Real-Time Surveillance With Optimized Image Processing Supervisory Technique

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

B. Tech(Hons)

in

Computer Science And Engineering

by

SANDEEP KUMAR YEDLA
(Roll No. 2016BCS0031)

Indian Institute of
Information Technology
Kottayam

to

DEPARTMENT OF CSE
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
KOTTAYAM - 686635, INDIA
April 2019

DECLARATION

I, **Sandeep Kumar Yedla** (**Roll No: 2016BCS0031**), hereby declare that, this report entitled **"Real-Time Surveillance With Optimized Image Processing Supervisory Technique"** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology(Hon)** in **Computer Science Engineering** is an original work carried out by me under the supervision of **Dr. Manikandan V. M** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Sandeep Kumar Yedla**

April 2019

CERTIFICATE

This is to certify that the work contained in this project report entitled **'Real-Time Surveillance With Optimized Image Processing Supervisory Technique'** submitted by **Sandeep Kumar Yedla** (**Roll No: 2016BCS0031**) to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology(Hon)** in **Computer Science Engineering** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635
April 2019

(Dr. Manikandan V. M)
Project Supervisor

ABSTRACT

Recent research in computer application has increasingly focused on building systems for supervisory activities, to take respective action to solve by creating sensible model of humans for supervisory purpose rather than a human appointed for continuous monitoring, The implemented system is developed by capturing the frame at fixed interval gap and detecting the difference among both the frames, if the difference occurs in the frame then the system is sent for next levels of image processing, object detection and respective object counting technique in a frame which ,might also consists of the multiple objects.

Contents

List of Figures

Chapter 1

Introduction

Human single glance at an image can tell what the object is all about.So, similarly it is difficult for a human to have a glance at n number of different objects / or monitoring simple objects and counting procedures. So using the widely using the frame difference and object detection techniques along with proposed architectural design, a system is designed which acts as a worker at a stores who monitors the items placed in the racks and warns a supervisory department to refill when all the racks are empty / less in number.

This motivation behind the project is to help the shopping complexes or supermarkets by scalable video surveillance and ping them if required items in the rack are in the less quantity.

Basically, this is done in four major steps which are 1) Frame difference, 2)Required Object detection, 3)Detected object count and 4) Ping warning if the count is less.

Chapter 2

Literature Survey

Literature Survey that done for the implementation of this system is upon the image capturing and storing in the local storage(continuously replacing to save the storage place), two images conversion to matrices and gray scaling, comparing for frame difference,required object detection using the YOLO and its trained models and basic object detection in making boxes using () -ordinates formation of boundary.

- ‘You Only Look Once:Unified, Real-Time Object Detection’, 2017. [5]
    - YOLO : you only look once is one of the good advancement for fast processing speed compared to other detection method,a fast yolo can process 155 frames approx per second, forming bounding boxes based on confidence and support calculation.YOLO makes

more localization errors but is less likely to predict false positives on background. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

  - Conclusion:

    You only look once is a good method for detection as the supervisory system needs speed and better accuracy.

- ‘Motion Detection Based on Frame Difference Method’
  November 2014. [8]

  - Recent research in computer vision has increasingly focused on building systems for observing humans and understanding their look, activities, and behavior providing advanced interfaces for interacting with humans, and creating sensible models of humans for various purposes. The goal of motion detection is to recognize motion of objects found in the two given images. Moreover, finding objects motion can contribute to objects recognition.

  - Conclusion:

    The obvious aim of the work is studying the principle of frame difference method and comparing different images and check for any motion in those images and to resolve the various problems.

- ‘Weakly Supervised Object Localization on grocery shelves using simple FCN and Synthetic Dataset’
  March 2018. [10]

  - It is a weakly supervised method using two algorithms to predict object bounding boxes given only an image classification dataset. First algorithm is a simple Fully Convolutional Network (FCN) trained to classify object instances. Here they enhance the FCN output mask into final output bounding boxes by a Convolutional Encoder-Decoder (ConvAE) viz. the second algorithm. ConvAE is trained to localize objects on an artificially generated dataset of output segmentation masks.
  - Conclusion:

    Pinhole Projection Formula can be used for distance estimation using a single camera input.

- 'EdgeFlow: a technique for boundary detection and image segmentation'

  May 2011. [4]

  - The study is regarding the smart surveillance's using .
  - Conclusion:

    Pinhole Projection Formula can be used for distance estimation using a single camera input.

- 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks'

  May 2015. [6]

  - State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals.

  - Conclusion:
    RPN and Fast R-CNN can be trained to share convolutional features

- 'R-FCN: Object Detection via Region-based Fully Convolutional Networks'
  March 2016 [1]

  - This is region-based, fully convolutional networks for accurate and efficient object detection. In contrast to previous region-based detectors such as Fast/Faster R-CNN that apply a costly per-region subnetwork hundreds of times, our region-based detector is fully convolutional with almost all computation shared on the entire image.
  - Conclusion:
    this is almost 2.5-20 times faster than the Faster R-CNN counterpart.

- 'Fast R-CNN'
  April 2015. [2]

  - This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy.

  - Conclusion:
    Compared to SPPnet, Fast R-CNN trains VGG16 3x faster, tests 10x faster, and is more accurate.

- 'An Overview of the Tesseract OCR Engine'
  May 2011. [9]

  - The Tesseract OCR engine, as was the HP Research Prototype in the UNLV Fourth Annual Test of OCR Accuracy.
  - Conclusion:
    Tesseract is now behind the leading commercial engines in terms of its accuracy. Its key strength is probably its unusual choice of features.

- 'An Automated Computer Vision System for Extraction of Retail Food Product Metadata'
  November 2018. [3]

  - Our study proposes an automation method to improve the extraction of unstructured product metadata from food product label images using computer vision (CV), machine learning (ML), optical character recognition (OCR), and natural language processing (NLP).Proposed an automatic image quality classification system to identify images that give a high degree of metadata extraction accuracy

  - Conclusion:

    Results show 95 % accuracy for attribute extraction from high-quality product images with machine-printed characters having contrasting backgrounds.

- ‘A Case Study on smart Surveillance Application System using WSN and IP webcam’
  May 2011. [7]

  - The study is regarding the smart surveillance’s using .
  - Conclusion:

    Pinhole Projection Formula can be used for distance estimation using a single camera input.

Chapter 3

Proposed Block Diagram

The block diagram gives clear idea about the information about the system, and eactly how the system works..

3.2 Methodology

1. Using IP Webcam App, creates server for wireless access within a range and allows for video surveillance.

    - Continuous surveillance is done using the camera .

    - Two frames from the Video is captured at fixed interval of time.

    - Captured two frames are replaced continuously for consequent comparison to satisfy the storage constraint.

- Captured frame is sent for comparison.

2. If the shape are equal and the frames are not equal then sent for processing.

    - Particular time gap is given and the again the frame is captured to check the count.

    - YOLO : you only look once object detection is used.

    - YOLO-COCO model is used for next level of processing.

3. Counting the number of bottles(items).

    - sinchms : if the count is less than the given limit then a message is triggered to the supervisor mobile number.

4. Else in the case when images are equal and image comparison is under certain threshold then the images are same, and process continuous.

3.1 Block-Diagram

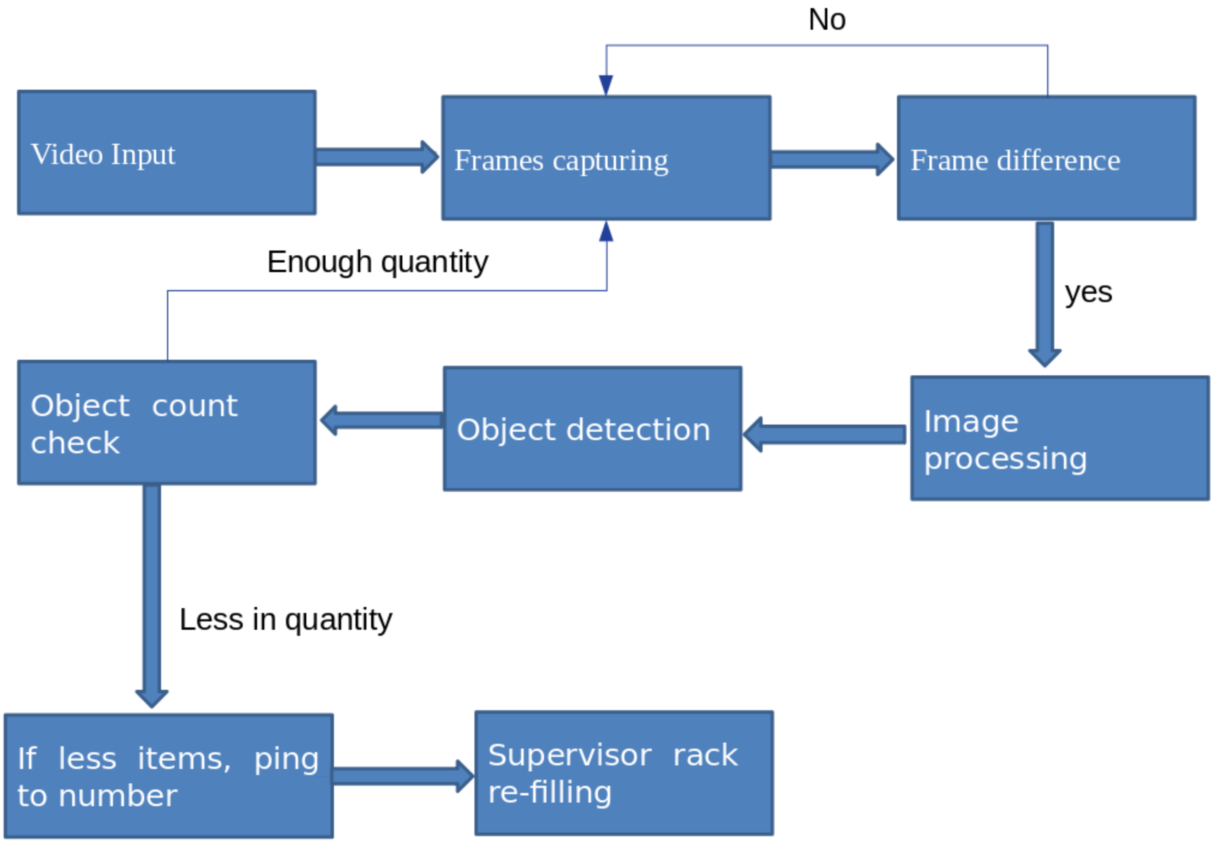


Figure 3.1: Block diagram for Real-Time Surveillance With Optimized Image Processing supervisory Technique.)

Chapter 4

Architecture of the System :

Aim for this architecture is, if once the frame captured then its value is compared to the threshold value which the threshold is calculated by taking some fixed frame average, and then sent of object detection and count.

- Step 1: Capture Two frames for the Video.
- Step 2 : Checking for Frame Difference.
- Step 3 : Comparing the Value of the Difference matrix with the threshold value.
- Step 4 : if frame difference matrix sum is less then the threshold then frames are termed as same frames.
- Step 5 : if frame difference is greater than the threshold, then frames are sent for next level processing, goto step 1.
- Step 6 : wait for certain time, then capture the frame.

- Step 7 : The objects are detected using YOLO by calculating the support and confidence values.

- Step 8 : The required objects in the frame are boxed and counted into a variable.

- Step 9 : Check the variable if the count is less.

- Step 10 : If the count is less then limit send sms to supervisor using sinchsms.

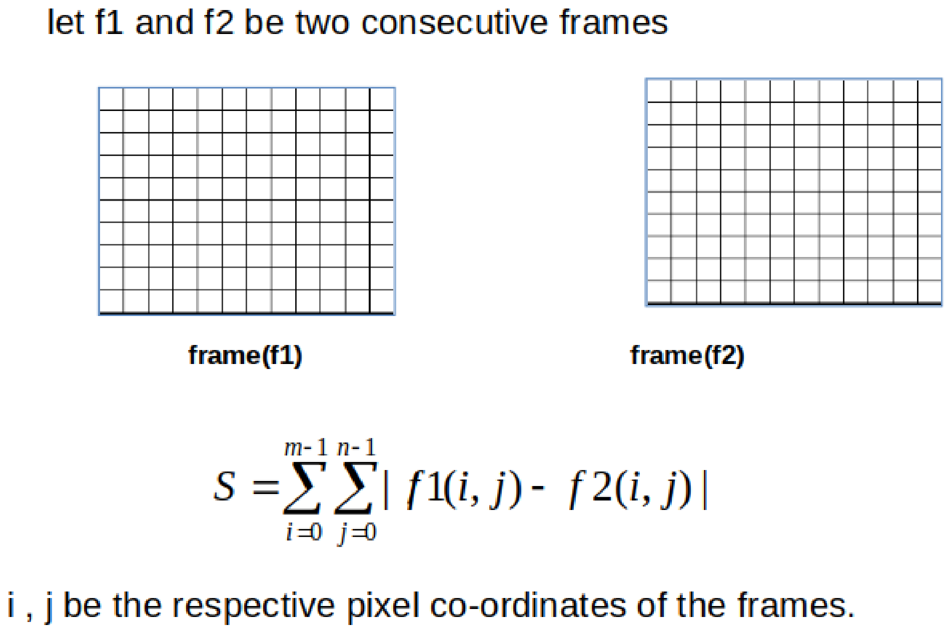
- Step 11 : goto step 1

Chapter 5

Modules of the System

5.1 Idea about Frame-Difference :

Frame difference is good technique in which two frames are compared for checking the equality by transforming the two images into the gray scale and then they are converted into matrix where the matrix difference is taken into account, the difference will be up-to some pixel threshold level, then both the frames are termed as same if it crosses the threshold then they are of different frames.



let f1 and f2 be two consecutive frames

$$S = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} | f1(i, j) - f2(i, j) |$$

i , j be the respective pixel co-ordinates of the frames.

Figure 5.1: Frame Difference formulae application.)

5.1.1 Formula application for the frame change detection

5.2 Idea about Image processing :

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it.

5.3 Idea about Object Detection :

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.

- Fast R-CNN
- Faster R-CNN
- YOLO : YOU ONLY LOOK ONCE

5.4 YOLO : You Only Look Once.

The YOLO is an advance object detection technique.The YOLO design enables end-to-end training and real- time speeds while maintaining high average precision.

Figure 5.2: Architecture of CNN

5.5 R-CNN

the R-CNN algorithm proposes a bunch of boxes in the image and checks if any of these boxes contain any object. R-CNN uses selective search to extract these boxes from an image (these boxes are called regions).

Lets first understand what selective search is and how it identifies the different regions. There are basically four regions that form an object: varying scales, colors, textures, and enclosure. Selective search identifies these patterns in the image and based on that, proposes various regions. Here is a brief overview of how selective search works:

- It first takes an image as input.
- It generates initial sub-segmentation's so that we have multiple regions from this image.
- The technique then combines the similar regions to form a larger region (based on color similarity, texture similarity, size similarity, and shape compatibility).
- Finally, these regions then produce the final object locations (Region of Interest).

5.5.1 Problems with R-CNN

So far, weve seen how R-CNN can be helpful for object detection. But this technique comes with its own limitations. Training an R-CNN model is expensive and slow thanks to the below steps:

- Extracting 2,000 regions for each image based on selective search.
- Extracting features using CNN for every image region. Suppose we have N images, then the number of CNN features will be N*2,000 The entire process of object detection using R-CNN has three models:
  - CNN for feature extraction
  - Linear SVM classifier for identifying objects
  - Regression model for tightening the bounding boxes

5.6 Fast R-CNN

In Fast R-CNN, we feed the input image to the CNN, which in turn generates the convolutional feature maps. Using these maps, the regions of proposals are extracted. We then use a RoI pooling layer to reshape all the proposed regions into a fixed size, so that it can be fed into a fully connected network.

Lets break this down into steps to simplify the concept:

- As with the earlier two techniques, we take an image as an input.
- This image is passed to a ConvNet which in turns generates the Regions of Interest.
- A RoI pooling layer is applied on all of these regions to reshape them as per the input of the ConvNet. Then, each region is passed on to a fully connected network.
- A softmax layer is used on top of the fully connected network to output classes. Along with the softmax layer, a linear regression layer is also used parallely to output bounding box coordinates for predicted classes.

5.6.1 Problems with Fast R-CNN

But even Fast R-CNN has certain problem areas. It also uses selective search as a proposal method to find the Regions of Interest, which is a slow and

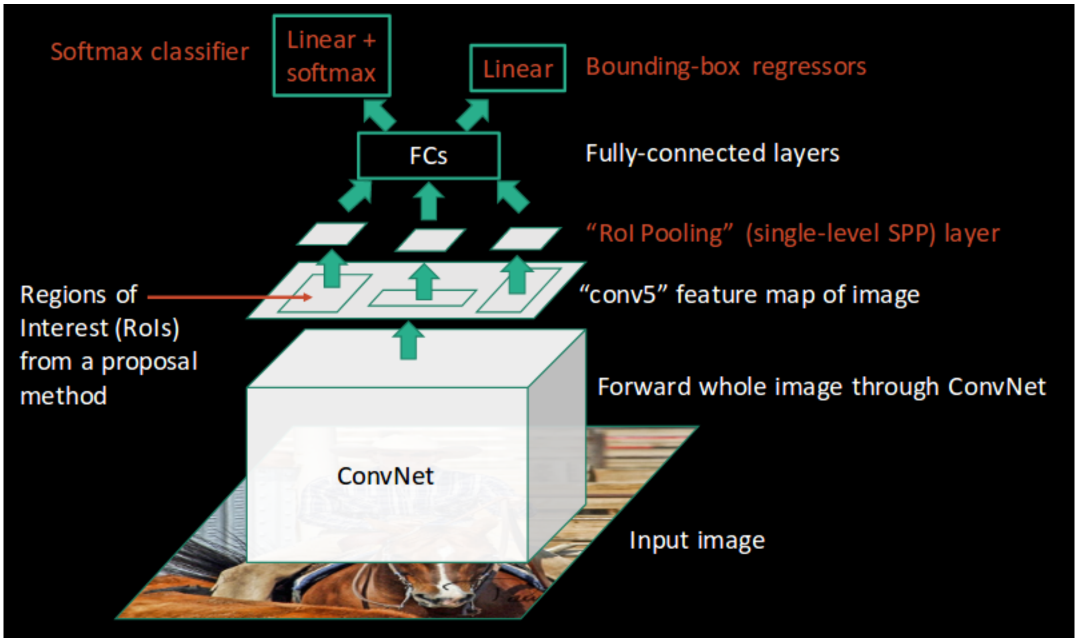

Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Figure 5.3: Working of Fast R-CNN

time consuming process. It takes around 2 seconds per image to detect objects, which is much better compared to R-CNN. But when we consider large real-life datasets, then even a Fast R-CNN doesnt look so fast anymore.

But theres yet another object detection algorithm that trump Fast R-CNN. And something tells me you wont be surprised by its name.

5.7 Faster R-CNN

Faster R-CNN [11] is the modified version of Fast R-CNN. The major difference between them is that Fast R-CNN uses selective search for generating

Regions of Interest, while Faster R-CNN uses RPN (Region Proposal Network), aka RPN. RPN takes image feature maps as an input and generates a set of object proposals, each with an objectness score as output.

5.7.1 Working of Faster R-CNN

The below steps are typically followed in a Faster R-CNN [11] approach:

1. We take an image as input and pass it to the ConvNet which returns the feature map for that image.

2. Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score.

3. A RoI pooling layer is applied on these proposals to bring down all the proposals to the same size.

4. Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

5.7.2 Problems with Faster R-CNN

All of the object detection algorithms we have discussed so far use regions to identify the objects. The network does not look at the complete image in one go, but focuses on parts of the image sequentially. This creates two complications:

- The algorithm requires many passes through a single image to extract all the objects.

- As there are different systems working one after the other, the performance of the systems further ahead depends on how the previous systems performed.

- A **list of bounding boxes**, or the (x, y)-coordinates for each object in an image

- The **class label** associated with each bounding box

- The **probability/confidence score** associated with each bounding box and class label

Chapter 6

Implementation and Results

The implementation part is done using open source computer vision (opencv python).

The setup of four Miranda bottles are kept for checking process.

The video surveillance is done using the Redmi Note3 (back camera 16 mp), frame captured is of 640 * 480 pixel resolution.

6.1 Input Images

Once the first frame is captured particular time interval gap is taken and then the second image is captured.

Here the frame difference is calculated and since the difference is occurred the image is sent for processing and detection of required objects.

Then the third frame is captured after around 30 seconds of gap which is given for a person to take the bottles away.then the third frame is as follows.

Then the third frame is sent for detection and object counting.

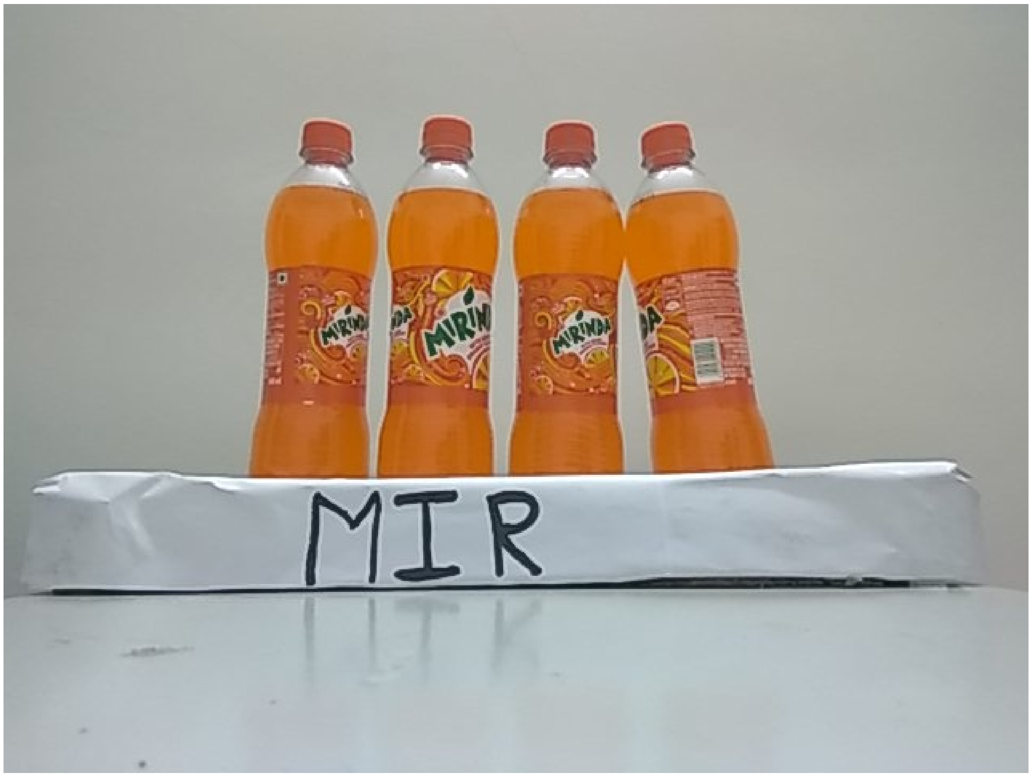


Figure 6.1: Four Miranda bottles with "MIR" written on it.

The First frame consists of four bottles with the name MIR written, which is 640 * 480 pixels.

6.2 Output

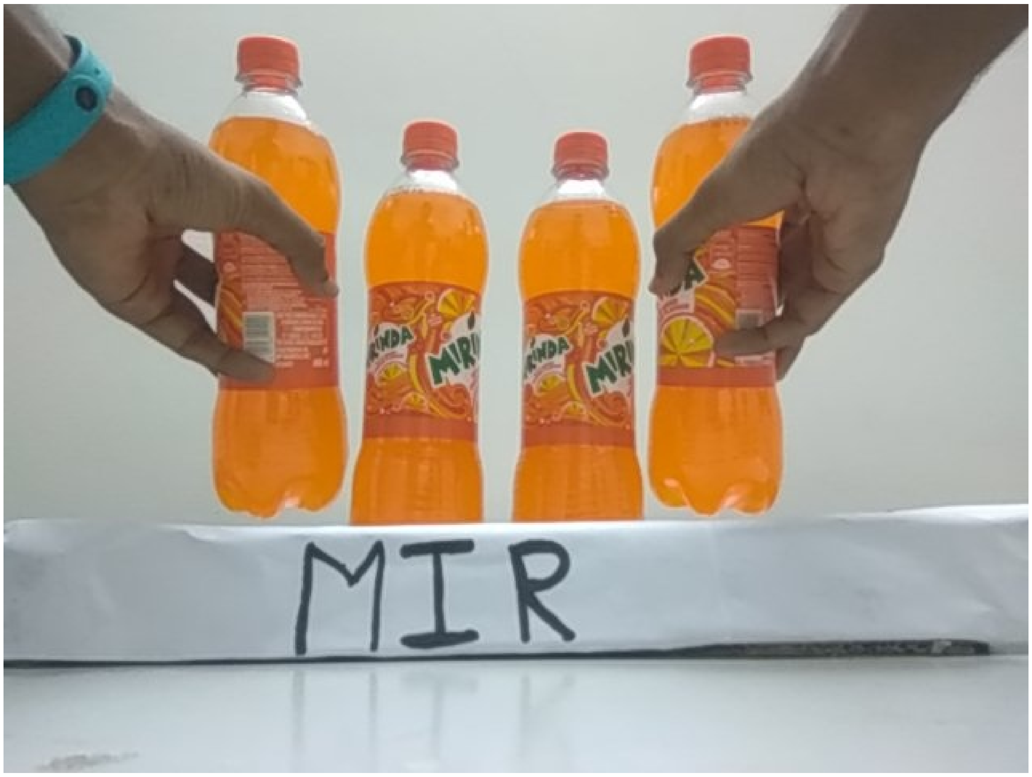


Figure 6.2: Miranda bottles with human hands in the frame.
The second

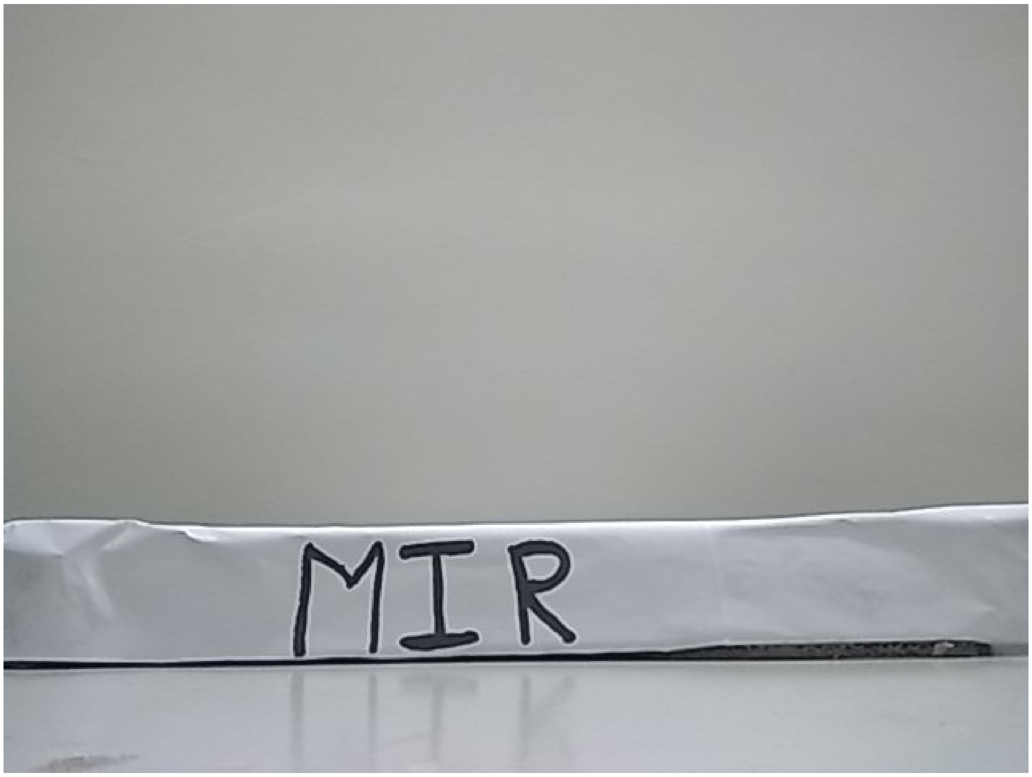
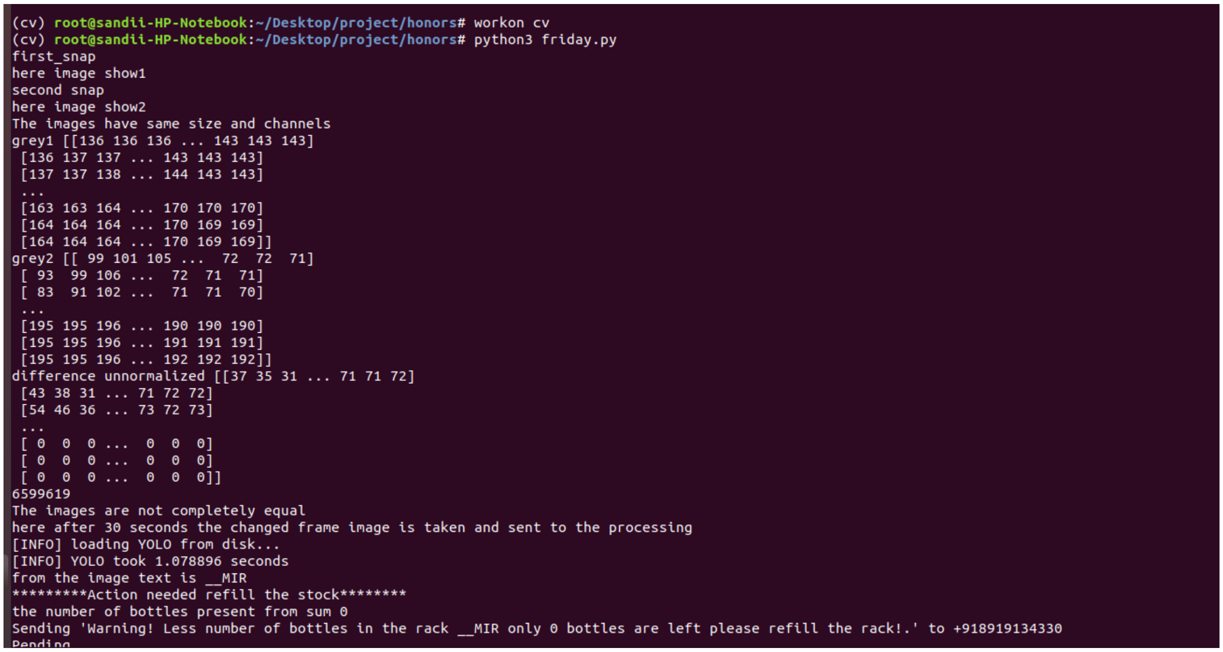
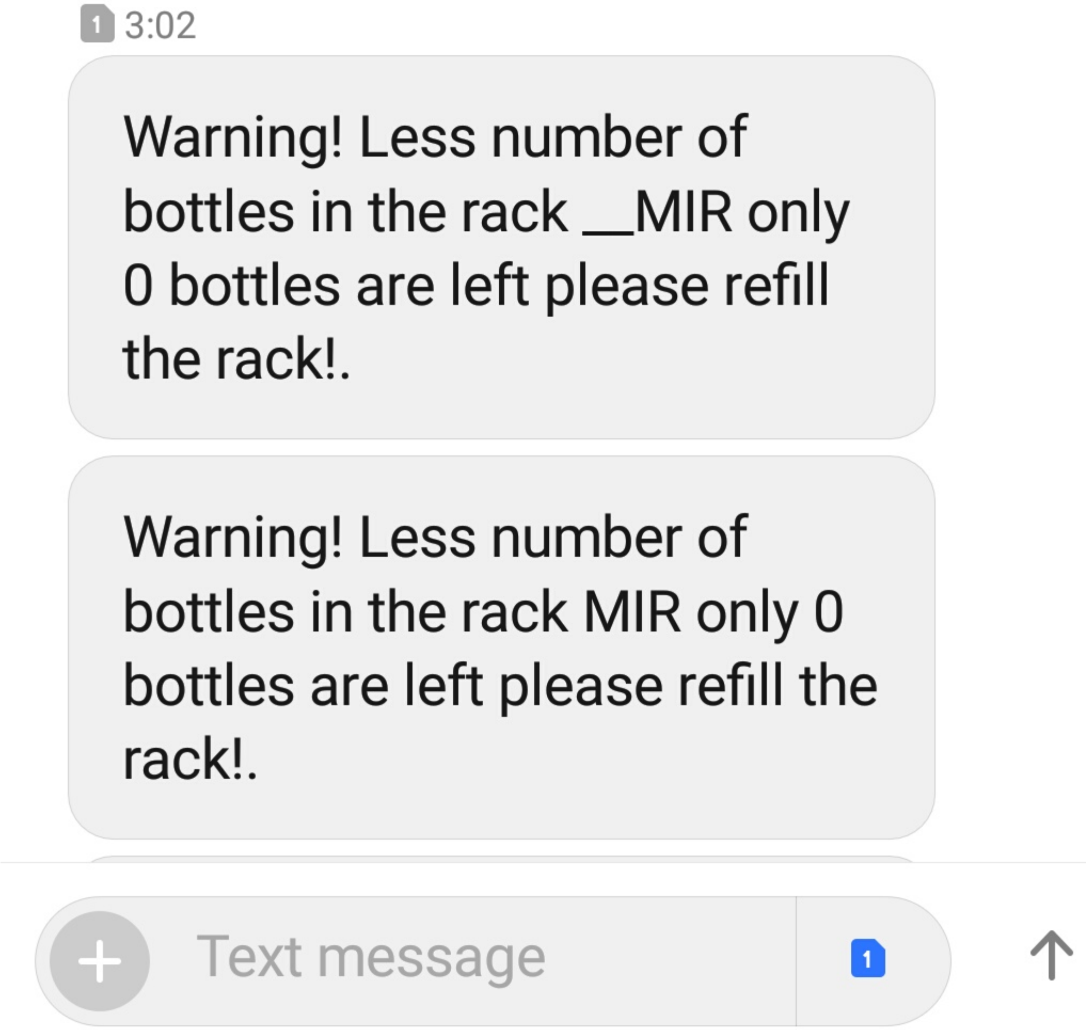


Figure 6.3: Emptied Miranda bottles with rack name "MIR"in the frame.



```
(cv) root@sandii-HP-Notebook:~/Desktop/project/honors# workon cv
(cv) root@sandii-HP-Notebook:~/Desktop/project/honors# python3 friday.py
first_snap
here image show1
second snap
here image show2
The images have same size and channels
grey1 [[136 136 136 ... 143 143 143]
 [136 137 137 ... 143 143 143]
 [137 137 138 ... 144 143 143]
 ...
 [163 163 164 ... 170 170 170]
 [164 164 164 ... 170 169 169]
 [164 164 164 ... 170 169 169]]
grey2 [[ 99 101 105 ...  72  72  71]
 [ 93  99 106 ...  72  71  71]
 [ 83  91 102 ...  71  71  70]
 ...
 [195 195 196 ... 190 190 190]
 [195 195 196 ... 191 191 191]
 [195 195 196 ... 192 192 192]]
difference unnormalized [[37 35 31 ... 71 71 72]
 [43 38 31 ... 71 72 72]
 [54 46 36 ... 73 72 73]
 ...
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]]
6599619
The images are not completely equal
here after 30 seconds the changed frame image is taken and sent to the processing
[INFO] loading YOLO from disk...
[INFO] YOLO took 1.078896 seconds
from the image text is __MIR
*********Action needed refill the stock********
the number of bottles present from sum 0
Sending 'Warning! Less number of bottles in the rack __MIR only 0 bottles are left please refill the rack!.' to +918919134330
Pending
```

Figure 6.4: Cmd output showing the results.



Warning! Less number of bottles in the rack __MIR only 0 bottles are left please refill the rack!.

Warning! Less number of bottles in the rack MIR only 0 bottles are left please refill the rack!.

Figure 6.5: Warning message about ”0” Miranda bottles sent to mobile.

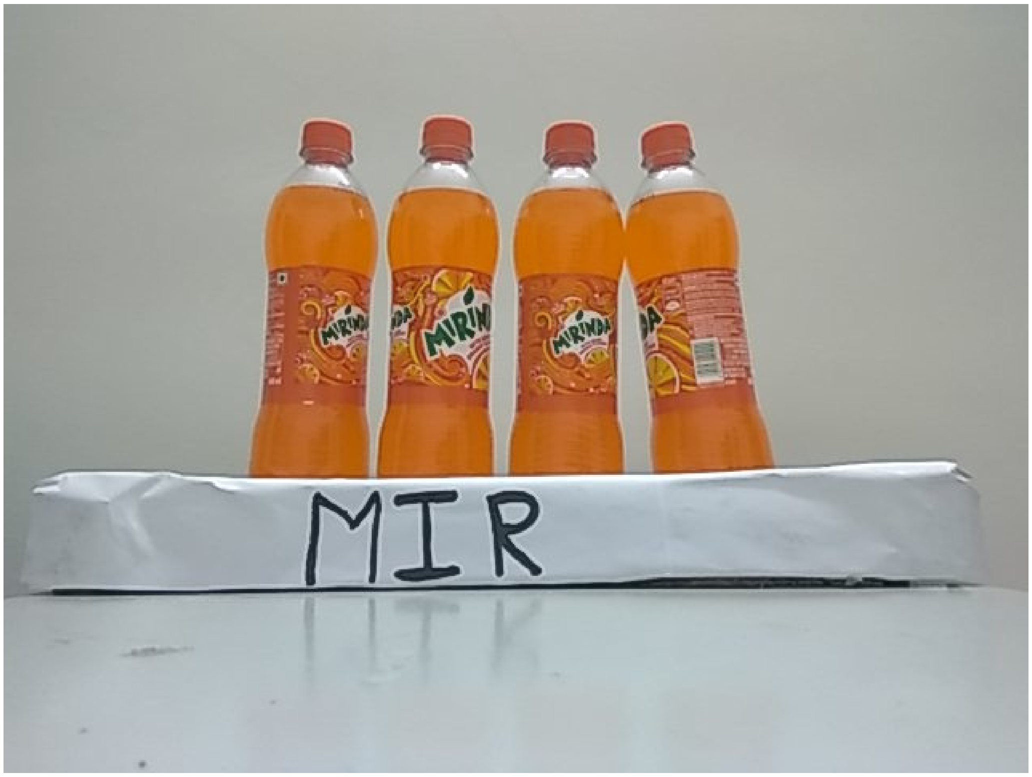


Figure 6.6: four Miranda bottles.

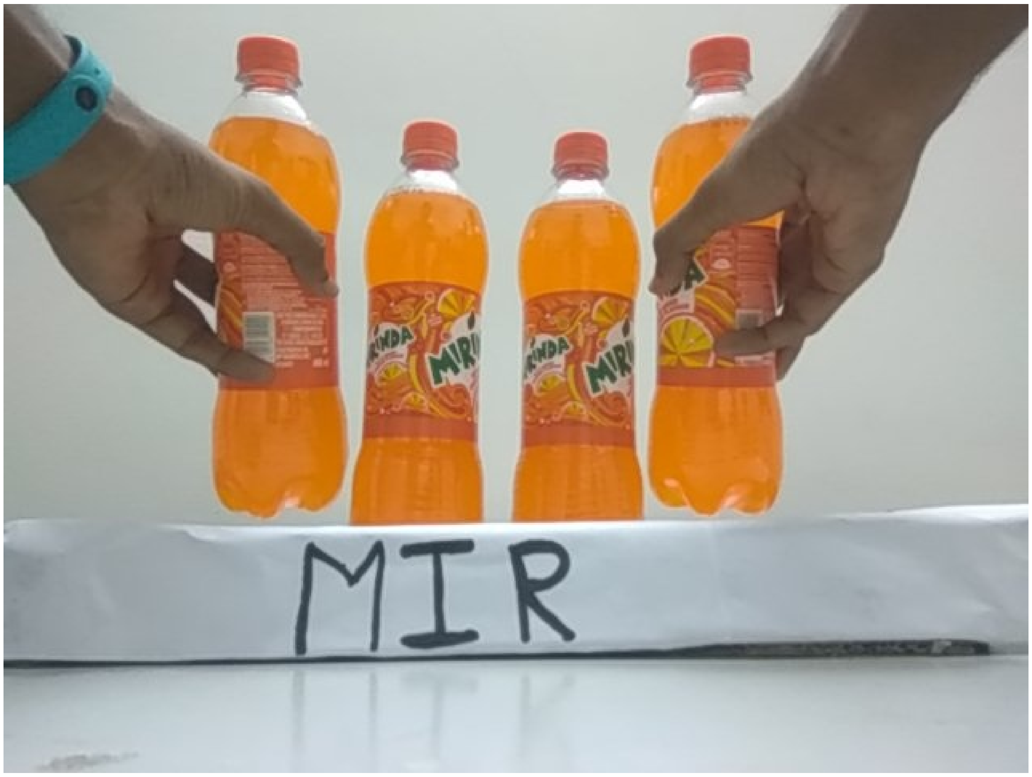


Figure 6.7: Second captured frame .

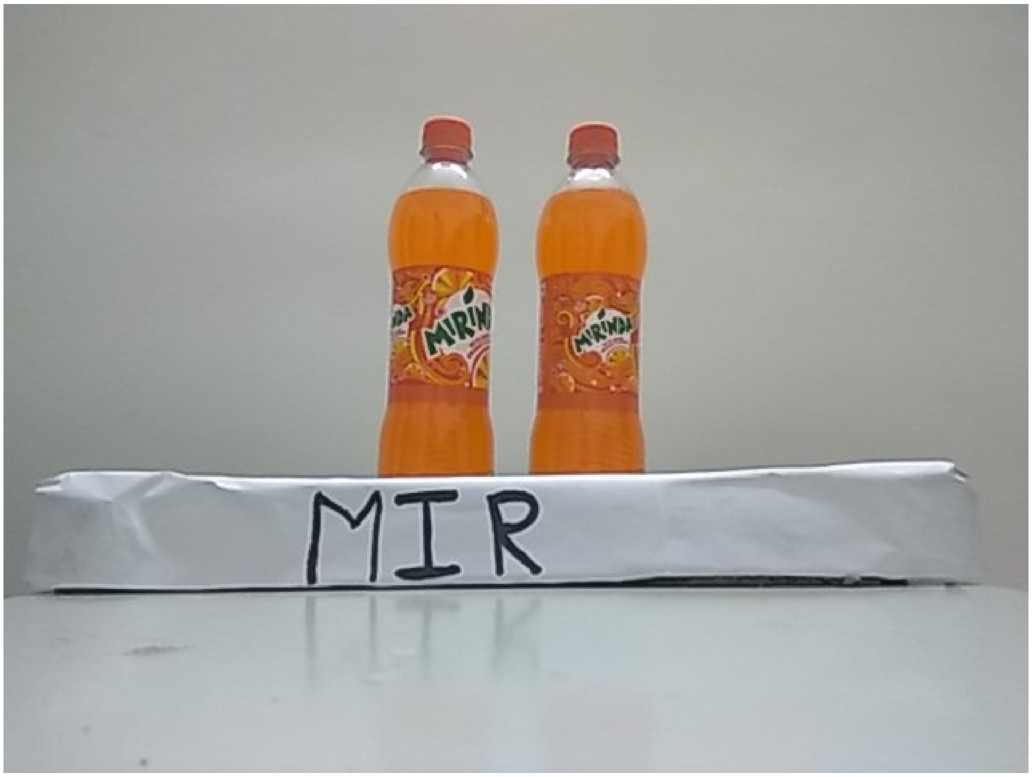


Figure 6.8: Third frame captured after interval gap.

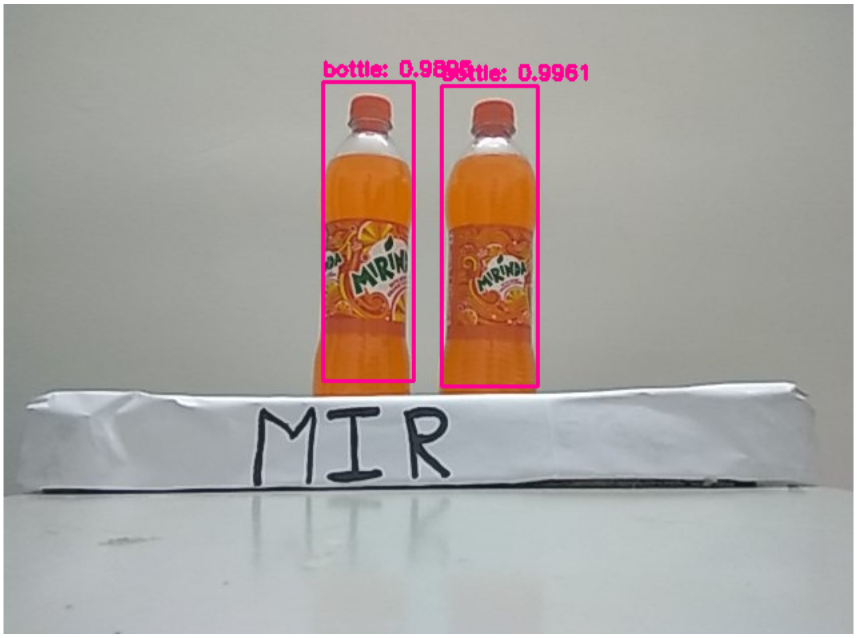
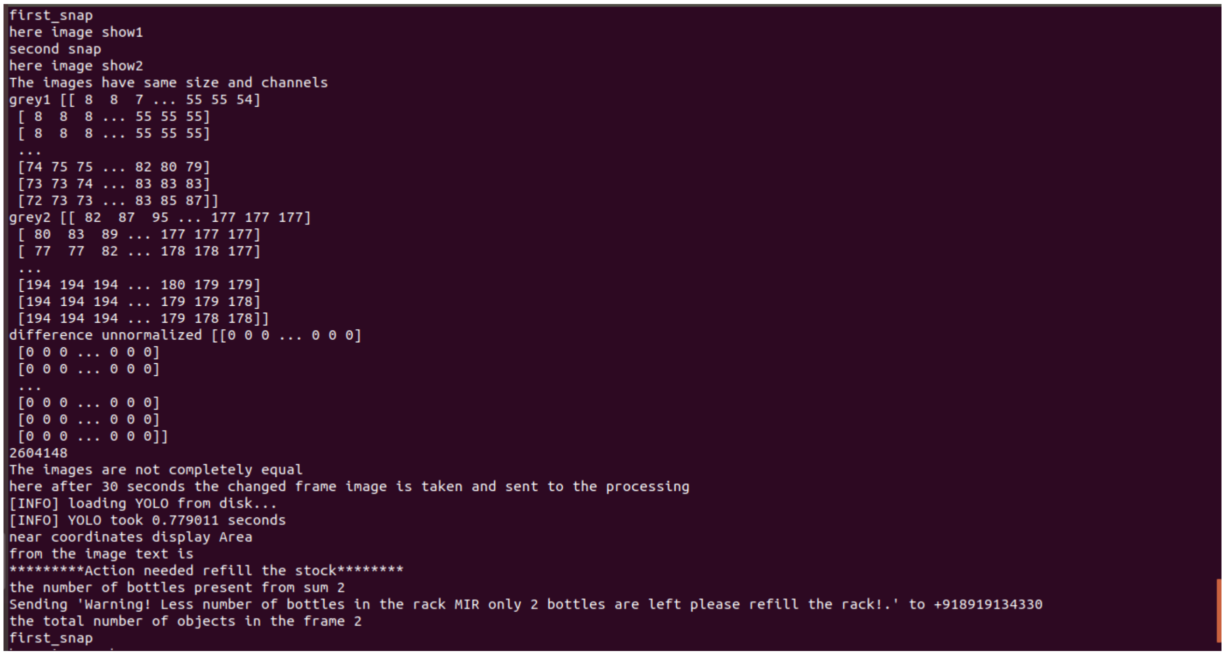
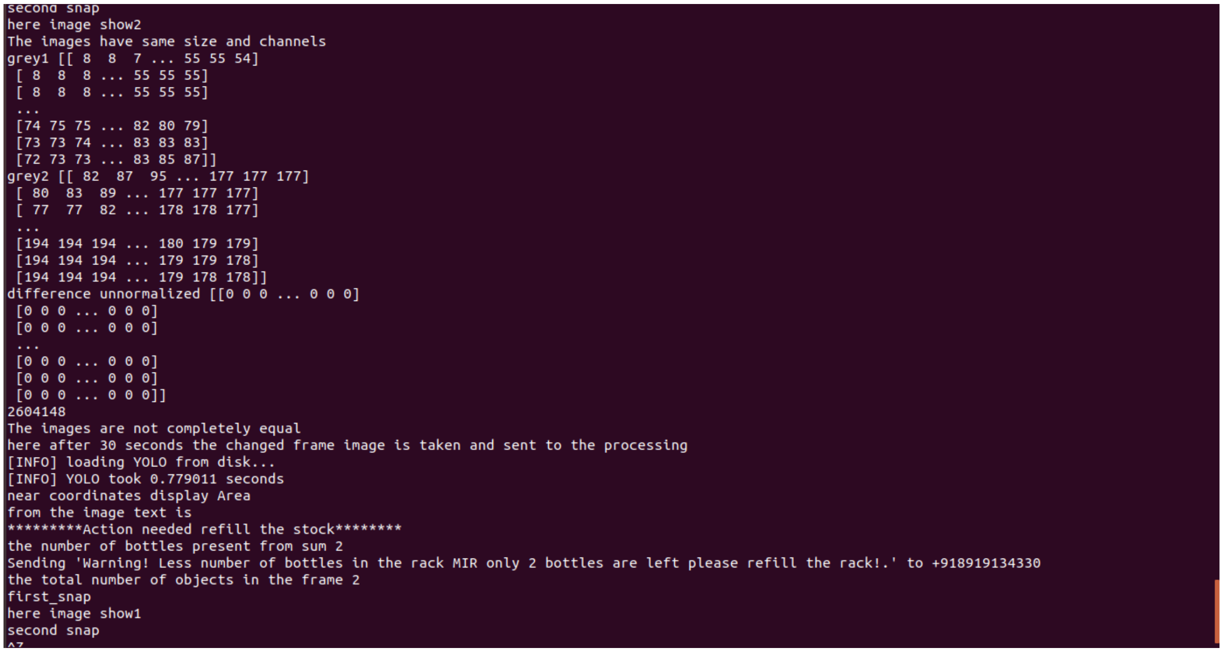


Figure 6.9: Cmd output showing the results.



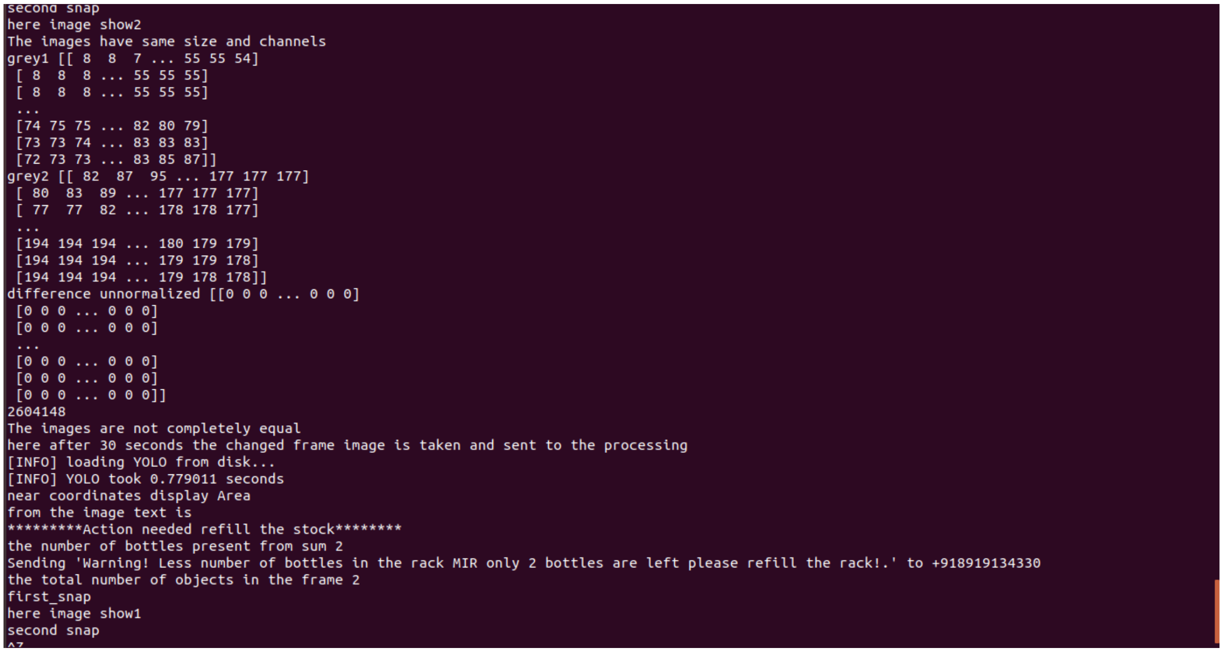
```
first_snap
here image show1
second snap
here image show2
The images have same size and channels
grey1 [[ 8  8  7 ... 55 55 54]
 [ 8  8  8 ... 55 55 55]
 [ 8  8  8 ... 55 55 55]
 ...
 [74 75 75 ... 82 80 79]
 [73 73 74 ... 83 83 83]
 [72 73 73 ... 83 85 87]]
grey2 [[ 82  87  95 ... 177 177 177]
 [ 80  83  89 ... 177 177 177]
 [ 77  77  82 ... 178 178 177]
 ...
 [194 194 194 ... 180 179 179]
 [194 194 194 ... 179 179 178]
 [194 194 194 ... 179 178 178]]
difference unnormalized [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
2604148
The images are not completely equal
here after 30 seconds the changed frame image is taken and sent to the processing
[INFO] loading YOLO from disk...
[INFO] YOLO took 0.779011 seconds
near coordinates display Area
from the image text is
*********Action needed refill the stock********
the number of bottles present from sum 2
Sending 'Warning! Less number of bottles in the rack MIR only 2 bottles are left please refill the rack!.' to +918919134330
the total number of objects in the frame 2
first_snap
```

Figure 6.10: Two bottles image .



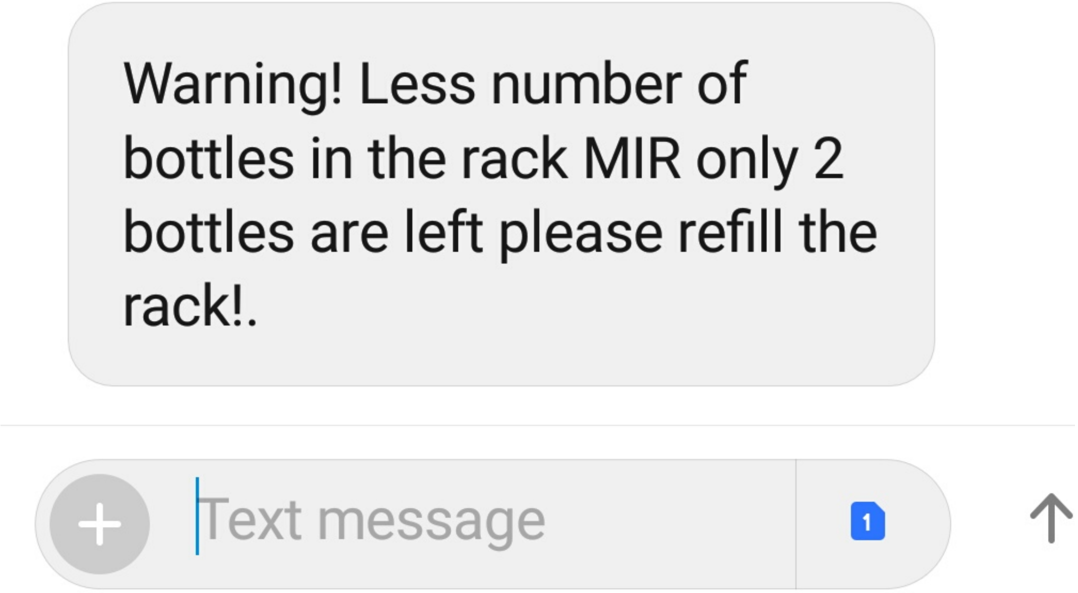
```
second snap
here image show2
The images have same size and channels
grey1 [[ 8  8  7 ... 55 55 54]
 [ 8  8  8 ... 55 55 55]
 [ 8  8  8 ... 55 55 55]
 ...
 [74 75 75 ... 82 80 79]
 [73 73 74 ... 83 83 83]
 [72 73 73 ... 83 85 87]]
grey2 [[ 82  87  95 ... 177 177 177]
 [ 80  83  89 ... 177 177 177]
 [ 77  77  82 ... 178 178 177]
 ...
 [194 194 194 ... 180 179 179]
 [194 194 194 ... 179 179 178]
 [194 194 194 ... 179 178 178]]
difference unnormalized [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
2604148
The images are not completely equal
here after 30 seconds the changed frame image is taken and sent to the processing
[INFO] loading YOLO from disk...
[INFO] YOLO took 0.779011 seconds
near coordinates display Area
from the image text is
*********Action needed refill the stock********
the number of bottles present from sum 2
Sending 'Warning! Less number of bottles in the rack MIR only 2 bottles are left please refill the rack!.' to +918919134330
the total number of objects in the frame 2
first_snap
here image show1
second snap
^Z
```

Figure 6.11: Cmd output showing the results.



```
second snap
here image show2
The images have same size and channels
grey1 [[ 8  8  7 ... 55 55 54]
 [ 8  8  8 ... 55 55 55]
 [ 8  8  8 ... 55 55 55]
 ...
 [74 75 75 ... 82 80 79]
 [73 73 74 ... 83 83 83]
 [72 73 73 ... 83 85 87]]
grey2 [[ 82  87  95 ... 177 177 177]
 [ 80  83  89 ... 177 177 177]
 [ 77  77  82 ... 178 178 177]
 ...
 [194 194 194 ... 180 179 179]
 [194 194 194 ... 179 179 178]
 [194 194 194 ... 179 178 178]]
difference unnormalized [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
2604148
The images are not completely equal
here after 30 seconds the changed frame image is taken and sent to the processing
[INFO] loading YOLO from disk...
[INFO] YOLO took 0.779011 seconds
near coordinates display Area
from the image text is
*********Action needed refill the stock********
the number of bottles present from sum 2
Sending 'Warning! Less number of bottles in the rack MIR only 2 bottles are left please refill the rack!.' to +918919134330
the total number of objects in the frame 2
first_snap
here image show1
second snap
^Z
```

Figure 6.12: Cmd output showing the results.



Warning! Less number of bottles in the rack MIR only 2 bottles are left please refill the rack!.

Chapter 7

Conclusion

As the project is related to Real-Time surveillance, this application will help the supervisors in their daily life with flexible environment with out any manual labour being appointed for monitoring. Image capturing, Frame difference, Image processing, Object detection, and counting number of required items in the frame has been successfully implemented as per requirement of the system. Once if the system is deployed in a big complexes like DMART, Reliance mart and More, kind of supermarkets it really gives a good result helping the managing staff saving the salary of the monitoring person, increasing the sales without losing the customers if the rack of required items are less or empty.

Chapter 8

Future Scope

Image processing and Object detection has good scope in the where where it can be used as survilliance and many other monitering applications.

This application can be future developed in the next phases and improved taking the data connecting it to database and some analysis part can be done(DATA-MINING), where at which racks in the complexes has more demand and at what time in which area supermarkets sales are more and which to be carefully taken into care for future analysis and increasing the sales.

Bibliography

[1] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 2016.

[2] Ross Girshick. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2015.

[3] Venugopal Gundimeda, Ratan S Murali, Rajkumar Joseph, and NT Naresh Babu. An automated computer vision system for extraction of retail food product metadata. *First International Conference on Artificial Intelligence and Cognitive Computing*, 2019.

[4] Wei-Ying Ma and Bangalore S Manjunath. Edgeflow: a technique for boundary detection and image segmentation. *IEEE transactions on image processing*, 2000.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *IEEE*, 2016.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015.

[7] Sayantani Saha and Sarmistha Neogy. A case study on smart surveillance application system using wsn and ip webcam. In *2014 Applications and Innovations in Mobile Computing (AIMoC)*, pages 36–41. IEEE, 2014.

[8] Nishu Singla. Motion detection based on frame difference method. *International Journal of Information & Computation Technology*, 2014.

[9] Ray Smith. An overview of the tesseract ocr engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).*

[10] Srikrishna Srivastava Varadarajan, Muktabh Mayank. Weakly supervised object localization on grocery shelves using simple fcn and synthetic dataset. *arXiv preprint arXiv:1803.06813*, 2018.