Convolutional Neural Network for Person Detection using YOLO Framework

M. H. Putra, Z. M. Yussof, S. I. Salim, K. C. Lim
Centre for Telecommunication Research and Innovation (CeTRI),
Faculty of Electronics and Computer Engineering (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM),
Durian Tunggal, Melaka, Malaysia
mohdhasbullahputra@gmail.com

***Abstract*—In this paper we present a real-time person detection system suitable for use in Intelligent Car or Advanced Driver Assistance System (ADAS). The system is based on modified You only Look Once (YOLO) which uses 7 convolutional neural network layers. The experimental results demonstrate that the accuracy of the person detection system is reliable for real time operation. The performance of the detection is benchmarked using the standard testing datasets from Caltech and measured using Piotr's Matlab Toolbox. The results benchmarking is emphasizing on the performance of false positive per image (FPPI) over the miss rate. ADAS demands both relatively good detection and accuracy in order to work in real time operation. A good detection result is marked by achieving low miss rate and low FPPI. This requirement was achieved by the modified YOLO with 28.5%, 26.4% and 22.7% miss rate at 0.1 FPPI and believed to be an excellent candidate for use in ADAS.**

***Index Terms* – ADAS; CNN; FPPI; YOLO**

I. INTRODUCTION

A vast growing topic among the computer vision research community is Vision-based object detection [13,14]. In particular, the person detection have a direct application especially in Advanced Driver Assistance System (ADAS) which is a future in intelligent self driving car. Numerous methods have been proposed for person detection, however most of the techniques are focusing on achieving high detection accuracy at the expense of high computational complexity. Thus many of these methods are not suitable for real time deployment such as being needed in ADAS.

Nowadays the vast emergence of convolutional neural network (CNN) has created an impressive performance in object classification and object detection. The remarkable result demonstrated by Girshick *et al*. [5] proposed that the region proposal network with convolutional neural network (R-CNN) for object detection. The result performance have become the new state of the art on standard detection benchmarks such as PASCAL VOC [9,10]) with a significant improvement compared to the traditional handcrafted state of the art methods which mainly used deformable part model (DPM) [1] and histogram of oriented gradients (HOG) [2] methods. R-CNN is using selective search in order to come up with region of proposals and uses CNN as the feature classifier for detection tasks. Each of the region of proposal will undergo the forward pass in the CNN network which makes R-CNN is computationally expensive. Then the author proposed Fast-RCNN [6] which reduces the computational complexity by introducing the ROI polling. ROI pooling will reduce the size of the region of proposal which will enhance the performance of R-CNN. With the excellent achievement of Fast-RCNN, the speed performance is still limited due to the bottleneck generated by the region of proposal. A faster version of Fast-RCNN is introduced. Faster R-CNN [7] replace the selective search with Region Proposal Network (RPN) which enabling the system to achieve better speed performance.

Despite of the excellent detection result achieve by Faster-RCNN, the intensive computation still make Faster-RCNN is not suitable for real time operation as needed by ADAS. In order to meet the requirement of high detection accuracy and high speed performance for real time operation, another approach of CNN based object detection by using unified detection is introduced by Redmon et. al. [8]. The proposed method, You Only Look Once (YOLO) predicts the bounding boxes and class probabilities directly from full images in a single evaluation.

Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. The YOLO model runs in real-time at 45 frames per second on nVidia Titan X with mean average precision (mAP) of 63.4% on the PASCAL VOC 2007 dataset. The fast YOLO achieves a mAP of 52.7% at 150 frame per second (fps) while Faster R-CNN runs at 7 fps and attains a mAP of 73.2% on the VOC 2007 test set.

Our person detection system is based on modified YOLO architecture, where the number of convolutional layers reduced to 7 and only detecting only one class (person). This will result in some reductions in computational complexity but accuracy is expected to degrade. We investigate the performance of the modified YOLO especially for detection of small size person by varying the grid cells from 7x7 to 11x11.

The remainder of this paper is organized as follows. A brief description of the YOLO architecture will be provided in Section 2. Section 3 briefly describes the datasets used and how training is performed. Section 4 describes experimental results using the system. Section 5 provides the conclusion of this paper.

.

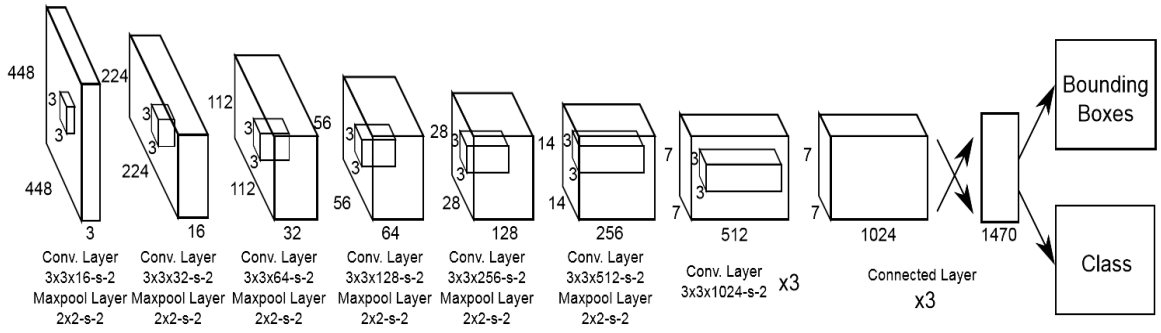


Figure 1: Original YOLO architecture [8]

II. You Only Look Once

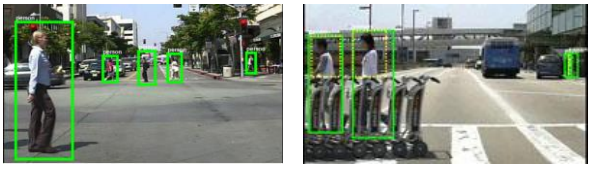
YOLO architecture is made up of 27 CNN layers, with 24 convolutional layers, followed by 2 Fully Connected layers and a final detection layer as shown in Figure 1. It divides the input image into S × S grid cells and each grid cell will predicts B bounding boxes and yield a score for each of the C classes. Each bounding box consists of 5 predictions which are center x, center y, width, height and confidence of the bounding box. For each grid cell, there will only be one set of class scores C for all bounding boxes in that region.

The fully connected layers use the features extracted from the convolutional layers and use the information to predict the probabilities of the object and at the same time for the bounding box constructions. YOLO final detection layer is a regression that maps the output of the last fully connected layer to the final bounding box and class assignments. The original YOLO network is trained on PASCAL VOC 2007 and PASCAL VOC 2012 dataset with 20 classes of objects with a grid size of 7x7.

III. Experimental Setup

A. The Datasets

CALTECH Dataset [12] is used to train our models. In this paper, one class of object which are person with varying shapes and colors are being used. The dataset is consists of 10 hours of videos in sequence format (seq) collected from a car dashboard driving through the city area. The annotations are also provided in the form of bounding boxes coordinates showing the ground truth of the pedestrian. Apart from that, the evaluated annotation is for pedestrian which is 50 pixel tall without any occlusions. Figure 2 shows some of the images in the CALTECH datasets used for the experiments.



(a) (b)

Figure 2: Examples of CALTECH dataset images (a) images of pedestrian without occlusion (b) images of pedestrian with occlusions

B. Annotation

The annotations containing the bounding box information of the ground truth is required. The ground truth annotation provided by Caltech is converted to the format accepted by YOLO. After the conversion, the bounding box annotation for a single image should contain the following parameters:

(Class id, x, y, Width, Height)

The *Class* indicates the id of the class to be detected. As for *x* and *y*, they represent the centre of the bounding box. *Width* and *Heights* represents the bounding box width and height in fraction of the original image. The formation of the ground truth annotations for YOLO is shown in Figure 3:

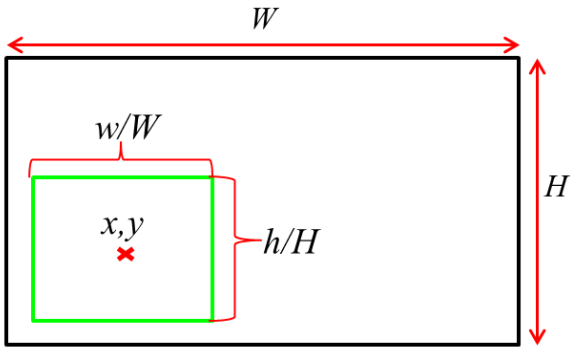


Figure 3: Bounding box ground truth labeling

C. Training

The convolutional neural network is trained by using the batch mode training. Batch mode training will randomly select *n* number of pair which consists of the dataset image (i_n) and ground truth label (g_n). The error obtained will be accumulate throughout the iterations until the iteration reach the number of the batch *n*. The operation is shown in Figure 4.

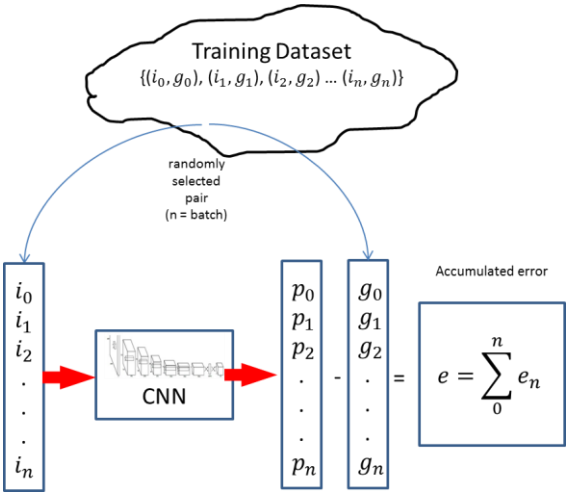


Figure 4: Batch mode training process

The training errors are represented by the intersection over union (IoU). The IoU will measure the accuracy of the bounding box prediction based on how much the prediction bounding box is overlapping with the labelled bounding box. At the early stage of training, the IoU is very small and expected reach more than 90% of overlaps when the training ends. The concept of IoU is shown in Figure 5.

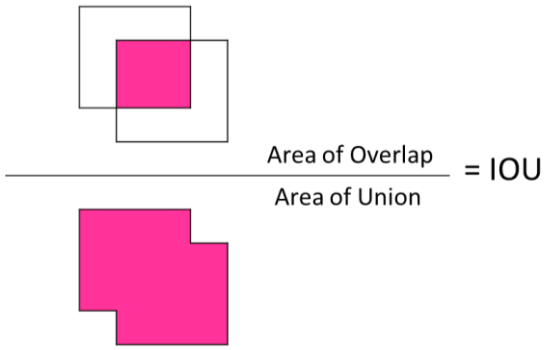


Figure 5: Methods to calculate the IOU for training error detection

In addition, the network training is accelerated using Nvidia K40 GPU which is faster than the normal CPU training speed. After every thousands of iterations, the weight files are stored to a backup directory and can be used as a checkpoint if the training needs to be stopped.

D. Inference

The method proposes reframes the detection problem as a single regression problem to bounding boxes and class probabilities. It requires just a single neural network evaluation for predicting multiple bounding boxes class probabilities. The input image is first resized to the input size of the network (448 x 448). Then the input image will be divided into S x S grid. The parameter of the last layer of the convolution is connected with the fully connected layer and the final outputs for prediction are varied from 7x7(2 x 5 +1) tensor, 9 x 9(2 x 5 +1) tensor and 11 x 11(2 x 5 + 1) tensor. The value of the tensor is calculated as follows;

$$Tensor = s \times s(B \times 5 + C) \quad (2)$$

where: s = Number of grid cells
B = Number of predictions in each cell
C = Number of classes

Then, each grid cells will predict B bounding box which yielding in total of 262, $(B \times s \times s)$ of bounding box predictions. Thresholding the box prediction will filter out prediction below the confidence threshold value. Each grid cells will only responsible to predict for one class and the non-maxima suppression will be applied to delete the duplicate of the bounding box. Figure 6, 7 and 8 shows the steps involves for the inference phase.

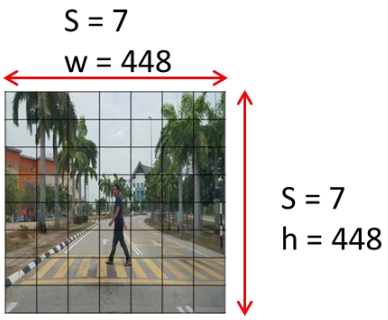


Figure 6: Image is divided into $s \times s$ grid cells

Figure 7: Each grid cells will predict 2 bounding boxes

Figure 8: After the prediction, the grid cell will only select one object for detection

IV. RESULT

The result of the prediction from each tensor will be evaluated qualitatively and quantitatively in term of accuracy and performance. To evaluate the qualitative performance of the person detection system, YOLO_11x11 is tested with images from the testing datasets to verify the functionality. Apart from that, the system is also tested with images captured from the dashboard camera to test the robustness of the system. The result of the detection is shown in Figure 9 ,10 and 11.

Figure 9: Person Detection results compared with the ground truth annotations using YOLO_7x7

Figure 10: Person Detection results compared with the ground truth annotations using YOLO_11x11

Figure 11: Detection of pedestrian with real life image captured from dashboard camera

Apart from that, modified YOLO detection is also quantitatively benchmarked using the Caltech testing datasets [12]. The testing datasets consist of video sequences (in seq format) and the ROC plot which consists of false positive per image versus miss rate graph was generated using piotr's matlab toolbox [11]. The generated graph is shown in Figure 12. As observed in Figure 10, the reliability of modified YOLO is considered as reasonable for pedestrian detection. Further details for the miss rate from the graph is summarize in Table 1.

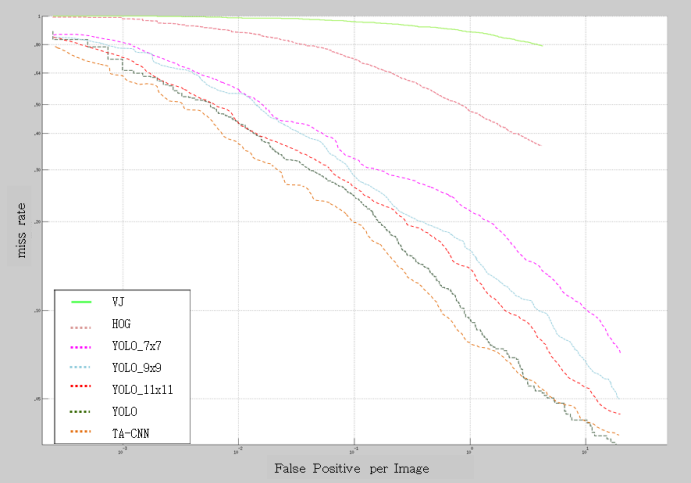


Figure 12: Graph of false positive per image (fppi) against miss rate

Table 1
Comparisons of Person Detection Results on the CALTECH dataset

Architecture	Miss Rate (MR%) at 0.1 fppi
VJ[12]	94.7
HOG[2]	68.5
TA-CNN[15]	20.9
YOLO	21.3
YOLO_7 x 7	28.5
YOLO_9 x 9	26.4
YOLO_11x11	22.7

The miss rate of the modified YOLO at 0.1 fppi is much lower if compared with the traditional person detection system such as VJ and HOG. The standard YOLO trained with only one class have the lowest miss rate compared to the modified YOLO. The performance of the miss rate is followed by YOLO_11x11 with difference of 1.4%, YOLO_9x9 and YOLO_7x7 by 5.1% and 7.2% respectively. The increase in the miss rate is considered as a trade off to the reduction of the numbers of layers in modified YOLO.

V. Conclusion

In this paper CNN-based person detector is presented with the focus on achieving lowest possible miss rate at 0.1 FPPI. Our real-time detector is based on modified YOLO which uses 7 convolutional layers. This reduction of number

of layers has the impact of reducing the computational complexity at the expense of acceptable loss in detection accuracy. The experimental results demonstrate that although the convolutional layers have been reduced to 7 layers, using larger 11x11 grid cells (or higher) can improve the detection accuracy on small objects. This makes the reduced number of convolutional layers in YOLO with higher number of grid cells a good candidate for use in ADAS which demands both relatively high detection accuracy and real time operation.

ACKNOWLEDGMENT

The authors would like to thank Centre for Telecommunication Research and Innovation (CeTRI), Faculty of Electronics and Computer Engineering (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), and the MyBrain15 program from the Ministry of Higher Education (MOHE) for sponsoring this study. The authors also would like to thank Centre for Research and Innovation Management Universiti Teknikal Malaysia Melaka (CRIM-UTeM) for sponsoring this paper. Apart from that, the authors also would like to thanks Collaborative Research in Engineering, Science and Technology (CREST) Malaysia for sponsoring the related work and equipments under the research grant (GLUAR/CREST/2015/FKEKK/I00005).

REFERENCES

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. "Object detection with discriminatively trained part based models" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[2] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". *In CVPR*, 2005.

[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. "Scalable object detection using deep neural networks" *In CVPR*, 2014.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, Alexander C. Berg. "SSD: Single Shot MultiBox Detector" arXiv:1512.02325v5 [cs.CV], Dec. 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation" In *CVPR*, 2014.

[6] R. Girshick. "Fast R-CNN" arXiv preprint arXiv:1504.08083, 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks" arXiv preprint arXiv:1506.01497, 2015.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection" arXiv preprint arXiv: 1506.02640, 2015.

[9] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[10] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010.

[11] P. Dollár, C. Wojek, B. Schiele and P. Perona "Pedestrian Detection: An Evaluation of the State of the Art" PAMI, 2012.

[12] Paul Viola and Michael J Jones. "Robust real-time face detection" IJCV, 57(2):137–154, 2004.

[13] C. Chen, A. Seff, A. Kornhauser and J. Xiao. "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving" Proceedings of 15th *IEEE International Conference on Computer Vision ICCV*, 2015.

[14] J. E. Hoo, and K. C. Lim. "Accuracy and Error Study Of Horizontal and Vertical Measurements with Single View Metrology for Road Surveying" ARPN Journal of Engineering and Applied Sciences. 11(12):7872-6, 2016.

[15] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Pedestrian detection aided by deep learning semantic tasks" *In CVPR*, 2015.