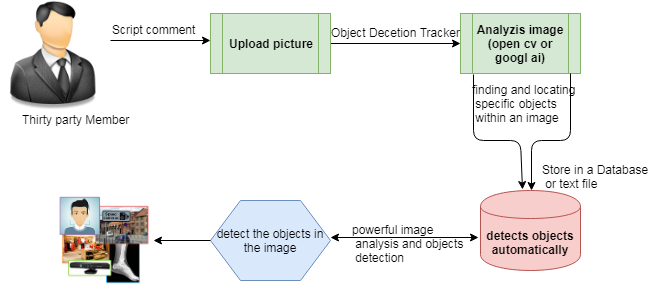
**OPEN IMAGES - OBJECT DETECTION TRACK**

**ABSTRACT:**

Object detection is a central task in computer vision, with applications ranging across search, robotics, self-driving cars, and many others. As deep network solutions become deeper and more complex, they are often limited by the amount of training data available. With this in mind, to spur advances in analyzing and understanding images, Open CV or Google AI has publicly released the Open Images dataset. Open Images follows the tradition of PASCAL VOC, Image Net and COCO, now at an unprecedented scale. In this project we to implement the best performing algorithm for automatically detecting objects. For example, we can take from the below image we can recognize notebook, glasses, laptop, coffee cup, and iPhone. But, computer cannot detect the objects in the image.

**ARCHITECTURE:**

****

**EXISTING SYSTEM**

Challenging to build an algorithm that detects objects automatically using an absolutely massive training dataset. Creating accurate Machine Learning Models which are capable of identifying and localizing multiple objects in a single image remained a core challenge in computer vision.

For detecting generic objects (like laptop, notebook, pen, glasses, coffee, car, person, table, tree, “small raw tomato" or a "large ripe tomato") we want an algorithm to detect very specific objects. We need to build the best performing algorithm for automatically detecting objects.

An image classification or image recognition model simply detects the probability of an object in an image. In contrast to this, object localization refers to identifying the location of an object in the image. An object localization algorithm will output the coordinates of the location of an object with respect to the image. In computer vision, the most popular way to localize an object in an image is to represent its location with the help of bounding boxes

**DISADVANTAGE:**

Need a large dataset

Because you need a large dataset, training time is usually significant

Takes lots of time to train and stuff

With the availability of large amounts of data, faster GPUs, and better algorithms

**PROPOSED SYSTEM:**

Object Detection track, the Challenge also includes a Visual Relationship Detection track to detect pairs of objects in particular relations, e.g. "person riding a bike” as shown below image. Identifying different objects (man and objects) is an important problem on its own, but identifying the relationship between them (holding) is critical for many real-world use cases. In this Visual Relationship Detection Track Challenge, we have to build an algorithm that detects pairs of objects in particular relations: things like “person riding a bike”, "water bottle on table," or "dog inside car." The Challenge dataset includes both objects bounding boxes and visual relationship annotations.

In this case of the Challenge, we have to build to build the best performing algorithm for automatically detecting relationships triplets.

**ADVANTAGES:**

Object detection is breaking into a wide range of industries, with use cases ranging from personal security to productivity in the workplace.

It can also be used within a visual search engine to help consumers find a specific item

Powerful image analysis and objects detection \

**MODULES:**

1. **UPLOAD IMAGES**

Uploading the image is done by user. Authorized person is uploading the new arrivals to system that are listed to users. Images can be uploaded with its attributes.

1. **ANALYSIS IMAGE**

Object detection in computer vision. Object detection is the process of finding instances of real-world objects such as faces, bicycles, and buildings in images or videos. Objectdetection algorithms typically use extracted features and learning algorithms to recognize instances of an object category.

1. **OBJECT DETECTION IMAGES**

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.

1. **OPENCV OR GOOGLE AI**

OpenCV is a library of programming functions mainly aimed at real-time computer vision. OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing,Google AI, formerly known as Google Research, is Google's artificial intelligence (AI) research and development branch for its AI applications.

**REQUIREMENT ANALYSIS**

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

**REQUIREMENT SPECIFICATION**

**Functional Requirements**

* Graphical User interface with the User.

**Software Requirements**

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. MySql
4. MySqlclient

**Operating Systems supported**

1. Windows 7
2. Windows XP
3. Windows 8

**Technologies and Languages used to Develop**

1. Python

**Debugger and Emulator**

* Any Browser (Particularly Chrome)

**Hardware Requirements**

For developing the application the following are the Hardware Requirements:

* Processor: Pentium IV or higher
* RAM: 256 MB
* Space on Hard Disk: minimum 512MB

**CONCLUSION:**

The main objective is to implement based on algorithms built. Intelligent visual surveillance systems to assist the human operators to detect the objects in image and responding to them rapidly. In this study, it was shown that the objects in the images/videos taken by camera effectively detected and successfully tracked using numerous techniques and their performance has been compared. The performance of these algorithms is excellent when applied to static camera views and certain video stabilization algorithms.