

IT Carlow - BSc.  
Software development

# Automatic Detection Of Brand Logos Design Manual

Student Name: Zhe Cui  
Student Number: C00266169  
Lecturer: Paul Barry  
Date: 30.04.2021



## **Abstract**

The purpose of this document is to provide a detail of the system design of the Automatic Detection of Brand Logos System. Automatic Detection of Brand Logos is a tool identifying brand logos in still and moving images, and calculate how long the logo is visible. This document provides detailed instructions for users.

# Table of Contents

1	Introduction.....	1
2	System design .....	1
3	Dataset .....	2
4	Algorithm .....	2
4.1	Faster R-CNN .....	2
4.2	Single Shot MultiBox Detector (SSD).....	3
4.3	You Only Look Once version3 (Yolov3).....	5
5	Video detection and timer .....	5
	References.....	7

## Table of Figures

Figure 2-1 System flow chart .....	1
Figure 4-1 Architecture of Faster R-CNN .....	3
Figure 4-2 Architecture of Single Shot MultiBox .....	4
Figure 4-3 MultiBox Detector .....	4
Figure 4-4 Architecture of Yolov3 .....	5
Figure 5-1 Defining a dictionary .....	6
Figure 5-2 Defining a set.....	6
Figure 5-3 Calculating how long the logo is visible.....	6

# 1 Introduction

Automatic Detection of Brand Logos is a tool identifying brand logos in images and videos, and calculate how long the logo is visible. It can help evaluate marketing campaigns, capture user reviews of the product, counterfeit detection and protect brands' intellectual property, personalize product recommendations, improve search algorithms.

## 2 System design

High-quality data sets are the basis of object detection. The quality of data sets determines the accuracy of results. The first step in this project is finding the appropriate data set, then looking for the most suitable algorithm for the requirements and data set of the project, then training the model, and finally testing.

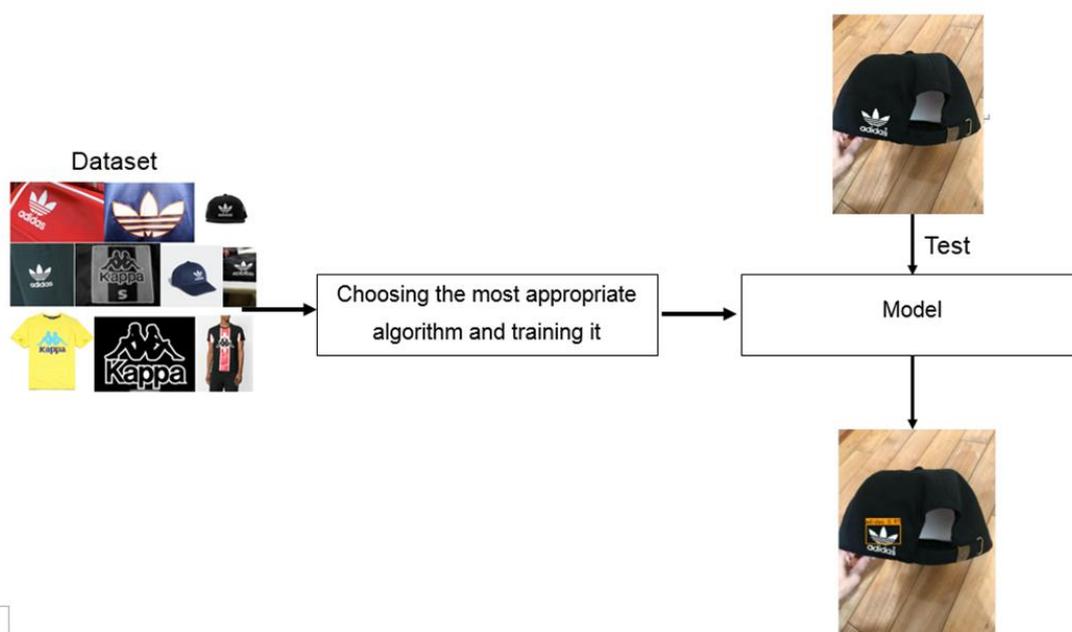


Figure 2-1 System flow chart

### **3 Dataset**

The first step is to find the public logo dataset and analyze whether the public data set is suitable for this project. If there is no suitable public data set, build the data set for this project. To build a dataset, a large number of pictures are needed. It is time-wasting and effort-wasting to download images one by one, using web crawlers to obtain data would be much easier, then manually filtering the data, and then manually labeling them.

### **4 Algorithm**

In the research stage, three algorithms, Faster R-CNN, SSD, and Yolov3, were preliminarily selected. After selecting the appropriate data set, the three algorithms are trained and evaluated respectively to select the most appropriate algorithm for this project.

#### **4.1 Faster R-CNN**

The Faster R-CNN architecture consists of the Regional Proposal Network (RPN) as a region proposal algorithm and the Fast R-CNN as a detector network [1]. The process of R-CNN is as follows:

- The input images are processed to obtain a large number of object candidate regions.
- The features of candidate regions are obtained through the convolutional network.
- SVM classifies features and the category of the target could be obtained.
- Finally, the overlapping candidate boxes were eliminated by non-maximum suppression to obtain an accurate boundary box.

Fast R-CNN makes an improvement based on R-CNN and inputs the candidate box and the whole image into the convolutional network. The convolutional layer learns the full image features, generates the global feature map, and then uses the ROI Pooling layer to intercept the corresponding candidate box regional features on the global feature map. Finally, the classification is carried out through the full connection layer.

Faster R-CNN is modified based on Fast R-CNN, eliminating the process of generating candidate boxes separately, directly generating the boundary boxes of candidate regions and global feature maps through an RPN network. Then, the full connection layer is used for classification and the boundary box of the candidate region is modified again.

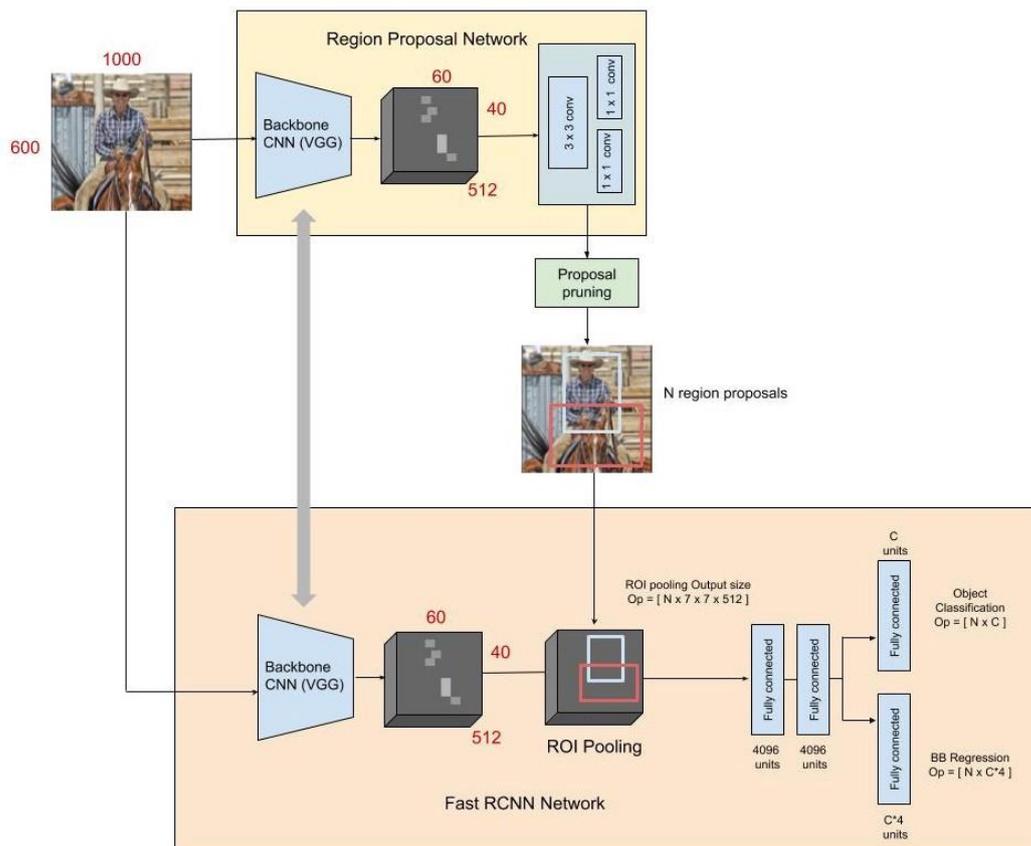


Figure 4-1 Architecture of Faster R-CNN

Source: <https://medium.com/@aviad10m/faster-r-cnn-tl-dr-7d202c2201d> (2018)

## 4.2 Single Shot MultiBox Detector (SSD)

SSD is a single-shot detector. It predicts the boundary boxes and classes directly from feature maps in one single pass [2]. The backbone feature extraction network of SSD is the traditional image classification network, such as VGG16. The process of SSD is similar to Yolo, which is also based on the calculation of object regression. The difference is that Yolo uses global features for prediction, while SSD uses local features. The SSD also uses the anchor mechanism of Faster R-CNN to establish

a correspondence between a location and its features. As shown in Figure 4-3, an  $8 \times 8$  feature map is set in Figure 4-3(b), and a  $3 \times 3$  sliding window is used for the region proposal. Figure 4-3(c) represents the result of regression calculation, including classification information and corresponding coordinate information.

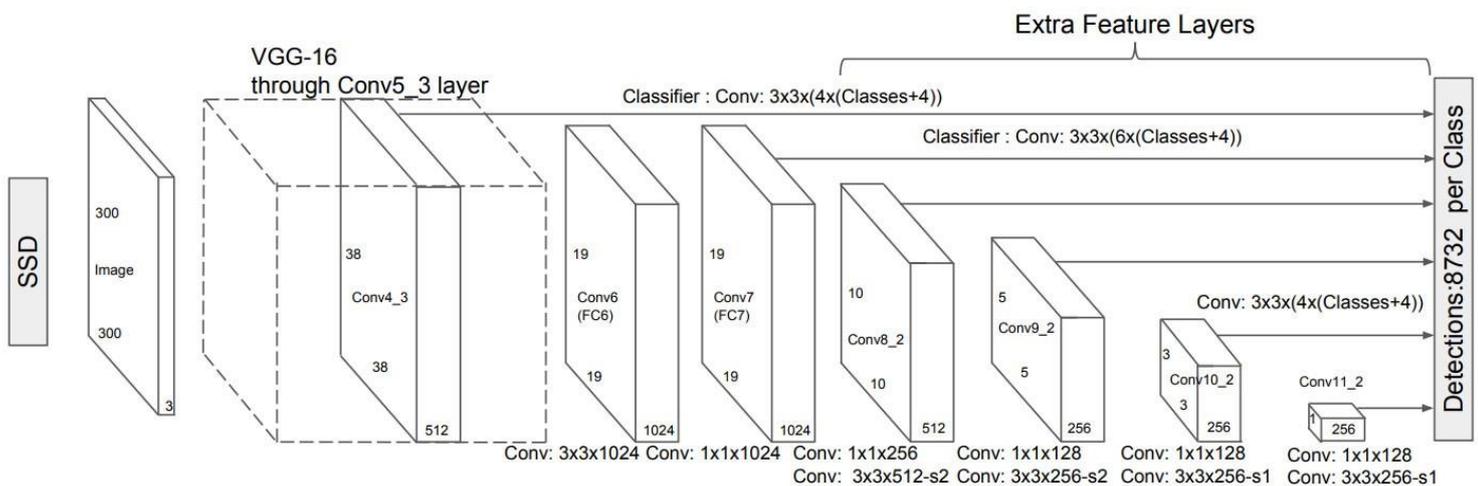


Figure 4-2 Architecture of Single Shot MultiBox

Source: Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October.

Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

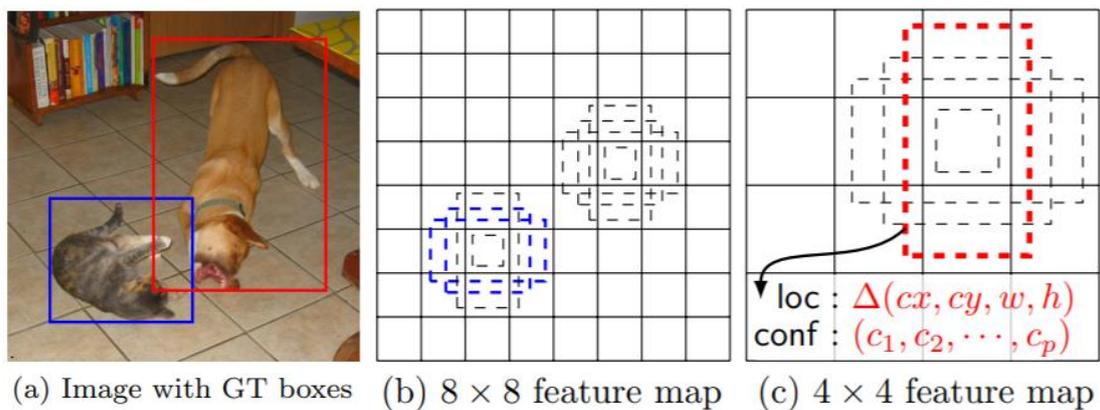


Figure 4-3 MultiBox Detector

Source: Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

### 4.3 You Only Look Once version3 (Yolov3)

The core idea of Yolo is to divide the image into regions for prediction. When the coordinate of the center point of the detected object is located in a grid of the feature graph, the grid is responsible for the output of the confidence and regression boundary box of the object. Yolo algorithm takes advantage of the idea of regression. It directly uses the regression method on the original input image and regresses the position and category of the bounding box on the possible objects.

Yolov3 uses Darknet 53 (with 52 convolutional layers) as the backbone network [3]. It used the Residual Network method, which is setting up shortcut connections between some layers. Yolov3 proposed a pyramid structure, which is similar to the Feature Pyramid Network (FPN). The feature maps output by the Feature Network were upsampled twice and then merged with the corresponding shallow Feature maps in the Feature Network, finally obtaining the multi-scale output tensor. In this way, the detection effect of small targets has been improved.

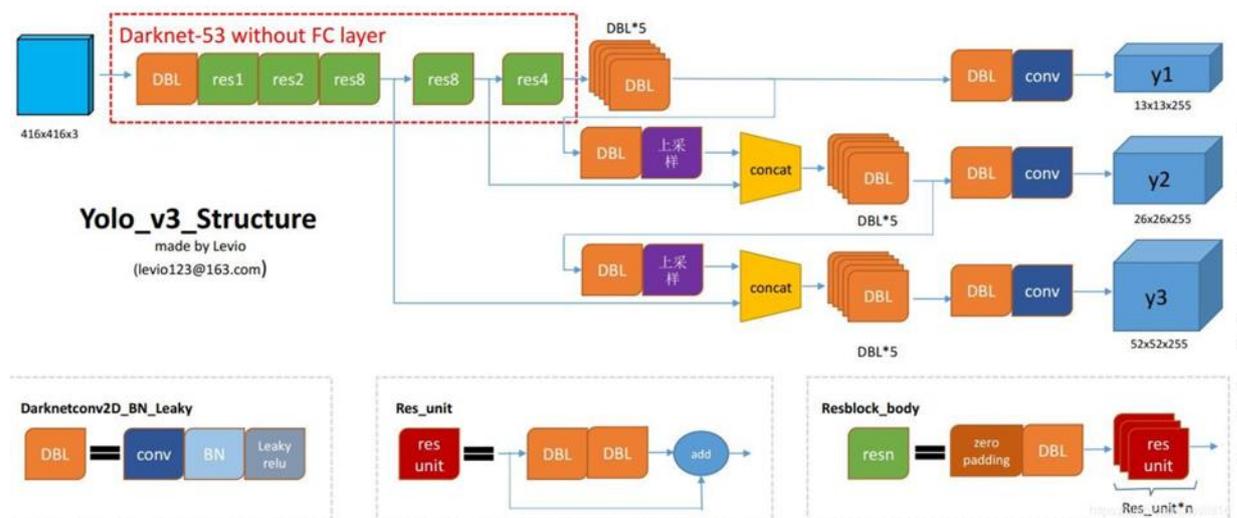


Figure 4-4 Architecture of Yolov3

Source: <https://blog.csdn.net/leviopku/article/details/82660381> (2018)

## 5 Video detection and timer

The VideoCapture Class in OpenCV can capture video, and read the video by frame. A frame is an image of each frame, which is a three-dimensional matrix.

Then detect each frame from the video stream, the logos in the video is identified.

Defining a dictionary to store each brand and duration as the key-value pairs. Then define a set that incrementing the logo's value by 1 every time a logo appears in the image. Sets cannot have two items with the same value, duplicate values will be ignored, which can help avoid the issue of double counting when multiple identical logos appear in a picture. A second is generally considered to be 24 frames. Dividing the value of each logo in the dictionary by 24 (Frames PerSecond) to get how long logo appears in the video.

```
self.band_dict = {  
    '361': 0.0,  
    'adidas': 0.0,  
    'anta': 0.0,  
    'erke': 0.0,  
    'kappa': 0.0,  
    'lining': 0.0,  
    'nb': 0.0,  
    'nike': 0.0,  
    'puma': 0.0,  
    'xtep': 0.0,  
}
```

Figure 5-1 Defining a dictionary

```
small_pic=[]  
set_out_classes = out_classes  
for i in set(set_out_classes):  
    set_predicted_class = self.class_names[i]  
    self.band_dict[str(set_predicted_class)] += 1
```

Figure 5-2 Defining a set

```
46 bd_list = (yolo.band_dict[k]//24.0 for k in yolo.band_dict.keys())  
47 bd_list = list(bd_list)
```

Figure 5-3 Calculating how long the logo is visible

## References

- [1] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.
- [2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
- [3] Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.