Computers, Materials & Continua
DOI:10.32604/cmc.2021.018781
Article

Tech Science Press

YOLOv2PD: An Efficient Pedestrian Detection Algorithm Using Improved YOLOv2 Model

Chintakindi Balaram Murthy[1], Mohammad Farukh Hashmi[1], Ghulam Muhammad[2,3,*] and Salman A. AlQahtani[2,3]

[1]Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, 506004, India
[2]Research Chair of Pervasive and Mobile Computing, King Saud University, Riyadh, 11543, Saudi Arabia
[3]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia
[*]Corresponding Author: Ghulam Muhammad. Email: ghulam@ksu.edu.sa

Received: 21 March 2021; Accepted: 04 May 2021

Abstract: Real-time pedestrian detection is an important task for unmanned driving systems and video surveillance. The existing pedestrian detection methods often work at low speed and also fail to detect smaller and densely distributed pedestrians by losing some of their detection accuracy in such cases. Therefore, the proposed algorithm YOLOv2 ("YOU ONLY LOOK ONCE Version 2")-based pedestrian detection (referred to as YOLOv2PD) would be more suitable for detecting smaller and densely distributed pedestrians in real-time complex road scenes. The proposed YOLOv2PD algorithm adopts a Multi-layer Feature Fusion (MLFF) strategy, which helps to improve the model's feature extraction ability. In addition, one repeated convolution layer is removed from the final layer, which in turn reduces the computational complexity without losing any detection accuracy. The proposed algorithm applies the K-means clustering method on the Pascal Voc-2007 + 2012 pedestrian dataset before training to find the optimal anchor boxes. Both the proposed network structure and the loss function are improved to make the model more accurate and faster while detecting smaller pedestrians. Experimental results show that, at 544 × 544 image resolution, the proposed model achieves 80.7% average precision (AP), which is 2.1% higher than the YOLOv2 Model on the Pascal Voc-2007 + 2012 pedestrian dataset. Besides, based on the experimental results, the proposed model YOLOv2PD achieves a good trade-off balance between detection accuracy and real-time speed when evaluated on INRIA and Caltech test pedestrian datasets and achieves state-of-the-art detection results.

Keywords: Computer vision; K-means clustering; multi-layer feature fusion strategy; pedestrian detection; YOLOv2PD

Abbreviations

AP	Average Precision
CV	Computer Vision
CUDA	Compute Unified Device Architecture
DPM	Deformable Part Model
FPS	Frames per second
FP	False Positive
FN	False Negative
HOG	Histogram of Oriented Gradient
IoU	Intersection over Union
MR	Miss-rate
MLFF	Multi-layer Feature Fusion
Pascal Voc	Pascal Visual Object Classes
RCNN	Regions Based Convolutional Neural Networks
SPPNet	Spatial Pyramid Pooling Network
SSD	Single Shot Multi-Box Detector
SOTA	State-of-the-art
TP	True Positive
TN	True Negative
YOLO	YOU ONLY LOOK ONCE
YOLOv2	YOU ONLY LOOK ONCE Version 2
YOLOv2PD	YOU ONLY LOOK ONCE Version 2 Based Pedestrian Detection

1 Introduction

One of the most important applications of Computer vision (CV) in self-driving cars is pedestrian detection. The field of pedestrian detection covers video surveillance, criminal investigations, self-driving cars, and robotics. Real-time pedestrian detection is an important task for unmanned driving systems. The vision system of autonomous vehicle technology was initially very difficult to develop in the field of CV; however, owing to continuous improvements of hardware computational power, many researchers have attempted to develop reliable vision systems for self-driving cars. Since 2012, deep learning has been developed and achieved tremendous progress in the field of CV. In the field of artificial intelligence, many deep learning-based algorithms have been introduced and used in a wide range of applications, such as in signal, audio, image, and video processing. In particular, deep learning-based algorithms play a groundbreaking role in fields such as image and video processing, for example, image classification and detection.

One of the direct applications of real-time pedestrian detection is that it should automatically locate pedestrians accurately with on-shelf cameras, since it plays a crucial role in robotics and unmanned driving systems. Despite tremendous progress having been achieved recently, this task still remains challenging due to the complexity of road scenes, such as them being crowded, occluded, containing deformations and exhibiting lighting changes. Currently, unmanned driving systems are among the major fields of research in CV, for which the real-time detection of pedestrians is essential to avoid possible accidents. Although deep learning-based techniques improve detection accuracy, there is still a huge gap between human and machine perception [1]. A complex background, low-resolution images, lighting conditions, and occluded and distant smaller objects reduces the model accuracy. To date, most researchers in this field have focused only on color-image-based object detection. Therefore, when detecting objects in a shadowy environment or objects captured at night, lower detection accuracy is achieved.

This is the major drawback of reliable vision-based detection systems since self-driving cars in real-time extremely complex environments should be able to detect objects in the daytime or at night. Nevertheless, current state-of-the-art (SOTA) real-time pedestrian detection still falls short of the fast and accurate human perception levels [2].

Currently, pedestrian detection methods are classified into two time slots: traditional and deep learning time slot methods. Traditional time slot methods cover various traditional machine learning algorithms such as Voila Jones detector [3], Deformable part model (DPM) [4], Histogram of oriented gradient (HOG) [5] and multi-scale gradient histograms [6]. These methods are time-consuming, require complex steps, are expensive, and require a high level of human interference. In the recent evolution of deep learning techniques since 2012, such techniques have become very popular and deep CNN-based pedestrian detection methods have achieved better performance than traditional time slot methods [7,8]. The first deep learning-based object detection model was RCNN [9]. This method generates a region of interest by using a selective search window for deep learning-based object detection, as implemented in all RCNN series. Deep learning time slot methods cover both two-stage detectors such as RCNN [9], SPPNet [10], Fast-RCNN [11], Faster RCNN [12] and Mask-RCNN [13] and single-stage detectors such as SSD [14] and YOLO [15]. Therefore, in the current scenario for real-time pedestrian detection, these methods are not quite suitable.

Generally, the speed of deep learning-based object detection methods is low, with these methods being unable to meet real-time requirements of self-driving cars. Therefore, to improve both speed and detection accuracy, Redmon et al. [15] proposed the YOLO network, a single end-to-end object regression framework. Later, Redmon et al. [16] implemented YOLOv2 to overcome the drawbacks of the YOLO [15] framework. YOLOv2 [16] improves the speed of the detection algorithm without losing any part of the detection accuracy. However, when detecting smaller objects in complex environments, it achieves low detection accuracy.

To improve both detection accuracy and speed when detecting smaller and densely distributed pedestrians, a new pedestrian detection technique is proposed, YOLOv2-based pedestrian detection (in short, YOLOv2PD). An efficient K-means clustering [17] algorithm is applied to select six different anchor box sizes while training the Pascal Voc-2007 + 2012 pedestrian dataset.

The contributions of the proposed work can be summarized as follows:

(1) The proposed YOLOv2PD model adopts the MLFF strategy to improve the model's feature extraction ability and, at the higher end, one convolution layer is eliminated.
(2) Moreover, intuitively, to test the effectiveness of the proposed model, another model referred to as YOLOv2 Model A is implemented and compared.
(3) The loss function is improved by applying normalization, which reduces the effect of different pedestrian sizes in an image, and which potentially optimizes the detected bounding boxes.
(4) Through qualitative and quantitative experiments conducted on Pascal Voc-2007 + 2012 Pedestrian, INRIA and Caltech pedestrian datasets, we validate the effectiveness of our algorithm, showing that it has better detection performance on smaller pedestrians.

The rest of the paper is organized as follows. Sections 2 covers related work. In Section 3, the proposed YOLOv2PD algorithm is illustrated. Section 4 covers the benchmark datasets Pascal Voc-2007 + 2012 Pedestrian, INRIA and Caltech; the experimental results and analysis are discussed. Finally, the conclusion is presented and future works are discussed.

2 Related Work

The research field of pedestrian detection has existed for several decades, in which different technologies have been employed for this detection, many of which have had significant impacts. Some methods aim to improve the basic features utilized [18–20], while others are intended to optimize the detection algorithms [21,22], while some other methods incorporate DPM [23] or use the advantage of context [23,24].

Benenson et al. [18] evaluated the complete performance of multifarious features and methods. Benenson et al. [20] implemented the fastest technique to achieve a frame rate of 100 frames per second (FPS) for pedestrian detection. After 2012, the deep learning era started, which has greatly improved the accuracy of pedestrian detection [21,24–26]. However, their run time on each image is slightly or markedly slower, taking a few seconds. Moreover, many remarkable techniques are now employed in CNNs. Paisitkriangkrai et al. [25] proposed new features constructed based on low-level vision features and incorporated spatial pooling to improve translational invariance which in turn improves the robustness of pedestrian detection process. The ConvNet [27] method uses convents for detecting pedestrians. It employs convolutional sparse coding to initialize each layer at the start and later performs fine-tuning to perform object detection. RPN-BF [28] is a perfect fusion of Region Proposal Networks (RPN) and Boosted Forest Classifier. RPN proposed in Faster RCNN [12] generates candidate bounding boxes, high-resolution feature maps, and confidence scores. To shape the Boosted Forest Classifier, it also employs the Real-boost algorithm for using the obtained information from RPN. This two-stage detector has shown good performance results on pedestrian test datasets. Murthy et al. [29] presented a study of pedestrian detection using various custom-made deep learning techniques.

Li et al. [30] proposed a network structure which integrates both region generation and prediction modules for accurate localization of real-time small-scale pedestrian detection. Li et al. [31] proposed scale-aware Fast-RCNN method for detecting pedestrians of various scales, and applied anchor box mechanism onto multiple feature layers. In addition, Ouyang et al. [32] proposed a unified deep neural network for jointly learning four key components, namely, feature extraction + deformation + occlusion and classification for pedestrian detection. Pang et al. [33] introduced a mask-guided attention network for detecting occluded pedestrians, which emphasizes only visible regions and suppresses occluded regions by modulating full body features. However, this method fails to achieve satisfactory results on heavily occluded pedestrians. Zhang et al. [34] proposed a simple and compact method by incorporating a channel-wise attention network on Faster RCNN detector while detecting occluded pedestrians.

Song et al. [35] proposed a novel method by integrating both somatic topological line localization and temporal feature aggregation for detecting small-scale pedestrians, which are relatively far from the camera. This method also eliminates ambiguities in occluded pedestrians by introducing a post-processing scheme based on Markov Random Field (MRF). Zhang et al. [36] proposed "key-point-guided super-resolution network" (KGSNet) for detecting small-scale and heavily occluded pedestrians. Initially, this network is trained to generate a super-resolution pedestrian image and then a part estimation module encodes the semantic information of four human body parts.

Lin et al. [37] proposed a graininess-aware feature learning method for detecting small-scale and occluded pedestrians. Attention mechanism is used to generate graininess-aware feature maps and then to enhance the features, a zoom-in-zoom-out module is introduced. Wu et al. [38] proposed a novel self-mimic loss learning method, to improve the detection accuracy of small-scale pedestrians. Hsu et al. [39] proposed a new ratio-and-scale-aware YOLO (RSA-YOLO) and achieves extremely better results while detecting small-pedestrians. Moreover, Han et al. [40] proposed a novel small-scale sense (SSN) network, which can generate some proposal regions and is effective when detecting small-scale pedestrians.

Specifically, two-stage deep learning-based object detectors offer advantages in achieving both higher localization accuracy and precision. The process requires huge resources and yet the computational efficiency is low. Owing to the unified network structures, one-stage detectors are much faster than two-stage detectors, even though the model precision decreases. Moreover, the amount of training data plays a vital role in deep learning-based object detectors. We present an end-to-end single deep neural network for detecting smaller and densely distributed pedestrians in real time inspired by YOLOv2. YOLOv2 ("You only look once version 2") [16] is an end-to-end single deep neural network that integrates feature extraction, bounding box extraction, object classification and detection. YOLOv2 is adopted as a basic model in order to achieve accuracy and higher speed when detecting smaller and densely distributed pedestrians. After making modifications in the YOLOv2 network structure and hyperparameters, it was adopted for the accurate detection of smaller and densely distributed pedestrians.

The proposed method YOLOv2PD adopts the YOLOv2 deep learning framework [16] as a base model and hyperparameters are adjusted to achieve better detection accuracy in real time. Additionally, at the higher end, some unwanted repeated convolution layers are eliminated in the proposed model, so it consumes less computational time than the YOLOv2 Model. Therefore, the YOLOv2PD model is the best method for accurate real-time detection of smaller and densely distributed pedestrians. The proposed model performance is evaluated on the Pascal Voc-2007 + 2012 Pedestrian dataset and its performance is compared with YOLOv2 and YOLOv2 Model A models. To test the robustness of the proposed model, YOLOv2PD is also evaluated on both INRIA [5] and Caltech [41] pedestrian datasets.

3 YOLOv2PD Proposed Algorithm

3.1 Anchor Boxes Selected Based on K-means Clustering

The proposed method applies a K-means clustering algorithm on the Pascal Voc-2007 + 2012 pedestrian dataset during training and selects the optimal number of anchor boxes of different sizes. It works by replacing traditional Euclidean distance with the distance function of YOLOv2 while implementing the K-means clustering algorithm. Therefore, the error obtained is made irrelevant with respect to anchor box sizes by adopting IoU as an evaluation metric, as shown in Eq. (1).

$$d\left(box, centroid\right) = 1 - IOU\left(box, centroid\right) \tag{1}$$

where box is the sample; centroid is cluster center point; IoU (box, centroid) is the overlap ratio between cluster and center boxes. Based on the clustering results analysis, the K value was chosen to be 6; therefore, six different anchor box sizes would be applied in order to improve the positioning accuracy. Finally, by implementing the K-means clustering algorithm on the training dataset, a suitable number of different anchor box sizes are selected for pedestrian detection, which in turn improves the positioning accuracy.

3.2 Improved Loss Function

Since images are captured using a video surveillance camera, some of the pedestrian images might be bigger, with pedestrians being nearer the camera, while some pedestrian images might be smaller, with pedestrians being located far away from the camera during detection. Therefore, pedestrians would appear smaller in the image when they are far from the camera, and vice versa. As such, sizes may vary in the captured images, even though the pedestrian is identical.

During YOLOv2 training, objects of different sizes show different effects on the network and produce large errors, particularly for images with smaller and densely distributed objects. To overcome this drawback, loss calculation for bounding box (BB) width and height is improved by applying normalization. Eq. (2) shows the improved loss function as:

$$\begin{aligned}
&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\prod_{ij}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(\mathrm{y}_i-\hat{\mathrm{y}}_i\right)^2\right]\\
&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\prod_{ij}^{obj}\left[\left(\frac{w_i-\hat{w}_i}{\hat{w}_i}\right)^2+\left(\frac{h_i-\hat{h}_i}{\hat{h}_i}\right)^2\right]\\
&+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\prod_{ij}^{obj}\left[\left(c_i-\hat{c}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\prod_{ij}^{noobj}\left(c_i-\hat{c}_i\right)^2\right]\\
&+\sum_{i=0}^{S^2}\prod_{i}^{obj}\sum_{c\in classes}\left[p_i(c)-\hat{p}_i(c)\right]^2
\end{aligned} \tag{2}$$

where (x_i, y_i) coordinates represent the center of the box, (w_i, h_i) coordinates are the width and height of the box, c_i is confidence prediction, and $p_i(c)$ is the conditional class probability for class c in cell i. $\hat{x}_i$, $\hat{\mathrm{y}}_i$, $\hat{w}_i$, $\hat{h}_i$, $\hat{c}_i$, and $\hat{p}_i(c)$ are the corresponding prediction values of x_i, y_i, w_i, h_i, c_i, and $p_i(c)$ and λ_{coord} corresponds to the weight of position loss, with a value of 5, λ_{noobj} corresponds to the weight of the classification loss, with a value of 0.5, S^2: $S \times S$ grid cells, B: bounding boxes (BBs), $\prod_{ij}^{obj}=1$, corresponds to the j^{th} BB in cell i that is responsible for detecting the pedestrian, else 0, $\prod_{i}^{obj}=1$, if the pedestrian is located in the cell i, else 0, From Eq. (2), the first term determines the BB localization loss error, the second term determines the BB confidence loss error with objects and without objects, and the third term determines the classification loss error. Eq. (2) in the proposed method is compared with original YOLOv2 [16] $\frac{w_i-\hat{w}_i}{\hat{w}_i}$ and $\frac{h_i-\hat{h}_i}{\hat{h}_i}$ term is used instead of $w_i-\hat{w}_i$ and $h_i-\hat{h}_i$, which would reduce the effect of different pedestrian sizes in an image, and which in turn potentially optimizes the detected BB.

3.3 Network Design

Multi-layer Feature Fusion (MLFF) Approach: In pedestrian detection, variations among pedestrians include occlusion, illumination changes, color, height, and contour, whereas local features exist only in the lower layers of CNN. Therefore, to use local features fully, an MLFF approach was implemented in YOLOv2PD. The Reorg aim is to keep feature maps of those layers the same. Part (a) passes through the following 3 × 3 and 1 × 1 convolution layers and then a down-sampling factor of Reorg/8 is applied, as shown in Fig. 1. Similarly, part (b) and part (c) perform the same operations, but with down-sampling factors of 4 and 2, respectively. Part (a), (b) local features, and part (c) global features of one layer are fused. This is done so that the network would distinguish the tiny differences among pedestrians and also it improves the network understanding of local features.

YOLOv2 is a fast and accurate object detection model. The YOLOv2 network can detect 9000 classes and variations among multiple objects are wider, such as cell phones, cars, fruits, sofas, and dogs. There are three repeated 3 × 3 × 1024 convolutional layers in the YOLOv2 network. Generally, at the higher end, repeated convolution operation deals with multiple classes and widely differing objects, such as fruits, animals, and vehicles. However, our main concern is only detecting the pedestrian class and feature differences among pedestrians are minute. Thus, the model performance may not improve due to repeated convolution layers at the higher end and, due to their presence, the model becomes more complex. Therefore, repeated convolution layers are removed from the higher end in the proposed models. This strategy would achieve almost competitive performance and reduce the time complexity of the Yolov2 network. Thus, three repeated $3 \times 3 \times 1024$ convolution layers are reduced to two in the proposed model, as shown in Fig. 1.

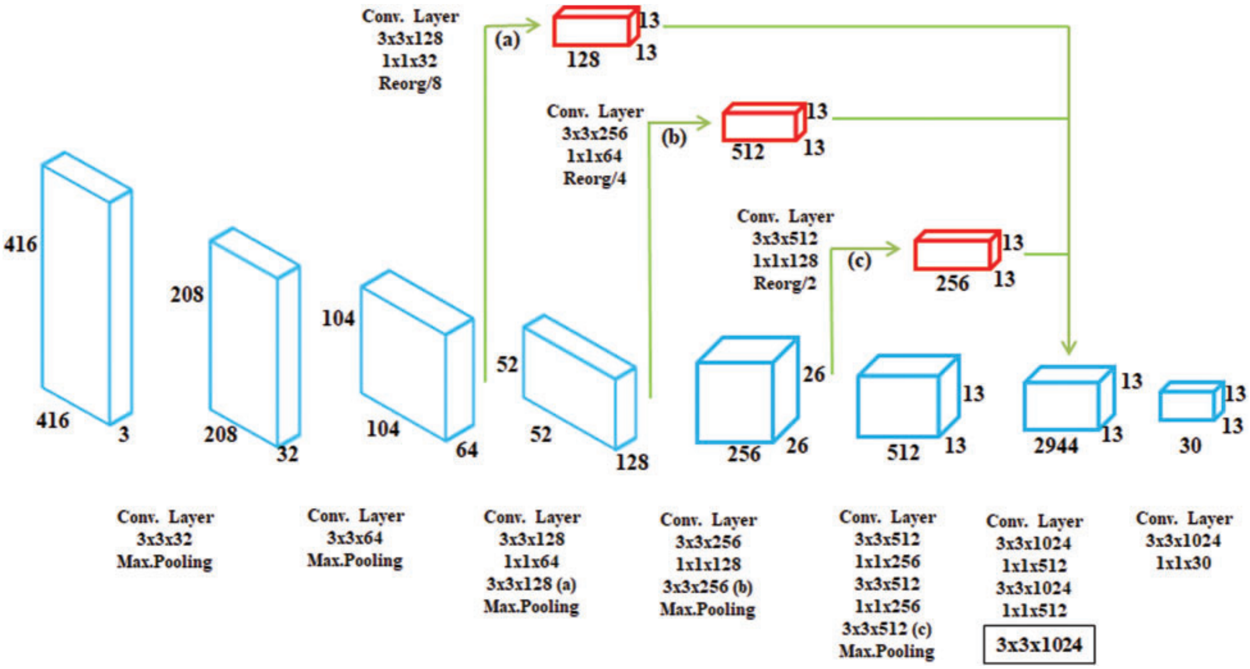


Figure 1: YOLOv2PD network architecture

A novel YOLOv2PD network structure is designed by adopting the MLFF approach and one unwanted convolutional layer is removed at the higher end. Moreover, intuitively, to test the effectiveness of the proposed model, another model, referred to as YOLOv2 Model A, was implemented and compared. The YOLOv2 Model A removed two $3 \times 3 \times 1024$ convolution layers and the YOLOv2PD model removed only one $3 \times 3 \times 1024$ convolution layer when compared with the YOLOv2 network. Tab. 1 shows the comparison between YOLOv2, YOLOv2 Model A, and YOLOv2PD network architecture.

Table 1: YOLOv2, YOLOv2 Model A, and YOLOv2PD network architecture

Layer No.	YOLOv2	YOLOv2 Model A	YOLOv2PD
L0	Conv_3*3_416*416*32	Conv_3*3_416*416*32	Conv_3*3_416*416*32
L1	Maxpool/2	Maxpool/2	Maxpool/2
L2	Conv_3*3_208*208*64	Conv_3*3_208*208*64	Conv_3*3_208*208*64
L3	Maxpool/2	Maxpool/2	Maxpool/2
L4	Conv_3*3_104*104*128	Conv_3*3_104*104*128	Conv_3*3_104*104*128
L5	Conv_1*1_104*104*64	Conv_1*1_104*104*64	Conv_1*1_104*104*64
L6	Conv_3*3_104*104*128	Conv_3*3_104*104*128	Conv_3*3_104*104*128
L7	Maxpool/2	Maxpool/2	Maxpool/2
L8	Conv_3*3_52*52*256	Conv_3*3_52*52*256	Conv_3*3_52*52*256
L9	Conv_1*1_52*52*128	Conv_1*1_52*52*128	Conv_1*1_52*52*128
L10	Conv_3*3_52*52*256	Conv_3*3_52*52*256	Conv_3*3_52*52*256
L11	Maxpool/2	Maxpool/2	Maxpool/2
L12	Conv_3*3_26*26*512	Conv_3*3_26*26*512	Conv_3*3_26*26*512
L13	Conv_1*1_26*26*256	Conv_1*1_26*26*256	Conv_1*1_26*26*256
L14	Conv_3*3_26*26*512	Conv_3*3_26*26*512	Conv_3*3_26*26*512
L15	Conv_1*1_26*26*256	Conv_1*1_26*26*256	Conv_1*1_26*26*256
L16	Conv_3*3_26*26*512	Conv_3*3_26*26*512	Conv_3*3_26*26*512
L17	Maxpool/2	Maxpool/2	Maxpool/2
L18	Conv_3*3_13*13*1024	Conv_3*3_13*13*1024	Conv_3*3_13*13*1024
L19	Conv_1*1_13*13*512	Conv_1*1_13*13*512	Conv_1*1_13*13*512
L20	Conv_3*3_13*13*1024	Conv_3*3_13*13*1024	Conv_3*3_13*13*1024
L21	Conv_1*1_13*13*512	Conv_1*1_13*13*512	Conv_1*1_13*13*512
L22	**Conv_3*3_13*13*1024**	**Conv_3*3_13*13*1024**	**Conv_3*3_13*13*1024**
L23	**Conv_3*3_13*13*1024**	Route-L16	**Conv_3*3_13*13*1024**
L24	**Conv_3*3_13*13*1024**	Conv_3*3_13*13*512	Route-L6
L25	Route-L16	Conv_1*1_13*13*64	Conv_3*3_13*13*128
L26	Conv_1*1*_13*13*64	Reorg	Conv_1*1_13*13*32
L27	Reorg	Route-L26 L22	Reorg
L28	Route-L27 L24	Conv_3*3_13*13*1024	Route-L10
L29	Conv_3*3_13*13*1024	Conv_1*1_13*13*30	Conv_3*3_13*13*256
L30	Conv_1*1_13*13*30	Detection	Conv_1*1_13*13*64
L31	Detection		Reorg
L32			Route-L16
L33			Conv_3*3_13*13*512
L34			Conv_1*1_13*13*64
L35			Reorg
L36			Route-L35 L31 L27 L23
L37			Conv_3*3_13*13*1024
L38			Conv_1*1_13*13*30
			Detection

4 Datasets and Experimental Results

4.1 Datasets

Pascal Voc-2007 + 2012 dataset [42]: This dataset contains 20 object classes and around 17,125 labeled images; it is a complete dataset generally used for object detection and classification. An unsupervised learning method (K-means clustering) is applied during training. Since manual annotation of a dataset is a complex and huge project, around 10,080 pedestrian and non-pedestrian images (referred to as the Pascal Voc-2007+2012 Pedestrian dataset) were extracted from Pascal dataset [42].

The INRIA Pedestrian dataset [5] contains 1826 pedestrians, with image resolution 64 × 128. The pedestrian images captured in this dataset possess a complex background, illumination changes, various degrees of occlusion, variations in human posture, and individuals wearing different clothes.

The Caltech pedestrian dataset [41] contains a set of video sequences of 640 × 480 in size captured from an urban environment. It includes training (set 00 to set 05) subsets and testing (set 06 to set 10) subsets. It contains 250 k video frames, 350 k bounding boxes and 2.3 k pedestrians ("person" or "people" labels) are annotated. The training dataset is formed by extracting every image after every 30 frames from set 00 to set 05 and testing images are extracted from set 06 to set 10. Tab. 2 shows the datasets used for both training and testing of the proposed algorithm.

Table 2: Datasets used by the proposed algorithm for Training & Testing

Datasets	Training Images	Testing Images
Pascal Voc-2007 + 2012 Pedestrian	9072	1008
INRIA	614	228
Caltech Pedestrian	4250	4024

4.2 Experimental Setup

The experiments were carried out on a workstation during the training phase; the testing phase was also performed on the same workstation. Darknet was chosen as a feature extractor for all of the models, which was trained on a huge ImageNet dataset. The experimental setup of the workstation is Windows 10 pro OS, Intel Xeon 64-bit CPU @3.60 GHz, 64 GB RAM, Nvidia Quadro P4000 GPU, CUDA 10.0 & CUDNN 7.4 GPU acceleration library and Tensorflow 1.x deep learning framework.

4.3 Training and Evaluation Metrics

The model training was carried out on Pascal Voc-2007+2012 Pedestrian dataset (9072) training images and tested on 1008 testing images, since we are only concerned with pedestrian images. The input image size is resized to 416 × 416 resolution and various data augmentation techniques are applied, such as color shifting, flipping, cropping, and random sampling, in order to enhance the training process. All of the three models are trained for 40 epochs, with an initial learning rate of 0.001, and later learning rate is divided by 10 at 60 and 80 epochs respectively. During the model training, it randomly selects a new input image of different resolution after every 20 epochs. Since multi-scale training strategy improves model robustness, so it can perform better prediction on images with different resolutions. While training, Caltech dataset, the original images are up-sampled to 1024×1024 pixels, one mini-batch contains 16 images, learning rate is 10-4 and the model training is stopped after 80 epochs.

Average precision (AP) and inference speed (FPS-Frames per second) are the standard techniques preferred to evaluate the model performance. Intersection over union (IoU) is a good evaluation metric used to measure the accuracy of the designed model on a test dataset. IoU is simply computed as the area of intersection divided by the area of union. IoU helps to determine whether a predicted BB is a True Positive (TP), False positive (FP) or False Negative (FN) by defining a threshold of ≥ 0.5.

Recall: A measure of how good the model is at finding all of the positives. Precision: A measure of the accuracy of our predictions. These two terms are inversely proportional to each other.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4}$$

AP: This is the area under the precision–recall curve, which shows the correlation between precision and recall at different confidence scores. A higher AP value indicates better detection accuracy.

The performance of the model while validating INRIA and Caltech test datasets was visualized using a plot between the number of false positives per image and the miss rate (MR). The ratio between the number of FNs and the total number of positive samples (N) is referred to as the MR.

$$\text{Miss rate (MR)} = \text{FN}/\text{N} \tag{5}$$

There is another relationship between the miss rate and recall expressed as:

$$\text{Recall} = 1 - \text{Miss rate (MR)} \tag{6}$$

4.4 Results and Analysis

Fig. 2 shows the analysis of the training stage of all three models. The y-axis indicates average loss and the x-axis indicates the number of iterations performed in training. It is clear from Fig. 2 that the average loss curve is not stable up to approximately 10000 iterations. Compared with all of the other models, the average loss curve of the YOLOv2PD model decreases faster initially, followed by that of YOLOv2 Model A. The reason for this is that both YOLOv2PD and YOLOv2 Model A adopted a multi-layered feature fusion strategy, so they obtained more local features, which accelerated the training convergence. During the training stage, initially the YOLOv2PD model first reached a minimum average loss value (overall lowest value = 0.54), followed by YOLOv2 Model A and YOLOv2 models. Therefore, the YOLOv2PD model is more suitable for detecting small pedestrians on the Pascal Voc-2007 + 2012 pedestrian dataset.

Fig. 3 shows the precision *vs.* recall (PR) curve obtained on the Pascal Voc-2007 + 2012 pedestrian dataset of all three models. The graph shows that, with increasing recall value at the convergence point, the precision gradually starts decreasing.

With different input image resolutions of 416×416, 544×544, and 608×608, YOLOv2PD achieves comparable detection performance when compared with YOLOv2 Model A and

YOLOv2. Tab. 3 compares the detection performance of all models for different image resolutions with respect to AP and inference speed (FPS) parameters. The proposed network YOLOv2PD achieves AP, that is, detection performance of 79.5, 80.7, and 82.3 respectively. From these results, it is clear that, as the applied input image resolution increases, the AP value increases but at the same time inference speed decreases.

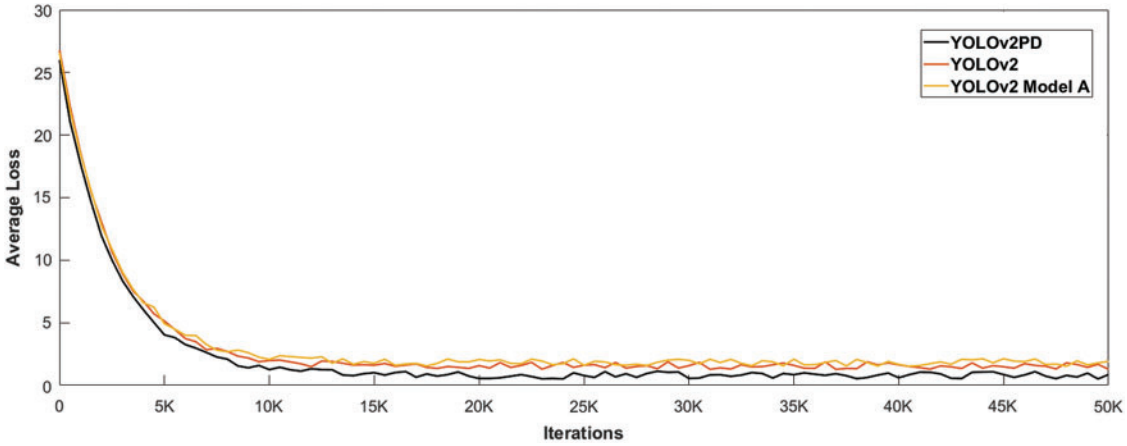


Figure 2: Analysis of training stage of all of the models

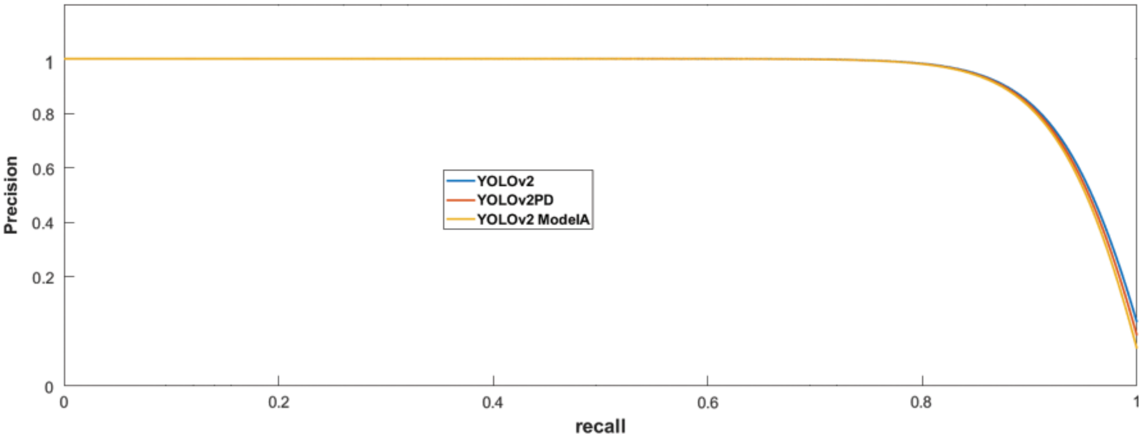


Figure 3: PR curves of all of the models on the Pascal Voc-2007 + 2012 pedestrian dataset

To have a model that runs at higher inference speed, an image size of 416 × 416 is the best choice. As the input image size increases, inference speed decreases since these terms are directly proportional to each other. However, we are concerned with detecting smaller and densely distributed pedestrians, so 416 × 416 images are not quite suitable as they miss the detection of many smaller objects. Therefore, we consider selecting a 544 × 544 image size for detecting smaller and densely distributed pedestrians. From the experimental results, our proposed algorithm runs at 36.3 FPS in real time on 544 × 544 image resolution. In this study, if the AP is considered, then an image size of 544 × 544 would be the best choice as the proposed model achieves 80.7% detection accuracy, which is 2.1% higher than that of YOLOv2 [16]. The proposed model runs at

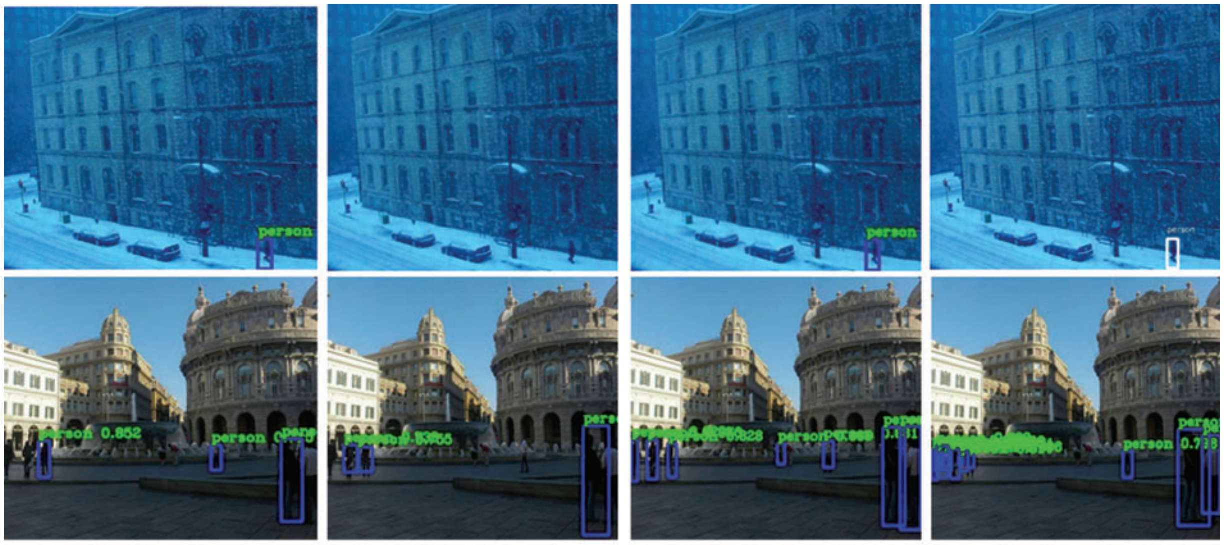
30.6 FPS for the 608 × 608 image resolution, but the inference speed falls by 5.7 FPS compared to 544 × 544 image resolution.

Table 3: Evaluation results of all of the models on the pedestrian test dataset (IoU@0.5)

Input Size	Model	Average Precision (AP)	Inference Speed (FPS)
416 × 416	YOLOv2	75.2	45.1
	YOLOv2 Model A	77.1	64
	YOLOv2PD	79.5	47.2
544 × 544	YOLOv2	76.5	32
	YOLOv2 Model A	78.3	38.2
	YOLOv2PD	**80.7**	**36.3**
608 × 608	YOLOv2	78.2	26.1
	YOLOv2 Model A	80.4	32.1
	YOLOv2PD	82.3	30.6

4.5 Small Pedestrian Detection

The Pascal Voc-2007 + 2012 pedestrian dataset contains 20 different classes and every class may have small objects. We were concerned with detecting smaller and densely distributed pedestrians in this dataset, so we manually picked up 330 images that mainly included smaller pedestrians to evaluate the model performance. Fig. 4 shows detection results of all models and compared with YOLOv3 [43] SOTA detector. From these detection results, it is evident that the proposed model can produce better prediction on smaller and densely distributed pedestrians than the other models.



(a) (b) (c) (d)

Figure 4: Detection results of YOLOv2, YOLOv2 Model A, YOLOv2PD and YOLOv3 Models

The evaluation results of all three models on the INRIA test dataset are expressed in terms of average precision and inference speed (milliseconds). Tab. 4 shows detected results on the INRIA test dataset for different image resolutions. At 544 × 544 test image resolution, the proposed model achieves 91.2% AP, which constitutes an improvement by 6.6% and 11.4% compared with YOLOv2 Model A and YOLOv2 models, respectively. This is because our model uses the MLFF strategy while detecting smaller pedestrians.

Table 4: Detection results of all of the models on the INRIA Test dataset. (IoU@0.5)

Input size	Model	Average precision (AP)	Inference speed (ms)
544 × 544	YOLOv2	79.8	27.4
	YOLOv2 Model A	84.6	24.7
	YOLOv2PD	**91.2**	**25.6**
608 × 608	YOLOv2	82.5	36.3
	YOLOv2 Model A	87.1	27.8
	YOLOv2PD	93.4	26.5

To test the robustness of the proposed model, we compared our model performance on the INRIA pedestrian test dataset with several SOTA algorithms.

Tab. 5 shows a comparison of the YOLOv2PD model performance with the advanced existing algorithms evaluated in terms of average MR and runtime (FPS) on a reasonable test dataset. Our model achieves better detection performance than YOLOv2 [16], Spatial Pooling [25] and Y-PD [44] and is improved by 4.7%, 3.4% and 1.3% respectively, but lags behind YOLOv3 [43] and F-DNN [45] by 0.6% and 1% respectively. Obviously, on the INRIA pedestrian test dataset, the proposed model achieves a better trade-off balance between speed and accuracy when detecting pedestrians.

Table 5: Comparison of YOLOv2PD results with recent SOTA methods on the INRIA test dataset

Models/Avg.MR (%)	Reasonable	Runtime (FPS)
VJ [3]	72.5	<1
HOG [5]	46	<1
YOLOv2 [16]	12.5	32
Very fast [20]	16	**>100**
Spatial pooling [25]	11.2	<1
RPN + BF [28]	6.9	~4
YOLOv3 [43]	7.2	20
Y-PD [44]	9.1	73
F-DNN [45]	**6.8**	~6
Proposed	7.8	36.3

Tab. 6 shows a comparison of the proposed model performance with the advanced existing algorithms on the Caltech test dataset, evaluated in terms of MR, average precision, and detection speed.

Table 6: Comparison of YOLOv2PD detection results with recent SOTA methods on the Caltech test dataset (IoU@0.75)

Models/LAMR (%)	Reasonable	Average Precision (AP)	Runtime (s)
RPN + BF [28]	9.580	0.324	0.50
SA-FastRCNN [31]	9.680	0.344	0.59
UDN + SS [32]	11.520	0.331	0.28
M-GAN [33]	**6.830**	–	–
Faster RCNN + ATT-Vbb [34]	10.330	–	–
TTL(MRF) + LSTM [35]	7.400	–	–
SSNet [40]	8.920	0.360	0.43
Y-PD [44]	18.4	0.321	–
SDS-RCNN [46]	7.360	0.355	**0.21**
CompactACT + Deep [47]	11.750	0.334	1.00
Proposed	7.480	**0.381**	0.29

From Tab. 6, it is clear that, on the Caltech test dataset, the proposed model has better detection performance than RPN + BF [28], SA-FastRCNN [31], UDN + SS [32], Faster RCNN + ATT-Vbb [34], SSNet [40], Y-PD [44] and CompactACT + Deep [47], and models on the reasonable subset [h ∈ (50, ∞)]. However, the proposed model average miss rate falls behind those of M-GAN [33], TTL (MRF) + LSTM [35] and SDS-RCNN [46] models by 0.65%, 0.80% and 0.12% respectively.

To show the findings more intuitively, regarding the real-time performance of the proposed algorithm to achieve a perfect balance between detection speed and accuracy, we fed a real-time test video to all models. The detection results of the randomly selected 79th frame for all of the models are shown in Fig. 5. We evaluated the running time for these three models on a real-time input test video. The detection speed on an input image of size 544 × 544 was 32 FPS for YOLOv2, 38.2 FPS for YOLOv2 Model A, 36.3 FPS for YOLOv2PD and 20 FPS for YOLOv3. Although the proposed model runs in real-time, it fails to detect smaller and similar occluded pedestrians. The use of the Internet of Things may make the method more efficient [48].

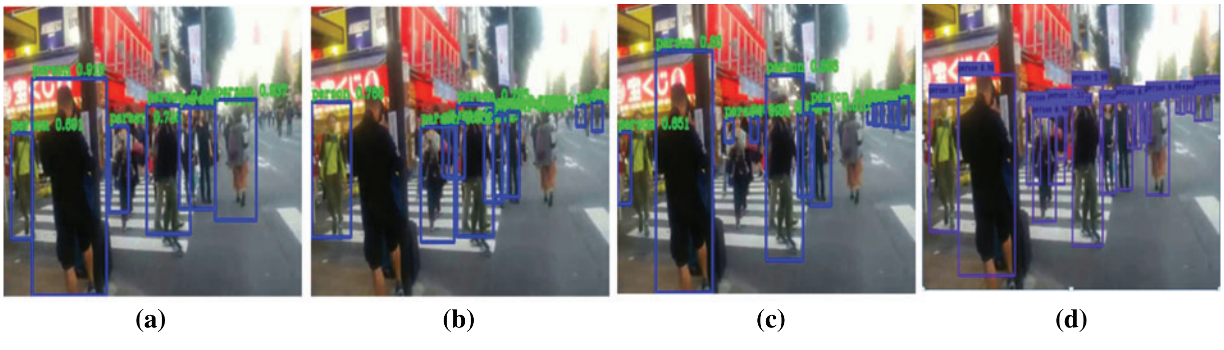


Figure 5: Real-time detection results of YOLOv2, YOLOv2 Model A, YOLOv2PD and YOLOv3 Models

5 Conclusion

A new advanced model named YOLOv2PD was proposed for the accurate detection of smaller and densely distributed pedestrians. The proposed network YOLOv2PD structure was designed to improve the network's feature extraction ability by adopting the MLFF strategy and, at the higher end, one repeated convolutional layer was removed. To improve the detection accuracy while detecting smaller and more densely distributed pedestrians, the loss function was improved by applying normalization. The experimental results show that, for an applied input image of 544 × 544 in size, the proposed algorithm achieves 80.7% AP, which is 2.1% higher than that of the YOLOv2 Model on the Pascal Voc-2007+2012 pedestrian test dataset. To test the robustness of the proposed algorithm, we captured a real-time video and fed it images at 544 × 544 resolution; it obtained 36.3 FPS and achieved 80.7% detection accuracy compared with the SOTA YOLOv2 Model. The experimental results show that the proposed model achieves 7.8 average MR on INRIA and 0.381 AP on Caltech pedestrian test datasets. Although the model was run in real time, there is still room for improvement of the speed, miss rate on INRIA test dataset and miss detection of small similar and occluded pedestrians.

Funding Statement: The authors are grateful to the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia, for funding this work through the Vice Deanship of Scientific Research Chairs: Research Chair of Pervasive and Mobile Computing.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Zhang, R. Benenson, J. Hosang and B. Schiele, "How far are we from solving pedestrian detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 1259–1267, 2016.

[2] S. Zhang, R. Benenson, M. Omran, J. Hosang and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 973–986, 2018.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, HI, USA, pp. I-I, 2001.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 886–893, 2005.

[6] N. Muhammad, M. Hussain, G. Muhammad and G. Bebis, "Copy-move forgery detection using dyadic wavelet transform," in *2011 Eighth Int. Conf. Computer Graphics, Imaging and Visualization*, Singapore, pp. 103–108, 2011.

[7] G. Muhammad, M. S. Hossain and N. Kumar, "EEG-based pathology detection for home health monitoring," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 603–610, 2021.

[8] G. Muhammad, M. F. Alhamid and X. Long, "Computing and processing on the edge: Smart pathology detection for connected healthcare," *IEEE Network*, vol. 33, pp. 44–49, 2019.

[9] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.

[10] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[11] R. Girshick, "Fast R-CNN," in *IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 1440–1448, 2015.

[12] 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, no. 6, pp. 1137–1149, 2016.

[13] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 2980–2988, 2017.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot multibox detector," in *European Conf. on Computer Vision*, Cham, Springer, pp. 21–37, 2016.

[15] J. Redmon, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779–788, 2016.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, pp. 6517–6525, 2017.

[17] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. of the 18th Annual ACM-SIAM Sym. on Discrete Algorithms*, New Orleans, LA, USA, pp. 1027–1035, 2007.

[18] R. Benenson, M. Omran, J. Hosang and B. Schiele, "Ten years of pedestrian detection, what have we learned," in *European Conf. on Computer Vision*, Cham, Springer, pp. 613–627, 2014.

[19] A. D. Costea and S. Nedevschi, "Word channel based multiscale pedestrian detection without image resizing and using only one classifier," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 2393–2400, 2014.

[20] R. Benenson, M. Mathias, R. Timofte and L. Van Gool, "Pedestrian detection at 100 frames per second," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 2903–2910, 2012.

[21] P. Luo, Y. Tian, X. Wang and X. Tang, "Switchable deep network for pedestrian detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 899–906, 2014.

[22] R. Appel and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Proc. of the European Conf. on Computer Vision*, Berlin, GA, Springer, pp. 645–659, 2013.

[23] J. Yan, X. Zhang, Z. Lei, S. Liao and S. Z. Li, "Robust multi-resolution pedestrian detection in traffic scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 3033–3040, 2013.

[24] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *IEEE Int. Conf. on Computer Vision*, Sydney, NSW, Australia, pp. 2056–2063, 2013.

[25] S. Paisitkriangkrai, C. Shen and A. Van Den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *European Conf. on Computer Vision*, Cham, Springer, pp. 546–561, 2014.

[26] X. Zeng, W. Ouyang and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *IEEE Int. Conf. on Computer Vision*, Sydney, NSW, Australia, pp. 121–128, 2013.

[27] C. Wojek, S. Walk and B. Schiele, "Multi-cue onboard pedestrian detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 794–801, 2009.

[28] L. Zhang, L. Lin, X. Liang and K. He, "Is faster R-CNN doing well for pedestrian detection?," in *Proc. of the European Conf. on Computer Vision*, Cham, Springer, pp. 443–457, 2016.

[29] C. B. Murthy, M. F. Hashmi, N. D. Bokde and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms-A comprehensive review," *Applied Sciences*, vol. 10, no. 9, pp. 3280, 2020.

[30] Z. Li, Z. Chen, Q. J. Wu and C. Liu, "Real-time pedestrian detection with deep supervision in the wild," *Signal Image and Video Processing*, vol. 13, no. 4, pp. 761–769, 2019.

[31] J. Li, X. Liang, S. Shen, T. Xu, J. Feng *et al.,* "Scale-aware fast R-CNN for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.

[32] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan *et al.,* "Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1874–1887, 2018.

[33] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. S. Khan *et al.,* "Mask-guided attention network for occluded pedestrian detection," in *IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea (South), pp. 4966–4974, 2019.

[34] S. Zhang, J. Yang and B. Schiele, "Occluded pedestrian detection through guided attention in CNNs," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp. 6995–7003, 2018.

[35] T. Song, L. Sun, D. Xie, H. Sun and S. Pu, "Small-scale pedestrian detection based on somatic topology localization and temporal feature aggregation," arXiv preprint arXiv: 1807.01438, 2018.

[36] Y. Zhang, Y. Bai, M. Ding, S. Xu and B. Ghanem, "KGSNet: Key-point-guided super-resolution network for pedestrian detection in the Wild," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 1–15, 2020.

[37] C. Lin, J. Lu, G. Wang and J. Zhou, "Graininess-aware deep feature learning for robust pedestrian detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 3820–3834, 2020.

[38] J. Wu, C. Zhou, Q. Zhang, M. Yang and J. Yuan, "Self-mimic learning for small-scale pedestrian detection," in *Proc. of the 28th ACM Int. Conf. on Multimedia*, Seattle, WA, USA, pp. 2012, 2020.

[39] W. Y. Hsu and W. Y. Lin, "Ratio-and-scale-aware YOLO for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 934–947, 2021.

[40] B. Han, Y. Wang, Z. Yang and X. Gao, "Small-scale pedestrian detection based on deep neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3046–3055, 2020.

[41] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian detection: A benchmark," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 304–311, 2009.

[42] X. Du, M. El-Khamy, V. I. Morariu, J. Lee and L. Davis, "Fused deep neural networks for efficient pedestrian detection," arXiv preprint arXiv: 1805.08688, 2018.

[43] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv: 1804.02767, 2018.

[44] Z. Liu, Z. Chen, Z. Li and W. Hu, "An efficient pedestrian detection method based on YOLOv2," *Mathematical Problems in Engineering*, vol. 1, no. 4, pp. 1–10, 2018.

[45] Z. Cai, M. Saberian and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 3361–3369, 2015.

[46] G. Brazil, X. Yin and X. Liu, "Illuminating pedestrians via simultaneous detection & segmentation," in *IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 4950–4959, 2017.

[47] M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[48] F. Alshehri and G. Muhammad, "A comprehensive survey of the Internet of Things (IoT) and AI-based smart healthcare," *IEEE Access*, vol. 9, pp. 3660–3678, 2021.