian benchmark demonstrate the effectiveness of the proposed architecture.

3. Proposed Architecture for Pedestrian Detection

Pedestrian detection consists of identifying and locating in the image the searched individuals. For a given entry, the

detection returns two information: the pedestrian classification but also the location.

The classification task is a part of the detection to be performed. This is to give the class of the object in the image. For Deep Learning, convolutional neural networks CNNs are the most used for classification. Thus, feature extraction filters are implemented as a hidden layer with shared weights that are optimized together with those of the classification component so that the total classification error is minimized.

The task of localization is to determine the extent of the individuals in the image by putting each pedestrian in a rectangular box. There are several methods for locating pedestrians. The localization problem is treated as a regression problem. There are external methods of the neural network, for example the selective search method and methods integrated into the neural network such as the Region Proposal Network (RPN).

Neural networks for detection give two sets of parameters at the output, the first is the class and the probability of the object and the second are the location parameters that are the coordinates (x, y) that are at center of the box, height and width. To determine the class of an object we use fully connected layers of type Softmax and to determine the parameters of the localization box we use fully connected layers of type linear regression.

The "You Only Look Once" algorithm, referred to as YOLO, is a powerful and fast Deep Learning algorithm [19]. It is one of the fastest algorithms but it is not the most accurate. If we want to carry out the pedestrian detection phase in real time, the use of this architecture presents the most robust and efficient solution. YOLOv2 [17] is the second version of the YOLO with the objective of improving the accuracy significantly while making it faster.

In this work, we present an implementation of a Deep Learning algorithm for pedestrian detection on a GPU graphics card. The detection phase in our approach is based on the Yolov2 algorithm [17], but we changed the feature extractor DarkNet19 CNN by the SqueezeNET network [18].

The architecture of SqueezeNet is proposed by Iandola et al. [18]. The idea of developing this network is to use filters (1 × 1) instead of filters (5 × 5). Subsequently, they implemented filters (1 × 1) as a bottleneck layer to reduce the depth to reduce the calculation of the following filters (3 × 3). Then, use Sub-sampling late to keep a large feature map.

The building element of SqueezeNet is called "Fire Module". This module contains two layers: a compression layer and an expansion layer. The "Fire" module has a squeeze convolution layer that includes filters (1 × 1), feeding an expansion layer composed of a mixture of convolution filters (1 × 1) and (3 × 3) as presented in Figure 1. The compression layer and the expansion layer retain the same feature map size while the other architectures reduce the map size. SqueezeNet uses a methodology similar to GoogleNet but with a different structure. These modules allow a 50-fold reduction in parameters compared to AlexNet and keep the same accuracy of AlexNet.

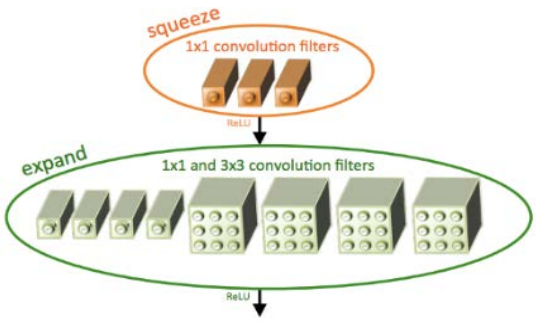

Fig. 1 Fire Module architecture

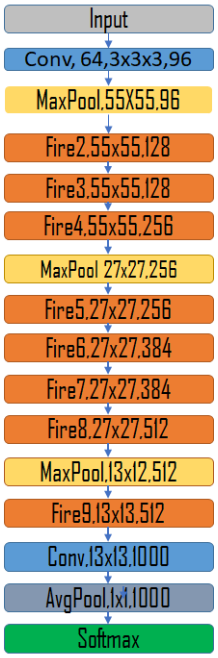

Fig. 2 SqueezeNet model

SqueezeNet presents one of the innovative architectures as a classifier, with a reduced architecture that has less than 737,868 parameters and a reduced size of about 4.7 MB. These features give it an advantage over other CNNs architectures like AlexNet which has about 132 million parameters [18]. We introduce SqueezeNet into our detection model to make it too fast and robust. Figure 2 explicitly describes the different layers of SqueezeNet.

The YOLO detection network has 24 convolution layers followed by two fully connected layers. Each of these

convolution layers uses (1 x 1) reduction layers alternatively to reduce the depth of the feature map. For the last convolution layer, it emits a tensor of form (7, 7, 1024). The tensor is then flattened. Then it generates (7 x 7 x 30) shape parameters. Then it transforms them into the form of (7,7,30).

YOLOv2 does classification and prediction in a single framework. It uses the anchor boxes, which are responsible for predicting bounding box, and this anchor boxes are designed for a given dataset by using k-means clustering.

In the YOLOv2, the localization process has been changed to direct location prediction technique. The direct location prediction technique was based on predicting offsets to each of the anchor boxes instead of predicting arbitrary boundary boxes. If the offset values are constrained, then the diversity of the predictions can be maintained and each prediction focuses on a specific shape. To perform this technique, YOLOv2 predicts 5 parameters (t_x, t_y, t_w, t_h, and t_o) and applies the sigmoid function to constraint its possible offset range. The parameters of the proposed bounding box (b_x, b_y, b_w, and b_h) are calculated as Eq. 1 and the box confidence scores was calculated as Eq. 2.

$$b_x = \sigma(t_x) \times c_x$$
$$b_y = \sigma(t_y) \times c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h} \quad (1)$$
$$\sigma(t_o) = p_r(object) \times IoU\ (b, object) \quad (2)$$

Where:

- (t_x, t_y, t_w, t_h, and t_o): YOLO generated predictions,
- (c_x, c_y) : top left corner coordinate of the anchor,
- (p_w, p_h): the width and the high of the anchor,
- (b_x, b_y, b_w, and b_h): (x, y) coordinate, width and high of the predicted bounding box respectively,
- $\sigma(t_o)$: box confidence score,
- $p_r(object)$: the probability the box contains an object,
- IoU: (Intersection over Union), the IoU between the predicted box and the ground truth.

For the final prediction,(p_w, p_h) and (c_x, c_y) are normalized using the high and width of the input image. Figure 3 illustrates the predicted bounding box (blue rectangle) and the tested anchor (dotted rectangle).

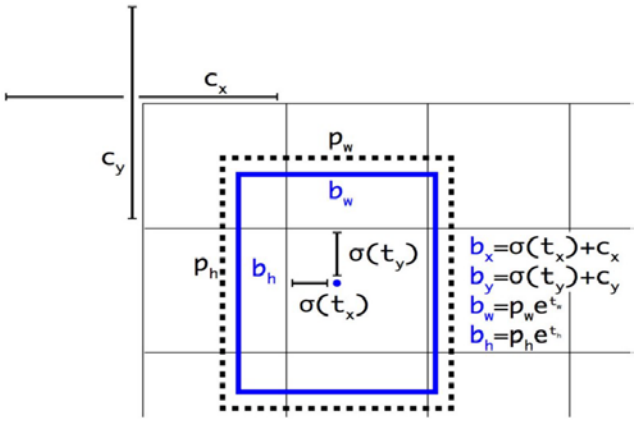

Fig. 3 Bounding boxes with dimension priors and location prediction [17]

4. Experiments and Results

For the development of our approach we use the Deep Learning framework TensorFlow, and Nvidia acceleration libraries (cuda 8.1, cudnn) to perform the learning and testing phases of our proposed model on a desktop equipped with Intel i7 CPU and Nvidia GTX960 GPGPU.

We used the Caltech Pedestrian Dataset [20], which contains about 250,000 images with a total of 350,000 bounding boxes, taken from a vehicle driving through an urban environment. To optimize our algorithm, we have divided this database into learning dataset and Test and Validation datasets, following the most used Deep Learning protocol as 80% for training and 20% for test and validation.

To train the proposed network, the loss function needs to be optimized. The loss function is the difference between the network output and the target value. The learning algorithm based on the back-propagation technique optimize the loss function by updating the value of the network parameters (weights and biases) until reaching the minimum difference between the network values and the target values. The proposed architecture was trained using the momentum optimizer which is a popular optimizer frequently providing faster convergence. It updates the weights very much like SGD (Stochastic Gradient Descent).

After model training, we obtain a minimum of objective function of the order of 0.198, with a reduced neural architecture a little closer to the Yolov2 model with DarkNet 19 network for feature extraction which obtained a minimum of objective function equal to 0.146 with the same learning dataset.

Based on the Deep Learning framework TensorFlow and the OpenCv image processing library, testing this proposed model achieves a Mean Average Precision of 75.8% mAP. Figure 4 illustrate example of detection on image chosen from the Caltech Pedestrian Dataset. The

proposed approach was tested also using images that do not belong to the dataset. An example is illustrated in figure 5.

Fig. 4 Detection results on an image from Caltech database (5 Boxes)

Fig. 5 Detection results on an image from Google (14 Boxes)

After testing, the model was deployed in a pedestrian detection inference on the GPGPU. The inference speed achieved was 32.4 fps.

A comparison with existing works for pedestrian detection is presented in Table 1. All these algorithms are learned on the same dataset, as well as we use transfer learning to test these algorithms with the same hardware resources used during the learning and testing phase of our proposed approach.

By analyzing the results presented in Table 1, we note that our proposed model outperforms the state-of-the-art architectures in term of speed and that allows us to perform real-time detection. Also it presents a reduced neural architecture that has less than 0.8 million parameters and that can be implemented on a mobile application or an embedded system for an Advanced Driver Assistance Systems.

In terms of accuracy and comparing our approach with that of YOLOv2 combined with DarkNet [17] or SSD+SqueezeNet [21], we note that our contribution with the addition of SqueezeNet gave a better precision. But our approach is not more accurate, compared with that of Fused DNN [10] which reaches an accuracy of the order of 89.05%.

Table 1. Comparison with existing works for pedestrian detection

Approach	mAP (%)	Inference speed (FPS)	Parameters Number (million)
Fused DNN [10]	89,05	8.15	≈ 147
SSD+SqueezeNet [21]	71.6	8.6	≈ 2.5
Faster R-CNN [22]	76,4	21	≈ 160
YOLOv2 + Darknet19 [17]	49.1	30	≈ 6
Mask R-CNN [23]	80,09	2.6	≈ 100
Our's (YOLOv2+ SqueezeNet)	75.8%	32.4	≈ 0.7

5. Conclusion

Pedestrian detection using deep learning techniques is a very dominant field in the most recent scientific research, thanks to its utility in the development of ADAS systems.

Based on the YOLOv2 model and with the objective of developing a real-time pedestrian detection algorithm, we proposed in this paper, a pedestrian detection structure based on SqueezeNet Convolotionel Neural Network. After the learning and testing phase, we find that reducing the number of parameters of a neural architecture can reduce the size of the structure and make it faster with acceptable accuracy. The proposed structure achieved high performance and proved its robustness and can be implemented on a mobile application or embedded system. Indeed, the appearance of new Nvidia platforms for autonomous cars, such as the Nvidia Drive AGx, make it easy to deploy a Deep Learning model in the vehicle computer. In our test, a hybrid system composed of an Intel CPU and a Nvidia GPU with approximately the same performance of the Nvidia Drive AGx platform, was used.

Acknowledgments

The authors wish to acknowledge the approval and the support of this research study by the grant N° ENG-2018-3-9-F-7654 from the Deanship of the Scientific Research in Northern Border University, Arar, KSA.

References

[1] Global status report on road safety 2018, https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/

[2] Estl, Hannes. "Paving the way to self-driving cars with advanced driver assistance systems." Worldwide Systems Marketing for Advanced Driver Assistance Systems (ADAS), Texas Instruments (2015).

[3] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems (2019).

[4] Kang, Seung Heon, Seung-Jae Lee, and In Kyu Park. "Parallelization and optimization of feature detection algorithms on embedded gpu." International IEEE Workshop on Advanced Image Technology. Vol. 108. 2014.

[5] Chang Chao Cai, Jue Gao, BianMinjie, Peichang Zhang and Honghalo Gao: “Fast Pedestrian detection with Adaboost Algorithm using GPU”. International Journal of Database Theory and Application Vol.8, No.6, pp.125-132, (2015).
[6] Said Yahia and Atri Mohamed. “Efficient and high-performance pedestrian detector implementation for intelligent vehicles”. IET Intelligent Transport Systems Journal (ISSN: 1751-956X), Vol. 10, Iss. 6, pp. 438–444, (2016).
[7] Komorkiewicz, M., Kluczewski, M., Gorgon, M.: Floating point HOG implementation for real-time multiple object detection. In: IEEE Field Programmable Logic and Applications (FPL), pp. 711–714 (2012).
[8] Song, H., Choi, I. K., Ko, M. S., Bae, J., Kwak, S., & Yoo, J. Vulnerable pedestrian detection and tracking using deep learning. In Electronics, Information, and Communication (ICEIC), 2018 International Conference on (pp. 1-2). IEEE, 2018.
[9] Sun, W., Zhu, S., Ju, X., & Wang, D. Deep learning based pedestrian detection. In 2018 Chinese Control and Decision Conference (CCDC) (pp. 1007-1011). IEEE, 2018.
[10] Du, Xianzhi, et al. "Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection." Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. IEEE, 2017.
[11] Li, J., Liang, X., Shen, S., Xu, T., Feng, J., & Yan, S. (2018). Scale-aware fast R-CNN for pedestrian detection. IEEE Transactions on Multimedia, 20(4), 985-996, 2018.
[12] Lin, C., Lu, J., Wang, G., & Zhou, J. Graininess-Aware Deep Feature Learning for Pedestrian Detection. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 732-747), 2018.
[13] Du, X., El-Khamy, M., Morariu, V. I., Lee, J., & Davis, L. (2018). Fused Deep Neural Networks for Efficient Pedestrian Detection. arXiv preprint arXiv:1805.08688, 2018.
[14] Calum Blair, Neil M Robertson, Danny Hume (2013) “Characterising a Heterogeneous System for Person Detection in Video using Histograms of Oriented Gradients: Power vs. Speed vs. Accuracy ", In IEEE Journal on Emerging and Selected Topics in Circuits and Systems 3 (2). pp: 236-247, 2013.
[15] S. Bauer, S. Kohler, K. Doll, and U. Brunsmann, “FPGA-GPU architecture for kernel SVM pedestrian detection,” in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 61 –68, June 2010.
[16] Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2018). Towards reaching human performance in pedestrian detection. IEEE transactions on pattern analysis and machine intelligence, 40(4), 973-986, 2018.
[17] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263-7271, 2017.
[18] Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." arXiv preprint arXiv:1602.07360(2016).
[19] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.
[20] P. Dollar, C. Wojek, B. Schiele, et P. Perona, « Pedestrian Detection: An Evaluation of the State of the Art », IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no 4, p. 743-761, avr. 2012.
[21] Verbickas, Rytis, Robert Laganiere, Daniel Laroche, Changyun Zhu, Xiaoyin Xu, and Ali Ors. "SqueezeMap: fast pedestrian detection on a low-power automotive processor using efficient convolutional neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 146-154. 2017.
[22] Kim, Jong Hyun, Ganbayar Batchuluun, and Kang Ryoung Park. "Pedestrian detection based on faster R-CNN in nighttime by fusing deep convolutional features of successive images." Expert Systems with Applications 114 (2018): 15-33.
[23] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969. 2017.

Yahia Fahem Said received the B.Sc., M.Sc., and Ph.D. degrees in Electronics from the Faculty of Science of Monastir, Monastir University, Tunisia in 2008, 2010, and 2016 respectively. He worked as an assistant professor at Faculty of Science of Monastir, Monastir University, Tunisia. Now, he is an assistant professor at Electrical Engineering Department, College of Engineering, Northern Border University, Arar, Saudi Arabia. His research interests are Pattern Recognition, Artificial Intelligence, Deep Learning, Convolutional Neural Networks, Image and Video Processing, Embedded Vision and SoC Design.

Mohammad Barr received the M.Sc., and Ph.D. degrees in Electronic Engineering from Faculty of Technology, De Montfort University-Leicester, United Kingdom, in 2013 and 2018, respectively. He is an assistant professor at Electrical Engineering Department, College of Engineering, Northern Border University, Arar, Saudi Arabia. His research interests are Image and Video Processing, Digital Signal Processing and Embedded Systems.