YOLO based Logo detection

Ashna Eldho
Dept. of AEI
Rajagiri School of Engineering and Technology
Kochi, Kerala, India

Tony Francis
Robert Bosch(RBEI)
Bangalore, India

Hari C. V.
Dept. of AEI
Rajagiri School of Engineering and Technology
Kochi, Kerala, India

***Abstract*—Artificial Intelligence(AI) is an art of creating machines which can perform functions that requires intelligence when performed by people. Object detection, which comes under AI, can be applied to various domains like traffic sign detection, face detection, video surveillance, ball tracking in sports, pedestrian detection, people counting etc. Recently, AI has brought many innovations in the retail sector such as shelf monitoring and self checkouts. This work introduces object detection to the retail domain, that is, detection of logos in various products. Dataset used for this purpose was not publicly available and hence the images are taken from different stores with different background, different angle and different lighting conditions. Some images are also taken from Internet sources. Logos of various objects are detected using You Only Look Once(YOLO). Instead of local features, it uses global features to predict labels and bounding boxes. Our trained network efficiently detects logos of different products with more than 95% accuracy.**

***Index Terms*—Computer Vision, YOLO, Logo Detection, CNN**

I. Introduction

Logo is a unique graphic word or combination of words and graphic symbols that identifies a company, a public organization, an institution or an individual's products and services [1]. It can be found in various shapes, fonts, colors and styles.. Logo detection is an application of object detection. It has various applications in many domains such as product brand recognition for intellectual property, protection in E-commerce platform, vehicle logo detection for intelligent transportation, product brand management on social media etc. Figure 1 shows an example of a logo.

Fig. 1. Example of a Logo

The problem description of object detection is to decide where objects are located in the image(location of the object) and to which category each object belongs (classification of the object) [2]. It is one of the main challenging areas in computer vision. It requires more complex methods to solve as the detection requires accurate localization of the objects. Earlier, the best performing detectors were based on a combination of hand crafted image feature representations such as Scale Invariant Feature Transform (SIFT) [3], Histogram of Oriented Gradient (HoG) [4] and Deformable Part Model (DPM) [5]. Profound Learning has been a hotspot as of late and got great application impact in the field of picture grouping, scene acknowledgment and item location and following. Real-time object detection is essential for many real world vision applications such as face detection, logo detection, pedestrian detection, traffic sign detection and self-driving cars.

978-1-7281-5523-4/19/$31.00 ©2019 IEEE

II. Literature Survey

Object detection models are mainly classified into traditional object detection models and deep learning models. In the traditional object detection models, the regions where the objects are likely to be found are first predicted and then the features are extracted. Scale Invariant Feature Transform(SIFT) [3], Histogram of Oriented Gradient(HoG) [4] and Haar-like features are the representative ones. Classification algorithms like Support Vector Machine(SVM) [6], AdaBoost, Deformable Part Model(DPM) [5] are then used to classify the objects. Viola *et. al* [8] has built an efficient moving detector, using AdaBoost to train a chain of increasingly complex region rejection rules based on haar-like wavelets and space-time differences. An efficient face detection model using this algorithm is proposed in [9]. Deep learning models can be categorized into region based models and classification / regression based models.

In [10], a multi scale sliding window approach which can be used for classification, localization and detection is proposed. A tuned CNN is used for localization and detection on Imagenet data [11]. In this method, for each location, bounding boxes are predicted using the features pre-trained on classification and then the confidence of each box is used during merging. The drawback of this method is that it is time-consuming and hence can not be used for real-time applications. Ross Girshick *et. al* [12] proposed Regional-Convolutional Neural Network(R-CNN) which adopts a method known as selective search [13] to generate 2000 region proposals for each region. Then, each proposed region is cropped/warped into a fixed resolution and the CNN model proposed in [11] is used to extract 4096-dimensional features as the final representations. Linear SVMs and greedy Non Maximum Suppression algo-

rithms are applied to detect the objects. Since 2000 region proposals are extracted, training is very slow and hence cannot be applied for real time applications. As the region proposals are warped or cropped to a particular size, there are chances of losing the contents. To solve this problem, Spatial Pyramid Matching(SPM) based CNN architecture is used in [14] which is named as SPP-Net.

R-CNN leaves open several questions such as whether Convolutional Neural Network contains sufficient geometric information to localize objects, whether R-CNN pipeline can be simplified and also whether R-CNN can be accelerated. In [15] , Karel *et. al* tried removing the region proposals generated from R-CNN, dropping SVM and integrating SPP. Thus, the whole detector reduced to the evaluation of a single CNN.

Girshick *et. al* [16] introduced Fast R-CNN which fixes the disadvantages of R-CNN and SPP-Net on their speed and accuracy. In this method, instead of feeding local proposals to the CNN, the CNN is fed the entire input image to produce a convolutional feature map. From these features, region proposals are generated and is then fed to a fully connected layer. Fast R-CNN is faster than R-CNN as 2000 region proposals are not fed to the CNN every time. However, both Fast R-CNN and R-CNN uses selective search [13] to find out the region proposals which is slow and time consuming. Hence, it cannot be applied for real-time applications. To solve these problems, Ren *et. al* [17] introduced Faster R-CNN which is similar to Fast R-CNN. Instead of using selective search [13] algorithm, a separate network called Region Proposal Network(RPN) is used to predict the regions. Li *et. al* [18] proposed a region based fully convolutional network (R-FCN) for object detection. Feature Pyramid Networks (FPN) were also developed for object detection to improve scale invariance. Liu *et. al* [19] proposed a Single Shot Multibox Detector (SSD) for object detection. Redmon *et. al* [21] proposed You Only Look Once(YOLO) algorithm which uses global features instead of local features to predict labels and bounding boxes. YOLO has got three versions. YOLO version 1 has got 24 convolutional layers and 4 maxpooling layers. However it cannot detect small objects. YOLO version 2 [22] is faster and more accurate than version 1. It has got 19 convolutional layers and 5 max pooling layers. Redmon *et. al* introduced latest version of YOLO, i.e., YOLOv3 [23] which is more faster than v2. Chung *et. al* [24] detected traffic signs using YOLOv1.

III. METHODOLOGY

You Only Look Once (YOLO) [21] is a state-of-the-art, real-time object detection. A single convolutional network can simultaneously predict multiple bounding boxes and class probabilities for each box. Using this method, you only look once at an image to know what object is present and where it is present. YOLO has got several benefits compared to other methods. First, YOLO is extremely fast which makes them useful for real-time object detection. Second, while making predictions, YOLO reasons globally about the image. Third, generalizable representations of objects are learned by YOLO.

TABLE I
TINY YOLOv2 ARCHITECTURE [22]

	INPUT	FILTERS/STRIDE	OUTPUT
Input	416x416x3		
Convolution	416x416x3	3x3x16/1	416x416x16
Maxpooling	416x416x16	2x2/2	208x208x16
Convolution	208x208x16	3x3x32/1	208x208x32
Maxpooling	208x208x32	2x2/2	104x104x32
Convolution	104x104x32	3x3x64/1	104x104x64
Maxpooling	104x104x64	2x2/2	52x52x64
Convolution	52x52x64	3x3x128/1	52x52x128
Maxpooling	52x52x128	2x2/2	26x26x128
Convolution	26x26x128	3x3x256/1	26x26x256
Maxpooling	26x26x256	2x2/2	13x13x256
Convolution	13x13x256	3x3x512/1	13x13x512
Maxpooling	13x13x512	2x2/1	13x13x512
Convolution	13x13x512	3x3x1024/1	13x13x1024
Convolution	13x13x1024	3x3x1024/1	13x13x1024
Convolution	13x13x1024	1x1x50/1	13x13x50

YOLO v2 [22] architecture consist of 19 convolutional layers and 5 maxpooling layers. Graphics Processing Unit(GPU) is needed for this implementation. If GPU is not available, a miniature version of YOLO known as Tiny YOLO can be used. It consists of 9 convolutional layers and 6 maxpooling layers. In this work, Tiny YOLO architecture is used to do the logo detection. All the images are resized to 416×416 pixels at both training and testing times. Tiny YOLOv2 architecture is shown in Table 1. For each convolutional layer, a 3×3 filter is used with stride 1.

In this algorithm, the input image is divided into *S* x *S* grid cells and within each grid cell, it predicts *B* bounding boxes and *C* confidence scores, that is, how much the network is sure that the box contains an object. If an object's core falls into a grid cell, the grid cell is responsible for predicting the object. For each of the *B* bounding boxes, there are 5 predictions: these are centre x, centre y, width of the bounding box, height of the bounding box and class to which the detected object belongs to.

$$Y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c \end{bmatrix}$$

where, p_c is probability of whether an object is present or not, b_x and b_y are the x and y coordinates of the bounding boxes, b_h and b_w are the height and width of the bounding boxes and c is the class of the detected object. If there is no object, then, p_c will be 0.

Once all bounding boxes are detected, Intersection over Union(IoU) [26] is calculated between the predicted bounding box and the ground truth bounding box. It is the intersection of the predicted and the ground truth bounding boxes divided by the union of both the boxes. IoU can be classified into poor, good and excellent. If IoU is less than 0.5, it is classified as poor and if IoU is greater than 0.7 but less than 0.9, it

is good and if it is greater than 0.9, it is excellent. Boxes with probability less than a particular threshold value will be eliminated. This is known as filtering by class scores. Threshold value can be set according to our wish. Usually it is set as 0.6. To this filtered output, Non-Maximum Suppression (NMS) algorithm [27] is applied. The following loss function is optimized during training.

$$\lambda_{coord}\Sigma_{i=0}^{S^2}\Sigma_{j=0}^{B}1_{i,j}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]+$$

$$\lambda_{coord}\Sigma_{i=0}^{S^2}\Sigma_{j=0}^{B}1_{i,j}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$$

$$+\ \Sigma_{i=0}^{S^2}\Sigma_{j=0}^{B}1_{i,j}^{obj}(C_i-\hat{C}_i)^2+\lambda_{noobj}\Sigma_{i=0}^{S^2}\Sigma_{j=0}^{B}1_{i,j}^{noobj}$$

$$(\mathrm{C}_i-\hat{C}_i)^2+\Sigma_{i=0}^{S^2}1_i^{obj}\Sigma_{cclasses}(p_i(c)-\hat{p}_i(c))^2 ...(1)$$

where, 1_i^{obj} will be 1, if an object appears in cell *i*, $1_{i,j}^{obj}$ is 1 if the j^{th} boundary box in cell *i* is responsible for detecting the object, $\hat{p}_i(c)$ denotes the conditional class probability for class *c* in cell *i* and $\hat{C}_i$ is the box confidence score of the box *j* in cell *i*. λ_{coord} is set to 5 and λ_{noobj} is set to 0.5.

TABLE II
DETAILS OF IMAGES COLLECTED

Products	Total	Stores	Internet
Dove	1500	1000	500
Himalaya Baby products	1500	1000	500
Nestle	1500	1000	500
Pepsi	1500	1000	500
Yippee	1500	1000	500

IV. DATASET CREATION

Logo detection from real-world images are challenging as the change in sizes, rotation, lighting conditions, rigid and non-rigid transformations etc. affects the detection. No such dataset is publically available. Images of Dove, Himalaya Baby products, Nestle, Pepsi and Yippee Noodles were collected from different stores. While collecting these images, different lighting conditions, different angles, different camera clarity and different background were to be considered. For each category, a total of 1500 images were collected out of which 1000 images were collected from different stores and the rest 500 were taken from Internet resources. Stores that were used for data collection were SPAR Hypermarket(Bangalore), Big Bazaar(Bangalore), More(Bangalore), Star Hyper(Bangalore), Mithra Mega Mart(Ernakulam) and Twenty20 Food Market(Ernakulam). Data Augmentation was done to increase the dataset. A total of 186000 images were used for training and 5000 images were used for testing.

Different types of cameras used and their specifications are mentioned below.

1) Redmi Note 6 Pro - Front camera comprises of 20MP(main camera) +2MP(depth sensor) sensors and back side has 12MP+5MP sensors
2) Honor 9 Lite - 13 MP + 2 MP camera setup on both the front and back
3) MotoG turbo 13 MP front camera and 5 MP back camera
4) Apple Iphone SE 12 MP front camera and 1.2 MP back camera
5) Redmi Mi 4 - 13 MP front camera and 5 MP back camera
6) Moto G5 plus 12 MP front camera and 5 MP back camera

Fig. 2. Yippee Noodles image taken from store

Fig. 3. Dove image taken from store

Fig. 4. Himalaya Baby Product image taken from store

Fig. 5. Nestle image taken from store

V. RESULTS

For Dove, Pepsi, Himalaya Baby products, Nestle and Yippee Noodles images, ground truth annotations are drawn. To draw the ground truth bounding boxes, the graphical

Fig. 6. Pepsi image taken from store

Fig. 7. Pepsi image taken from Internet resources

Fig. 8. Yippee Noodles image taken from Internet resources

Fig. 9. Nestle image taken from Internet resources

Fig. 10. Himalaya Baby product image taken from Internet resources

Fig. 11. Dove image taken from Internet resources

annotation tool LabelImg [25] is used and the results are saved as a *.txt* file. It consist of 5 values:- logo class, x coordinate corresponding to the boundary box center, y coordinate corresponding to the boundary box center, boundary box width and boundary box height. Each of the images are divided into 13×13 grid cells. To obtain more accurate bounding boxes, 5 anchor boxes are used. Thus, the total number of filters is 50 (i.e., [(classes + Output Y) x number of anchor boxes], [(5+5)x5]). If the centre of an object falls into a grid cell, then, that grid cell is responsible for predicting the bounding box. Intersection over Union is calculated between the ground truth and the predicted bounding boxes and are then filtered by class scores. IoU computes the area of overlap between the ground truth bounding box and the predicted bounding box. The higher the IoU, more accurate is the bounding box. Non-maximum suppression algorithm is then applied to this output and finally the correct object is detected.

TABLE III
RESULTS

Figure Number	Logo detected	Accuracy
12	Pepsi	97%
13	Yippee	98%
14	Nestle	95.8%
15	Himalaya Baby	99%
16	Dove	99%
17	Dove Himalaya Baby	99%, 97%
18	Yippee, Himalaya Baby, Dove	98%, 96%, 96%

A total of 186000 images were used for training and 5000 images were used for testing. Accuracy is obtained by comparing the predicted bounding boxes with the ground truth bounding boxes. Using LabelImg [25], ground truth of the training and testing set is created. It includes the coordinates of the bounding boxes. This is compared with the predicted boxes. The overall testing accuracy is 96%. This approach can be used for gamification in the retail domain. In such cases, the network should be able to detect each logos correctly. Table 3 shows the results obtained when tested on single images. In Figure 12, Pepsi logo was detected with 97%accuracy. Logo detection for Yippee Noodles(98%accuracy) is shown in Figure 13. Similarly, in Figure 14, Nestle logo was detected with 95.8%accuracy. Figure 15 shows the Himalaya Baby Product logo detection with 99%accuracy. Dove logo was detected with 99%accuracy in Figure 16. Figure 17 shows the detection of both Dove soap and Himalaya Baby soap. This implies that our network is able to detect multiple logos at the same time. Dove soap was detected with 99%and Himalaya Baby soap with 97%accuracy. Similarly, Figure 18 shows the detection of Yippee Noodles(98%), Himalaya Baby product(96%) and Dove(96%). Figure 20 shows the loss versus iteration number plot while training and the corresponding values of loss for various iteration number are shown in the Table 4. From the table, it is clear that after each iteration, the loss decreases. We get a loss of 0.0692 at iteration 25,000. In order to test whether the network will be able to run in real-time, a video was taken and given as input to the network. Our network was

Fig. 12. Pepsi output

Fig. 13. Yippee Noodles output

Fig. 14. Nestle output

Fig. 15. Himalaya Baby output

Fig. 16. Dove image output

Fig. 17. Combination output

Fig. 18. Combination output

Fig. 19. Combination output

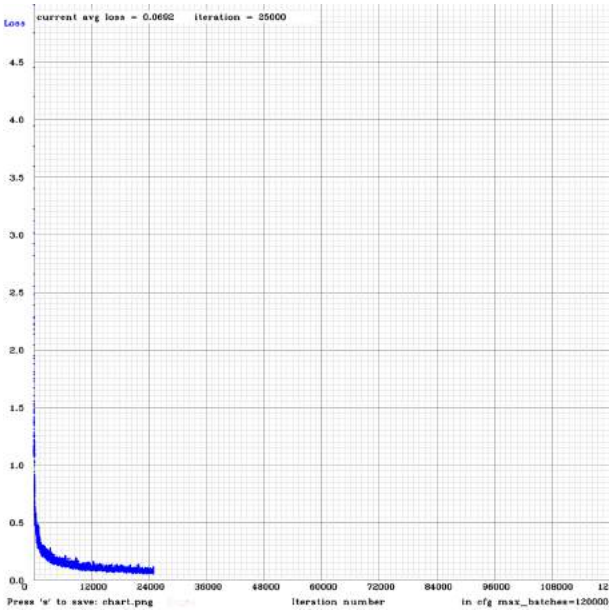

Fig. 20. Loss versus Iteration Number plot

able to detect the logos in it with higher accuracy and higher speed making it suitable for real-time applications.

Inorder to show that this YOLO network can be fine-tuned for object detection in retail domain, detection of Lays was done. Lays images were collected from different stores with different background, different angle and different cameras. A total of 2000 images were collected from different stores and data augmentation was done to increase the dataset. Training was done on 6000 images and testing on 500 images. The overall accuracy obtained was 94%. In Figure 21, the side view of Lays was shown to the camera. The network was able to detect Lays with 99%accuracy. A person holding the Lays packet is shown in Figure 22. Our network correctly identified the Lays in his hand with 96%accuracy. In Figure 23, different Lays were detected with accuracies 95.5%, 98%, 99%, 100%, 96%, 100%and 98%. Lays in Figure 24 was detected with

TABLE IV
LOSS VERSUS ITERATION NUMBER VALUES

Iteration Number	Loss
1	15.535951
500	0.44584
1000	0.352658
4600	0.298553
6750	0.15837
10000	0.110130
15000	0.089001
20000	0.074990
22800	0.072008
25000	0.0692

98%accuracy. Figure 25 shows both Lays and Kurkure hanging in a shop. Our network successfully identified different lays from this with 99%, 99%, 96%,98%and 95%accuracies. Lays in Figure 26 was detected with 98%accuracy. If the dataset is further increased, 100%accuracy may be achieved for every image. Table 5 shows the above results in a glimpse.

Fig. 21. Side view of Lays image

Fig. 22. A person holding Lays

Fig. 23. Group of Lays image

VI. CONCLUSION

We designed a novel logo detection network using YOLO for retail domain. Detection of Pepsi, Dove, Himalaya Baby products, Yippee Noodles and Nestle was done here. Unavailability of the dataset was the most challenging problem. Database was created by collecting images from different stores and from other Internet resources. Diffrent conditions such as different background, different lighting conditions, different angles and different camera clarity were to be considered while creating the database. The network was able to detect all the logos shown in an image with more than 95%accuracy. When a video was shown to the network, it was able to detect the logos in it with higher accuracy and higher speed making it suitable for real-time applications. Logo detection can be extended to more products. Also, the dataset can be increased to improve the accuracy. Once fine tuned for logo detection, the same approach can be used for object detection in retail for self checkouts.

Fig. 24. Side view of Lays image

Fig. 25. Lays and Kurkure image

Fig. 26. Lays image

TABLE V
RESULTS OBTAINED FOR LAYS

Figure Number	Accuracy
21	99%
22	96%
23	95.5%, 98%, 99%, 100%, 96%, 100%, 98%
24	98%
25	99%, 99%, 96%, 98%, 95%
26	98%

REFERENCES

[1] Alireza Alaei and Mathieu Delalandre, "A Complete Logo Detection/Recognition System for Document Images", *11th IAPR International Workshop on Document Analysis Systems, 2014.*

[2] Zhong-Qiu Zhao, Shou-tao Xu and Xindong Wu, "Object Detection with Deep Learning: A Review", *IEEE Transactions on Neural Networks and Learning Systems.*

[3] D. G. Lowe, "Distinctive image features from scale-invariant key points", *International Journal of Computer Vision, Vol. 60, No. 2, 2004.*

[4] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection", *CVPR, 2005.*

[5] Pedro Felzenszwalb, David McAllester and Deva Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model", *IEEE Trans. Pattern Anal. Mach Intell, Vol. 32, 2010.*

[6] C. Cortes and V. Vapnik, "Support Vector Machine", *Machine Learning, Vol. 20, No. 3, 1995.*

[7] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application of boosting", *J. of Comput. and Sys. Sci., Vol. 13, No. 5, 1997.*

[8] P. Viola, M. J. Jones and D. Snow, "Detecting pedestrians using patterns of motion and appearance", *The 9th ICCV, Nice, France, Vol. 1, 2003.*

[9] P. Viola, M. J. Jones and D. Snow, "Robust Real-Time Face Detection", *International Journal of Computer Vision, Vol. 57, 2004.*

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks", *International Journal of Computer Vision, Vol. 57, 2004.*

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *NIPS, 2012.*

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

[13] J. R. Uijlings, K. E. Van De Sande, T. Gevers and A. W. Smeulders, "Selective Search for Object Detection", *International Journal Of Computer vision, Vol.104, No.2, 2013.*

[14] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE Trans. Anal. Mach. Intell., Vol.37, No.9, 2015.*

[15] Karel Lene and Andrea Vedaldi, "RCNN-R", *International Journal of Computer Vision, Vol.37, No.9, 2015.*

[16] R. Girshick, "Fast r-cnn", *ICCV, 2015.*

[17] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", *IEEE Trans. Pattern Anal. Mach. Intell, Vol. 39, no.6, 2017.*

[18] Y. Li, K. He and J. Sun, "r-fcn: object detection via region-based fully convolutional networks", *NIPS, 2016.*

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, "Ssd: Single shot multibox detector", *ECCV, 2016.*

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *NIPS, 2014.*

[21] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection", *CVPR, 2016.*

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger", *CVPR, 2016.*

[23] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", *CVPR, 2017.*

[24] Chung Yu Wang and Royce Cheng-Yue, "Traffic Sign Detection using You Only Look Once Framework", *CVPR, 2017.*

[25] "LabelImg", https://github.com/tzutalin/labelImg.

[26] "Intersection over Union ", https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[27] "Non-Maximum Suppression algorithm ", https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/.