SHIVAJI UNIVERSITY, KOLHAPUR

DEPARTMENT OF TECHNOLOGY

A

MINOR PROJECT REPORT

ON

"OBJECT DETECTION IN IMAGES"

COMPUTER SCIENCE AND TECHNOLOGY PROGRAM

SUBMITTED BY

Sr. No.	Name	Roll No.	Exam Seat No.
1.	Shantanu Y. Balse	1	
2.	Aniket R. Bhosale	2	
3.	Prasad M. Chaudhari	3	

Under the Guidance of

Mr. H. A. Tirmare

Year

2018-2019

CERTIFICATE

This is to certify that the project report entitled

"OBJECT DETECTION IN IMAGES"

Submitted To

DEPARTMENT OF TECHNOLOGY, SHIVAJI UNIVERSITY, KOLHAPUR

Has been completed under my guidance and supervision. To the best of my knowledge and belief, the matter presented in this project report is original and has not been submitted elsewhere for any other purpose.

Submitted by

Name	Roll No.	Exam Seat No.
1) Shantanu Y. Balse	1	
2) Aniket R. Bhosale	2	
3) Prasad M. Chaudhari	3	

H. A. Tirmare

Asst. Professor
DOT, SUK

R. J. Deshmukh

Coordinator (CST)
DOT, SUK

ACKNOWLEGEMENT

.

It is our foremost duty to express our deepest gratitude to the guide **"Mr. H. A. Tirmare"** for his **enthusiastic** guidance and support without which this project might not have come to fruition.

We are also grateful to Mrs. R. J. Deshmukh **(**Coordinator of Computer Science & Technology) for providing all necessary facilities to carry out the project work and whose encouraging part has been a perpetual source of information.

We are highly indebted to **Director Mr. J. S. Bagi** for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We also thank all staff members of our department for their timely help and encouragement, in completion of our project work.

We are indebted to the library personnel's for offering all the help in completing the project work. Last but not only the least we are thankful to our colleagues, our parents and all those who helped us directly or indirectly throughout this project work.

DECLARATION

We, the undersigned, hereby declare that the project report entitled **"OBJECT DETECTION IN IMAGES"** written and submitted by us to **Computer Science and Technology** program, under the guidance of **Mr. H. A. Tirmare** is our original work. The empirical results in this project report are based on the data collected by us.

Sr. No.	Name	Roll No.	Exam Seat No.	Signature
1)	Shantanu Y. Balse	1		
2)	Aniket R. Bhosale	2		
3)	Prasad M. Chaudhari	3		

ABSTRACT

Object Detection is the process of finding real-world object instances like car, bike, TV, flowers, and humans in still images or Videos. It allows for the recognition, localization, and detection of multiple objects within an image which provides us with a much better understanding of an image as a whole. It is commonly used in applications such as image retrieval, security, surveillance, and advanced driver assistance systems (ADAS).

This project was an attempt at developing an object detection system using modern computer vision tools. The result is an easy to use python based application that takes in an image as input and attempts to detect and recognize the various objects in the image. Objects recognized at a confidence rate of greater than 50% are enclosed in a bounding box labelled with the name of the object that the program has guessed. Objects such as person, laptop, bowl, car, dog, cat, clock, spotlight, cup, bicycle, motorcycle, etc. can be detected.

TABLE OF CONTENTS

1. Introduction

1.1 Overview

Object recognition has recently become one of the most exciting fields in computer vision and AI. The ability of immediately recognizing all the objects in a scene seems to be no longer a secret of evolution. With the development of Convolutional Neural Network architectures, backed by big training data and advanced computing technology, a computer now can surpass human performance in object recognition task under some specific settings, such as face recognition.

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos

Methods for object detection generally fall into either machine learning-based approaches or deep learning-based approaches. For Machine Learning approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification. On the other hand, deep learning techniques that are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN).

This project employs the use of deep learning to perform object detection

1.2 Problem Statement

Many problems in computer vision were saturating on their accuracy before a decade. However, with the rise of deep learning techniques, the accuracy of these problems drastically improved. One of the major problem was that of image classification, which is defined as predicting the class of the image. A slightly complicated problem is that of image localization, where the image contains a single object and the system should predict the class of the location of the object in the image (a bounding box around the object). The more complicated problem (this project), of object detection involves both classification and localization.

In this case, the input to the system will be a image, and the output will be a bounding box corresponding to all the objects in the image, along with the class of object in each box. An overview of all these problems is depicted in Fig. 1.

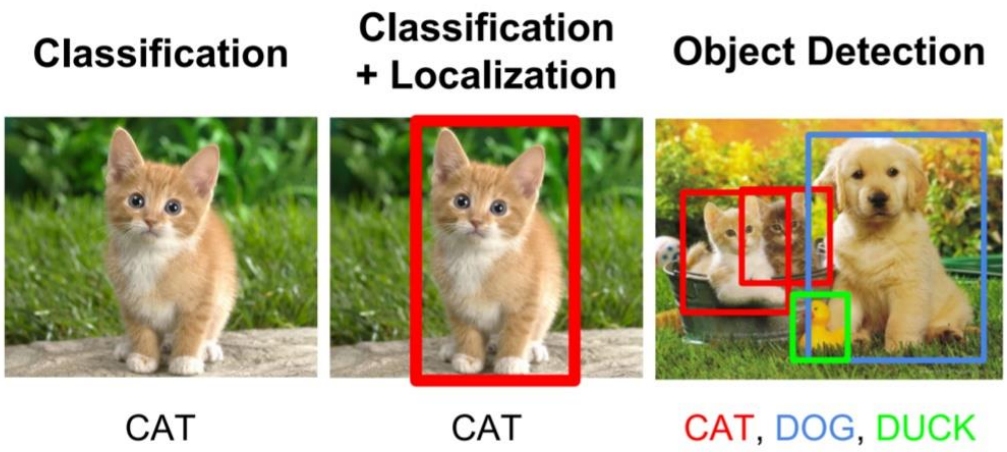


Figure 1: Computer Vision Tasks

Figure 1.1: Computer Vision Tasks

2. Literary Review

Object detection and recognition is an important computer vision problem which has gained increasing attention due to its academic and commercial potential. Although different kinds of approaches have been proposed to tackle this problem, it still remains challenging due to factors like abrupt appearance changes and severe object occlusions.

3. Software Requirements Specification

3.1 Perspective

Most object detection systems available publicly are too complex for non-computer professionals. Most users do not wish to go through the long and arduous process of setting up a development environment on their home computers. It is far more convenient to offer them an interface through which they can get their computer to detect objects in images with the click of a button.

The program must follow the following requirements:

3.2 Requirements

- Must be easy to use.
- Must come with a simple interface.
- Cross platform (must work out of the box on Windows, Mac OS and Linux systems with no or minimal changes).
- The user need not concern themselves with the internal working of the program.

3.3 Working

It should allow the user to input an image file. This will typically be a .jpeg or a .png file. When the user clicks on the "Detect" button, the application should work in the background and output a new image. This image will have bounding boxes drawn across all the objects that the program has managed to detect. The bounding boxes will also have a label attached to them that tells the kind of object it is. This may include objects such as person, dog, cat, car, etc.

The following images show the input and output images before and after passing through the program respectively:

Before:

Figure 3.1 Dog and bicycle, before object detection

After:

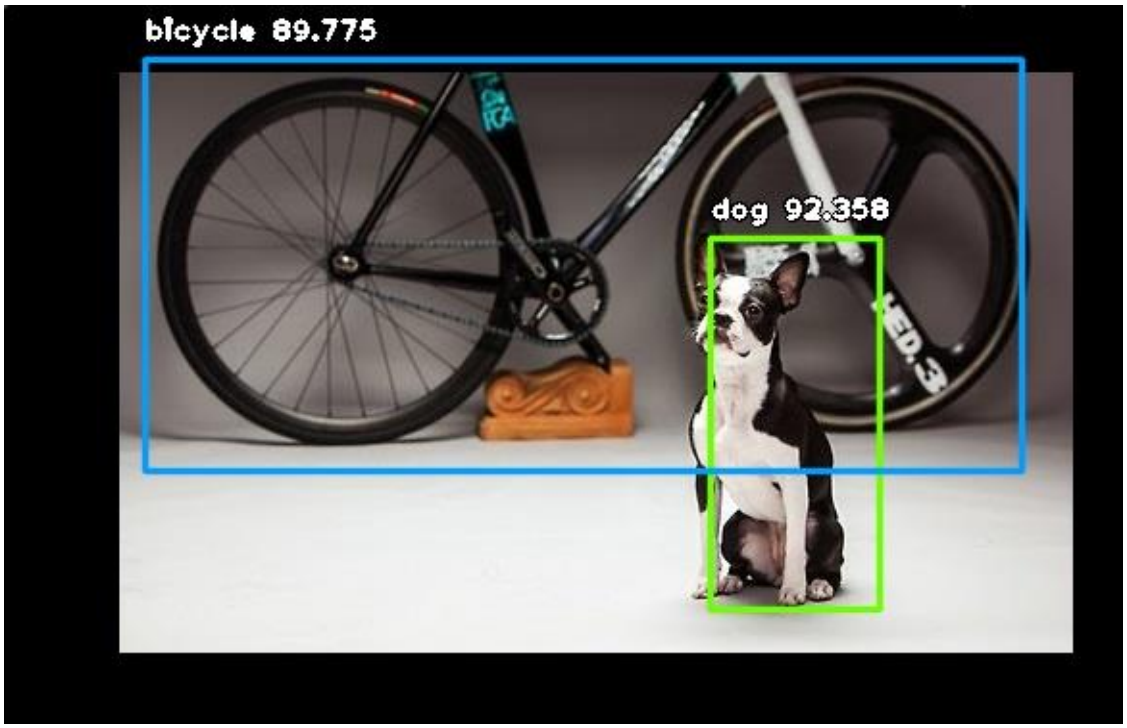


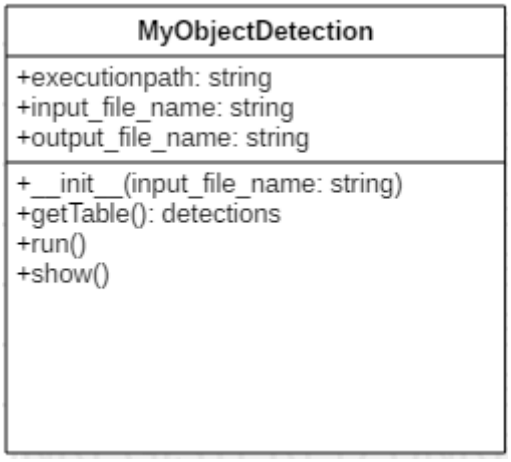
Figure 3.2 Dog and bicycle, after object detection

4. System Design

4.1 System Architecture

The program consists of two Python source files. The source code is a combination of procedure-oriented and object-oriented paradigms.

4.2 Class Diagram:



The program consists of a single class called **MyObjectDetection** defined in **MyObjectDetection.py**.

The following are the methods offered by this class:

- **Constructor**: The constructor takes in one argument: the name of the image file to perform object detection on.
- **getTable()**: This method returns a dictionary containing the names and confidence score of all the objects detected after the run() method is called. This dictionary is used to display names of objects with their corresponding confidence score in a tabular format in the **App.py** file which contains the starting code of the application.
- **run()**: This method performs the actual object detection algorithm from the ImageAI framework. Depending on the clock rate of the CPU it typically takes about 30 seconds for it to finish executing.
- **show()**: This method displays the image with objects detected along with the original image so that the user can compare them side by side.

5. Coding Techniques and Implementation Details

The project is implemented in python 3 with the help of the ImageAI framework for object detection and the AppJar library for the user interface.

PYTHON

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

IMAGEAI

A python library built to empower developers to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code.

An **AI Commons** project https://aicommons.science Developed and Maintained by Moses Olafenwa and John Olafenwa, brothers, creators of TorchFusion and Authors of Introduction to Deep Computer Vision

Built with simplicity in mind, **ImageAI** supports a list of state-of-the-art Machine Learning algorithms for image prediction, custom image prediction, object detection, video detection, video object tracking and image predictions trainings. **ImageAI** currently supports image prediction and training using 4 different Machine Learning algorithms trained on the ImageNet-1000 dataset. **ImageAI** also supports object detection, video detection and object tracking using RetinaNet, YOLOv3 and TinyYOLOv3 trained on COCO dataset.

OPENCV

OpenCV (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

While OpenCV is very powerful and offers a range of facilities for image processing, in this project we mostly use it for displaying, resizing and saving images.

APPJAR

AppJar is a cross-platform, open source Python library for developing GUIs (graphical user interfaces). It can run on Linux, OS X, and Windows. It was conceived, and continues to be developed with educational use as its focus, so is accompanied by comprehensive documentation, as well as easy-to-follow lessons.

The program can be launched by executing the provided **build.bat** file.

Immediately, the user sees the following screen:

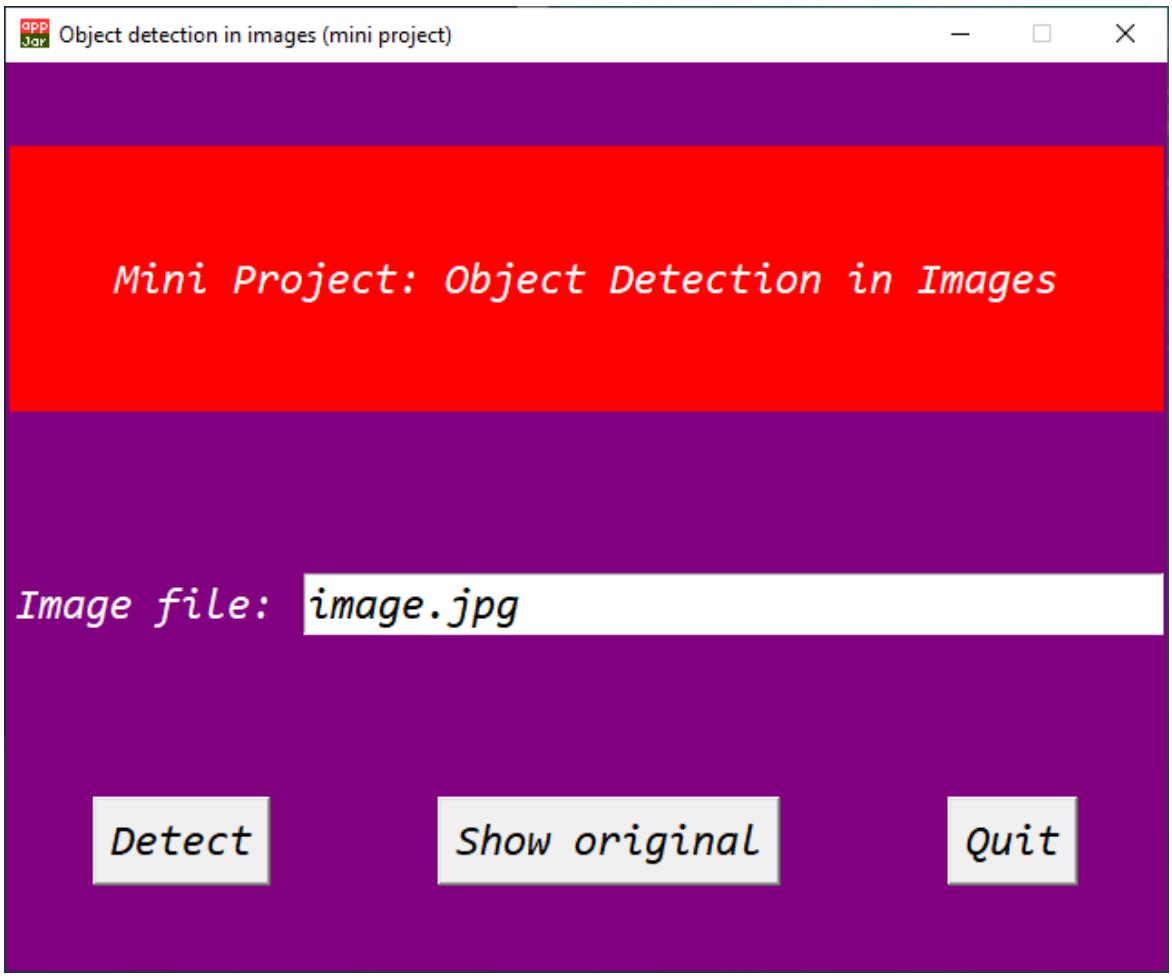


Figure 5.1 Main window

This is the main and only interface of the object detection program. The user can enter the name of the image file in the textbox.

The "**Show Original**" button simply displays the image file through OpenCV. This is so that the user won't have to browse the folder to search the image.

The "**Quit**" button quits the program.

The "**Detect**" button is where all the work of object detection takes place. When the user clicks on this button, the program reads the name of the image file and passes it to the constructor of the **MyObjectDetector** class as seen in the class diagram section. It then calls the **run()** method of the class. This method processes the image and detects objects. The resulting image is stored as a new file called "ORIGINALNAME_output.jpg" where "originalname" is the original

name of the image as passed in the image file name textbox. Once the **run()** method is finished executing, the program calls the **show()** method of the **MyObjectDetector()** class. This method uses OpenCV to show both the original as well as the output image side by side for comparison as shown below:

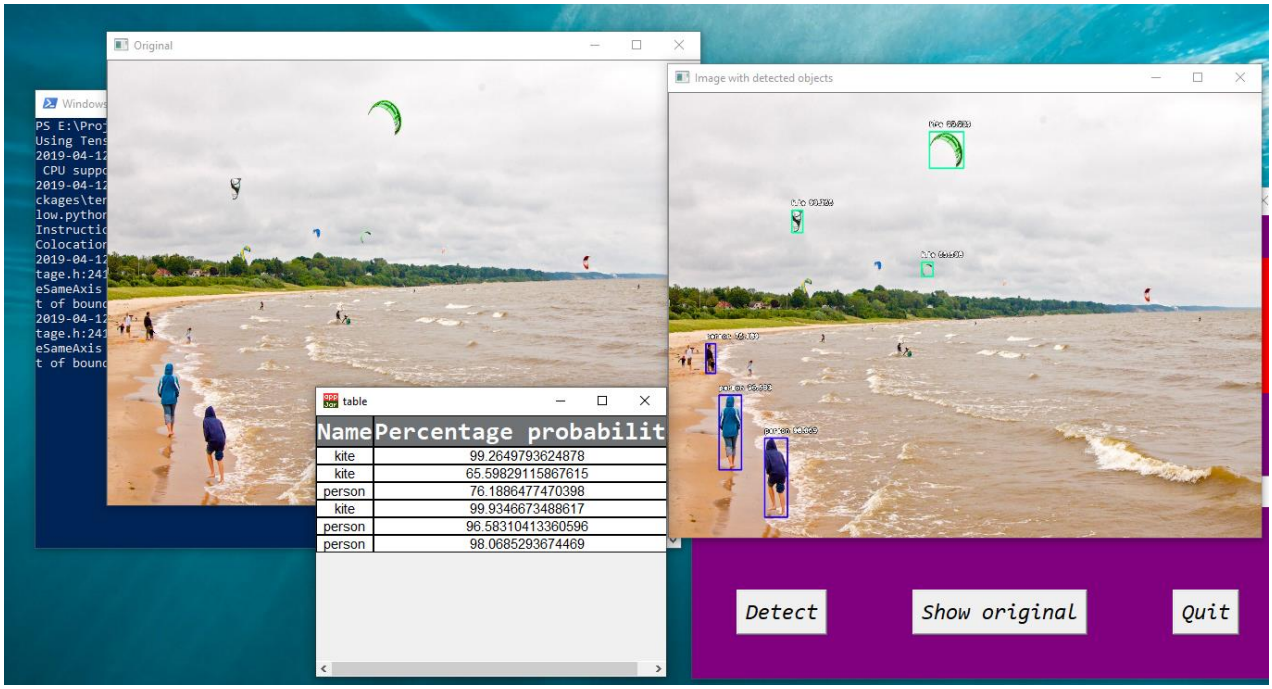


Figure 5.2. The application in working

Along with the original and the output images, the program also shows a table as seen in the screenshot above. This table lists the percentage probability or the confidence score of each detected object in the image. The confidence score is a percentage value that indicates the accuracy of a detection. The higher this value is, the more accurate the detection is. The algorithm generally detects lots of objects but only those with a percentage probability above or equal to 50% are accepted as correct detections.

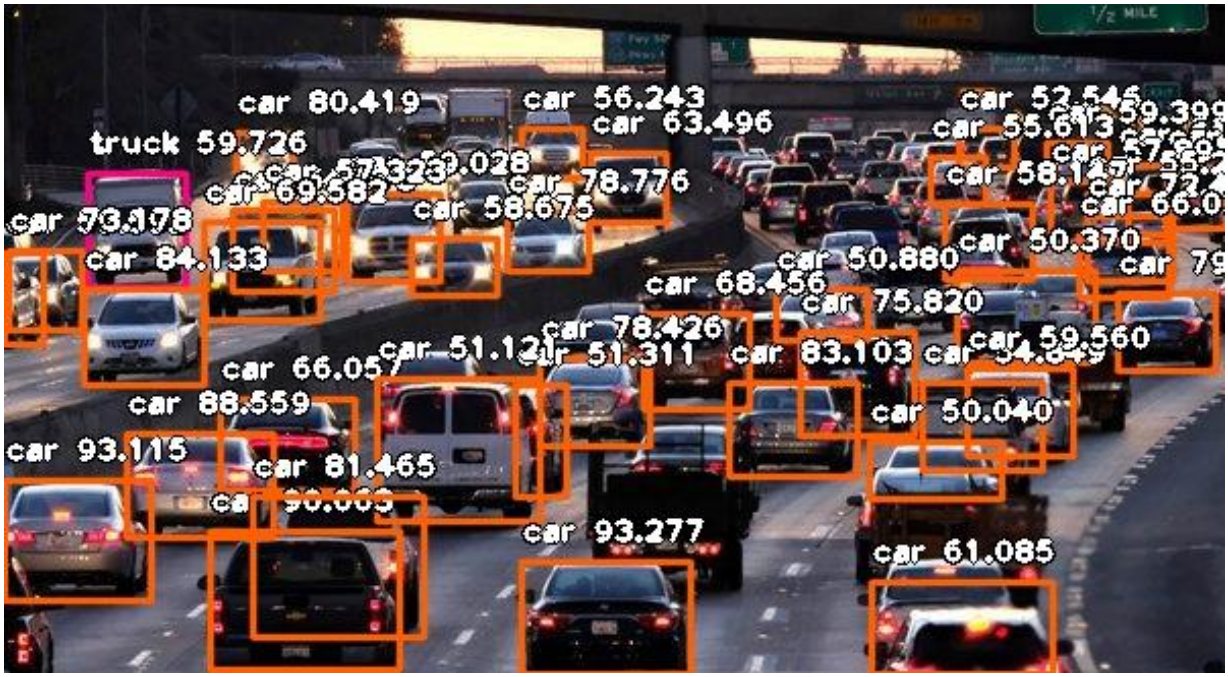
Name	Percentage Probability
kite	99.2649793624878
kite	65.59829115867615
person	76.1886477470398
kite	99.9346673488617
person	96.58310413360596
person	98.0685293674469

Table 5.1. Name-Percentage Probability table for image4.jpg

To demonstrate the working of the program, the following are a few of the images that we ran through it along with the outputs that we obtained:

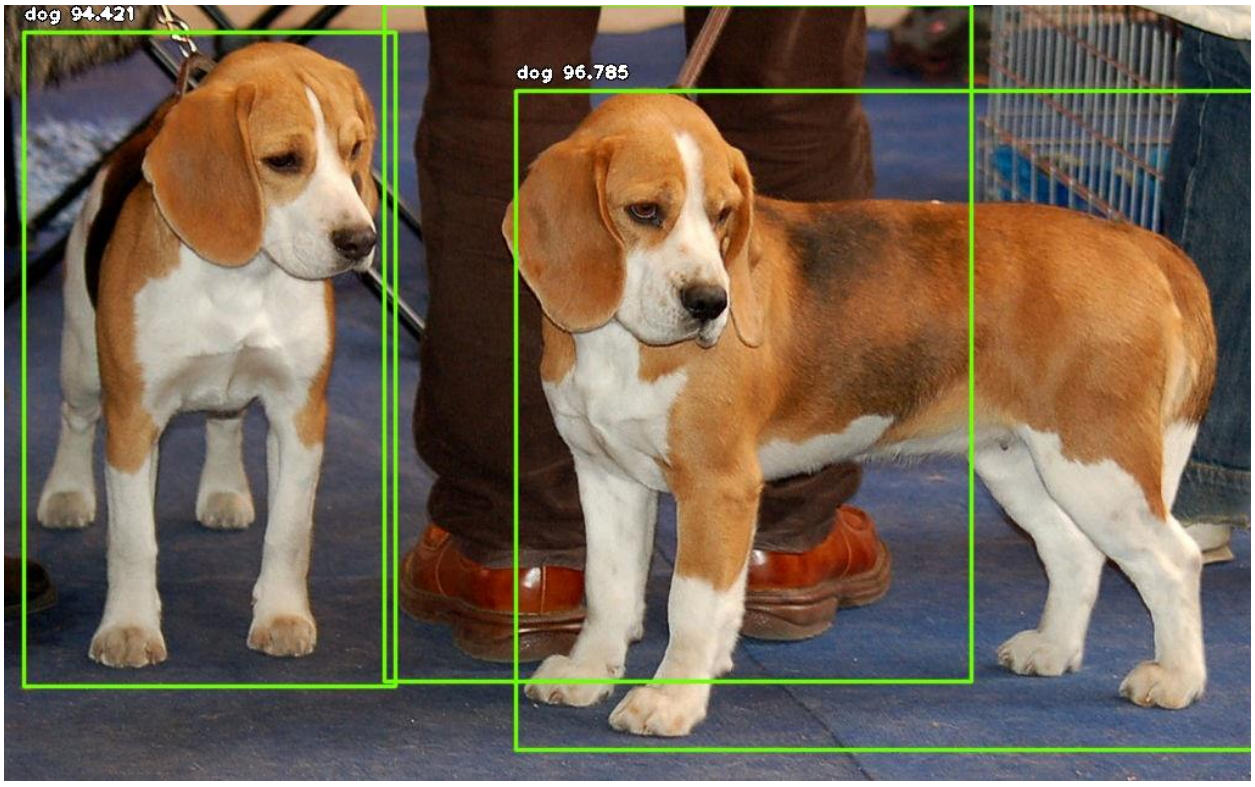
Before:

After:



Before:

After:



Before:

After:

6. Applications of Object Detection

A well-known application of object detection is face detection that is used in almost all the mobile cameras. A more generalized (multi-class) application can be used in autonomous driving where a variety of objects need to be detected. Also it has an important role to play in surveillance systems. These systems can be integrated with other tasks such as pose estimation where the first stage in the pipeline is to detect the object, and then the second stage will be to estimate pose in the detected region. It can be used for tracking objects and thus can be used in robotics and medical applications. Thus this problem serves a multitude of applications.

Facial Recognition:

A deep learning facial recognition system called the "**DeepFace**" has been developed by a group of researchers in the **Facebook,** which identifies human faces in a digital image very effectively. **Google**uses its own facial recognition system in Google Photos, which automatically segregates all the photos based on the person in the image. There are various components involved in Facial Recognition like the eyes, nose, mouth and the eyebrow.

People Counting:

Object detection can be also used for people counting, it is used for analyzing store performance or **crowd statistics** during festivals. These tend to be more difficult as people move out of the frame quickly.

It is a very important application, as during crowd gathering this feature can be used for multiple purposes.

Industrial Quality Check:

Object detection is also used in industrial processes to identify products. Finding a specific object through visual inspection is a basic task that is involved in multiple industrial processes like sorting, inventory management, machining, quality management, packaging etc.

Inventory management can be very tricky as items are hard to track in **real time.** Automatic object counting and localization allows improving inventory accuracy.

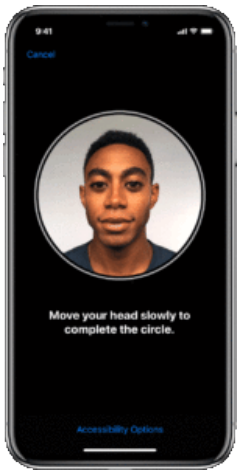
Self-Driving Cars:

Self-driving cars are the Future, there's no doubt in that. But the working behind it is very tricky as it combines a variety of techniques to perceive their surroundings, including radar, laser light, GPS, odometry, and computer vision.

Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and once the image sensor detects any sign of a living being in its

path, it automatically stops. This happens at a very fast rate and is a big step towards driverless cars.

Security:



Object Detection plays a very important role in Security. Be it face ID of Apple or the retina scan used in all the sci-fi movies.

It is also used by the government to access the security feed and match it with their existing database to find any criminals or to detect the robbers' vehicle.

The applications are limitless.

7. Conclusion

The program successfully detects various different kinds of objects in an image. We were able to successfully incorporate external libraries. The performance of the program was acceptable. Object detection is still very much a progressing field in computer vision. This project helped us gain an insight into this rapidly progressing field and pave way for more advanced projects in the future.

8. Future Work

There is still much that can be done in the project. In its present state, it only depicts a basic working of object detection. In the future we would like to make the following advancements to the project:

- Presently, the program takes about 20 seconds to perform on an Intel i7 quad core processor with 8 gigs of RAM. We would like to optimize the program to reduce this time.
- The project currently makes use of pre-trained models. If possible, we would like to train our own model using a custom dataset to detect only a few types of objects with extreme accuracy; for example detection of only electronic devices.
- We would like to develop an object detection system for videos. The present system works only for images.
- We would also like to develop a real-time object detection system that works smoothly even on embedded systems.

9. Bibliography/References

- Tariq Rashid, *Make Your Own Neural Network*, CreateSpace Independent Publishing Platform, USA ©2016
- www.krazytech.com
- en.wikipedia.org
- www.w3school.com
- www.appjar.info
- https://docs.opencv.org/master/d9/df8/tutorial_root.html