Object Detection & Count in Image

Prince Shah[1], Pratham Shah[2], Mit Thakkar[3]

Guided By

Prof. Sejal Thakkar[4]

[DEPARTMENT OF COMPUTER ENGINEERING]

[1]Prince Shah IU1741050052 CE Indus University,
[2]Pratham Shah \IU1741050051 CE Indus University
[3]Mit Thakkar IU1741050059 CE Indus University
[4]Professor Sejal Thakkar Dept. of Computer Engineering, Indus University, Gujarat , Ahmedabad

---***---

Abstract - *In this project, we have studied three to four research papers and came across different methods to implement object count. Firstly, we implemented a simple object detection using Open CV and Tensor flow in Python. Secondly, we studied YOLO (You Only Look Once) and its various applications thoroughly, using which we implemented YOLO framework for detecting various objects and finally getting a count of them. Our main objective was to get the yield (count) of different types of fruits on a tree .*

***Key Words*: Project, YOLO, Look, Tensor, Flow, Object, Detection.**

1. INTRODUCTION

The project implemented here is all about learning about different algorithms to implement the object detection and finally getting a count, using various techniques like Convolutional Neural Networks (CNN), ANN, and finally using YOLO.

1.1 Project Scope

We have developed an algorithm which helps us to detect the particular fruit on a tree and after that it counts the total number of that particular fruit on the tree. We have been able to get a decent amount of accuracy, approximately 97% in counting the fruits. Finally, complete content and organizational editing before formatting.

1.2 Objective of Study

The objective of this project is to develop a machine learning algorithm that allows us to detect different types of fruits on trees and to get the count with maximum accuracy.

1.3 Features of the Project

This object detection and count can be applied on any image irrespective of its size and its format like jpg or png. Using YOLO framework has proven to be quite advantageous as it has a very high efficiency, it only looks once at the image and is able to detect and count the objects. Hence, we are also able to get very high speed.

1.4 Advantages

CNN has been and is one of the finest techniques for object detection as it uses various layers, it also has various updates in the form of RCNN, and many other neural network based algorithms. YOLO (You Only Look Once) is a framework which is very fast and takes minimal amount of the CPU.

1.5 Disadvantages

It takes a lot of amount of time for the people and also a higher disk spaced and high-speed RAM is required for implementing CNN (for training and labeling data). 2. As in the case of YOLO, it is difficult to label and detect objects of smaller dimension, also overlapping objects are difficult to identify as different and sometimes object not present in enough intensity of light could also not be detected, which are some of the disadvantages of YOLO framework.

2. - LITERATURE SURVEY

Literature survey was done on 3-4 research papers which were read by us and we decided to take into scope only 1-2 out of them. One of the researches has implemented this project using blob detection neural network and the other one was implemented using Convolutional Neural Network by testing on synthetic images generated by Gaussian Filter. These research papers will be discussed later on in this report

2.1 Literature Review

Many different researches are carried out on these topic in many different domain. Along with the theoretical approach, it is necessary to practically implement it. We have made attempts to implement it using the machine learning model.

3. – APPROACH

Libraries that can be used: Tensor Flow - It is an open source framework that makes it easy to construct train and deploy object detection models. Image AI - It has made possible for anyone with basic knowledge of Python to build applications and systems that can detect objects using only few lines of code. Pillow - This library provides various image filters including the edge detection filters and edge enhancement filters. Steps followed: • Installing Anaconda Virtual Environment. • Install various libraries like Tensor Flow, SciPy, Keras • Activating Tensor Flow • Installing Jupyter Notebook.

3.1 Research Paper Study

We have undertaken study of two research papers which we have gone through in-depth and made an analysis of these two as shown below.

3.2 Counting Apples and Oranges With Deep Learning: A Data Driven Approach

In this research paper we have studied about implementing the detection using blob detection which is based on CNN. The broad steps that were followed in that methods were: 7

1. We collected human generated labels from the set of fruits images.

2. Implement image segmentation by training the images by blob detection fully convolutional neural network.

3. Then use this segmented images to generate an intermediate count of fruits.

4. Then training a linear regression so that we can map intermediate counts with the ground truth of human data.

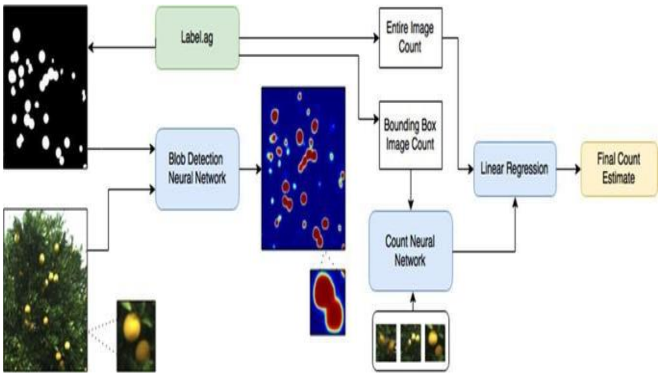


Figure 3.1 Working Of The Deep Learning Approach

The preparation pipeline begins with a given picture. Label.ag then creates the comparing ground truth name map, and these two data sources are utilized to prepare the mass recognition neural system. This neural system yields a fragmented picture, and the pipeline extricates the directions of the bouncing boxes around each mass. These directions are then used to remove the comparing window in the first picture. Label.ag produces the relating ground truth means each jumping box , and these are utilized as contributions to prepare the check neural system. The check neural system at that point gauges and aggregates up the include for each mass in the divided picture to create a middle of the road tally gauge. The middle of the road check gauge is relapsed on the whole picture ground truth tally given by label.ag to create the last tally gauge.

3.4 Deep Count: Fruit Counting Based on Deep Simulated Learning

In this paper we have studied the yield detection by using deep convolutional networks. The manual counting of fruits and yield estimation is very much time consuming and expensive. The network has been trained by synthetic images.

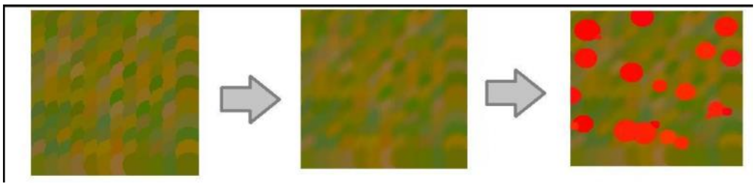


Figure 3.2 Synthetic Image for Testing

Filled the entire image with green and brown spots to copy the background, then applied Gaussian filter for blurring the image and then drawing red circles of different sizes which show us the apple.

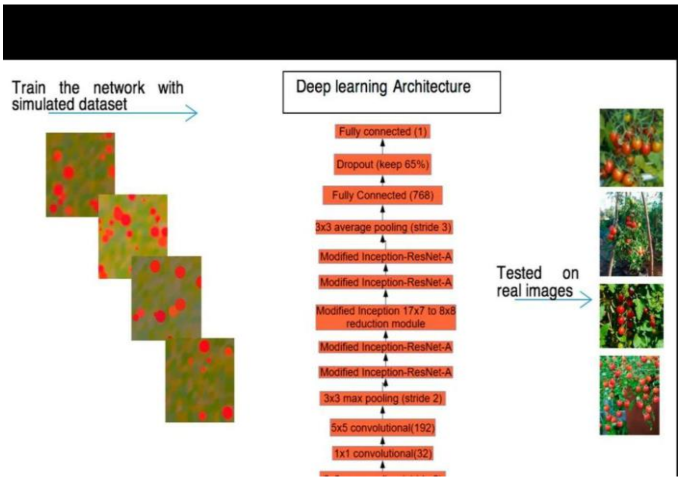
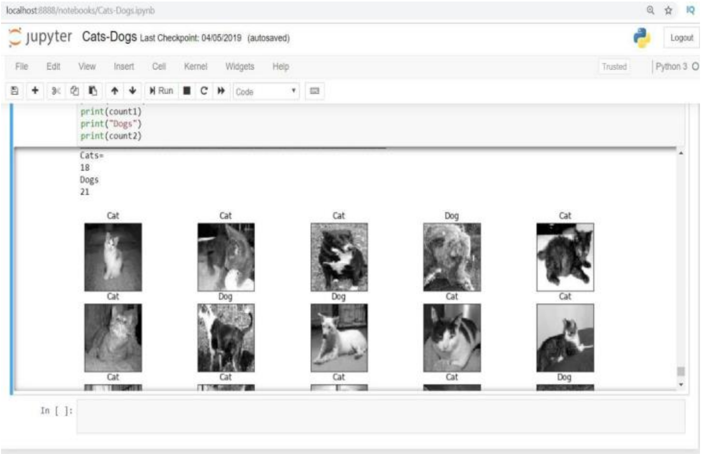

Train the network with simulated dataset
Deep learning Architecture
Fully connected (1)
Dropout (keep 65%)
Fully Connected (768)
3x3 average pooling (stride 3)
Modified Inception-ResNet-A
Modified Inception-ResNet-A
Modified Inception 17x7 to 8x8 reduction module
Modified Inception-ResNet-A
Modified Inception-ResNet-A
3x3 max pooling (stride 2)
5x5 convolutional(192)
1x1 convolutional(32)
Tested on real images

Figure 3.2 Framework of this research

4. –SYSTEM DESIGN & IMPLEMENTATION

In order to implement object detection and count in a particular image, we first came across CNN using Open CV and Tensor Flow. We download two major datasets namely- Cats & Dogs and Cars dataset, on which we ran our algorithm for object detection. Using CNN, we firstly needed to train our dataset appropriately. In the cars dataset we had around 8,144 train images and after training these we had to test the *remaining to make sure that we got the correct output. The cars data sets' test images were around 8,041 out of which only 30-40 of them were tested by us. The accuracy obtained by us was almost close to 100% and we were able to identify Car module as 'Car' and rest of the objects as 'Not a Car'.*

In the same way we performed the training and testing processes on the Cats & Dogs dataset. We performed training on approximately 6,000 images and finally tested our algorithm on around 40 odd images out of the other 6,000 which were available. We identified a Cat image as 'Cat' and a Dog image as 'Dog'.



```
print(count1)
print("Dogs")
print(count2)
```

```
Cats=
18
Dogs
21
```

```
In [ ]:
```

Fig 4.1 Cats & Dogs

4.2 Analysis of Open CV

After implementing both the datasets, on our algorithm of object detection which utilizes CNN, we analyzed that running this algorithm was painstakingly long ,it took almost 45-60 minutes for training and testing the data, and if there would be any error we had to make a small change and run the entire algorithm again. Hence, we decided we had to move to a better and a faster working algorithm. Although Neural Networks being the best method for object detection and the most efficient out of all, we know we were giving up a bit on accuracy while moving to a different piece of code.

4.3 YOLO Implementation

YOLO (You Only Look Once) is a condition of the art ,real time object discovery framework which is very quick and exact. Earlier location frameworks repurpose classifiers or localizers to perform recognition. They apply the model to a picture at numerous measurements and scales. High scoring elements of the picture are viewed as discoveries.

Technique .A solitary neural system is connected to the whole picture. This system partitions the whole picture into various districts and predicts bouncing boxes and probabilities for every area. These cases are weighted by anticipated probabilities.

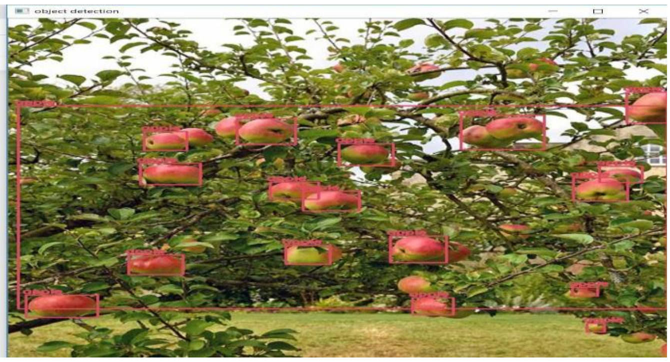
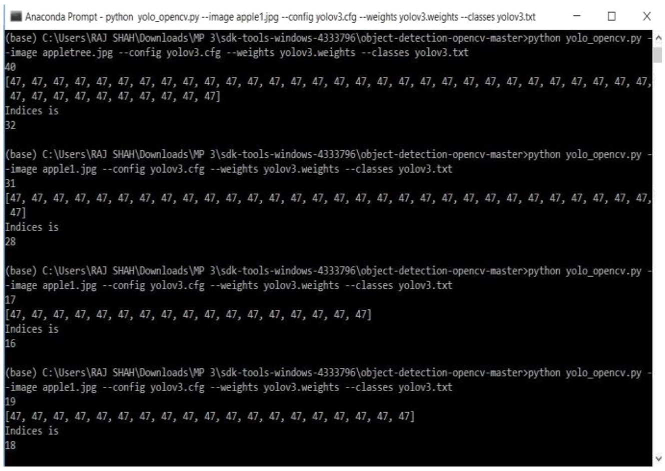


Fig 4.3.1 Apple detection & count using YOLO

5. CONCLUSIONS


Anaconda Prompt - python yolo_opencv.py --image apple1.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt

```
(base) C:\Users\RAJ SHAH\Downloads\MP 3\sdk-tools-windows-4333796\object-detection-opencv-master>python yolo_opencv.py -
-image appletree.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt
40
[47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,
 47, 47, 47, 47, 47, 47, 47, 47, 47, 47]
Indices is
32

(base) C:\Users\RAJ SHAH\Downloads\MP 3\sdk-tools-windows-4333796\object-detection-opencv-master>python yolo_opencv.py -
-image apple1.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt
31
[47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,
 47]
Indices is
28

(base) C:\Users\RAJ SHAH\Downloads\MP 3\sdk-tools-windows-4333796\object-detection-opencv-master>python yolo_opencv.py -
-image apple1.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt
17
[47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47]
Indices is
16

(base) C:\Users\RAJ SHAH\Downloads\MP 3\sdk-tools-windows-4333796\object-detection-opencv-master>python yolo_opencv.py -
-image apple1.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt
19
[47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47]
Indices is
18
```

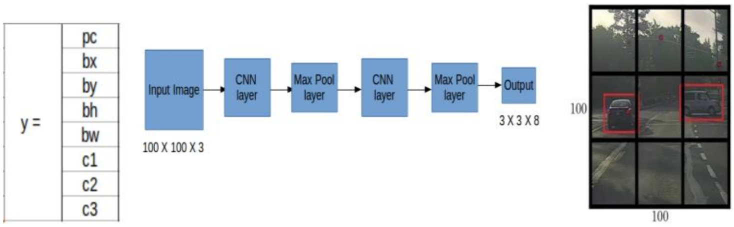
Fig 4.3.2 Output of Count

As shown in the figures, we have run the code which uses blob detection network in order to detect and count the number of objects and uses YOLO framework to give us the final count. We have implemented this code on only one image that is shown in the Fig 4.3.1 which is apples on a tree, and the output is as shown in the Fig 4.3.2. We have used Anaconda Prompt for its implementation and inbuilt YOLO libraries for its weight and different layers' configuration.

5.1 Analysis of YOLO

Some of the main segments of the code that need to be analyzed are as follows:

- *Yolo Framework and encode bounding boxes:*



- *Intersection Over Union(IoU) and Non-Max Suppression(nms):*

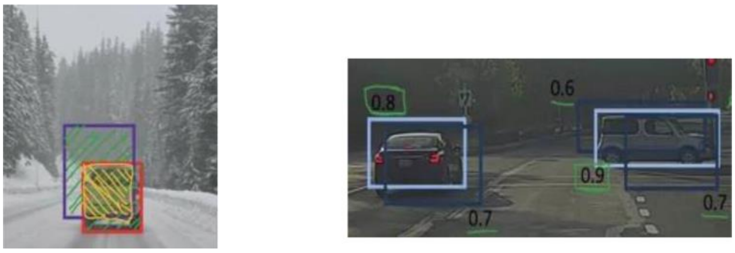
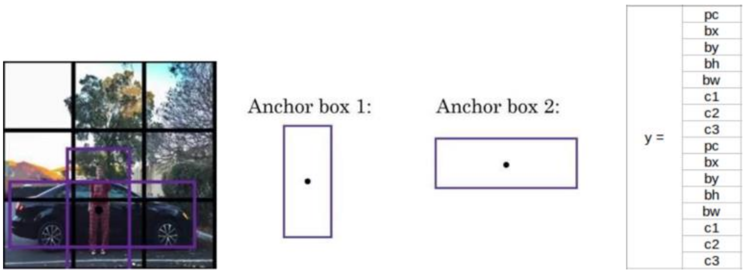


Fig 4.4.2

- *Anchor Boxes:*



- *Blob Detection Network:*

  How does Blob Detection work?

  It is basically a group of connected pixels that share a common property (eg : Gray scale) that is controlled by a set of parameters as mentioned below:

- *Thresholding*
- *Grouping*
- *Merging*
- *Center and Radius Calculation*

1. *Thresholding : Convert the source pictures to a few paired pictures by thresholding the source picture with limits beginning at minThreshold. These edges are augmented by thresholdStepuntil maxThreshold. So the primary edge is minThreshold, the second is minThreshold + thresholdStep, the third is minThreshold + 2 x thresholdStep, thus on.*

2. *Grouping: In every double picture, associated white pixels are gathered. How about we call these paired blobs.*

3. *Merging : The focuses of the double masses in the parallel pictures are processed, and masses found nearer than minimum Distance Between Blobs are merged*

4. *Center & Radius Calculation: The* focuses and radii of the new blended masses are figured and returned.

6 –Conclusions

Subsequent to contemplating the exploration papers we comprehended the diverse strategies by how we can identify and perceive objects utilizing different system models and structures.

We have actualized 2 essential codes for perceiving feline and pooch from different pictures and furthermore object acknowledgment like recognizing an individual, a transport or a cycle from a specific picture.

Subsequent to contemplating some more research papers we went over mass identification neural system which is used to bunch in pixels identifying with some basic property of a picture in the wake of changing over it to gray scale or high contrast.

In the wake of discovering that we connected YOLO on the Coco dataset of different items and saw precisely how the code and different parameters used in the code work.

We additionally went over specific troubles utilizing the Yolo framework since it predominantly works for huge articles like vehicles, pets, truck, bikes and so forth.

We connected to other little items like including apples in an apple tree, oranges and so forth.

There are many practical uses of object detection even though it is somewhat of a new tool in the market, some of them are:

1) Face Detection-Since a long time now everyone points cameras towards faces and hence a feature of detecting faces has become widespread for a more efficient auto-focus.

2) Counting- To count the number of objects in a particular project (this Project in particular), also to count the number of people from a live camera feed.

3) Visual Search Engine- It uses object detection to pipeline various parts of a particular product, let's say you want a specific bag, then you can find lots of examples of related bags. This is much more powerful than Google's Search Engine.

4) Aerial Image Analysis- This has become possible for high quality detection of object like cars, trees and other types of objects which can be viewed from a height. Companies use drone footage to count and detect various objects.

This data would then be utilized for further analysis.

ACKNOWLEDGEMENT

We, Prince Shah (IU1741050052), Pratham Shah (IU1741050051) and Mit Thakkar (IU1741050059), would firstly like to thank Prof. Sejal Thakkar for her guidance and knowledge in the making and development of our Project. We would also like to exclusively thank Dr. Seema Mahajan, our department H.O.D. We also thank our friends, classmates and colleagues who have helped us throughout this semester to understand various application of machine learning and in turn helping to implement this project of object detection.

REFERENCES

1. *Counting Apples and Oranges with Deep Learning: A Data-Driven ApproachSteven W. Chen, Shreyas S. Shivakumar, Sandeep Dcunha, Jnaneshwar Das, Edidiong Okon, Chao Qu, Camillo J. Taylor, and Vijay Kumar.*

2. *Deep Count: Fruit Counting Based on Deep Simulated Learning Maryam Rahnemoonfar and Clay Sheppard*

3. *https://appsilon.com/object-detection-yolo-algorithm/*

4. *https://www.edureka.co/blog/tensorflow-object-detection-tutorial/*