

Real-Time Object Detection Using Convolutional Neural Network

G. Kiran Kumar

*Department of Computer Science and
Engineering Anurag Group of Institutions,
Email - kirankumarcse@cvsr.ac.in*

Veerubhotla Mohan

*Department of Computer Science and
Engineering Anurag Group of Institutions,
Email - 16h61a05n0@cvsr.ac.in*

P. Preethi Sree

*Department of Computer Science and
Engineering Anurag Group of Institutions,
Email - 16h61a05k9@cvsr.ac.in*

R. Shanker

*Department of Computer Science and
Engineering Anurag Group of Institutions,
Email - 16h65a05p4@cvsr.ac.in*

E. Divya

*Department of Computer Science and
Engineering Anurag Group of Institutions,
Email - 16h65a05p4@cvsr.ac.in*

Abstract-Object detection and recognition systems have gained significant interest of researchers due to vast advancement in the field of computer vision technology. Although there are number of object recognition systems implemented in past researches, there still remains a constant demand for new, better and accurate recognition systems. Current detection systems make use of classifiers to perform detection. We are implementing a machine learning model which can detect objects using the concept of Convolutional neural networks (CNNs). This enables us to detect objects with very high accuracy.

Keywords- Object Detection, Computer Vision, Convolution Neural Network

I. INTRODUCTION

During the last years, there has been a rapid and successful expansion on computer vision research. Parts of this success have come from adopting and adapting machine learning methods, while others from the development of new representations and models for specific computer vision problems or from the development of efficient solutions. One area that has attained great progress is object detection. The present works gives a perspective on object detection research. Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems.

Given a set of object classes, object detection consists in determining the location and scale of all object instances, if any, that are present in an image. Thus, the objective of an object detector is to find all object instances of one or more given object classes regardless of scale, location, pose, view with respect to the camera, partial occlusions, and illumination conditions.

II. PROPOSED ALGORITHM

The proposed algorithm is divided into the following 3 modules:

1. Design of User Interface
2. Implement YOLOv3 neural network
3. Non-Max Suppression

Design of User Interface:

Python “tkinter” is used to create a user interface. This interface consists of 2 buttons: “upload” and “Detect objects and Display”. The “upload” button upon clicking will open a window where we can browse and upload the image as input. The “detect objects and display” button will run the neural network and objects will be detected. The final step is non-max suppression algorithm and finally the result will be displayed.

Build, Train and Test the CNN:

YOLOv3 architecture has been created and the corresponding weights are saved. The “coco” images dataset has been used.

Object Detection:

OpenCV module has been used in order to implement the neural network. The non-max suppression algorithm has also been implemented using OpenCV. The initial step of the process is dividing the input image into 13 x 13 grid. In each of the cells, objects will be detected. There is a possibility that one object can span across multiple cells and this may result in same object getting detected multiple times. This produces multiple boxes around the same object. Non- max suppression comes into picture in this step and removes all the additional boxes but leaves the one which perfectly fits the object.

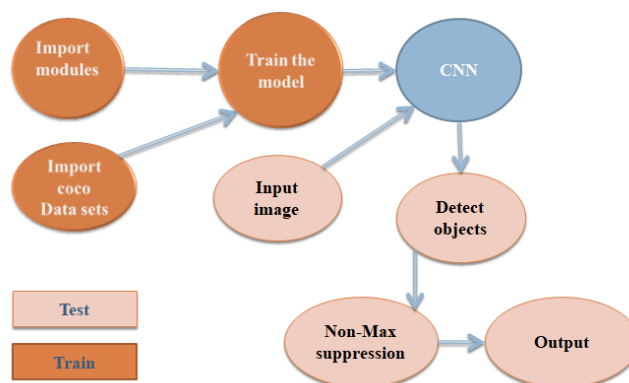


Fig.2.1. Working of proposed algorithm

Convolution Neural Network:

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see $h \times w \times d$ (h = Height, w = Width, d = Dimension). Eg., An image of $6 \times 6 \times 3$ array of matrix of RGB (3 refers to RGB values) and an image of $4 \times 4 \times 1$ array of matrix of grayscale image.

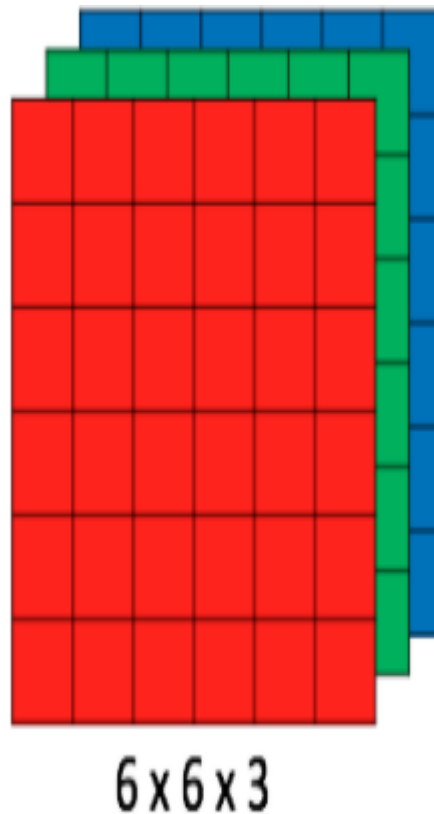


Fig.2.2. An image of $6 \times 6 \times 3$ array of matrix of RGB

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

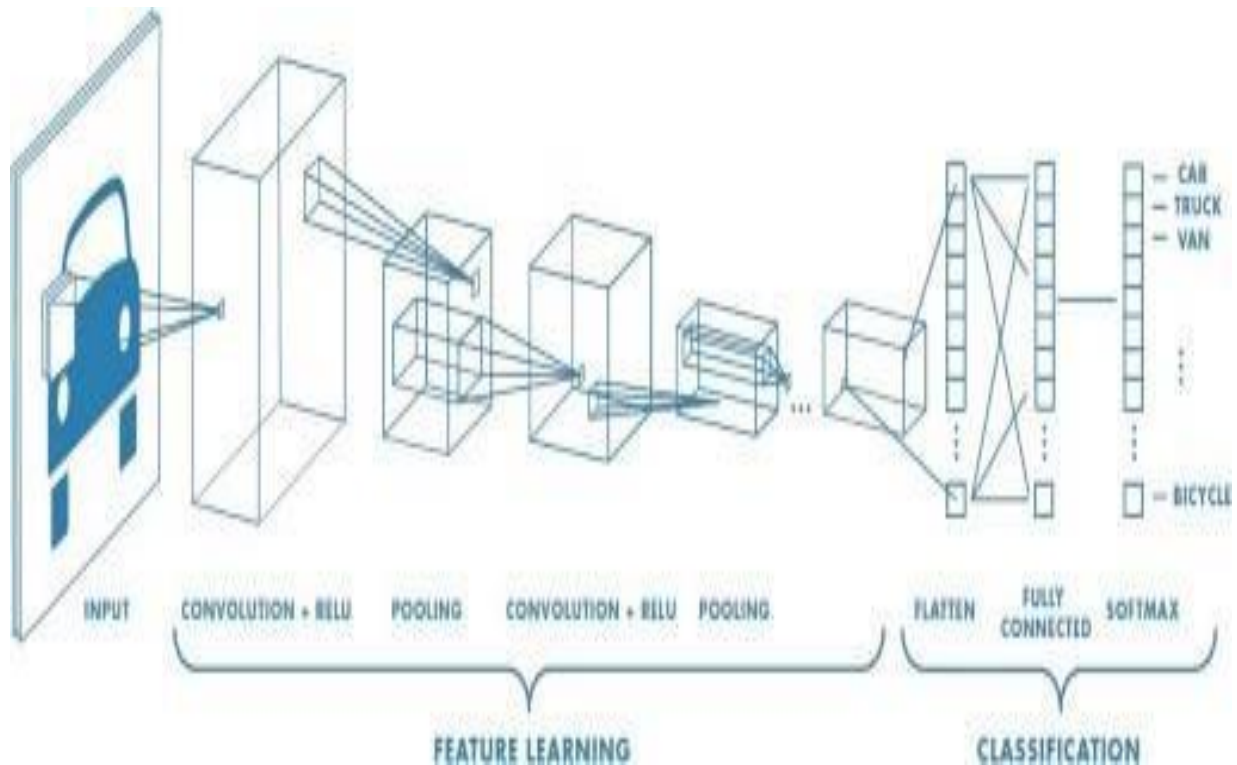


Fig.2.3. Processes involved in Convolution Neural Network

III. EXPERIMENT AND RESULT

The aim of the program is to detect all the objects present in a given image.

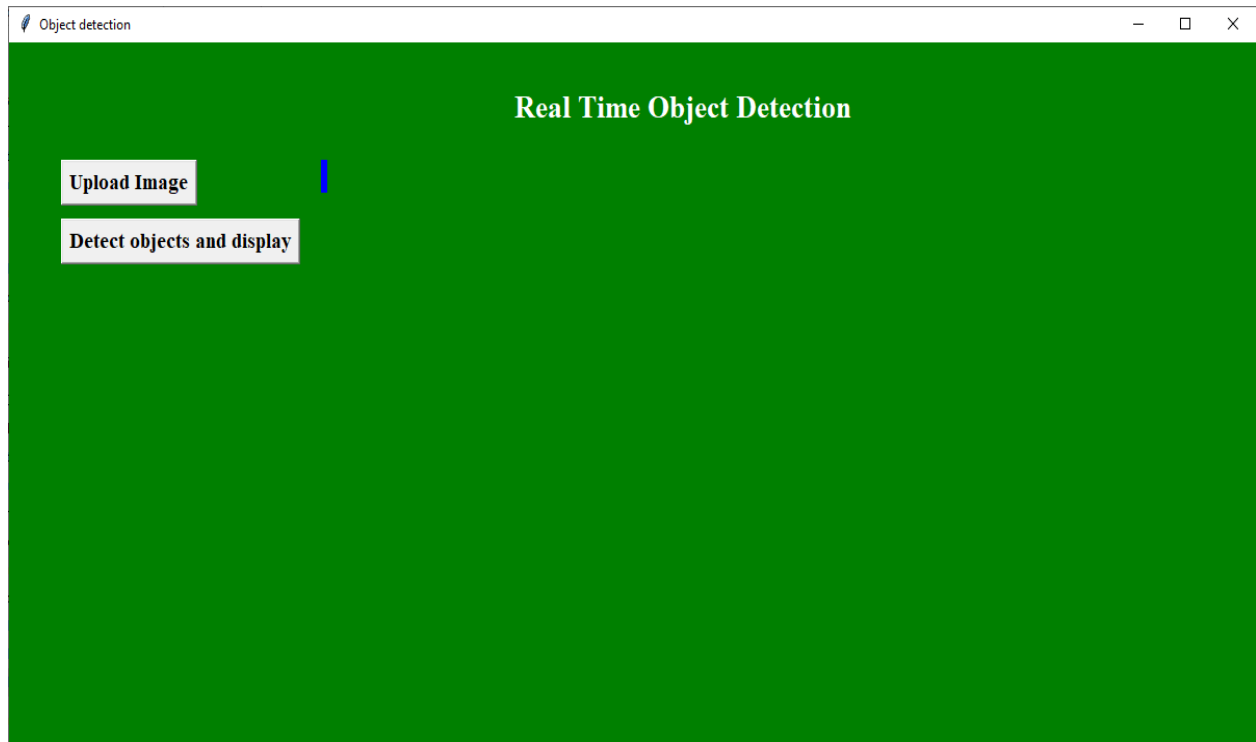
To achieve this we need to install python 3.5 version (or higher) in 64 bit laptop. While Python installation we need to select checkbox saying 'Add path to system variable'. This option will appear at the first during Python installation. Once Python is successfully installed, we need to execute the below commands in the command prompt. Make sure that the system is connected to internet. "pip" command is used to install some packages.

- pip install numpy
- pip install opencv-python

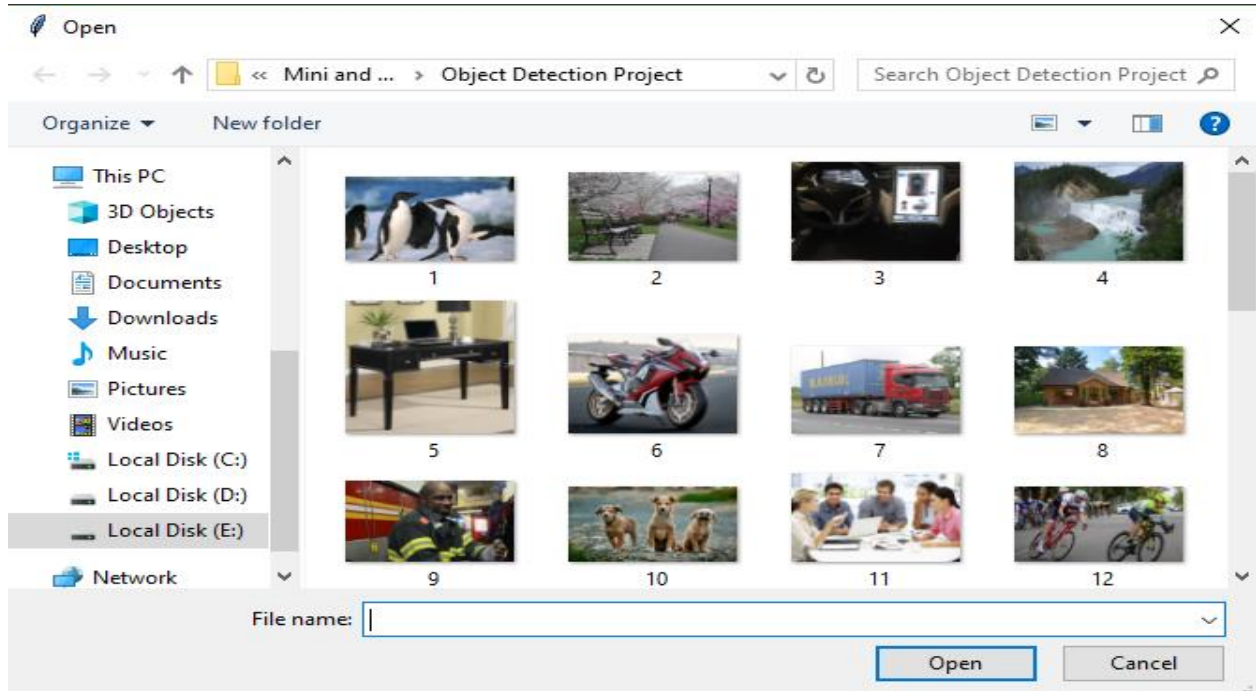
After executing above commands, all the required packages for the further steps will be installed.

Now follow the below steps:

Double click on 'run.bat' file. We will get below screen



Click on 'Upload Image' button to upload image from the computer



Select the image by browsing in the above window.



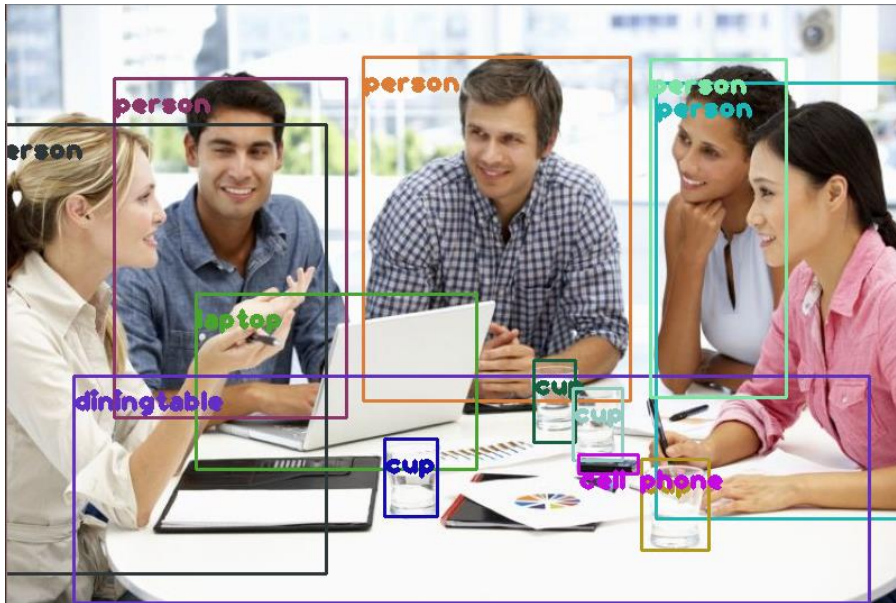
Once the image is successfully uploaded, the location of the image file will be displayed as shown above. Now click on 'Detect objects and display' button.

Test case 1:

Input Image:



Output image:



Test case 2:

Input image:



Output image:



IV.CONCLUSION

Object detection is breaking into a wide range of industries, with use cases ranging from personal security to productivity in the workplace. Object detection and recognition is applied in many areas of computer vision, including image retrieval, security, surveillance, automated vehicle systems and machine inspection. Significant challenges stay on the field of object recognition. The possibilities are endless when it comes to future use cases for object detection. Some of the applications are **optical character recognition, self driving cars, tracking objects, face detection and face recognition, activity recognition**. As we can foresee many applications of object detection, designing of systems to detect objects of an image has been given a lot of importance. Our project can detect 80 day-to-day objects in real-time and also with high accuracy.

REFERENCES

- [1] [1] You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon , Santosh Divvala, Ross Girshick , Ali Farhadi University of Washington, Allen Institute for AI , Facebook AI Research Published at **arXiv.org** on 8 April 2018 Dhananjay Garud and Dr. S.S. Agrwal , “Face Liveness Detection ”, International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) International Institute of Information Technology , Pune, 2016.
- [2] G. Pan, L. Sun, Z. Wu, and S. Lao, “Eyeblink -based anti-spoofing in face recognition from a generic webcam,” in Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV), Oct. 2007, pp. 1–8.



- [3] J. Maatta, A.Hadid, M.Pietikainen, "Face Spoofing Detection From Single images Using MicroTexture Analysis", Proc. Intn Joint Conference on Biometrics, 2011, Washington, D.C., USA.
- [4] C.Anuradha, R.Kavitha, A.V.Allin Geo, S.R.Srividhya, "Liveness Detection with Opencv", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8, Issue-9S3, July 2019.