Evaluation of Deep Learning YOLOv3 Algorithm for Object Detection and Classification

Weal A. Ezat
Dept. of Computer Science and Engineering, Faculty of Electronic Engineering Menoufia University.
waelezat2@hotmail.com

Mohamed M. Dessouky
Dept. of Computer Science and Engineering, Faculty of Electronic Engineering Menoufia University.
waelezat2@hotmail.com

Nabil A. Ismail
*Dept. of Computer Science and Engineering, Faculty of Electronic Engineering*Menoufia University email address or nabil_is@hotmail.com

***Abstract* — You Only Look Once version 3 (YOLOv3) is a deep learning model for object detection and classification. It is a single neural network architecture model that uses features from the feeding images and predicts bounding box for all classes of image simultaneously. This paper descript an experimental work for train the deep learning model based on YOLOv3 architecture implemented using Tensor Flow as a deep learning framework. The training process had been done using the data-set PASCAL VOC 2007 and data-set PASCAL VOC 2012 and using The Adaptive Moment Estimation Optimizer (ADM optimizer). The trained model is then tested by using the VOC 2007 test data-set. The final results evaluate the YOLOv3 deep learning model performance for object detection and classification.**

Keywords — Deep learning, YOLOv3, Object detection.

I. INTRODUCTION

. The human visual system is accurate and so fast it gives the human the ability to perform complicated tasks like driving car with high performance. Deep learning algorithms for object detection and classification would allow computerized systems to simulate human performance in complex tasks like driving cars and eliminate using specialized sensors and complicated systems.

You Only Look Once (YOLO) is a deep learning model for object detection and classification. YOLO had been created and tested by Joseph Redmon and others in 2016 [1] and achieved high speed and accuracy than other methods. YOLO is a single neural network that predicts bounding boxes and class probabilities directly from full image in one evaluation. It sees the entire image during training and testing time is not like sliding region techniques or proposal-based window techniques. YOLO learns general representations of objects, be trained on natural images, and then it will be tested on artwork. It have the exultant level of generalization as it little to break down when it is applied to unexpected inputs [1].

YOLO9000 is a new version from YOLO detect over a 9000 object categories. YOLO9000 is called YOLOv2, which had been introduced by Joseph Redmon and Ali Farhadi in 2016 [2] as enhanced version from YOLO. YOLOv2 has various improvements to the YOLO. The improvement is mainly on recall and localization as maintaining the classification accuracy [2].

YOLOv3 is an enhanced version from YOLO. It had been created by Joseph Redmon and Ali Farhadi in 2018 [3]. YOLOv3 is a hybrid approach combining the neural networks that had been used in YOLOv2 and Darknet-19. Darknet-19 is a neural network with nineteen convolutional layers and five max pooling layers [3]. It uses softmax layer type for object classification. The YOLOv3 can predict bounding boxes using dimension clusters as anchor boxes. It predicts an object score for each bounding box using logistic regression [3].

The contribution of this paper is

- Performance Comparison between the YOLOv3 and other classification algorithms and methods for multi-class image classification.
- Using Adaptive Moment Estimation Optimizer (ADAM) as an optimization algorithm to train the deep learning model.

This paper is organized as follows. Section 2 presents the related work in detail and the related deep learning models. Section 3 presents the proposed approach and model in detail. Section 4 introduces the model tuning process with analysis of the data-set used. Section 5 presents the experimental results with diagrams and tables. Finally, the conclusion is present in Section 6.

II. RELATED WORK

The Convolutional Neural Network (CNN) is one of the leading methods for image classification. For the deep learning CNN model, every input image pass through a series of convolution layers with filters, Pooling and fully connected layers finally the method use Soft-max function to classify the object with probabilistic values between 0 and 1 [4]. The deep learning algorithm solves wide problems of classification task. The ability to process large clusters of images quickly, state the CNN deep learning model as the most important method for images classification. The ability and flexibility for changeable in the CNN deep learning model with a wide range of data-sets make the deep learning algorithm as the most important technique for the classification task [5].

Region-based Convolutional Neural Network (R-CNN) [6] is a method proposed to bypass the problem of selecting a vast number of regions. The R-CNN uses selective search to extract just 2000 regions from the image and called them region proposals. The 2000 candidate region proposals are warped into the square and fed into a CNN that produces a 4096 feature vector as output. The R-CNN cannot be implemented in real-time as it takes around 47 seconds for each test image [6].

Fast Region-based Convolutional Neural Network (Fast R-CNN) [7] is a model for object detection that has been designed to rich real-time performance. The Fast R-CNN feeding images and a set of object proposals. The network processes the whole image to produce a feature map. For each object proposal, a region of interest pooling layer extracts a fixed-length feature vector from the feature map.

Each feature vector is fed into sequences of fully connected layers. The fully connected layers branch into two sibling output layers. One layer is that it produces soft-max probability over k objects classes. The other layer outputs a four real-valued number for the K object classes. The four values encode a refined bounding box position [7] where the K is exertion for number of classes has been used by Fast R-CNN.

Faster R-CNN is presented by combining the feature of two type modules. The first module is a deep, fully convolutional network that proposes regions. The second module is the fast R-CNN detector that uses the proposed regions. The proposed Region's Neural Network (RPN) module gives the guided to the Faster R-CNN module to look for. The Faster R-CNN uses the processing power of GPU and CPU as the searching algorithm run in CPU [8].

YOLO is an approach based on a single neural network and designed for object detection and classification. As the whole detection pipeline is a single neural network, detection performance can be optimized end-to-end directly. YOLO implicitly encodes contextual information about classes as well as their appearance. YOLO outperforms most detection methods like R-CNN by a wide margin. YOLO can quickly identify objects in images; it struggles to precisely localize some objects, especially small ones [1].

III. Proposed Method

The proposal tested model is based on YOLOv3 implemented using Tensor Flow framework as shown in Figure 1. The model definition has been written with C++. Interfacing with deep learning framework has been written with python code.

A. Bounding Box Prediction

YOLOv3 network follows the YOLO9000 system and predicts bounding boxes using dimension clusters as anchor boxes [2]. The YOLOv3 network predicting four coordinates for the bounding box t_x , t_y , t_w , t_h as shown in Figure 2. If the cell is offset from the top left corner of the image by (c_x , c_y) the bounding box prior has a width p_w , p_h and the predictions correspond to equation [3].

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t_w} \quad (3)$$

$$b_h = p_h e^{t_h} \quad (4)$$

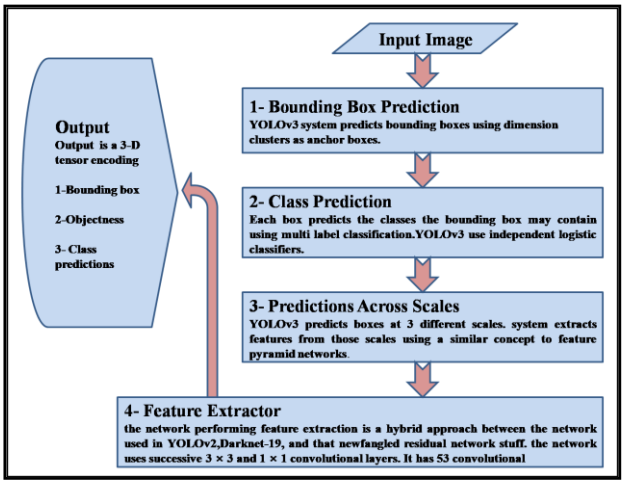

Figure 1: YOLOv3 layers specification and data flow

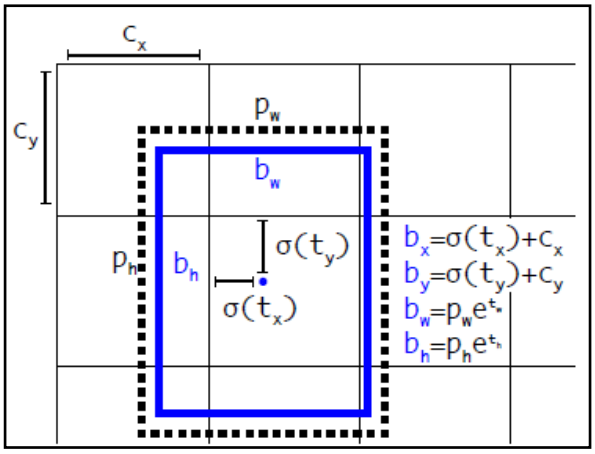

Figure 2:YOLOv3 Bounding boxes with dimension priors and location Prediction [3].

B. Class Prediction and Predictions Across Scales

YOLOv3 predicts boxes at 3 different scales. System extracts features from those scales using a similar concept to feature pyramid networks.

C. Feature Extractor

The YOLOv3 network uses a model which is a hybrid approach between the network used in YOLOv2 and Darknet-19. The network is 53 convolution layers, as shown in Table 1 [3].

Table 1: YOLOv3 feature extractor network [3]

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3/2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avg-pool		Global	
	Connected		1000	
	Soft-Max			

IV. Model Tuning Producers

The experiment is carried out on the YOLOv3 model using the two data-sets PASCAL VOC 2007 and PASCAL VOC 2012 for training the model. The data-set PASCAL VOC 2007 is used for the model test process [9-10] because most modern and classic models are tested with Pascal 2007 data set for models performance comparisons.

A. *ADAM Optimization Algorithm*

The ADAM optimization algorithm is an extension to stochastic descent that has recently seen broader adoption for deep learning applications in computer vision. The ADAM is different from classical stochastic gradient descent. It combines the advantages of two other extensions of stochastic gradient descent; root mean square propagation that also maintains per-parameter learning rates. The ADAM optimizer had been used as an optimization algorithm for training the deep learning YOLOv3 model which used in the experiment [11]. The mathematical model [12] of the ADAM defined as below.

t in range number of iterations

g = compute gradient (x, y)

m = beta_1 * m + (1 - beta_1) * g

v = beta_2 * v + (1 - beta_2) * np.power (g, 2)

m_hat = m / (1 - np.power (beta_1, t))

v_hat = v / (1 - np.power (beta_2, t))

w = w - step_size * m_hat / (np.sqrt (v_hat) + epsilon)

B. *Batch Size and Epochs*

Using Tensor Flow [13] the model had been trained using a batch size of 6 images with 2759 batches and for six epochs. The machine used for the training and testing process is CPU Intel Celeron 2.16 GHz with 4 GB RAM. The experimental work time is 27 days for training and testing process.

C. *Train Data –Set*

The model had been trained using 16551 images and 40058 objects, as shown in Figures 3 and 4. The model has been tested with 4952 files, as shown in Figure 5.

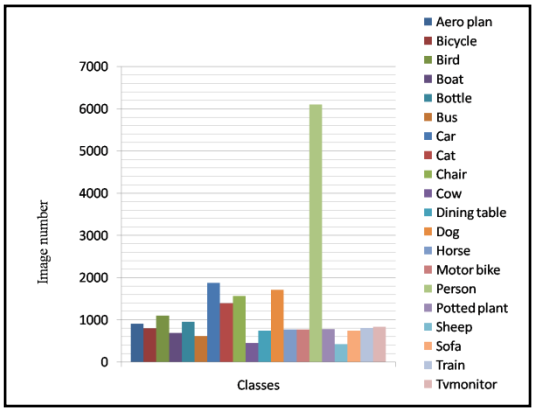

Figure 3: Train images number over classes

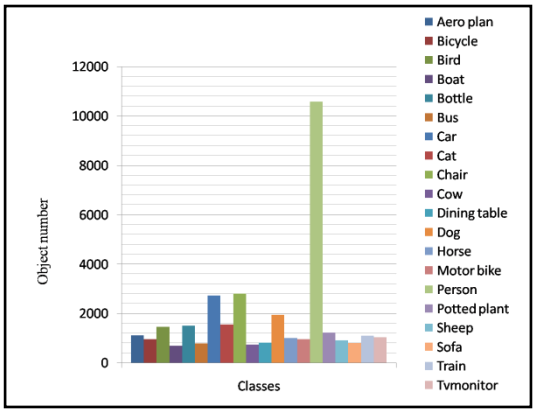

Figure 4: Train object number over classes

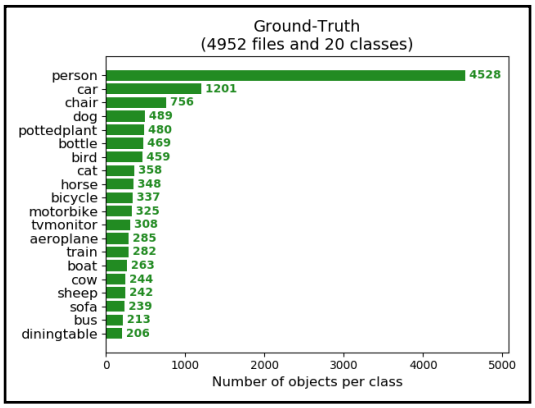

Figure 5: Ground truth of the test data-set

V. Experiment Results

The experimental result had been depicted in Table 2 and Figure 6 by applying the average precision rate. The average precision rate of YOLOV3 over the Pascal 2007 data set is 80.07%. The true prediction and false prediction ratio over classes are shown in Figure 7. The recall diagram for classes (Bus, train, and potted plant) is shown in Figures 8-10.

Table 2: YOLOv3 performance over Pascal 2007 data-set classes

Class	Rate %	Class	Rate %
Bus	95	Aero plane	79
Car	92	TV/monitor	76
Bicycle	90	Sofa	74
Horse	90	Dining table	73
Cat	90	Bottle	73
Person	89	Sheep	66
Motorbike	89	Cow	64
Dog	85	Boat	62
Train	85	Chair	51
Bird	82	Potted plant	79.11

Based on Average Precision (AP), the performance of YOLO v3 over Pascal 2007 data-set has been judged by the precision / recall curve [8]. Detections have been considered true positives or false positives based on the area of overlap with ground truth bounding boxes. Intersection over Union (IOU) is evaluation value used to measure the detection accuracy of an object detector. To be true positive value the IOU value = a_0 (area of overlap) value exceeds 50 % where a_0 the area of overlap between the predicted bounding box B_P and ground truth bounding box B_{gt} must by the formula:

$$IOU = a_0 = \frac{\text{area}\,(B_P \cap B_{gt})}{\text{area}\,(B_P \cup B_{gt})} \qquad (5)$$

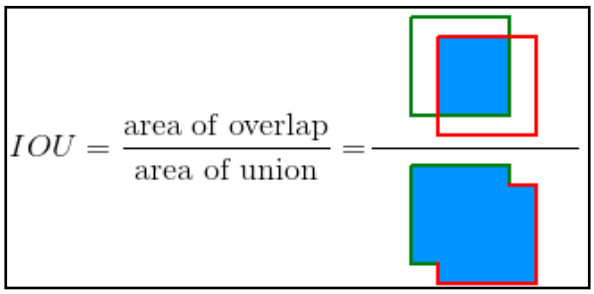

Figure 6: Intersection over Union **(**IOU)

True Positive Rate (TPR) is a synonym for Recall and defined as follows:

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

False Positive Rate (FPR) is a synonym for Precision defined as follows:

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

Receiver Operating Characteristic Curve (ROC) is a graph showing the performance of the classification model at all classification thresholds. This curve plots two parameters, True Positive Rate and False Positive rate. Area under Curve (AUC) measures the entire two-dimensional area underneath the entire ROC curve. The performance of YOLO v3 over Pascal 2007 classes is depicted in Table 3 and Figures 11 – 14

The Error rate is defined as:

$$\text{Error Rate/Misclassification rat} = \frac{\text{False Predection}}{\text{Total Population}} \qquad (8)$$

The accuracy is defined as:

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{\text{Total Population}} \qquad (9)$$

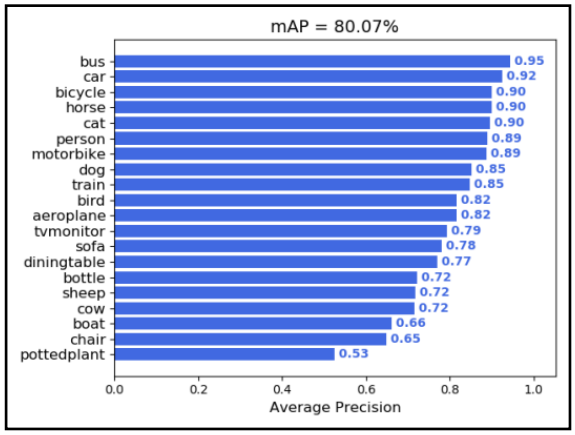

Figure 7: Average precision rate over classes

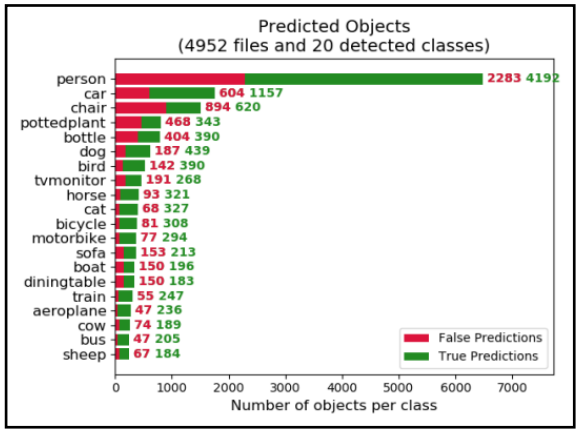

Figure 8: Test predictions rate over classes

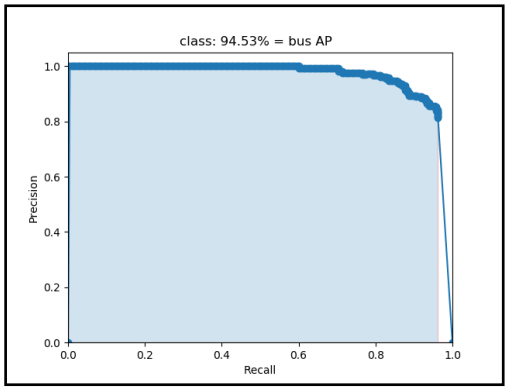

Figure 9: ROC and AUC for Bus class

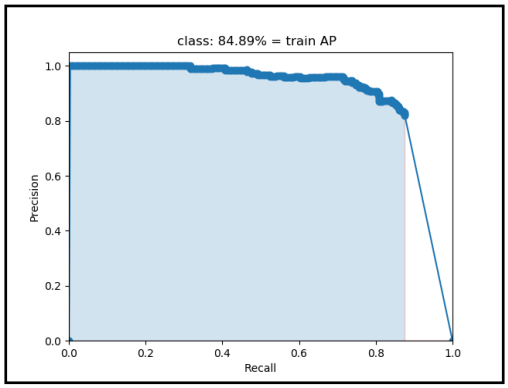

Figure 10: ROC and AUC for Train class

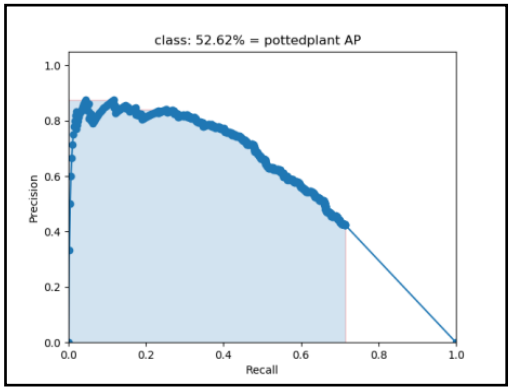

Figure 11: ROC and AUC for Potted plant class

Table 3: The performance of YOLOv3 over Pascal 2007

Rate (%)	map	Error	Precision	Recall	Accuracy
Bus	95	.28	81.3	96.2	99.72
Car	92	3.57	65.7	96.33	96.43
Bicycle	90	.48	79.1	91.39	99.52
Horse	90	.65	77.5	92.24	99.45
Cat	90	.41	82.78	91.34	99.59
Person	89	13.48	65.34	92.57	86.52
Motorbike	89	.46	79.24	90.46	99.54
Dog	85	1.11	70.12	89.77	98.89
Train	85	.33	81.78	87.58	99.67
Bird	82	.84	73.3	84.96	99.16
Aero plane	79	.28	83.39	82.8	99.72
TV/monitor	76	1.13	58.38	87.12	98.87
Sofa	74	.81	58.19	89.12	99.09
Dining table	73	.89	54.95	88.83	99.11
Bottle	73	2.39	73.3	83.15	97.61
Sheep	66	.4	73.3	76	99.6
Cow	64	.44	71.8	77.45	99.56
Boat	62	.89	40.95	74.52	99.11
Chair	51	5.28	44.95	82.01	94.72
Potted plant	79.11	2.87	42.29	71.45	97.23

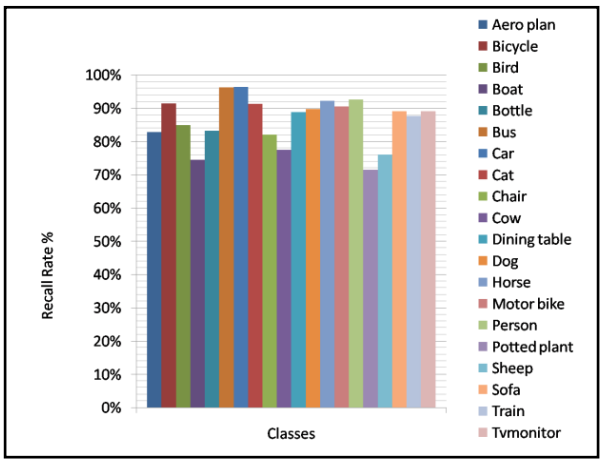

Figure 12: Recall Rate over Pascal 2007 classes

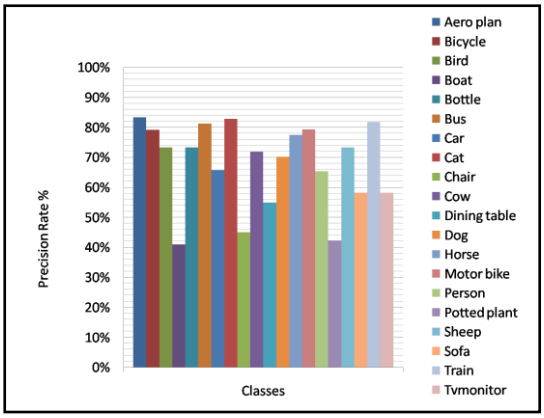

Figure 13: Precision Rate over Pascal 2007 classes

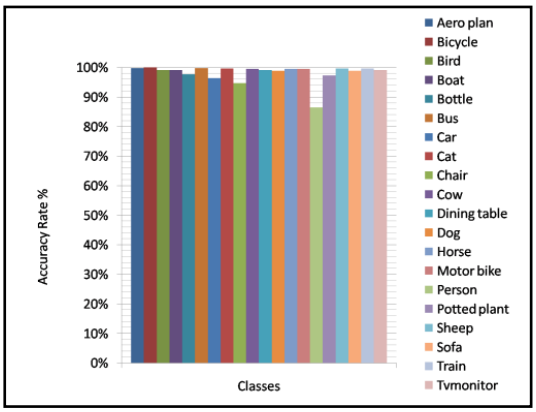

Figure 14: Accuracy Rate over Pascal 2007 classes

The comparison performance of the detection methods over Pascal 2007 data-set is shown in Table 4 and Figure 15.

Table 4: The performance of detection methods over Pascal 2007

Method	YOLOv3	YOLO [1]	FRCNN [8]
AP (%)	80.07	63.4	70

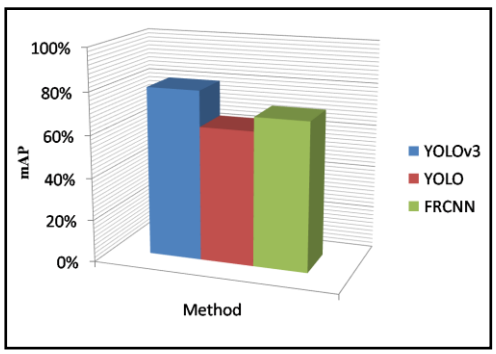

Figure 15: Detection methods Performance over Pascal 2007 data-set.

ACKNOWLEDGMENT

The results of the experimental work had been evidenced pointed out that the deep learning algorithm YOLOv3 model is an effective solution for the object detection and classification task. The deep learning YOLOv3 model had been trained using the ADAM optimizer algorithm and with the data-set Pascal 2007 train and the data-set Pascal 2012 train. Finally, the trained model had been tested with Pascal 2007 test data-set. The performance of the model had been measured with an average precision rate. The performance of YOLOv3 is compared with the R-CNN model, Fast R-CNN, Faster R-CNN, and YOLO models, which tested with Pascal 2007 test data set. The YOLOv3 is more efficient than other models, with an average precision rate of 80.07%.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[3] "YOLOv3: An Incremental Improvement - pjreddie.com." [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf. [Accessed: 24-Sep-2019].

[4] "CS231n: Convolutional Neural Networks for Visual Recognition," Stanford University CS231n: Convolutional Neural Networks for Visual Recognition. [Online]. Available: http://cs231n.stanford.edu/. [Accessed: 25-Sep-2019].

[5] W. A. Ezat, M. M. Dessouky, and N. A. Ismail, "Multi-class Image Classification Using Deep Learning Algorithm," Journal of Physics: Conference Series, vol. 1447, p. 012021, 2020.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[7] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[8] 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, Jan. 2017.

[9] "The PASCAL Visual Object Classes Challenge 2007," The PASCAL Visual Object Classes Challenge 2007 (VOC2007). [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/. [Accessed: 25-Sep-2019].

[10] The PASCAL Visual Object Classes Challenge 2012 (VOC2012) host.robots.ox.ac.uk/Pascal/VOC/voc2012.

[11] "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning," Machine Learning Mastery, 06-Aug-2019. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/. [Accessed: 25-Sep-2019].

[12] D. P. Kingma and J. L. Ba∗ , "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," arXiv, Jan. 2017.

[13] "First Steps with Tensor Flow: Toolkit | Machine Learning Crash Course," Google. [Online]. Available: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit. [Accessed: 28-Sep-2019].