Multiple Pedestrian Detection Using Faster Region-based Convolutional Neural Network (RCNN)

Ghalia . S. Shariha[1], Mohammed .M. Elmogy[2], Asma. g. Tawil[3]

High Institute of science and technology kabaw, Kabaw, Libya [1]
Faculty of Computersand Information,Mansoura University[2]
College of Engineering Technology, Janzour, Libya[3]

Abstract

Pedestrian detection plays a crucial role in security, intelligent surveillance, vehicles, and robotics. Occlusion handling is a challenging worry in tracking multiple people. The tracking is based on highest accuracy object detectors. This paper is aimed at proposing a structure that has the ability to identify many pedestrians in an image, and is operated on the R-CNN algorithm (Faster Region Based Conventional Neural network. The program was developed using a VGG19 deep network which was also pre-trained using Image-net thus extracting the feature map with small learning rate. Depending on the trained weights, the training time was reduced using transfer learning. The structure was tested and trained on the Penn Fudan database for pedestrians. The research further evaluated the accuracy of pedestrian detection by use of ROC (Receiver Operating Characteristic) curve AUC that was able to attain 91.9%. Furthermore, the projected structure attained specificity of 91.3%, 91.9% F1 score, 6.6% sensitivity, and 96.9% accuracy (ACC).

Keywords: Pedestrian Detection; Multiple Pedestrians; Deep learning; Faster R-CNN (Region-based Convolutional Neural Network).

1. Introduction

Pedestrian detection falls under the division of target direction, and the pedestrian is the tag that it is supposed to sense. Target detection is aimed at perfectly getting an object's location in a particular image and also marks the group of the object. Additionally, it is considered as the initial procedure for certain applications, including driving assistance structures, intelligent digital management of content and smart video surveillance. Target direction is aimed at solving the challenge of the status of the target and where it is. Nevertheless, it is not easy to get a solution for this challenge because it can also appear in different categories and also because of differences in color, illumination, pose and scale.

This procedure can be undertaken through conducting a search and indexing images that are still which have an object in different positions, sizes and backgrounds. The target for detection contains two diverse directions, deep learning algorithms and traditional algorithms. Archetypal representations of traditional algorithms include a HOG (Histogram of Oriented Gradients) that was formulated by Dalal et al [1]. HOG aspects are computationally economical and are helpful for most

challenges that are faced in the real world. On every window that is obtained from operating the sliding window through the pyramid, it computes HOG aspects that are transmitted to a support vector device (SVM) with an aim of creating classifiers. Through the program real time videos were run and indicated face detection, pedestrian detection as well as other items for object detection.

Another invention that was conducted by Viola et al [2] developed a person detector that had the ability to move efficiently by use of AdaBoost algorithm [3]. A chain of complex and progressive location rejection regulations was trained on the foundation of Haar-like aspects [4], space-time variations and wavelets. According to Papageorgiou et al. [5] the description of a device that could detect pedestrians on the basis of SVM polynomial by use of Haar wavelets that are recited as descriptors of input indicated sub windows that were variant-based [6].

Pedestrian detection that is done on the basis of deep learning is characteristically seen as a representation of RCNN algorithm series and is mainly used for purposes of deep learning that is founded on the algorithm of pedestrian tracking. According to Girshick et al [7], the Region-based Convolutional Neural Network (RCNN) is made of two procedures. The first process is the SS (selective search) [8] which are responsible for the identification of bounding boxes that are manageable in figures on candidates of the object region. Additionally, it autonomously takes out CNN aspects from every location for purposes of categorization. Another concern is that RCNN was found to be slow since operating CNN on the 2000 location proposals affected the generation of SS. With the use of SPP-net (Spatial Pyramid Pooling), Zhang et al. [9] computed the representation of CNN for the whole image just once which was later used in the calculation of CNN representation for every piece that was spawn by SS. This procedure can be undertaken by doing a pooling form of operation on only one segment of the feature maps in the Conv (last convolution layer) that matches up to the specific location. The area that is rectangular and which relates to a location can be computed through a projection of the location on the Conv layer through considering the occurrence of down sampling in the intermediate location which can be attained through simple division of the coordinates by sixteen in case there is VGG.

Nevertheless, the system presented a major challenge when SPP net was used because it failed to do back-propagation through the layer of spatial pooling. Additional ideas of R-CNN and SPP-net to boost the process of training through the unification of three diverse models into a single joint framework thus increasing computation results that are shared known as Fast R-CNN were used by Girshik et al. [10]. In place of the extraction of CNN aspect vectors in an independent way for every location that was proposed, the model combined them into a single CNN forward and later passes them into the entire image whereby the region proposals divide the matrix feature. Later, the matrix feature is divided for use in learning the classification of objects and also the repressor of bounding-box.

According to Lin et al. [16], Calculations can be used in increasing R-CNN. This claim was proved using an uncomplicated structure that developed feature pyramids within FPN (ConvNets). This

methodology illustrates essential developments over certain baselines which are long. Therefore, it offers a realistic remedy for applications and research of other pyramids, without necessarily calculating image pyramids. The study finally indicates that regardless of the coherent representative authority of ConvNets that are deep as well as their inherent sturdiness to the difference in scale, it remains significant to address multi-scale challenges in an explicit way using the representations of

pyramids. Another proposal that was made by He et al. [15] suggested the extension of Mask R-CNN to further the quicker R-CNN to the division of pixel-image representations. The main idea was to ensure the decoupling of pixel-level and classification tasks of prediction mask. On the basis of Faster R-CNN program, it ensured the addition of the third piece for object mask prediction which worked parallel to the current branches for localization and classification. The mask division is considered as an insignificant network that is entirely connected and is applied to every location of interest (RoI), thus forecasting the amount of alignment required compared to bounding boxes. Consequently, mask R-CNN was used in the improvement of the RoI layer of pooling that precisely and in a better way map locations on the original figure.

When conducting an appraisal for the works illustrated above in the object detection location, the most significant difficulties should be taken into consideration when developing a pedestrian tracking system that is new. These aspects include;

1. Increased accuracy in the detection of pedestrians particularly in scenes that are cluttered as well as those that have variations that are illuminated.
2. Quick achievement to ensure that real-time applications are used.

Our distribution use faster-RCNN algorism for detection pedestrian with a little dataset and reduced the train time when take weight learning from VGG19 architecture of CNN.

The remaining part of this paper includes Section 2 which explains the basic concept of the program, section 3 which explains the projected structure applied during the presentation in the construction of the projected structure and is applied in the research to ensure construction of the proposed structure. The paper also has section 4 which is a description of the experimental findings. The paper ends with section 5 which is a summary and conclusion of the entire presentation as well as future directions in this study.

2. Basic Concept

The pedestrian detection system includes many methods, even though these methods have a pipeline through which they can be compared and which is separated into three key stages; feature extraction, region proposal, and classification of the region.

2.1 Convolutional Neural Network (CNN)

CNN is regarded as a fashionable algorithm for classification of an image and characteristically includes the layer of activation function, complication layer, the layer of activation function and pooling which involves average and max-pooling with the objective of reducing dimensionality without wasting most aspects. There is an aspect map that is created by the final stratum of an involvedness layer. Most models that are pre-trained are created for direct application without undergoing the process of training replicas because of limitations in computation. Most replicas became fashionable similar to the VGG19 [12], VGG16 ResNet101 [18], ImageNet [17], ResNet50, AlexNet [19] that contains more than a million images.

The prototypes that have been identified for use in this paper were VGG19 and VGG16. The network used in VGG19 includes 47 levels, and three (FC) layers that are entirely connected. Furthermore, the VGG16 has 41 levels; whereby 16 of them have weights that are learnable, 13 have Conv layers and 3 have FC layers. Both can be classified as images into 1000 groups, including mouse, keyboard, and

human, many animals, and pencil. As a consequence, the network was able to learn the prominent aspect representations for many images.

2.2 Faster-RCNN

The Faster RCNN [11] was partitioned into two distinct phases with the first one being aimed at generating a thin set of object locations using a minimal network for convolution known as Region Proposal Network (RPN) [13]. The subsequent phase categorizes the location of every candidate as the foreground class or a background through the use of CNN.

The main implication in faster R-CNN by using RPN is to ensure generation of candidate locations that are faster than edge box (EB) [20] and SS. Through irregular training, the RPN that uses Fast-RCNN networks have the ability to share restrictions.

Fast R-CNN system assumes as input the whole image as well as a collection of proposal objects. The network initially recommends the entire image with certain Conv and layers of max-pooling thus producing a map for Conv feature. After RPN, we are able to attain the proposed locations in diverse sizes. The different locations which have dissimilar sizes indicate that there are diverse CNN feature maps. The location of interest pooling is used for the simplification of the challenge through reduction of the aspect maps into similar sizes. Dissimilar to Max-Pooling which contains a size that is fixed, ROI Pooling tears the input aspect map into a number that is fixed of segments that are almost equal, thus applying Max-Pooling on all locations. Consequently, the effect of ROI Pooling is usually standard, despite the size of its input.

Additionally, every feature vector is placed into a series of FC levels that are grouped into two distinct layers of output; one which is responsible for the production of estimates for soft-max probability over the class of pedestrians and another consequent level that is responsible for the output of four digits that have real values for every pedestrian who is identified. Every set of four values that is encoded is refined with positions of bounding boxes for pedestrians who are detected.

Pseudo Code: The following pseudo-code summarizes Faster-RCNN method.

```
1. Inputs: Images
2. feature_maps = process(image)
3. ROIs = region_proposal(feature_maps)
4. for ROI in ROIs
5.    patch = roi_pooling(feature_maps, ROI)
6.    class_scores, box = detector(patch)
7.    class_probabilities =
   softmax(class_scores)
8. end
9. Outputs: Classifications and bounding box
   coordinates of objects in the images.
```

3. The Proposed Framework

The function of this system is to detect a pedestrian. The most significant idea of the entire structure relies on the Trained Faster-RCNN [11] algorithm which has a dataset that is labeled for used pedestrians. The VGG19 deep architecture network that is pre-trained where we usually retrain a network that is pre-trained for the classification of Penn-Fudan dataset for pedestrians [14] replaces and images the last levels with improved layers that are acclimatized to the improved data set. Additionally, we alter the amount of classes from 1000 into 2 classes; the initial one for background and the consequent one for pedestrians. The planning of Faster R-CNN is complicated since it has some portions. As shown in Fig. 1.

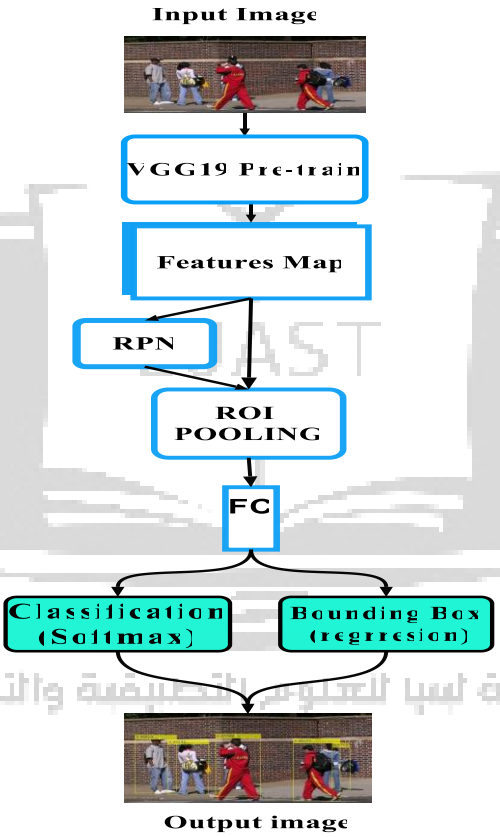


Fig.1: The proposed framework for multiple pedestrians based on Faster RCNN.

The method that is proposed in this paper can be used for the detection of pedestrians on small-scale as well as person regions on a high resolution. The procedure for detection can be changed through the provision of information about the probable pedestrian size in the frame of the video with an aim of elevating the rate of detection.

a) The initial phase constitutes the function of the RPN network which is image input and consequently leads to the output of a group of rectangular locations for candidates. The image for input is illustrated as $heigh \times width \times Depth$ (of any size) which undergoes through pre-trained VGG-19 CNN for the duty of categorization (Image-Net) thus indicating a group of convolution aspect maps in the last convolution level.

b) The anchor is considered as the center of RPN network. A window that slides is spatially run on these maps. The dimension of the sliding window is$n \times n$. For every sliding window, there is a set of nine anchors that are produced and have the same midpoint (x_a, y_a)even though with three diverse aspect ratios (AR) as well as other three distinct scales as shown below Fig. 2. Note

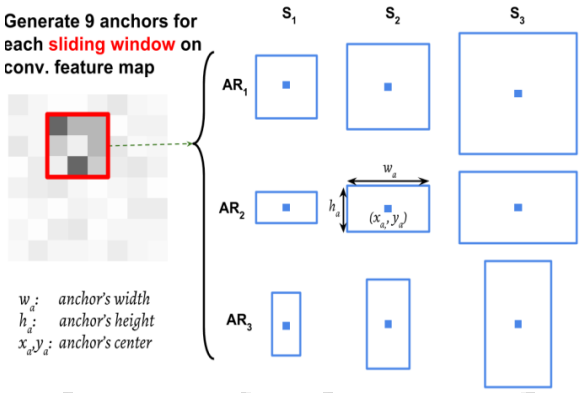


Fig. 2: The generation of the anchors.

That all these coordinates are computed with respect to the original image.

c) p^* Is a value that is calculated for every anchor, whereby there is an indication of how much the anchors overlie with bounding boxes with the ground-truth.

$$p^* = \begin{cases} 1 \ if \ IoU > 0.7 \\ -1 \ if \ IoU < 0.3 \\ 0 \quad otherwise \end{cases} \quad (1)$$

Whereby IoU is considered as the connection over the union and is labeled as

$$IoU = \frac{Anchor \cap GTBox}{Anchor \cup GTBox} \quad (2)$$

d) The RoI pooling level makes use of max pooling to facilitate the conversion of aspects that are within any region of interest that is valid into an insignificant feature map that has a permanent spatial extent of $H\,X\,W$ whereby W and H are levels hyper-parameters which have independent RoI. Every RoI is classified by a four-tier (c, r, w, h) that indicates its top corner at the left (c, r) as well as its width and height (w, h).

e) The 3×3 spatial features extracted from those convolution feature maps are fed to a smaller network which has two tasks: classification layer (cls), and regression layer (reg). The output of regression determines a predicted bounding-box (x, y, w, h), the output of classification sub-network is a probability p indicating whether the predicted box contains a pedestrian (1) or (0) for background.

f) Faster-RCNN utilizes multi-task function of cost that compiles the bounding box regression and classification losses:

$$\boldsymbol{L = L_{cls} + L_{reg}} \quad (3)$$

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \boldsymbol{\lambda} \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (4)$$

In the above equation, i is considered as the anchor index in a smaller batch and p_i considered as the calculated probability of the anchor with i being regarded as the object. The ground-truth label $p_i^*$is 1 if the anchor is positive and is 0 if the anchor is negative. t_i Is a vector representing the 4 parameterized coordinates of the predicted bounding box, and t_i^* is that of the ground-truth box associated with a positive anchor. The two terms are normalized by N_{cls} and N_{reg} and weighted by a balancing parameter $\boldsymbol{\lambda}$.

The cost function has two major parts, corresponding to the two branches of RPN, namely the classification error of the target or not and the regression error of bbox, where $L_{reg}(t_i,t_i^*) = R(t_i - t_i^*)$ is used in Fast-RCNN. Note that L_{reg} is multiplied by p_i in the regression error, so the bbox regression only calculates the error for the anchor containing the target. In other words, if the anchor does not contain the target, the box output location does not matter. So, for bbox's ground truth, only consider the anchor that is determined to have a target and use the coordinates of the label as ground truth. In addition, when calculating the bbox error, it is not the

Coordinates of the four corners, but t_x, t_y, t_w, and t_h. The specific calculation is as follows:

$$t_{x=(x-x_a)/w_a}\ ,\ t^*_{x=(x^*-x_a)/w_a},$$

$$t_{y=(y-y_a)/h_a},\ t^*_{y\ =(y^*-y_a)/h_a} \quad (5)$$

$$t_{w=\log(w/w_a)},\ t^*_{w=\log(w^*/w_a)}$$

$$t_{h=\log(h/h_a)},\ t^*_{h=\log(h^*/h_a)},$$

4. The Experimental Results

To test the proposed system, Penn-Fudan dataset. It has images that can be applied in detection of pedestrians. There are 170 images with 345 labeled pedestrians. The heights of labeled pedestrians in this database fell into [180,390] pixels and divided the dataset into 60% for training, 30% for testing and 10% for validation. We implemented our system on a Windows 10 using Intel(R) Core (TM) i7-7700K at 4.20GHz, 16GByte RAM, and also single 11GB memory GPU NVIDIA GeForce GTX1080Ti. All algorithms are implemented by using Matlab2018a.

We evaluated the performance of the proposed system by using the area under the curve (AUC). We are plotting the performance curve based on True positive rate and false positive rate. These metrics were find using the True Positives (TP), False Positives (FP), True Negative (TN) and False Positives (FP). Figure 3 shows some scenario of different possibilities.

Fig. 3: Samples of the used images.

Here, a detection is said as a TP, if the detected portion is an actual pedestrian area. A TN means, the area is detected as negative is actually an area without any pedestrian. When a background region is detected, it is a False positive, and if an actual pedestrian area is missed, it is treated as an FN.

True Positive Rate (TPR) is obtained using the ratio

$$\boldsymbol{TPR = \frac{TP}{TP+FN}} \quad (6)$$

Similarly, False Positive Rate (FPR) and True Negative Rate(TNR) is obtained as

$$\boldsymbol{FPR = \frac{FP}{FP+TN}} \quad (7)$$

$$TNR = \frac{TN}{TN+FP} \quad (8)$$

From (7) and (8) calculated (AUC)

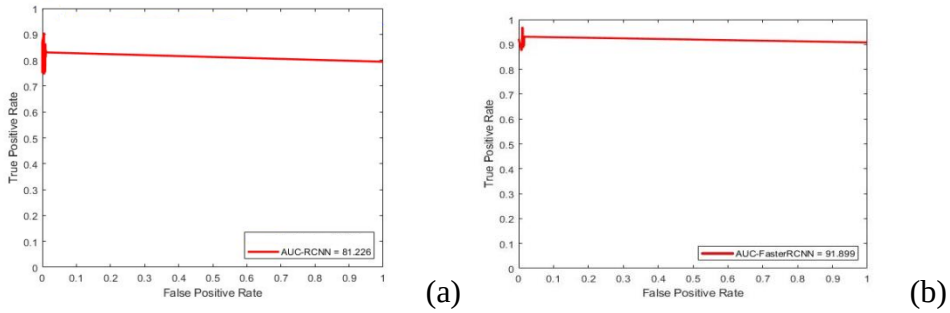
$$AUC = 0.5\,(TPR + TNR) \quad (9)$$

Calculating the accurateness (ACC), and F1 score is the harmonic mean of sensitivity and precision using Eqs as indicated below

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}\% \quad (10)$$

$$F1 = \frac{2TP}{2TP+FP+FN}\% \quad (11)$$

Fig. 4 shows Receiver operating characteristic (ROC) plots and AUC for our proposed



(a) (b)

Fig. 4: The ROC curve: **(a)** for VGG16, **(b)** for VGG19**.**

We use VGG16 and VGG19 to train the RPN and achieved from using RPN with VGG16 to 81.22% and using RPN with VGG19 to 91.89%. This is a promising result because it suggests that the proposal quality of RPN+VGG19 is better than that of RPN+VGG16. In Table 1, we summarized the results of the system with the area under the (ROC) curve when used VGG-16 and trained with VGG-19.

Table 1. The performance evaluation of Faster RCNN with two architecture

Model (Feature extraction)	Training and Test data	AUC
Faster RCNN, VGG-16	Penn-Fudan	81.22%
Faster RCNN, VGG-19	Penn-Fudan	91.89%

We run the experiment result ten times for training and testing with faster RCNN and the average two hour when use transfer learning and without transfer learning take 48 hour.

5. Conclusion

In this paper, we present multiple pedestrian detection**s** that Faster R-CNN is applied to extract pedestrian and get their localization and knowledge of its classification. We trained Faster R-CNN model with the Penn-Fudan pedestrian dataset. We used pre-trained VGG19 convolutional architecture. We reached somewhat satisfactory results at in **(**AUC) that is achieved 91.89%. Specificity = 91.255%, Sensitivity = 6.585%, (ACC) = 96.9%, and F1 score = 91.9%. In the future work, we will train the R-CNN with more data. We are planning to apply the Kernelized Correlation Filter algorithm to tracks of each pedestrian will be localized in the image frame, To improve the existed algorithm components, we may look for a more reliable prediction algorithm.

6. REFERENCES

[1] D. Navneet and T .Bill, "Histograms of Oriented Gradients for Human Detection". IEEE Computer Society, pp 886-893, 2005.

[2] V. Paul, J. Michael and S. Daniel, "Detecting Pedestrians using patterns of motion and appearance". ICCV, pp 734-741, 2003.

[3] F. Yoav and E. Robert, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. " Journal of computer and system sciences, vol 55, pp 119_139, 1997.

[4] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme. " Mathematische Annalen, vol 69, pp 331-371, 1910.

[5] P. Constantine and P. Tomaso, " A trainable system for object detection. " IJCV, vol 38(1), pp15-33, 2000.

[6] M. Anuj, P. Constantine and P. Tomaso, "Example-based object detection in images by components. " TPAMI, vol 23(4), pp 349-361, 2001.

[7] G. Ross, D. Jeff, D. Trevor and M. Jitendra, "Region-based convolutional networks for accurate object detection and segmentation. " TPAMI, 2015.

[8] J. Uijlings, K. van de Sande, T. Gevers and A. Smeulders, "Selective search for object recognition. " IJCV, 2013.

[9] H. Kaiming, Z. Xiangyu, R. Shaoqing and S. Jian, "Spatial pyramid pooling in deep convolutional networks for visual recognition. " arXiv:1406.4729v4, 2015.

[10] G. Ross, "Fast R-CNN. " IEEE International Conference on Computer Vision (ICCV),2015.

[11] R. Shaoqing, H. Kaiming, G. Ross and S. Jian, "Faster R-CNN: Towards real-time object detection with region proposal networks. " arXiv:1506.01497v3, 2016.

[12] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition. ", ICLR, 2015.

[13] K. Daniel, A. Michael, J. Christian, L. Georg, N. Heiko and T. Michael, "Fully Convolutional Region Proposal Networks for Multispectral Person Detection. ", IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 243-250, 2017.

[14] W. Liming, S. Jianbo, S. Gang and S.I-fan, "Penn-Fudan Database for Pedestrian Detection and Segmentation. " ACCV [Online]. Available: https://www.cis.upenn.edu/~jshi/ped_html/, 2007.

[15] H. Kaiming, G. Georgia, D. Piotr and G.Ross , "Mask R-CNN. " arXiv:1703.06870v3, 2018.

[16] Y. Tsung, D. Piotr, G. Ross, H. Kaiming, H. Bharath and B. Serge , "Feature Pyramid Networks for Object Detection. " CVPR, pp 2117-2125, 2017.

[17] **ImageNet**. http://www.image-net.org.

[18] K. He, Z. Xiangyu, R. Shaoqing and S. Jian, "Deep residual learning for image recognition. " Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770-778, 2016.

[19] O. Russakovsky, J. Deng and H. Su, "ImageNet Large Scale Visual Recognition Challenge. " **International Journal of Computer Vision (IJCV)**. Vol 115, Issue 3, pp 211-252, 2015.

[20] Z. Lawrence and D. Piotr, "Edge boxes: Locating object proposals from edges. " European Conference on Computer Vision (ECCV), 2014.

Fig.5: Some experiment results of pedestrian detection on the Penn-Fudan pedestrian Dataset using the Faster R-CNN system. The model is VGG-19 Each output box is associated with a softmax score in [0; 1]. A score threshold of 0.7 is used to display these images.