**Mini Project Report on**



**Real Time People Counting in OpenCV**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Anant Bhandari**  **2016628**

***Under the Mentorship of***

**Mr.Ankit Tomar**

**Faculty, CSE Dept.**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Real Time People Counting in OpenCV”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mr. Ankit Tomar, Faculty**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

****

Anant Bhandari 2016628 **signature**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1** |
| Chapter 2 | Literature Survey | **4** |
| Chapter 3 | Methodology | **6** |
| Chapter 4 | Result and Discussion | **9** |
| Chapter 5 | Conclusion and Future Work | **10** |
|  | References | **11** |

**Chapter 1**

**Introduction**

* 1. **Problem Statement**

**Real Time People Counting in OpenCV :** Elaborating on the problem statement; we are required to develop a model in python that can identify the number of people present in the camera frame and display the counter with utmost precision and accuracy using opencv.

OpenCV or ‘Open Source Computer Vision Library’ is a library in python that is 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 Apache 2 License**. Starting with 2011, OpenCV features **GPU acceleration** for real-time operations.

* 1. **Introduction of project**

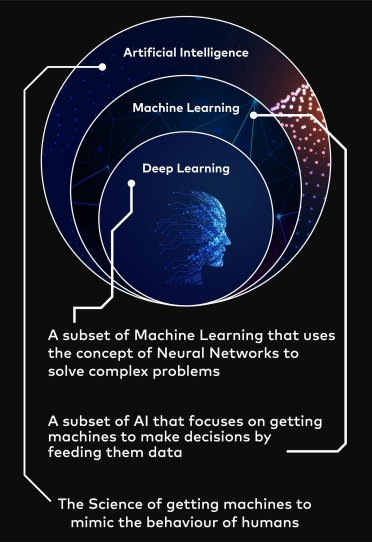
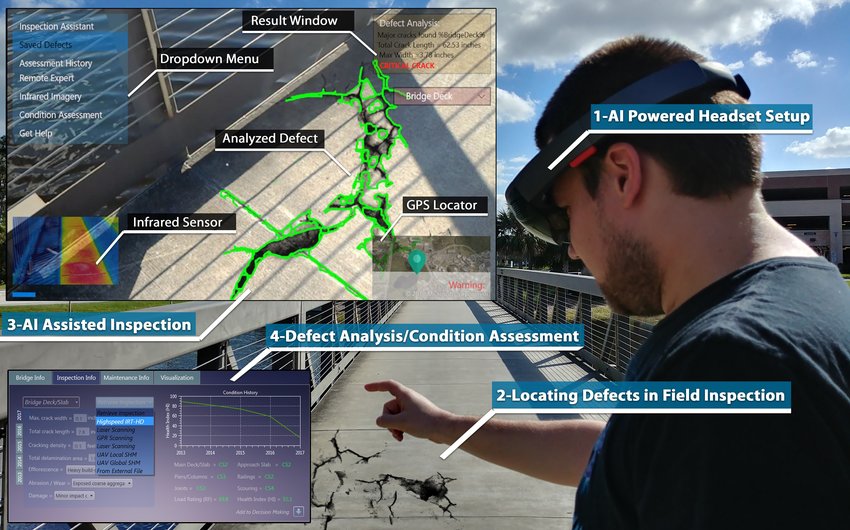
With this project, we will understand what is the best way to count people from a CCTV camera which might be useful for monitoring the customer movement in a shop or a Mall.

Before going on and starting with the project and coding the solution, let us discuss and know in brief about the core concepts/technologies that are involved in this project. They are:-

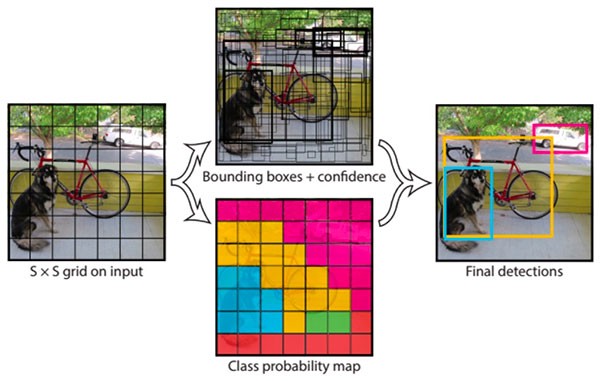
* **AI(Artificial Intelligence)**
* **YOLO Algorithm**

**Artificial Intelligence:** Artificial intelligence (AI) is intelligence—perceiving, synthesizing, and inferring information—demonstrated by machines, as opposed to intelligence displayed by animals and humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs. The Oxford English Dictionary of Oxford University Press defines artificial intelligence as:

“The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.”



**YOLO Algorithm for Object Detection:** YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.



YOLO is an abbreviation for the term ‘**You Only Look Once**’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

**Object Detection**: Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals.

This phenomenon seeks to answer two basic questions:

* What is the object? This question seeks to identify the object in a specific image.
* Where is it? This question seeks to establish the exact location of the object within the image.

Object detection consists of various approaches such as **fast R-CNN, Retina-Net**, and **Single-Shot MultiBox Detector (SSD)**. Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. YOLO algorithm has gained popularity because of its superior performance over the aforementioned object detection techniques.

And in this project, we will be using SSD as the most reliable and precise approach for detecting an object in the camera frame.

**Chapter 2**

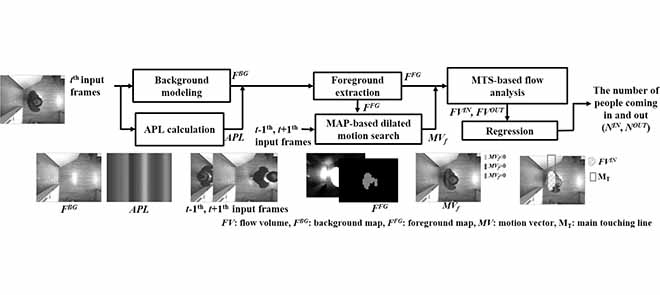
**Literature Survey**

While researching on the project, I got to know a variety of approaches to this problem.

Many research papers claimed to have successfully achieved the solution to the problem statement as stated above but they lacked in accuracy.

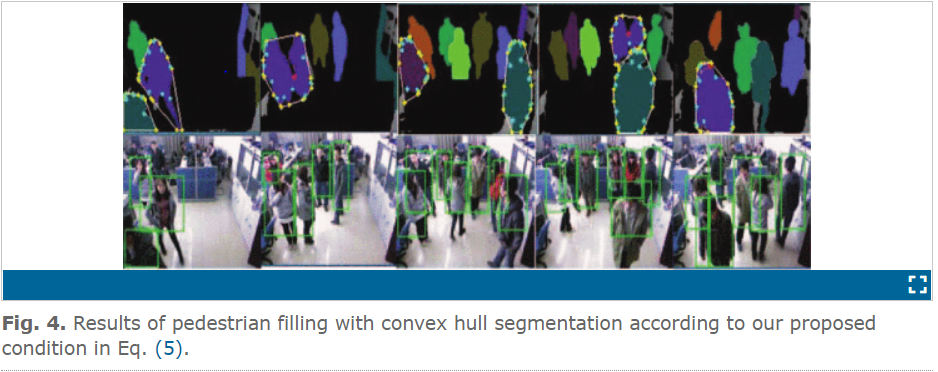
Some of the newest approaches that were found in the scholarly articles have been discussed below:-

**APPROACH-1**

We propose a real-time people-counting system that can be applied in a retail store to estimate the number of people entering and exiting. The proposed method consists of three main procedures: foreground extraction based on **the average picture level (APL)**, a dilated motion search based on the **maximum a posteriori probability (MAP)**, and flow analysis based on **multiple touching sections (MTS)**. We first produce a background model to extract the foreground by using line-wise APL. A dilated motion search with the MAP-based approach is then used to estimate the motions on the line of interest. Next, the flow generated by the foreground on the MTS is analyzed. Finally, the results of the motion estimation and flow analysis are incorporated to produce the number of people entering and exiting the store. We used a low-cost microcomputer to implement the system, which is capable of wireless transmission and is easy to install in a retail environment. Experimental results show that the proposed method provided the best F1 score and accuracy values for the people count results with much lower computational complexity than benchmark methods. In addition, it successfully estimated the number of people entering and exiting the store in real time.

The overall architecture of the proposed method.

**APPROACH-2**

Real-time accurate crowd counting is one of important tasks in intelligent visual surveillance systems. Most previous works can only count passing people robustly without heavy occlusions which are very common in the practical surveillance scenes. To solve this difficult problem, we propose a new method for crowd counting for RGB-D (RGB plus depth) data using a commodity depth camera. In our method, we first detect each head-shoulder of the passing or still person in the surveillance region with fast template matching based on depth information including pedestrian filling with convex hull segmentation. Then, we track and count each detected head-shoulder based on RGB information bi-directionally. By using this approach, we have built a practical system for robust and fast crowd counting. Extensive experimental results show that our method achieves significant improvement comparing to states-of-the-art approach, and the built system is not only robust to heavy occlusions, but also can be deployed in the real time crowd counting application scenes.

**Chapter 3**

**Methodology**

When using YOLO, we can Identify and count people by two ways:-

1. Identify the full-length of the people
2. Identify just their heads

After the research and literature survey, I have concluded that the best way to go forth is the 2nd one, i.e., to identify just the heads of the people.

**Identifying the heads of people**

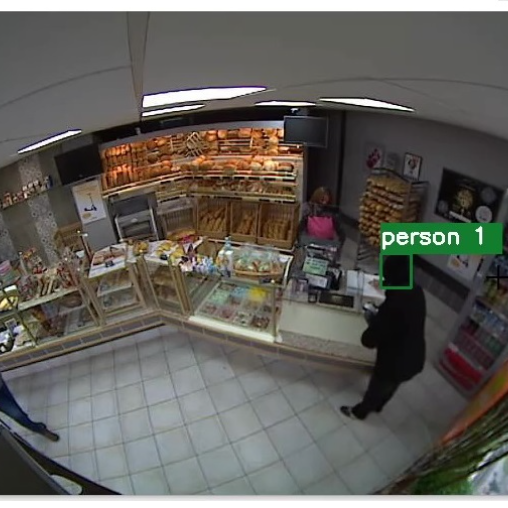
The explanation is very simple with this type of CCTV camera and due to its position, it is easier to identify the head than the whole body. Of course, there will also be fewer problems of overlapping people.

In this case, we will need a custom model of YOLO.

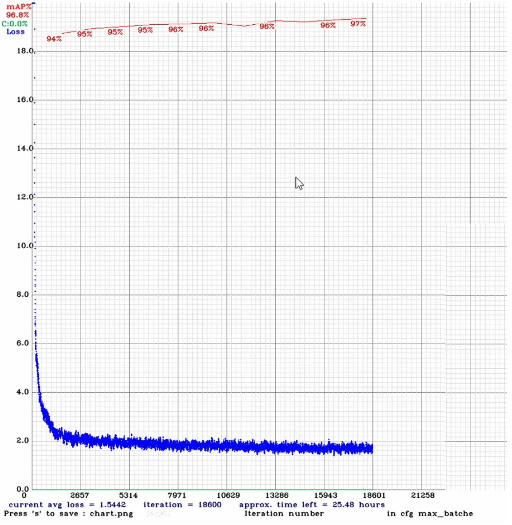
I had to train the computer by extracting the images from the sample video.

As you know, to train a model you first need a dataset of images which will then be labeled and passed to the algorithm. Where to find these images? There are several solutions and let’s see which is the best:

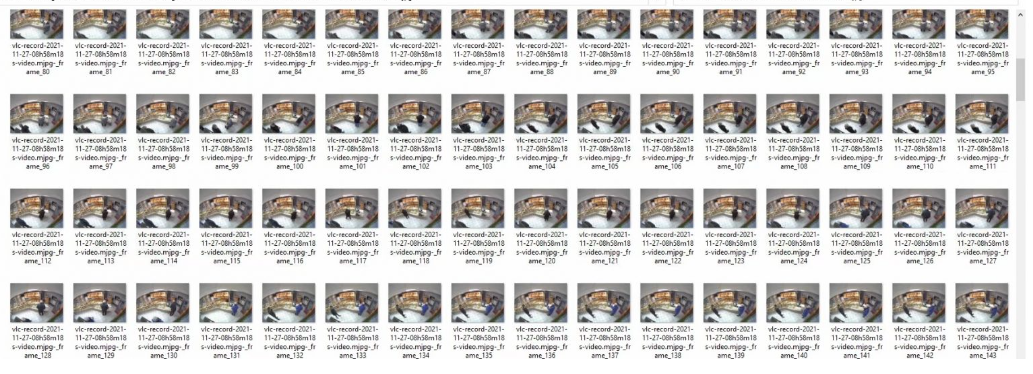
* **Google image Dataset [bad result]**
* **Images from video footage [good result]**

**Images from video footage – [good result]**

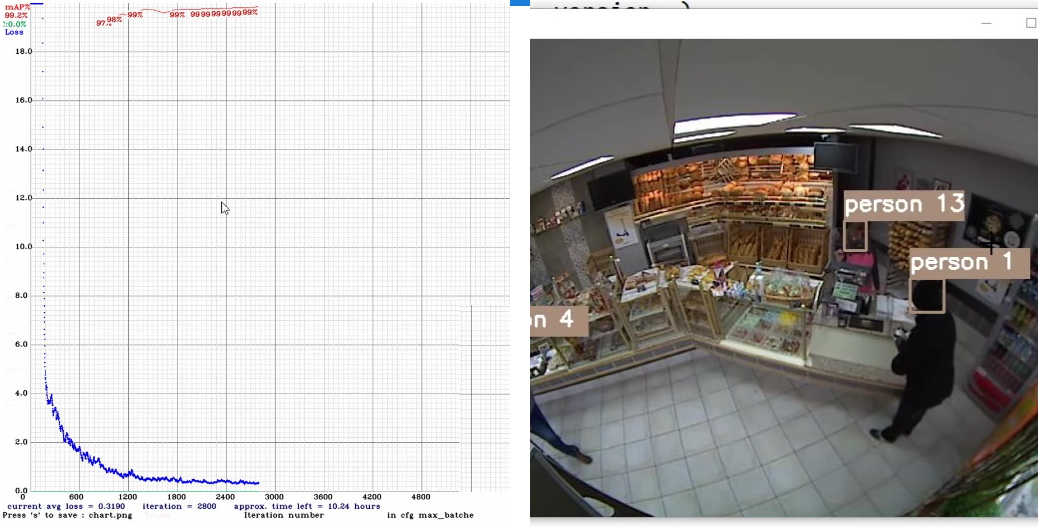
To get a good result in the Count people from a CCTV camera we have to try and extract the images from the sample footage itself, because using generic images didn’t work. If we compare the images downloaded from the Google Open image dataset we can see the difference with the heads present in the CCTV camera images.



We repeat the training process but first, it is necessary to extract thousands of images from the video footage.



Also in this case, by making the annotations and subsequently the training, we see the graph and the result. At first, it already seems an excellent result.



**Count people from CCTV camera from a Region of Interest (ROI)**

When the person enters the ROI it updates the count and it does not matter if it moves continuously because the count is associated with the ID of each individual user. For this procedure, I recommend defining an area of interest as small as possible to increase accuracy.

**Chapter 4**

**Result and Discussion**

**Accuracy in Count people from CCTV**

This is already an excellent result but there is certainly still a lot of room for improvement. For example, if a person moves to an area not covered by the camera, they may lose track.

This is a custom model; tweaked up to best fit the requirements of the sample video.

But, a more generalized model with a large data set can be made to fit any condition whatsoever.



**Chapter 5**

**Conclusion and Future Work**

**Conclusion**

We have achieved an excellent result for this project but I always remember that an analysis of the cameras, position, additional tools such as Deep cameras or Deep Learning models created specifically for that type of scenario is necessary.

It was due to the system limitations and deadline limitations that this model could not be generalized. Rest assured, with the proper training of the fully equipped machine, this model could be generalized to cover every aspect and to fit in every situation for counting the number of people from a live feed.

This project, if implemented with more than 95% accuracy, can prove to be very useful in Malls, Banks, and all other public places where keeping a track of the people visiting that place is a necessity.

**References**

[1] Huiyuan Fu; Huadong Ma; Hongtian Xiao, “Real-time accurate crowd counting based on RGB-D information” *2012 19th IEEE International Conference on Image Processing.*

[2] Sung In Cho; Suk-Ju Kang, “Real-Time People Counting System for Customer Movement Analysis”, IEEE Access ( Volume: 6)

[3] Github ID- saimj7, “People-Counting-in-Real-Time”

[4]Grace Karimi,” Introduction to YOLO Algorithm for Object Detection”.