REAL TIME LOGO RECOGNITION USING YOLO ON ANDROID

Primbetov Abbaz[1], Saidova Fazilat[2,] Primbetov Aziz[3], and Yembergenova Ulmira[4]
[1,2]*Tashkent University of Applied Sciences, Gavhar Str. 1, Tashkent 100149, Uzbekistan*
[3,4]*Nukus Branch of Tashkent University of Information Technologies Named After Muhammad Al-Khwarizmi, 74, A. Dosnazarov street, Nukus, Uzbekistan*
abbaz0203@mail.ru,aziz2212@mail.ru
https://doi.org/10.5281/zenodo.10471718

Keywords: Object detection, Convolutional Neural Network (CNN), You Only Look Once (YOLO), Faster R-CNN (Region-based Convolutional Neural Networks).

Abstract: Humans can easily detect and identify objects present in an image. The human visual system is fast and accurate and can perform complex tasks like identifying multiple objects and detect obstacles with little conscious thought. For a long time, humans have been trying to make computers understand what is on the images. With the availability of large amounts of data, faster Graphics Processing Unit (GPU)s, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy. The goal of this paper is to implement an object detection model suitable in terms of size and speed to run on an Android device and detect logos in real-time. The proposed approach is based on YOLOv2 (You Only Look Once) state-of-the-art, real-time object detection for logos and this project used the FlickrLogos-32 dataset. The experimental results show that we obtained a final accuracy of 82.3% and a speed of 35 fps (frames per second) on the NVidia GeForce GTX 1070.

1. INTRODUCTION

A logo is a graphical mark used to identify a company, organization, product or brand. Logos are used to represent a company's name, a particular product or service and are used heavily vin the marketing of products and services. Logos have become an integral part of a company's videntity and a well-recognized logo can increase a company's goodwill. A logo usually has a recognizable and distinctive graphic design, stylized name or unique symbol for identifying an organization. It is affixed, included, or printed on all advertising, buildings, communications, literature, products, stationery, vehicles, etc. Logo can be seen anywhere in the surrounding in our vdaily life, such as in the streets, supermarkets, on the products or services, on administrative documents, etc. Examples of different logos are shown in Figure 1. Logo detection is a challenging object recognition and classification problem as there is no clear definition of what constitutes a logo. A logo can be thought of as an artistic expression of a brand, it can be either a (stylized) letter or text, a graphical figure or any combination of these. Furthermore, some logos

Figure 1: Some figures illustrate that logos appear everywhere in our surrounding.

have a fixed set of colors with known fonts while others vary a lot in color and specialized unknown fonts. Additionally, due to the nature of a logo (as brand identity), there is no guarantee about its context or placement in an image, in reality logos could appear on any product, background or advertising surface. Also, this problem has large intra-class variations e.g. for a specific brand, there exist various logos types (old and new Adidas logos, small and big versions of Nike) and inter-class variations e.g. there exists logos which belong to different brands but look similar (see Figure 2).

Figure 2: Logo variations exemplar images
Left variations of brands Adidas. Notice, different graphical figures. Right variations of brands Chanel - Gucci, Vodafone, Target, beats, Bebo and Pinterest. Notice, similar looking logos but belong to different brands.

2. RELATED WORK

The problem of logo recognition itself has a rich research history. In the 1990's the problem was mainly explored in information retrieval use-cases. An image descriptor was generated using affine transformations and stored in a database for retrieval. There were also some neural network-based approaches but the networks were not as deep nor the results as impressive as recent work.In the 2000's, with the advent of SIFT and related approaches better image descriptors were possible. This method provides representations and transformations to image gradients that are invariant to affine transformations and robust when facing lighting conditions and clutter. A Recent initiative in logo recognition uses deep neural networks, which offer superior performance with end to end pipeline automation, i.e. from image and logo identification to recognition. Multiple methods for object detection using CNNs have been presented this recent year. The Region-Based Convolutional Neural Network (R-CNN) is an architecture that locates and classifies multiple objects by combining a CNN and an external region proposal method. A region proposal method is an algorithm that outputs thea set of regions of interest, typically defined with bounding boxes. A commonly used region proposal method is Selective Search. This algorithm proposes regions of interest by using similarity measures based on color and visual features. R-CNN method crops and resize each region of interest and classifies them using a CNN. The original architecture uses a CNN with five convolutional layers and two fully connected layers, although any CNN classifier can be used.Some more complex methods for object detection include Fast R-CNN and Faster R-CNN. Fast R-CNN is a method based on R-CNN in which the full image is processed by the convolutional layers and then, regions of the output of the last convolutional layer are cropped and classified. The network is formed by a set of convolutional layers, fully-connected layers, an external region proposal method (typically Selective Search) and a Region of Interest (RoI) pooling layer. The RoI pooling layer applies max-pooling to each region of interest using a grid of a fixed size (typically 7×7).

Fast R-CNN also introduces a bounding box regressor, a layer that outputs a fine-tuned location of bounding boxes. Faster R-CNN is based on Fast R-CNN but substitutes the external region proposal methods by a Region Proposal Network (RPN). RPN is a neural network that generates regions of interest using the features of the output of the last convolutional layer. RPN is formed by a 3×3 sliding window that outputs a set of bounding boxes (typically 9) with different sizes and aspect ratios and a fully connected layer that assigns a binary class (foreground or background) to each bounding box.

Many other object detection algorithms, including the previous ones described, output several overlapping bounding boxes. In order to merge them, the Non-Maximum Suppression (NMS) algorithm is used. NMS removes a bounding box if it largely overlaps with another bounding box of the same class with a higher confidence score. New methods for object detection based on deep learning are constantly appearing. Some of them include: Single Shot Detector (SSD) or You Only Look Once (YOLO) and YOLOv2. This method typically provides faster performance than Faster R-CNN but obtains a lower accuracy. YOLO is a recent, unified CNN based

object detection model, proposed by Joseph et. in 2016. It explores using a single network to predict both objects' positions and class scores at one time. The motivation is to reframe the detection problem as a regression problem, which regresses from the input image directly to class probabilities and locations. Benefit from the unified design, YOLO's detection speed is many times faster than other state-of-the-art methods .

3. NETWORK ARCHITECTURE

YOLOv2 is an improved version of YOLOv1 introduced in (Redmon et al. 2016b). We applied our project with YOLOv2 because compared to YOLOv1, YOLOv2 is a more accurate and faster detection method. However, the development team also came up with a "tiny" variation which is much smaller than the original. This tiny model-based implementation is called Tiny YOLOv2. Tiny YOLOv2 has 11 layers. Out of these 9 are convolutional and 2 are fully connected. This is much smaller than the regular model which is perfect for android. Figure 3 shows the structure of Fast YOLO. The tiny version is composed of 9 convolution layers with leaky relu activations. Observe that after 6 maxpool the 446x446 input image becomes a 13x13xD image

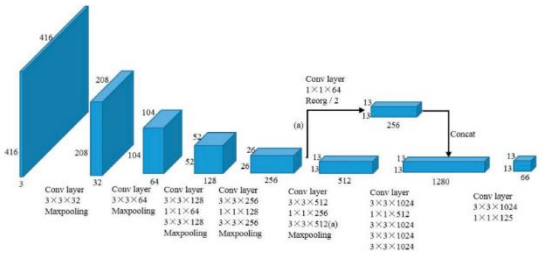


Figure 3: The network of YOLOv2

YOLO divides up the image into a grid of 13 by 13 cells. In object detection, we also have to predict the location and the shape of an object, not only classification. Therefore, the output of an object detection network becomes a little bit complicated. In our case of YOLOv2, the output is a 3-dimensional array (or Tensor in TensorFlow). Particularly in YOLOv2, the shape of output is 13x13xD, where D varies depending on how many classes of objects we want to detect (For example D=5 for a single class). The first 2-dimensional array (13x13) is called grid cells. So, there are 169 grid cells in total.One grid cell is 'responsible' for detecting 5 bounding boxes, that is we can detect up to 5 boxes on a grid cell. This means that the network can detect up to 169 x 5 = 845 boxes at once. This number of bounding boxes a grid cell can detect is actually the number of Anchor-Boxes we prepare, and we can change this number to whatever we want. So, for example, if we want to detect humans and cars and think that just two Anchor-Boxes (vertical rectangle for humans, and horizontal rectangle for cars) are enough to detect them, then the number 5 above becomes 2. (In the paper of YOLOv2, this number is denoted as 'B'). Figure 4: shows the output of the network for YOLOv2 looks like this.

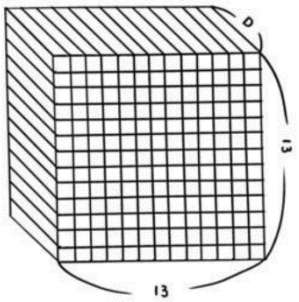


Figure 4: The output of the network for YOLOv2

Each grid cell has depth of D. The value of D depends on the number of classes we want to detect. When we have C classes of object, D is D=B(5+C) The output of the network looks like this. There are 13x13 = 169 grid cells in total, and each grid cell can detect up to B bounding boxes. One bounding box has 5 + C properties, therefore a grid cell has D = Bx(5+C) values (this is depth) Tensor=SxSxSx(5+c) In our case classes number C=30 and B=5

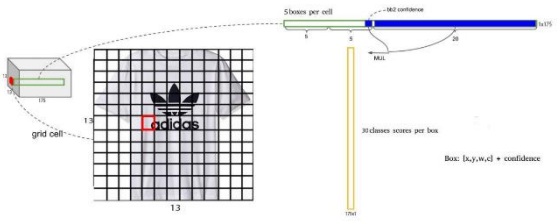


Figure 5: This 13x13 tensor can be considered as a 13x13 grid representing the input image, where each

cell of this tensor will hold the 5 box definitions and 30 class probabilities.

The input to the network is 416x416x3 image in YOLOv2-tiny. There is no fully connected layer in it. (Table 1)

Layer	kernel	Stride/Filters	Output shape
Input			416x416x3
Convolution	3×3	1/16	416x416x16
MaxPooling	2×2	2	208x208x16
Convolution	3×3	1/32	208x208x32
MaxPooling	2×2	2	104x104x32
Convolution	3×3	1/64	104x104x64
MaxPooling	2×2	2	52x52x64
Convolution	3×3	1/128	52x52x128
MaxPooling	2×2	2	26x26x128
Convolution	3×3	1/256	26x26x256
MaxPooling	2×2	2	13x13x256
Convolution	3×3	1/512	13x13x512
MaxPooling	2×2	1	13x13x1024
Convolution	3×3	1/1024	13x13x1024
Convolution	3×3	1	13x13x1024
Convolution	1×1	1/175	13x13x175

Table 1: Details of Network

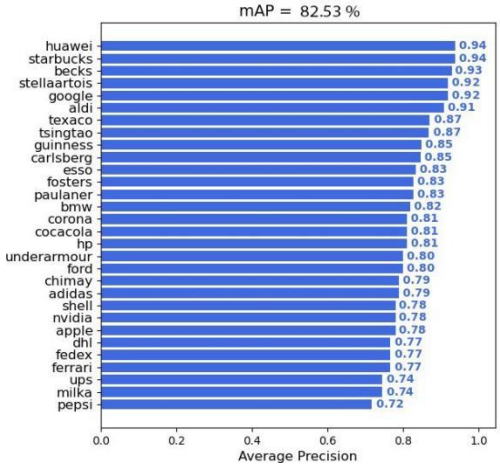
4. EXPERIMENTAL RESULTS

In our project we used FlickrLogos-32 dataset. The FlickrLogos-32 dataset contains photos showing brand logos and is meant for the valuation of multi-class logo recognition as well as logo retrieval methods on real-world images. Logos of 32 different logo classes and 6000 negative images were collected by downloading them from Flickr. The dataset includes images, ground truth, annotations (bounding boxes plus binary masks), evaluation scripts and pre computed visual features. The dataset FlickrLogos-32 contains photos depicting logos and is meant for the evaluation of multi-class logo detection/recognition as well as logo retrieval methods on real-world images. One of the most time-consuming and costly processes in constructing the Flickrlogos-32 database is to annotate logo objects from the collected product images. For each product image,a logo annotator needsto identify the logo objects, annotate the bounding box of each logo object, and then tag it with the corresponding logo class id. Figure 5 shows examples of logo object annotation on product images.

Figure 5: Instruction example of logo object annotation.The left-hand side is rejected due to too loose bounding box.

4.1 Metric

mAP (mean Average Precision) is a popular metric in measuring the accuracy of object detectors like YOLO, SSD, etc. Average precision computes the average precision value for recall value over 0 to 1.
Using this criterium, we calculate the precision/recall curve. Then we compute a version of the measured



precision/recall curve with precision monotonically,

Figure 6: Show us the result of mean average precision (mAP)

by setting the precision for recall r to the maximum precision obtained for any recall r' > r. Finally, we compute the AP as the area under this curve by numerical integration. No approximation is involved since the curve is piecewise constant and finally, we can calculate mean average precision object detection(mAP), resulting in a mAP value from 0 to 100%

(Mean average precision) of 82% and it can track logos very smoothly. In mobile android phones

(Honor 9) we have made the process result as shown in Figure 12 by conducting a series of experiments, the quantitative performance measure of logo detection. Training dark flow and our custom CNN architecture took an immense amount of time. We trained our models in batches of 64 in 8 mini-batches. This allowed us to efficiently train 64 images every step.

Training on a NVidia GeForce 1070, each step took 0.5 seconds. This allowed us to train each model for 2000 epochs, so we can observe the early stopping point and the weights that gave us the best accuracies. YOLO's implementation allowed us to save our weight files every 10000 steps, so we just let it continually train overnight so we can scrap the accuracy in the morning using a script. We have significant results that show our model works better with our dataset above with a little less than 2000 epochs. We trained up to 2000 epochs and the accuracy peaked at epoch 1500. We experimented with running different learning rates our accuracy never got any better.

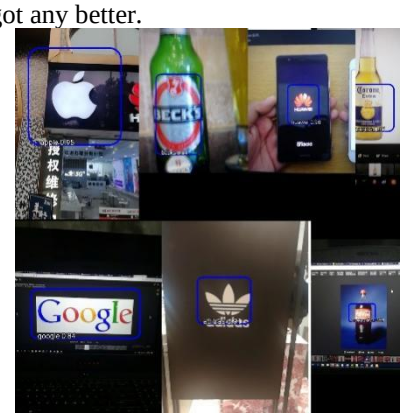


Figure 7. Shows the logo detection through Honor 9.

CONCLUSIONS

I have trained the model on the FlickrLogos-32 dataset and experiment results to show that YOLOv2 performs very well in real-time logo detection. By performing a comprehensive analysis of YOLOv2 over FlickrLogos-32 dataset, we found that the experiment result showed that we managed to achieve a final mean average precision (mAP) 82.53% and 30-35 FPS (frames per second) speed on an NVIDIA GeForce Gtx 1070 and our models performed well at the detection, with very low false-positive rates possible for a fairly reasonably. The application runs smoothly on the current test hardware. However, the main part of the goal was successfully implemented, a working application that utilizes a neural network model for object detection.

REFERENCES

[1] [1] Feh´erv´ari, I., Appalaraju, S. (2019, January). Scalable logo recognition using proxies. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 715-725). IEEE.

[2] [2] Su, Hang, Xiatian Zhu, and Shaogang Gong. "Open logo detection challenge." arXiv preprint arXiv: 1807.01964 (2018).

[3] [3] Oliveira, G., Fraz˜ao, X., Pimentel, A., Ribeiro, B. (2016, July). Automatic graphic logo detection via fast region-based convolutional networks. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 985-991).IEEE.

[4] [4] Hoi, S. C., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., Wu, Q. (2015). Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. arXiv preprint arXiv: 1511.02462.

[5] [5] Shafiee, M. J., Chywl, B., Li, F., Wong, A. (2017). Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. arXiv: Computer Vision and Pattern Recognition.

[6] [6] Feh´erv´ari, Istv´an, and Srikar Appalaraju. "Scalable logo recognition using proxies." 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019.

[7] [7] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. Neural information processing systems.

[8] [8], S., He, K., Girshick, R., Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[9] [9] Le, Viet Phuong. "Logo detection, recognition and spotting in context by matching local visual features." PhD diss., Universit´e de La Rochelle, 2015.

[10] [10] Eggert, C., Brehm, S., Winschel, A., Zecha, D. and Lienhart, R., 2017, July. A closer look: Small object detection in faster R-CNN. In 2017 IEEE international conference on multimedia and expo (ICME) (pp. 421-426). IEEE.