A

PROJECT REPORT

ON

"Automatic Grocery Item Detection and Recognition"

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

Electrical and Electronics

Submitted by:

Himani Maheshwari(2016546)
Muskan Gangil(2016577)
Samiksha Agarwal(2016601)
Vratika Kataria(2016629)

Supervised by:

Dr. Abdullah Bin Queyam

SCHOOL OF AUTOMATION

BANASTHALI VIDYAPITH RAJASTHAN

May 2023

BANASTHALI VIDYAPITH RAJASTHAN

SCHOOL OF AUTOMATION

CERTIFICATE

I hereby submit the project entitled **"Automatic Grocery Item Detection and Recognition"** in the **School of Automation** of the Banasthali Vidyapith, under the supervision of **"Dr. Abdullah Bin Queyam"**, School of Automation, Banasthali Vidyapith, Rajasthan, India.

Himani Maheshwari (2016546)
Muskan Gangil(2016577)
Samiksha Agarwal (2016601)
Vratika Kataria (2016629)

The project report is hereby approved for submission.

Dr. Abdullah Bin Queyam
(Mentor)

Prof. Shailly Sharma
(HOD School of Automation)

Date:

ABSTRACT

Image processing and Object detection are one of the most popular areas in the field of engineering and science. Various platforms are being used for designing and implementation of object detection algorithms. It includes C programming, MATLAB, Simulink, open CV etc. Among these, MATLAB is most popular because of its extensive features and ease of use. The Automatic Grocery Items Detection is an open-source business solution used by supermarkets, department stores, and more. In addition to payment, many billing systems in supermarkets also aid in accounting and inventory-related processes. If you are a local supermarket, then this is a good fit for you. As it provides a single interface to manage these different major stores. Billing system for grocery shops is built to calculate and display bills for different items and serve the customer in a faster and efficient manner. Various operations are carried out and the moving objects are detected. Thresholds at various phases decide the possibility of identifying the moving object of certain sizes. Moving objects are also tracked in it. MATLAB is used for implementation of the algorithm. Grocery shop is the place where customers come to purchase their daily using products and pay for that. So, there is a need to calculate how many products are sold and to generate the bill for the customer. This system is built for fast data processing and bill generation for customers. The user will consume less time in calculation and sales activity will be completed within fraction of seconds whereas in a manual system will make a user to write it down which is a long procedure and it also consumes a lot of time.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete the report.

A special gratitude I gave to my Supervisor **Dr. Abdullah Bin Queyam, whose** contribution in stimulating suggestions and encouragement, helped me to coordinate my project for this session.
I sincerely thank our head of department **Prof. Shailly Sharma,** Dean. School of Automation. Banasthali University, for her guidance and encouragement in carrying out this work. I also wish to express my gratitude to other staff members who rendered their help.

My thanks and appreciation also goes to my friends who helped me in my work on the project and people who have willingly helped me out with their abilities.

Himani Maheshwari (2016546)
Muskan Gangil (2016577)
Samiksha Agarwal (2016601)
Vratika Kataria (2016629)

Contents

List of figures

CHAPTER – 1

INTRODUCTION

Human quest for an automatic detection system of everyday occurrence led to the necessity of inventing an intelligent surveillance system which will make lives easier as well as enable us to compete with tomorrow's technology and on the other hand it pushes us to analyze the challenge of the automated video surveillance scenarios harder in view of the advanced artificial intelligence. It also deals with detecting instances of semantic objects of a certain class (such as humans, buildings, grocery items or cars) in digital images and videos. It is a computer vision technique that allows us to identify and locate objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene and determine. and track their precise locations, all while accurately labeling them.

In object detection and recognition, Object detection is a critical step. The performance of this step is important for scene analysis, object matching and tracking, and activity recognition. Over the years research is flowing towards innovating new concepts and improving or extending the work done in this field till now. Various object detection approaches have been developed based on statistical, fuzzy, neural networks etc. Most approaches involve complex theory. Object recognition is to describe a collection of related computer vision tasks that involve activities like identifying objects in digital photographs. Image classification involves activities such as predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing a bounding box around their extent. Object detection does the work of combining these two tasks and localizes and classifies one or more objects in an image. When a user or practitioner refers to the term “object recognition“, they often mean “object detection“. It may be challenging for beginners to distinguish between different related computer vision tasks.

The retail industry has been integrating technological innovations throughout its product chain in order to reduce costs and, cumulatively, improve customer experience, since increasing profit margins and attracting customers are the primary goals of any entrepreneur. In the retail context, the automatic detection and recognition of products has allowed a more efficient use of resources and revolutionized the way customers buy. They want the purchase process to be simple and fast; therefore, they value payout and tools that allow them to find desired items, know the product's availability and avoid payment queues.

Automatic self-checkout systems are the solution to fill the last need. Real-time inventory management, in particular automatic shelf monitoring, is also possible with computer vision tools, with out-of-stock shelves being detected in real-time by capturing images of racks. By the same philosophy, there is an opportunity to verify if the product displays and store layout follow the plan known as a planogram. Self-service technology is being applied to create autonomous stores and to permit a self-guided shopping experience, and product recognition devices can add value to the customer experience by assisting them in the correct purchase of the product.

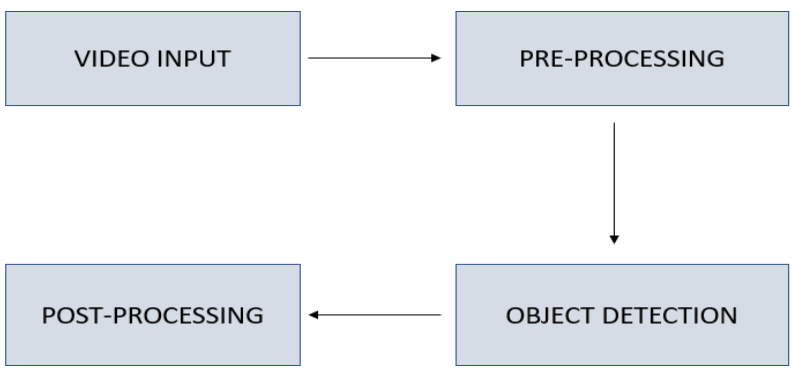

Figure 1: Basic block diagram of object detection.

The Function of each block (in Fig 1):

- Video Input: It can be a stored video or a real time video.
- Pre-processing: It mainly involves temporal and spatial smoothing such as intensity adjustment, removal of noise, contrast adjustment etc. For real time systems, frame – size and frame – rate reduction is commonly used. It highly reduces computational cost and time.

- Object detection: It is the process of change detection and extracts appropriate change for further analysis and qualification. Pixels are classified as foreground, if they changed. Otherwise, they are considered as background. This process is called background subtraction. The degree of "change" is a key factor in segmentation and can vary depending on the application. The result of segmentation is one or more foreground blobs, a blob being a collection of connected pixels. Post-processing: - Remove false detection caused due to dynamic conditions in the background using morphological and speckle noise removal.
- Post-processing: This step involves choosing the boxes that best localize objects such that each object has only 1 corresponding bounding box. Post processing methods are usually employed for this purpose. They help suppress superfluous boxes and aid in selecting the best possible boxes. This article provides an overview of some of the popular post processing mechanisms.

1.1-BACKGROUND

Humans can easily locate and identify objects of interest (such as people, animals, buildings or cars) in an image or video. Object detection and recognition are computer vision tasks that intend to reproduce this ability. The interest in this area dates back more than three decades and the progress of objection detection and recognition can be divided into two periods based on the main strategies: traditional handcrafted-feature-based methods and deep-learning-based methods. Manual feature extraction suffers from a lack of robustness due to the diversity of aspects of the class and strongly depends on the experience of the researcher.

The aim of object detection is to detect all instances of objects from a known class, such as people, grocery items, cars or faces in an image. Generally, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored.

Each detection of the image is reported with some form of pose information. This is as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. In some other situations, the pose information is more detailed and contains the parameters of a linear or non-linear transformation. For example, face detection in a face detector may compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face. Object detection systems always construct a model for an object class from a set of training examples. In the case of a fixed rigid object in an image, only one example may be needed, but more generally multiple training examples are necessary to capture certain aspects of class variability.
We have tried different techniques like Cascade objectdetector, trainACFobjectdtector ,SURFfeature, Alexnet, featuring Points.

INTRODUCTION TO MATLAB-

Like other well-known programming languages like Java, C#, etc., MATLAB has its own Integrated Development Environment (IDE) and collection of libraries. Since it was initially known as the matrix programming language, MATLAB is an acronym for "Matrix Laboratory." It is a programming language of the fourth generation. It is a multi-paradigm, MATLAB. Therefore, it can be used with a variety of programming paradigms, including functional, Visual, and Object-Oriented.

The various uses of MATLAB are:

- Developing algorithms
- Performing linear algebra that is linear
- Graph plotting for larger data sets
- Data visualization and analysis
- Numerical Matrix Computation
- Image Processing

IMPORTANCE OF GRAY SCALE IN IMAGE PROCESSING-

- It helps in simplifying algorithms and as well eliminates the complexities related to computational requirements.

- It makes room for easier learning for those who are new to image processing. This is because grayscale compressors an image to its barest minimum pixel.

- It enhances easy visualisation. It differentiates between the shadow details and the highlights of an image because it is mainly in 2 spatial dimensions (2D) rather than 3D.

- Colour complexity is also reduced. A typical 3D image requires camera calibration on brightness among others. The grayscale conversion option is very useful for captured images which do not need to match coloured detail.

1.2-MOTIVATION

Barcodes are widely used in many grocery supermarkets like Big Bazar, Easy Day for billing and statement generation. Check-out counters use laser bar-code readers in such stores but the space between the sensor and the object should be nearly zero when the reader is applied. The billing personnel have to manipulate either the reader or the objects. This makes the task tedious for the human worker. Each object has to be scanned individually taking much time and making the task monotonous for the billing personnel especially in big stores where hundreds of customers turn up in a day and thousands of objects have to be scanned in a day. In the modern era, the people have more income to spend and lesser time to spare, so they generally choose supermarkets for grocery and other shopping rather than local shops. Actually, the customer is free to choose products from huge available varieties which attract the large customers mainly in big cities so long queues of customers are seen at these stores. In many cases, the barcode is either damaged or there may be a problem in reading it due to lighting effects, occlusion, low resolution etc. A bar code based billing system is also costly as it requires bar coding of all products.

The world is moving towards an era of automation and human capital is a great asset which should be used in more intellectual works rather than manual, monotonous works. The advancements in technology have led to high-speed computers with excellent processors and storage capacity. The concepts of Image processing and Object detection can be used to develop an application for real time automated billing to ease laborious human work. For the purpose of automation, the human operator needs to be removed from the process. Computer vision based systems could be developed and deployed for such an automated billing application where minimum human interference and lesser wait time is required leading to customer satisfaction. The recognition in tangled real-world scenes needs to be unaffected by nearby tangle, fractional obstruction, orientation change, scale change.

Local descriptors are generally engaged in various real-world applications like image retrieval and object recognition because they are resistant to fractional obstructions, relatively unaffected to deviations in view and can be computed capably. There are two things to be kept in mind while using local descriptors. First, the interest point in position and scale must be chosen to safeguard only those points that are most probable to remain stable over transformations. Second, the interest point descriptor must be built distinctive, concise, and invariant over transformations. Thus,the main steps in object recognition are computation of local interest points, computation of descriptors and indexing/matching- In detection, some operators are applied to find typical key points during the feature detection stage to match well in other images. In description, the detected features are termed on the basis of the neighbor pixels around it during the feature description stage. In matching, each demanded feature is matched to the similar features with the referred one during this stage. Various types of local feature detector and descriptor have been developed but different recognition rates, efficiency, computation times, memory requirements are obtained by using various descriptor interest point algorithms for detection, description and matching. The challenge is to find the algorithms which are suitable for the automated billing application and finding the balance between various parameters to best suit the application.

CHAPTER 2

LITERATURE REVIEW

As a part of this survey review, some papers were downloaded in order to present a systematic technical analysis of the object detection techniques from various digital libraries. After studying the paper title, abstract, introduction, experiment and future scope, 10 most suitable papers for the review have been identified and systematically arranged in this section of the paper.

The literature review discusses previous attempts made in the field of research and development of object detection and usage of various machine learning and deep neural networks. The discussion on literature survey includes significant contributions made by different scholars in this discipline.

- A Practical Implementation of Face Detection by Using MATLAB Cascade Object Detector

This paper presents a prototype model of face detection in an image. The algorithm which allowed face detection, imposing new standards in this area, was the Viola – Jones algorithm. In this paper, a practical implementation of a face detector based on Viola-Jones algorithm using MATLAB cascade object detector is presented. Employing the system type object vision. CascadeObjectDetector,eight face detectors were developed using the trainCascadeObjectDetector function and tuning the number of cascade layers and the False Alarm Rate. For different tuning parameters, the performances of the face detectors were analyzed.

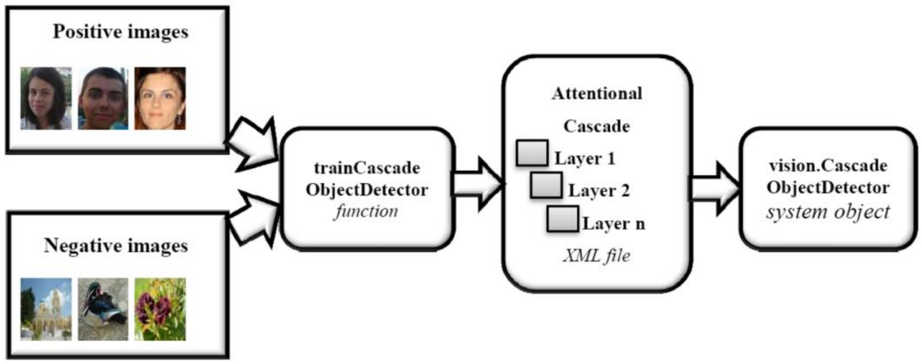

Fig 2.1: The attentional cascade training

- Object detection and classification by cascade object training

This research was done to train a model to detect some objects of interest among diversified objects, which in this case, were cups with saucers with differing colors. Our developed algorithm in terms of CV successfully detected and labeled the test images with varying degrees of accuracy – neutralizing the deficiencies of the primordial Viola-Jones' cascade classifier while maintaining almost zero incorrect refusal; however, there are scopes for improvement. The principles utilized for recognizing the cups could be employed for detecting other objects as well, in a multitude of scenarios.

Fig 2.2: The MATLAB Output

- Moving object detection and tracking Using Convolutional Neural Networks

A novel approach for object detection and tracking has been presented using convolutional neural networks. The moving object detection is performed using TensorFlow object detection API. The object detection module robustly detects the object. The detected object is tracked using CNN algorithm. Considering human tracking as a special case of detection of objects, spatial and temporal classes the facilities were learned during offline training. The shift variant architecture has extended the use of conventional CNNs and combined the global features and local characteristics in a natural way.

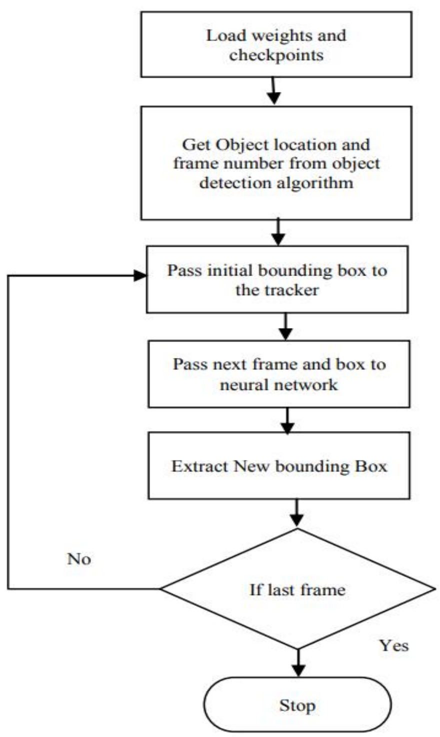

Fig 2.3: Flowchart of Object Detection.

- A Survey Paper on Object Detection Methods in Image Processing

The author has presented a review study on object detection. This survey paper provides study of various methodologies for object detection. This paper provides systematic analysis of various existing object detection techniques with precise and arranged representation. This survey also provides theoretical knowledge about different object detection approaches. Based on this study, AI based approaches particularly Deep Neural Networks and CNN have been found to be effective in this paper.

An effective learning strategy for cascaded object detection

The author has addressed the problem of class imbalance in the node classifier learning within the standard Viola–Jones cascade framework and proposed a new learning strategy aimed at maximizing the node classifiers ranking capability rather than their accuracy. Such an approach was revealed to be an effective solution to the class asymmetry problem and provided good performance in experiments when compared with other approaches.

- Application of MATLAB in Moving Object Detecting Algorithm

. The detection and extraction of moving object was carried out on images according to frame difference-based dynamic background refreshing algorithm. Finally, the desired video image was synthesized through adding image casing on the moving objects in the video. The results showed that using computer language Matlab to perform moving object detecting algorithm has favorable effects. From analysis and examples the author has done that the computer language Matlab has the characteristics of simple programming, easy operation and high processing rate, etc. when used in series of processing of moving object detecitng algorithm. The image processing toolkit for language Matlab has powerful functions, with which all commonly used techniques and methods in image processing can be implemented. As a result, the implementation of the algorithm becomes very fast.

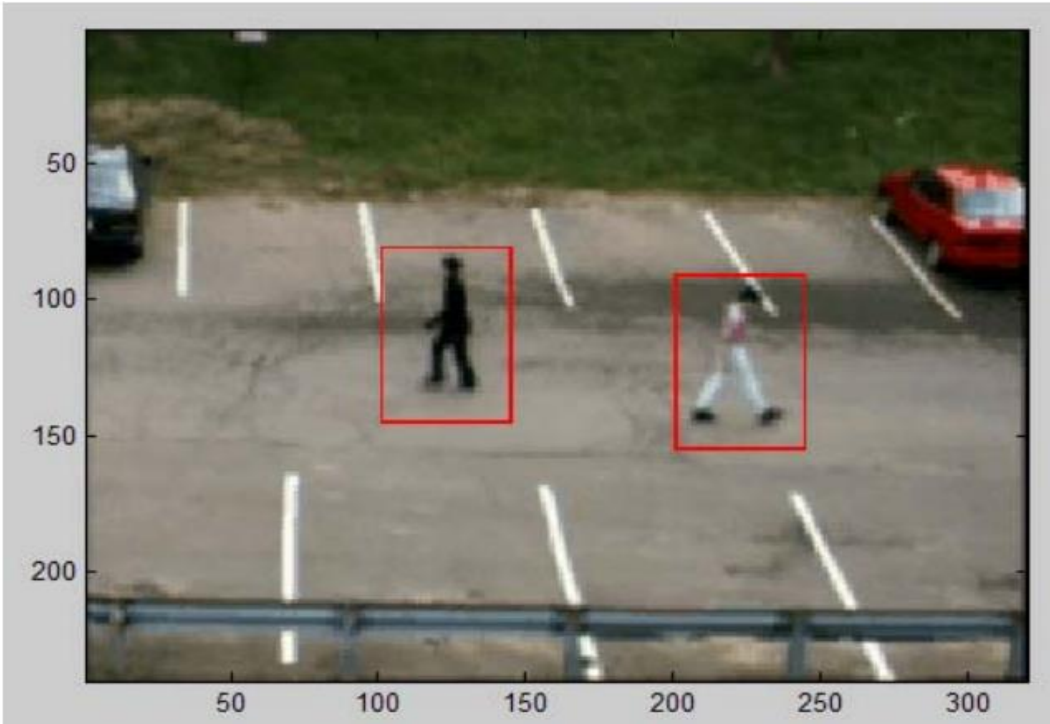

Fig.2.4: The experimental results when the motion of illegal visitors emerged in monitoring video.

CHAPTER-03

OBJECT DETECTION

<u>How to Capture any Image:</u>

```
% Create a webcam object
cam = webcam("USB2.0 HD IR UVC WebCam");

% Set the resolution of the webcam
cam.Resolution = '1280x720';

% Create a preview window
preview(cam);

% Wait for the user to press a key.
pause;

% Capture an image from the webcam
img = snapshot(cam);

% Save the image to a folder
imwrite(img, 'captured_image.jpg');

clear all
clc
```

3.1 Object Detection for cluttered image:

This method of object detection works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly colored objects or for objects containing repeating patterns.

This algorithm is designed for detecting a specific object.

an algorithm for detecting a specific object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amount of out-of-plane rotation.

So, the steps of object detection using SURF feature are:

So, the steps of object detection using SURF feature are:

Step 1: Read Images

```
I1=imread('Ketchup.JPG');
I1=rgb2gray(I1);
figure;
imshow(I1);
title('Ketchup');

```

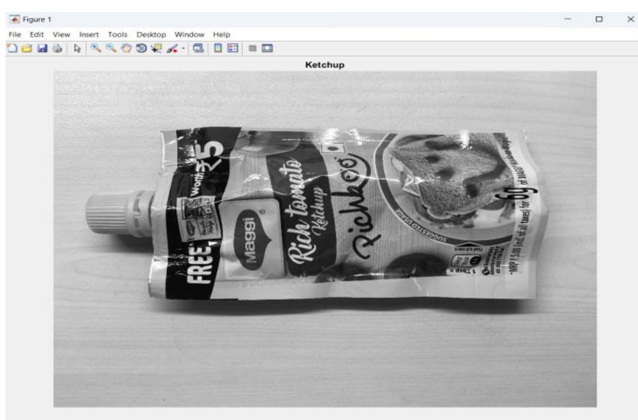

Fig 3.1.1: Image of ketchup

```
I2=imread('ClutteredImage.JPG');
I2=rgb2gray(I2);
figure;
imshow(I2);
title('Cluttered');
```

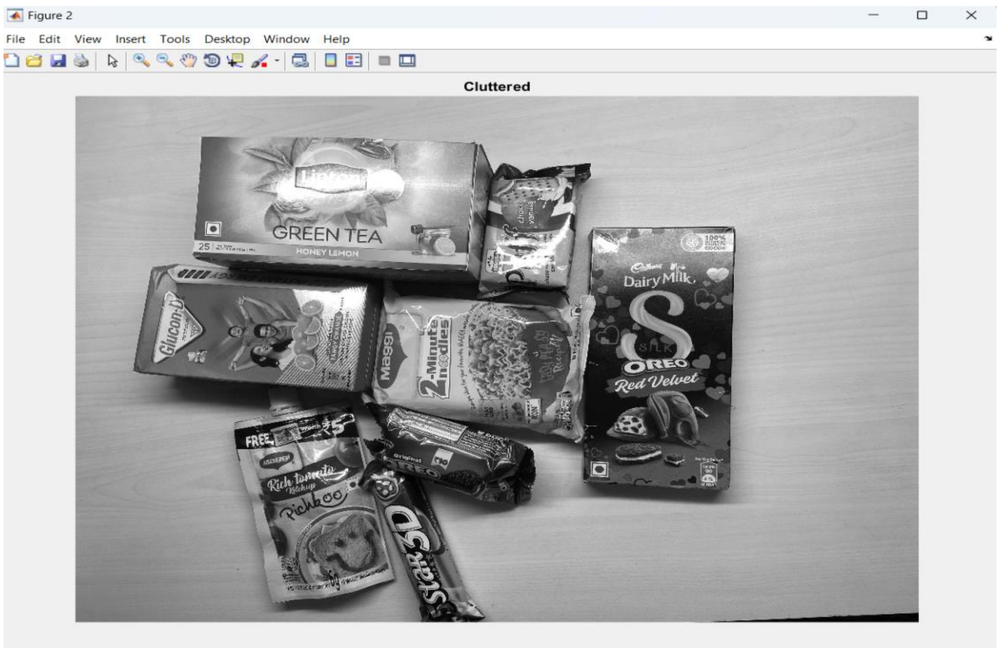

Fig 3.1.2: Cluttered image

Step 2: Detect Feature Points

```
boxPoints = detectSURFFeatures(I1);
scenePoints = detectSURFFeatures(I2);
```

```
figure;
imshow(I1);
title('100 Strongest Feature Points from Box Image');
hold on;
plot(selectStrongest(boxPoints, 100));
```

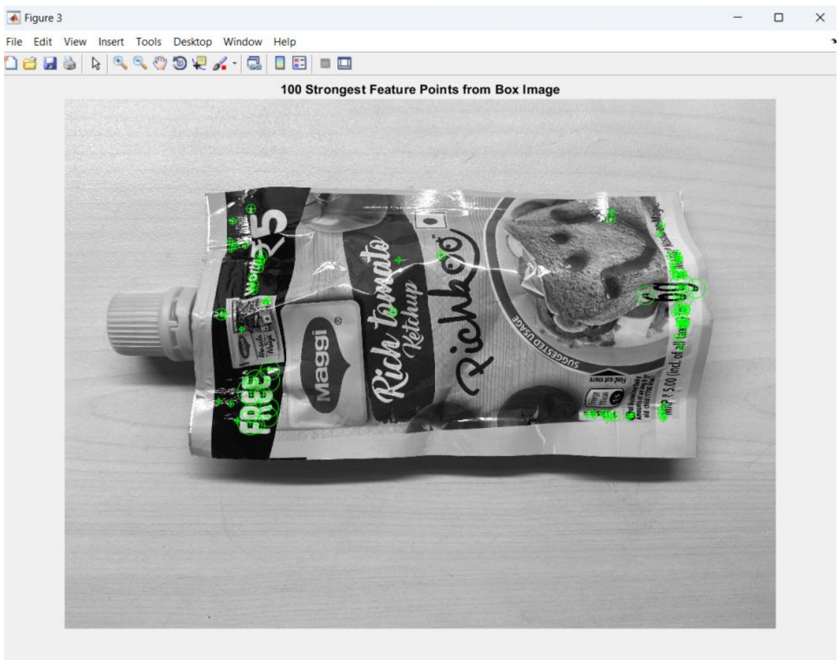

Fig 3.1.3: Feature detection

```
figure;
imshow(I2);
title('300 Strongest Feature Points from Scene Image');
hold on;
plot(selectStrongest(scenePoints, 300));
```

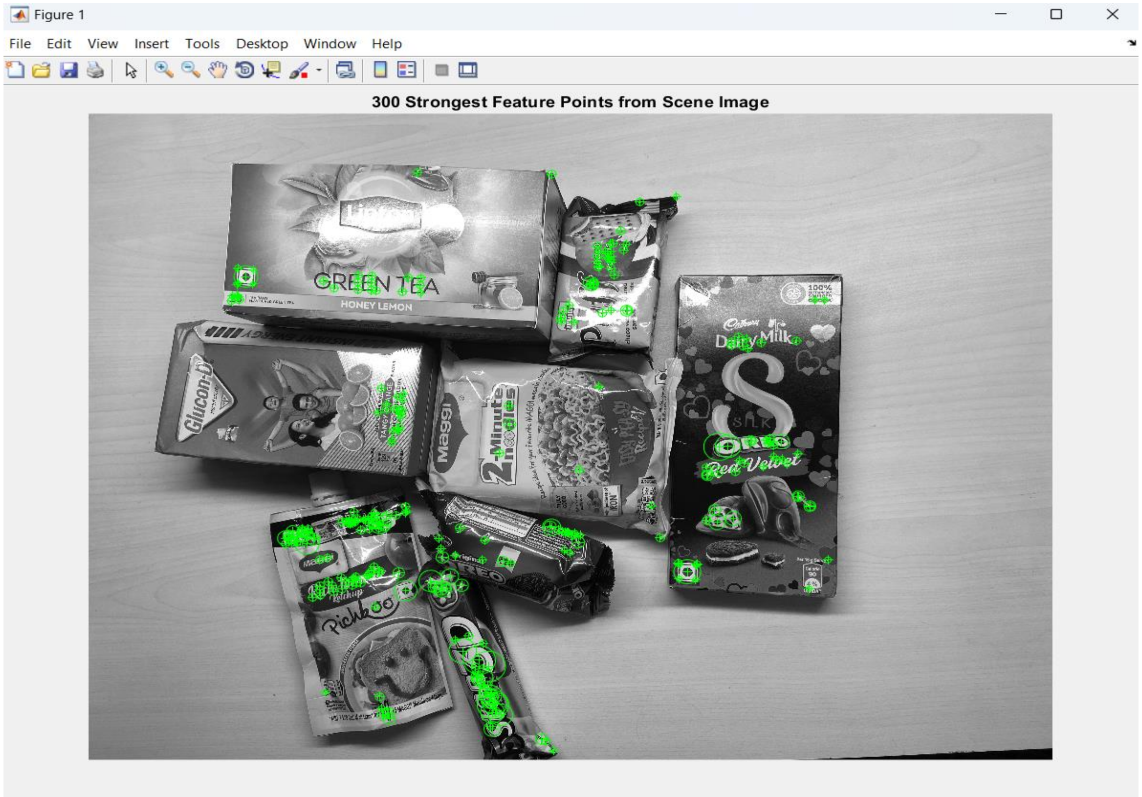

Fig 3.1.4: Feature detection of cluttered image

Step 3: Extract Feature Descriptorsand find Putative Point Matches

```
[boxFeatures, boxPoints] = extractFeatures(I1, boxPoints);
[sceneFeatures, scenePoints] = extractFeatures(I2, scenePoints);
boxPairs = matchFeatures(boxFeatures, sceneFeatures);
matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
matchedScenePoints = scenePoints(boxPairs(:, 2), :);
figure;
showMatchedFeatures(I1, I2, matchedBoxPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

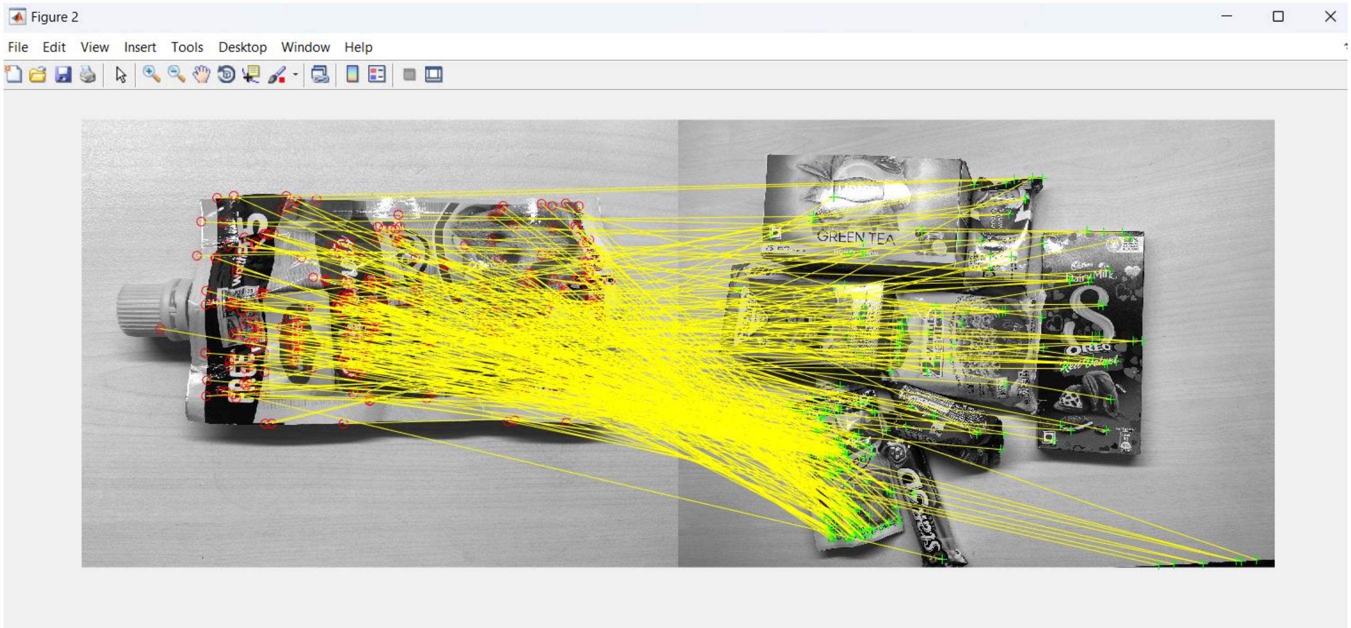

Fig 3.1.5: Matching points

Step 4: Locate the Object in the Scene Using Putative Matches

```
[tform, inlierBoxPoints, inlierScenePoints] = ...
    estimateGeometricTransform(matchedBoxPoints, matchedScenePoints, 'affine');
figure;
showMatchedFeatures(I1, I2, inlierBoxPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```

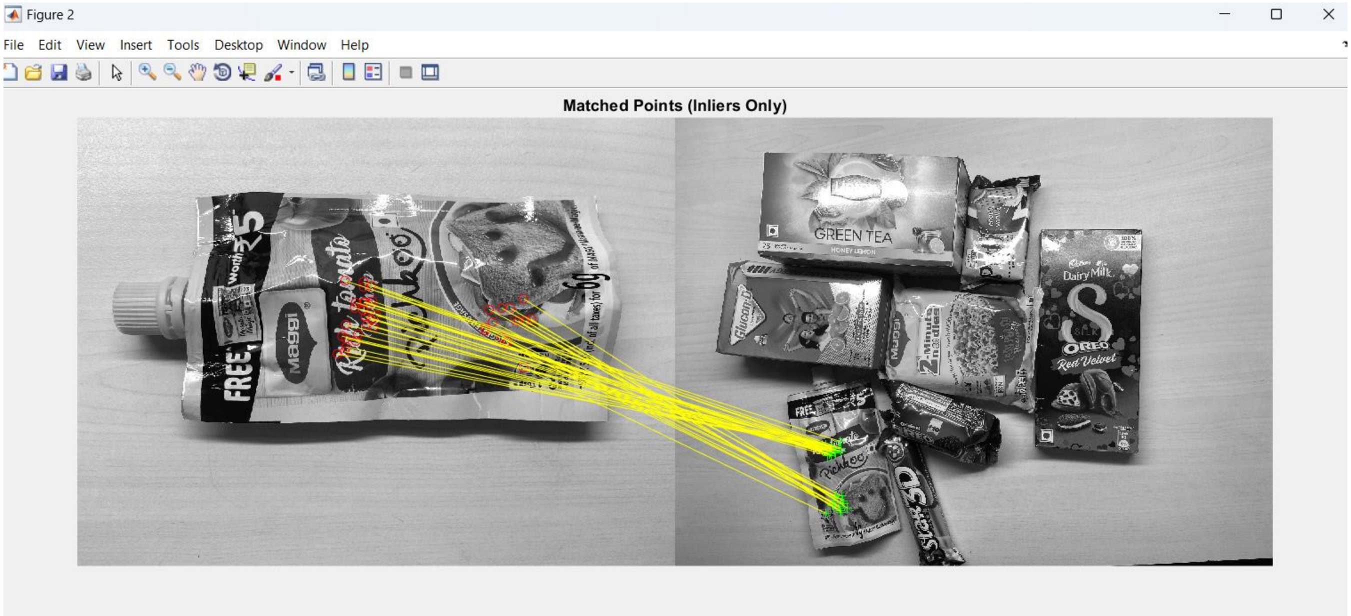

Fig 3.1.6: Matching points

```
boxPolygon = [1, 1;...                                % top-left
    size(I1, 2), 1;...                         % top-right
    size(I1, 2), size(I1, 1);... % bottom-right
    1, size(I1, 1);...                         % bottom-left
    1, 1];                     % top-left again to close the polygon
newBoxPolygon = transformPointsForward(tform, boxPolygon);
```

```
figure;
imshow(I2);
hold on;
line(newBoxPolygon(:, 1), newBoxPolygon(:, 2), 'Color', 'y');
title('Detected Box');
```

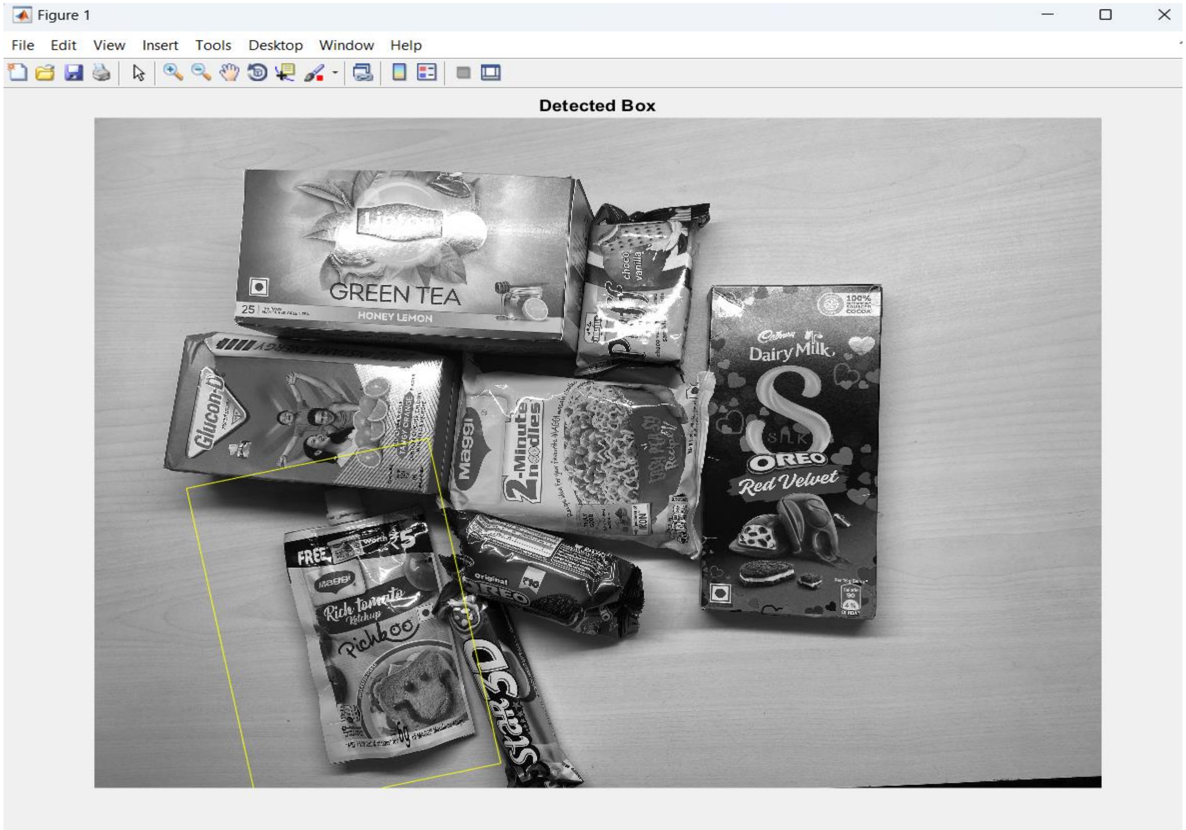

Fig 3.1.7: Detected item

Step 5: Detect another object

```
I3= imread('Maggie.JPG');
I3=rgb2gray(I3);
figure;
imshow(I3);
title('Maggie');
```

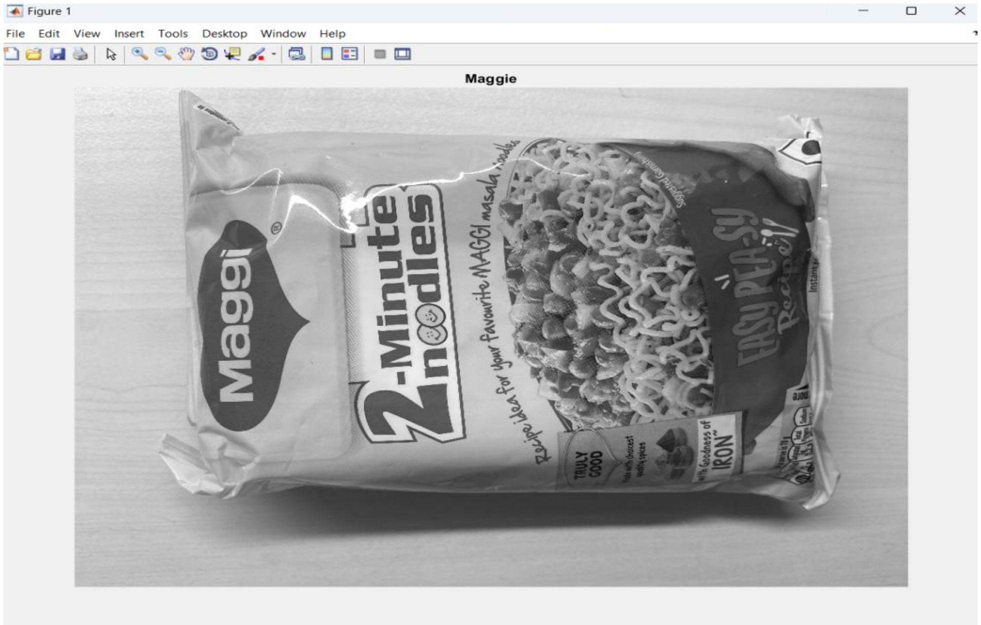

Fig 3.1.8: Image of Maggie

```
MaggiePoints = detectSURFFeatures(I3);
figure;
imshow(I3);
hold on;
plot(selectStrongest(MaggiePoints, 100));
title('100 Strongest Feature Points from Maggie Image');
```

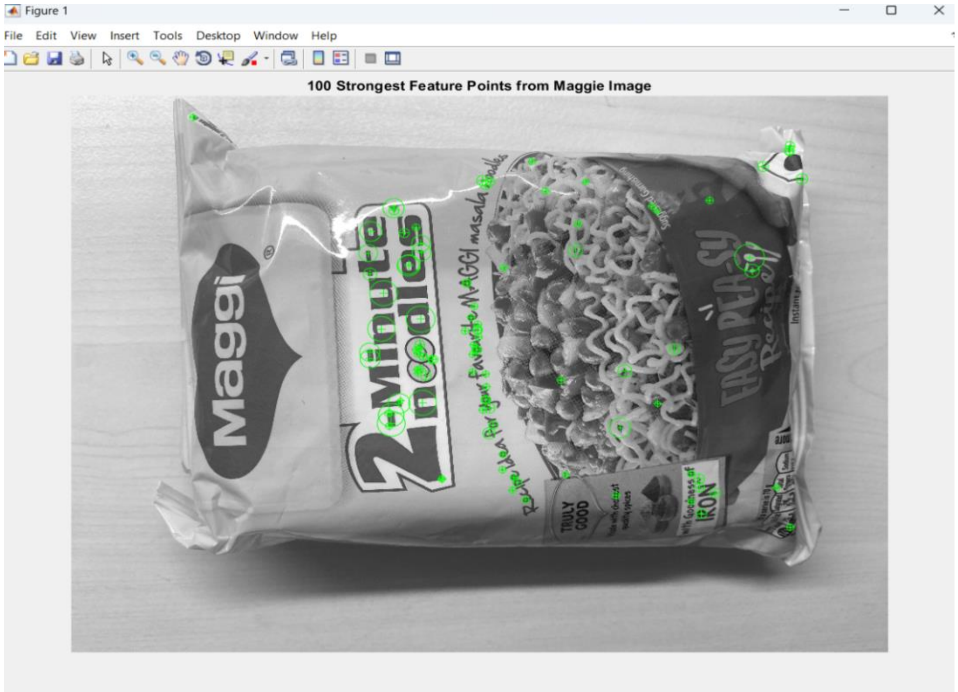

Fig 3.1.9: Featuring point

```
[MaggieFeatures, MaggiePoints] = extractFeatures(I3, MaggiePoints);
MaggiePairs = matchFeatures(MaggieFeatures, sceneFeatures, 'MaxRatio', 0.9);
matchedMaggiePoints = MaggiePoints(MaggiePairs(:, 1), :);
matchedScenePoints = scenePoints(MaggiePairs(:, 2), :);
figure;
showMatchedFeatures(I3, I2, matchedMaggiePoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

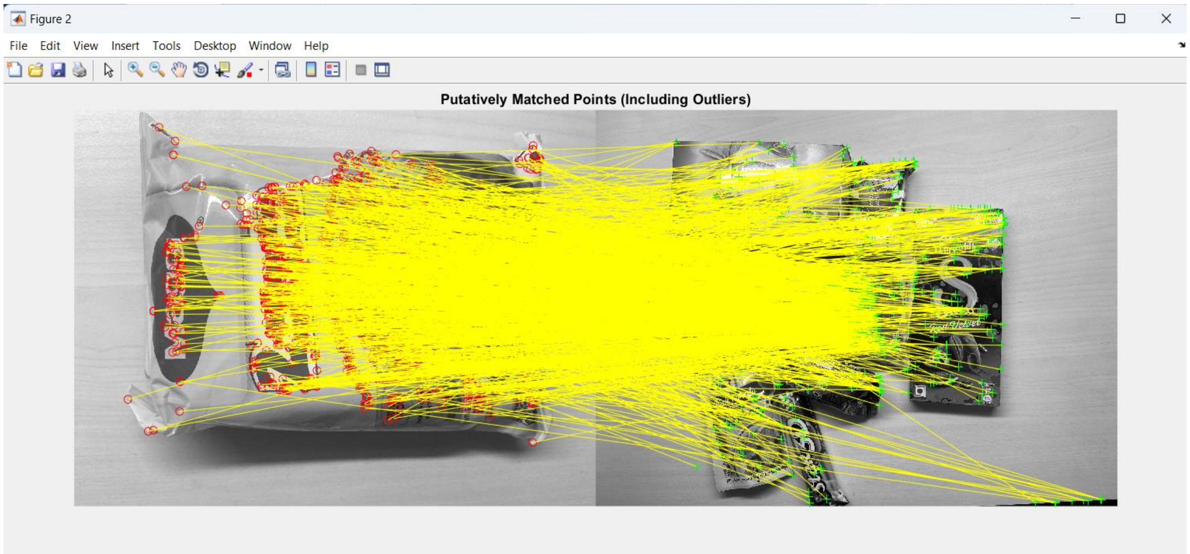

Fig 3.1.10: Matching Points

```
[tform, inlierMaggiePoints, inlierScenePoints] = ...
    estimateGeometricTransform(matchedMaggiePoints, matchedScenePoints, 'affine');
figure;
showMatchedFeatures(I3, I2, inlierMaggiePoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');

```

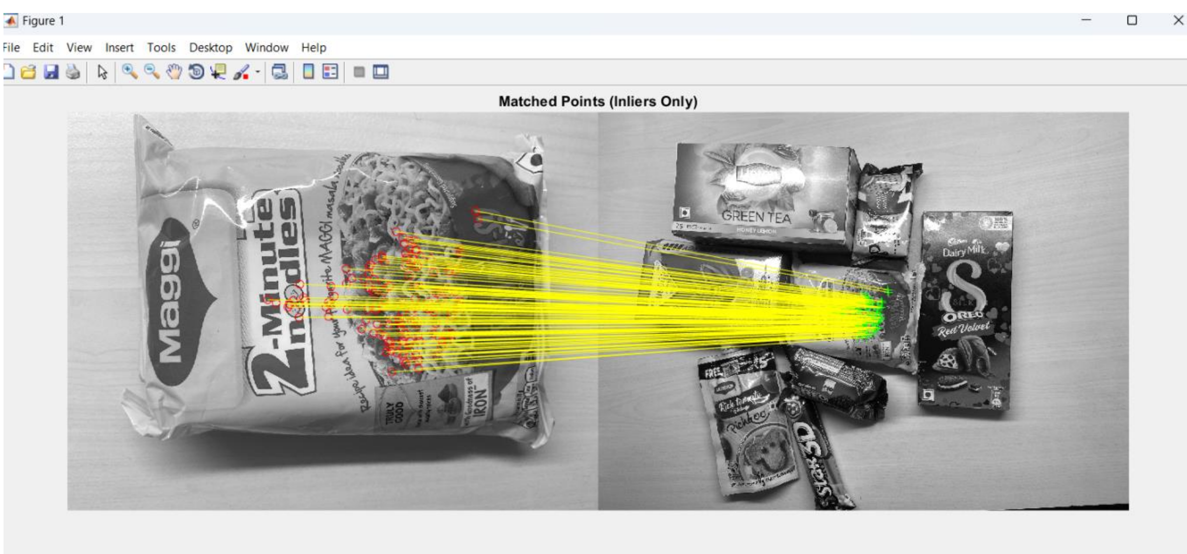

Fig 3.1.11: Matched Points

```
MaggiePolygon = [1, 1;...                                   % top-left
        size(I3, 2), 1;...                         % top-right
        size(I3, 2), size(I3, 1);...  % bottom-right
        1, size(I3, 1);...                         % bottom-left
        1,1];                         % top-left again to close the polygon

newLiptonPolygon = transformPointsForward(tform, MaggiePolygon);

figure;
imshow(I2);
hold on;
line(newBoxPolygon(:, 1), newBoxPolygon(:, 2), 'Color', 'y');
line(newLiptonPolygon(:, 1), newLiptonPolygon(:, 2), 'Color', 'g');
title('Detected Ketchup and Maggie');
```

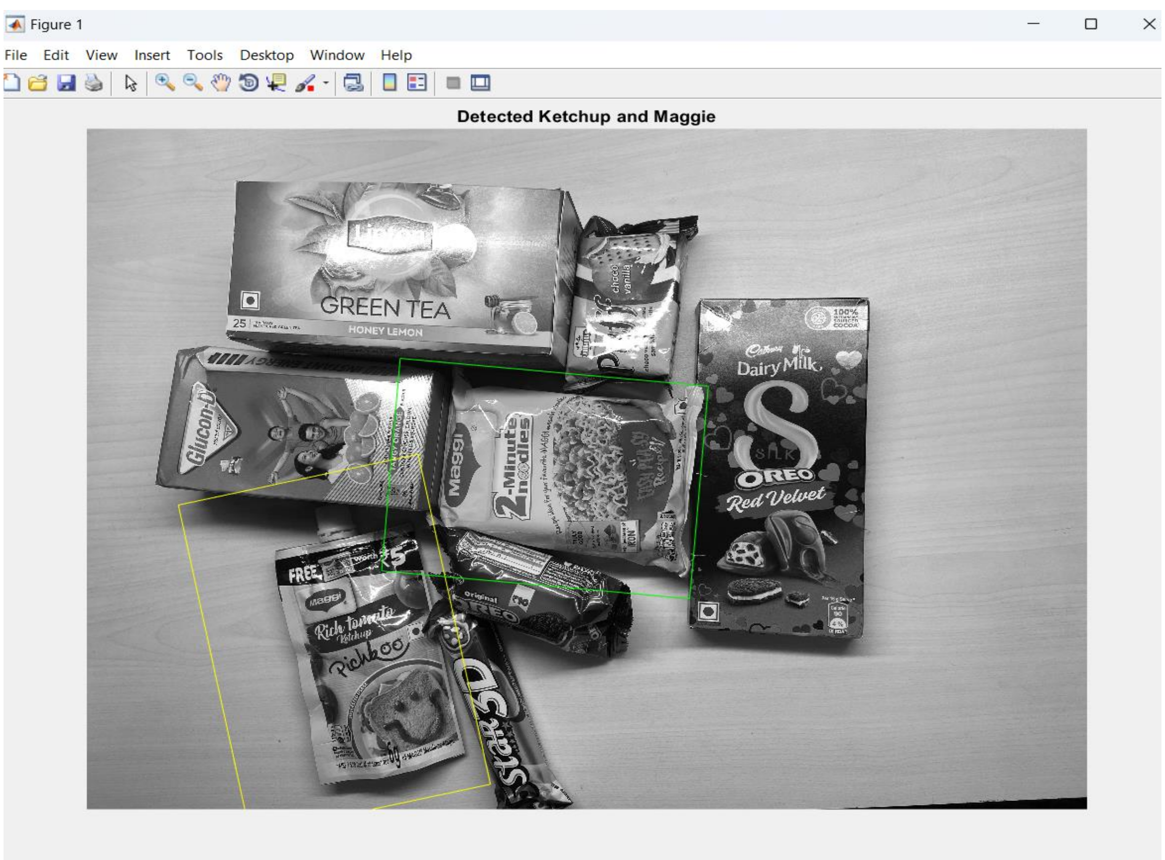

Fig 3.1.12: Detected item

Step 6: Detecting third object

```
I4 = imread('OreoSilk.JPG');
I4=rgb2gray(I4);
figure;
imshow(I4);
title('Oreo Chocolate');
```

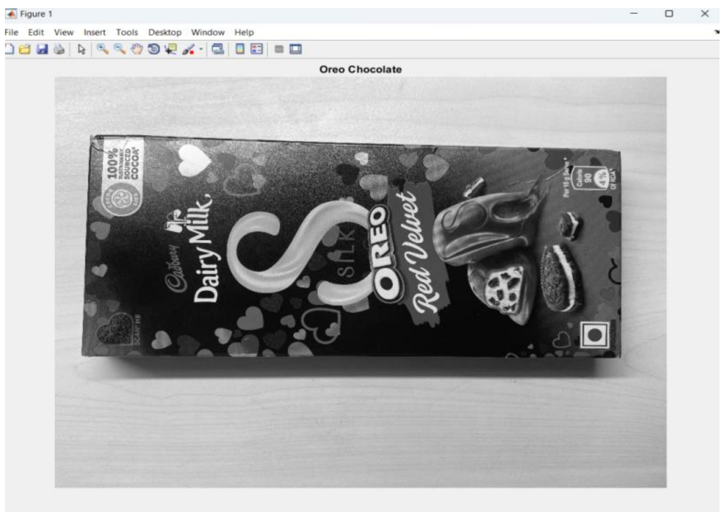

Fig 3.1.13: Image of Chocolate

```
OreoPoints = detectSURFFeatures(I4);
figure;
imshow(I4);
hold on;
plot(selectStrongest(OreoPoints, 100));
title('100 Strongest Feature Points from Oreo Image');
```

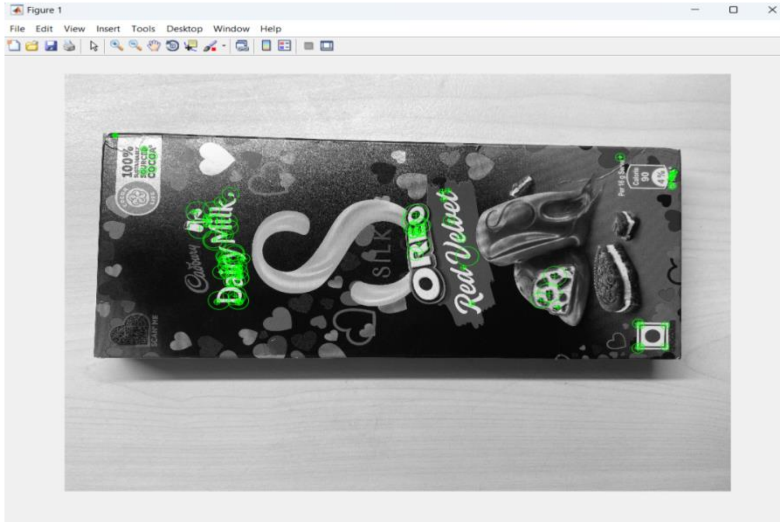

Fig 3.1.14: Featuring points

```
[OreoFeatures, OreoPoints] = extractFeatures(I4, OreoPoints);
OreoPairs = matchFeatures(OreoFeatures, sceneFeatures, 'MaxRatio', 0.9);
matchedOreoPoints = OreoPoints(OreoPairs(:, 1), :);
matchedScenePoints = scenePoints(OreoPairs(:, 2), :);
figure;
showMatchedFeatures(I4, I2, matchedOreoPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

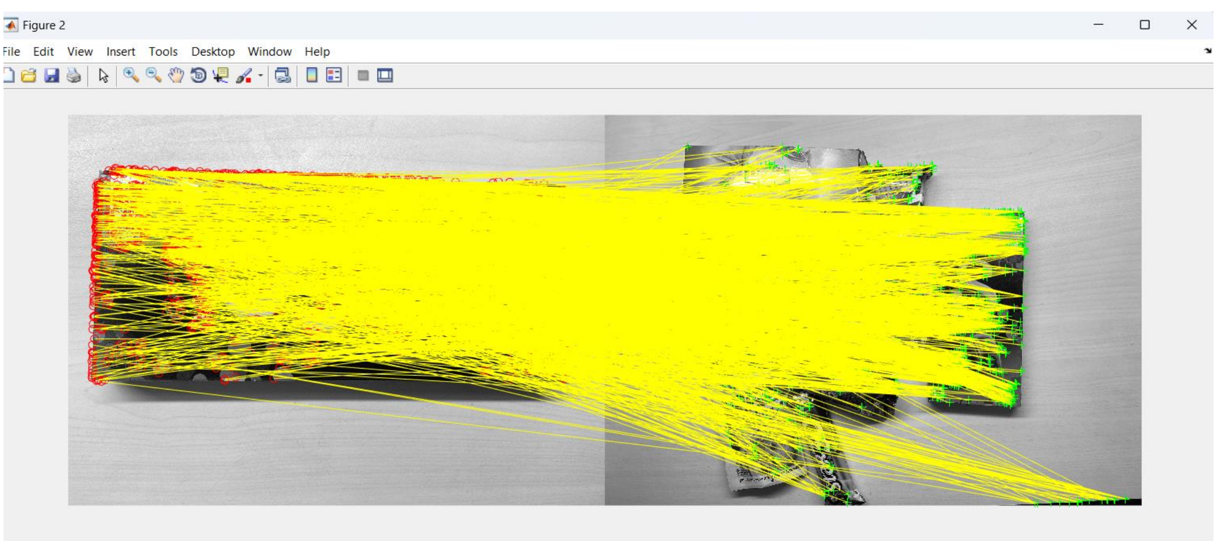

Fig 3.1.15: Matching points

```
[tform, inlierOreoPoints, inlierScenePoints] = ...
    estimateGeometricTransform(matchedOreoPoints, matchedScenePoints, 'affine');
figure;
showMatchedFeatures(I4, I2, inlierOreoPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```

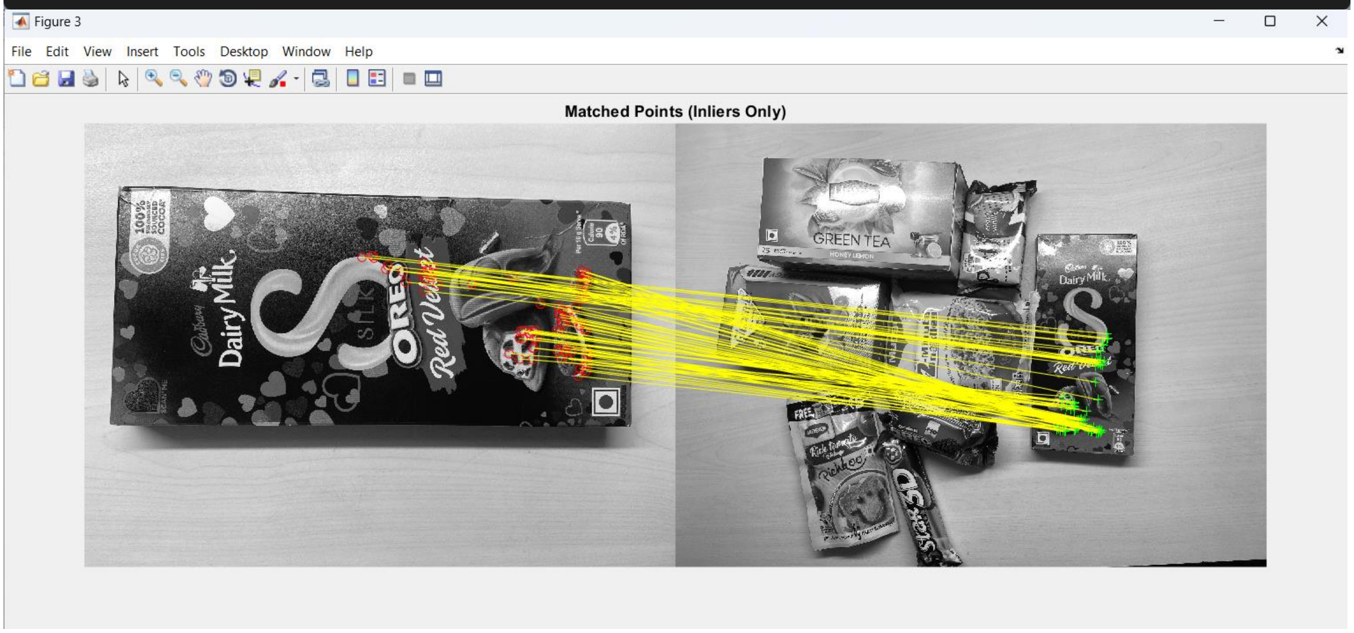

Fig 3.1.16: Matched points

```
OreoPolygon = [1, 1;...                              % top-left
    size(I4, 2), 1;...                          % top-right
    size(I4, 2), size(I4, 1);...   % bottom-right
    1, size(I4, 1);...                          % bottom-left
    1,1];                          % top-left again to close the polygon

neworeoPolygon = transformPointsForward(tform,OreoPolygon);
```

```
figure;
imshow(I2);
hold on;
line(newBoxPolygon(:, 1), newBoxPolygon(:, 2), 'Color', 'y');
line(newLiptonPolygon(:, 1), newLiptonPolygon(:, 2), 'Color', 'g');
line(neworeoPolygon(:, 1), neworeoPolygon(:, 2), 'Color', 'r');

title('Detected Ketchup and Maggie and Oreo Choclate');

```

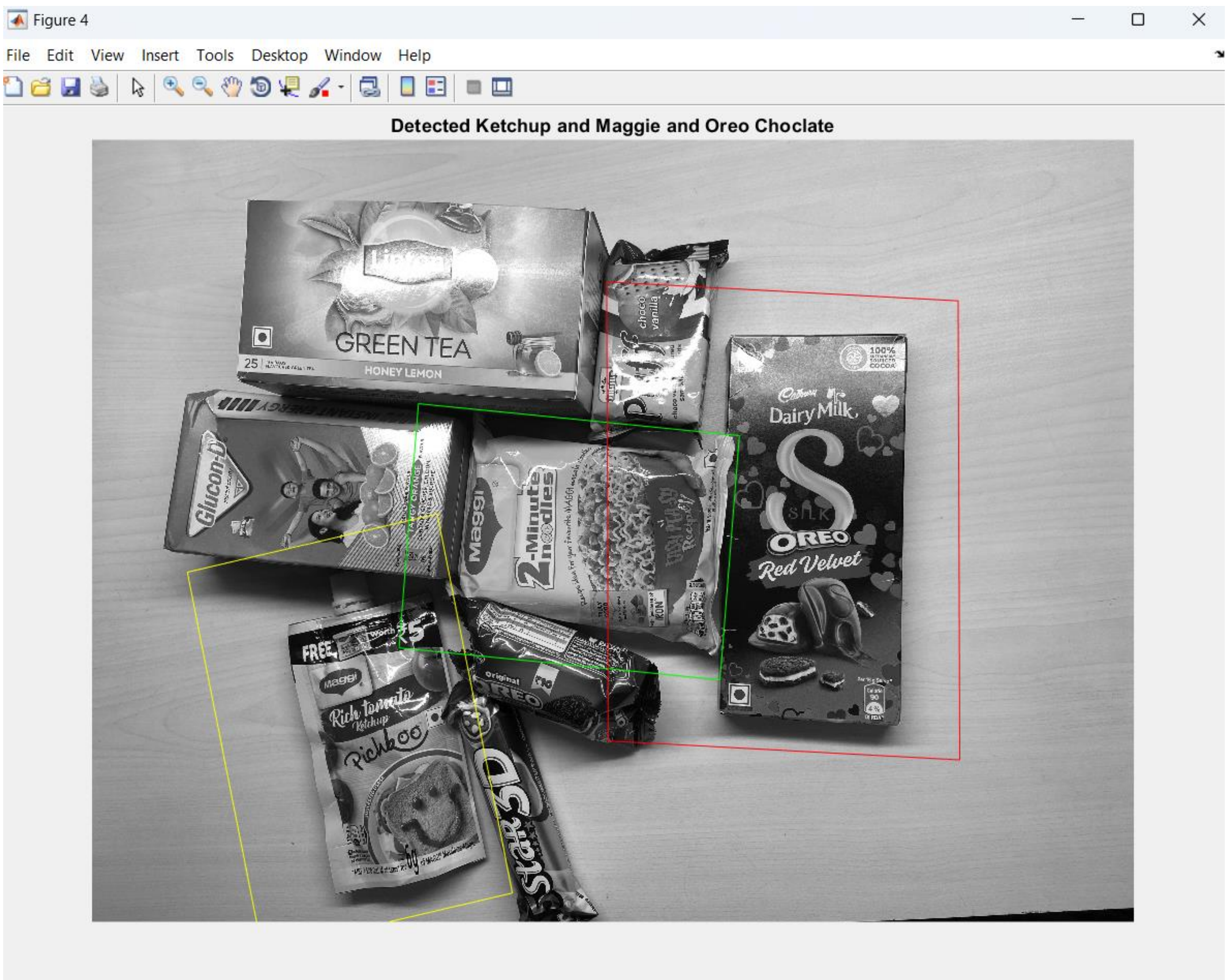

Fig 3.1.17: Detected item

3.2 Real time object detection

Real-Time Object Detection is a computer vision task that involves identifying and locating objects of interest in real-time video sequences with fast inference while maintaining a base level of accuracy. This is typically solved using algorithms that combine object detection and tracking techniques to accurately detect and track objects in real-time. They use a combination of feature extraction, object proposal generation, and classification to detect and localize objects of interest.

We have used iVCam for this real time detection program in mobile and just connected it with the laptop.

<u>Code-</u>

```
%% Detect Feature Points on All Item to be detected
clc
clear all
close all

I1 = rgb2gray(imread('xpert.JPG'));
points1 = detectSURFFeatures(I1);
[f1,vpts1] = extractFeatures(I1,points1);

I2 = rgb2gray(imread('seasoning.JPG'));
points2 = detectSURFFeatures(I2);
[f2,vpts2] = extractFeatures(I2,points2);

I3 = rgb2gray(imread('passpass.JPG'));
points3 = detectSURFFeatures(I3);
[f3,vpts3] = extractFeatures(I3,points3);

I4 = rgb2gray(imread('kitkat.JPG'));
points4 = detectSURFFeatures(I4);
[f4,vpts4] = extractFeatures(I4,points4);

I5 = rgb2gray(imread('khakhara.JPG'));
points5 = detectSURFFeatures(I5);
[f5,vpts5] = extractFeatures(I5,points5);

```

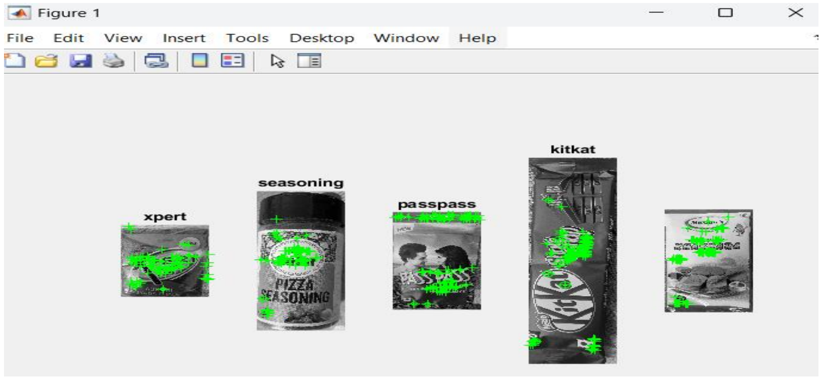

Fig3.2.1.- Detected feature points

```
% Create a new figure
figure;
subplot(1,5,1);
imshow(I1);
hold on;
plot(selectStrongest(points1, 100));
title('xpert');

subplot(1,5,2);
imshow(I2);
hold on;
plot(selectStrongest(points2, 100));
title('seasoning');

subplot(1,5,3);
imshow(I3);
hold on;
plot(selectStrongest(points3, 100));
title('passpass');

subplot(1,5,4);
imshow(I4);
hold on;
plot(selectStrongest(points4, 100));
title('kitkat');

subplot(1,5,5);
imshow(I5);
hold on;
plot(selectStrongest(points5, 100));
title('khakhara');
```

```
%% Select Image to detect
I = rgb2gray(imread('kitkat.JPG'));
points = detectSURFFeatures(I);
[f,vpts] = extractFeatures(I,points);

figure;
imshow(I);
hold on;
plot(selectStrongest(points, 100));
title('Item to be Detected');
```

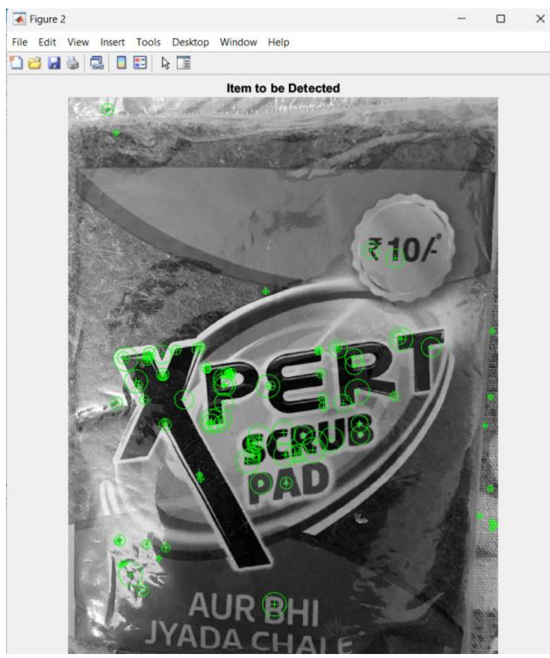

Fig3.2.2- Item to be detected

```
%% Start Real-Time Video
cam = webcam("e2eSoft iVCam");

duration = 10;  % Set the duration of the recording to 10 seconds
tic;
while toc < duration
 frame=snapshot(cam);
 I2=rgb2gray(frame);

 points2 = detectSURFFeatures(I2);
 [f2,vpts2] = extractFeatures(I2,points2);
 indexPairs = matchFeatures(f1,f2) ;
 matchedPoints1 = vpts1(indexPairs(:,1));
 matchedPoints2 = vpts2(indexPairs(:,2));

 figure(1);
 subplot(1,1,1)
 showMatchedFeatures(I1,I2,matchedPoints1,matchedPoints2,'montage');
end
```

Applying the condition-

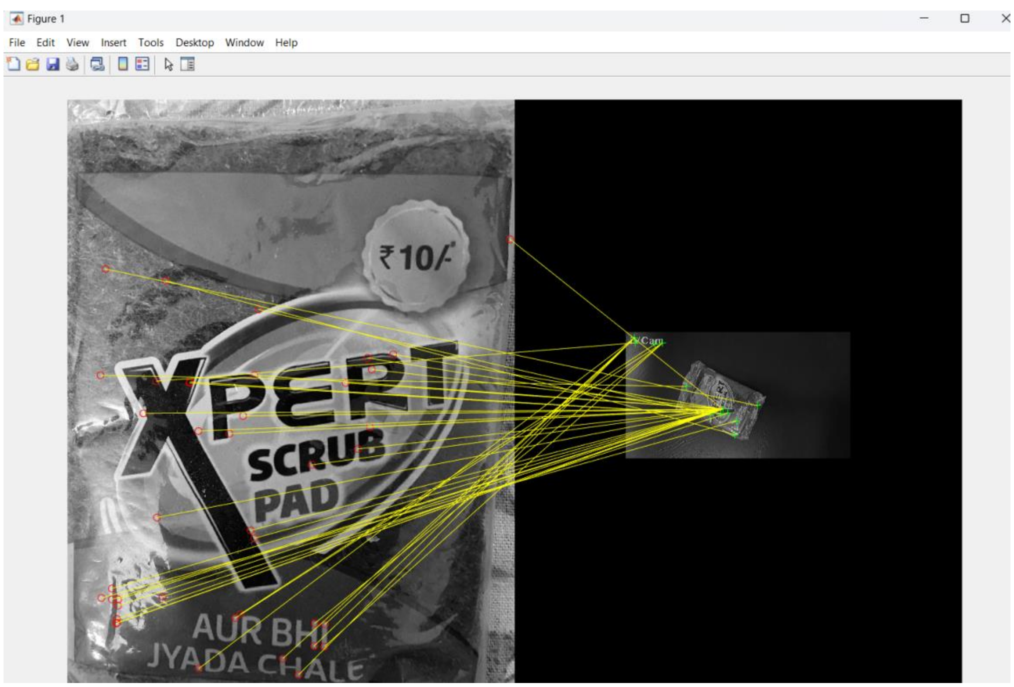

Fig-3.2.3Detecting the features

```
%% Matched Item
 I_n=rgb2gray(frame);
 points_n = detectSURFFeatures(I_n);
 [f_n,vpts_n] = extractFeatures(I_n,points_n);
 indexPairs = matchFeatures(f,f_n) ;
 matchedPoints = vpts(indexPairs(:,1));
 matchedPoints_n = vpts_n(indexPairs(:,2));

 [tform, inlierIdx] = estgeotform2d(matchedPoints, matchedPoints_n, 'affine');
 inlierItemPoints   = matchedPoints(inlierIdx, :);
 inlierScenePoints = matchedPoints_n(inlierIdx, :);

 figure;
 showMatchedFeatures(I, I_n, inlierItemPoints, inlierScenePoints, 'montage');
 title('Matched Item');
```

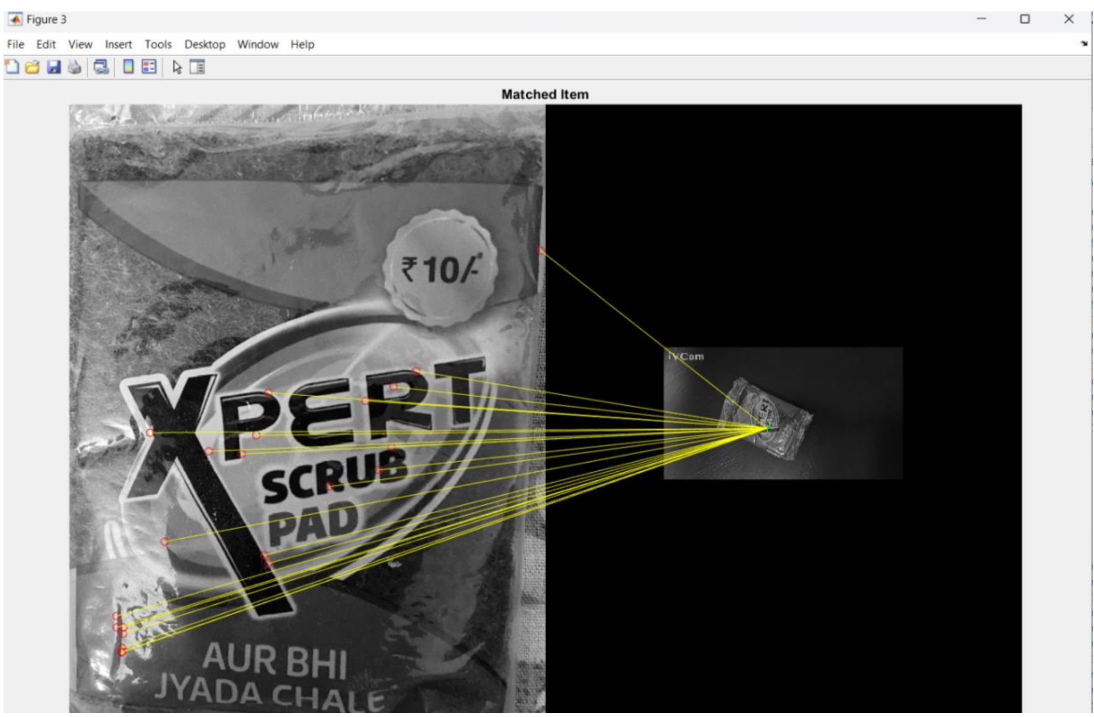

Fig3.2.4- Matched Item

CHAPTER-04

OBJECT DETECTION USING ALEXNET

Alex Net is a convolutional neural network that is 8 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database.

Alex Net has been trained on over a million images and can classify images into 1000 object categories (such as keyboard, coffee mug, pencil, and many animals). The network has learned rich feature representations for a wide range of images. The network takes an image as input and outputs a label for the object in the image together with the probabilities for each of the object categories.

Transfer learning is commonly used in deep learning applications. You can take a pretrained network and use it as a starting point to learn a new task. Fine-tuning a network with transfer learning is usually much faster and easier than training a network with randomly initialized weights from scratch. You can quickly transfer learned features to a new task using a smaller number of training images.

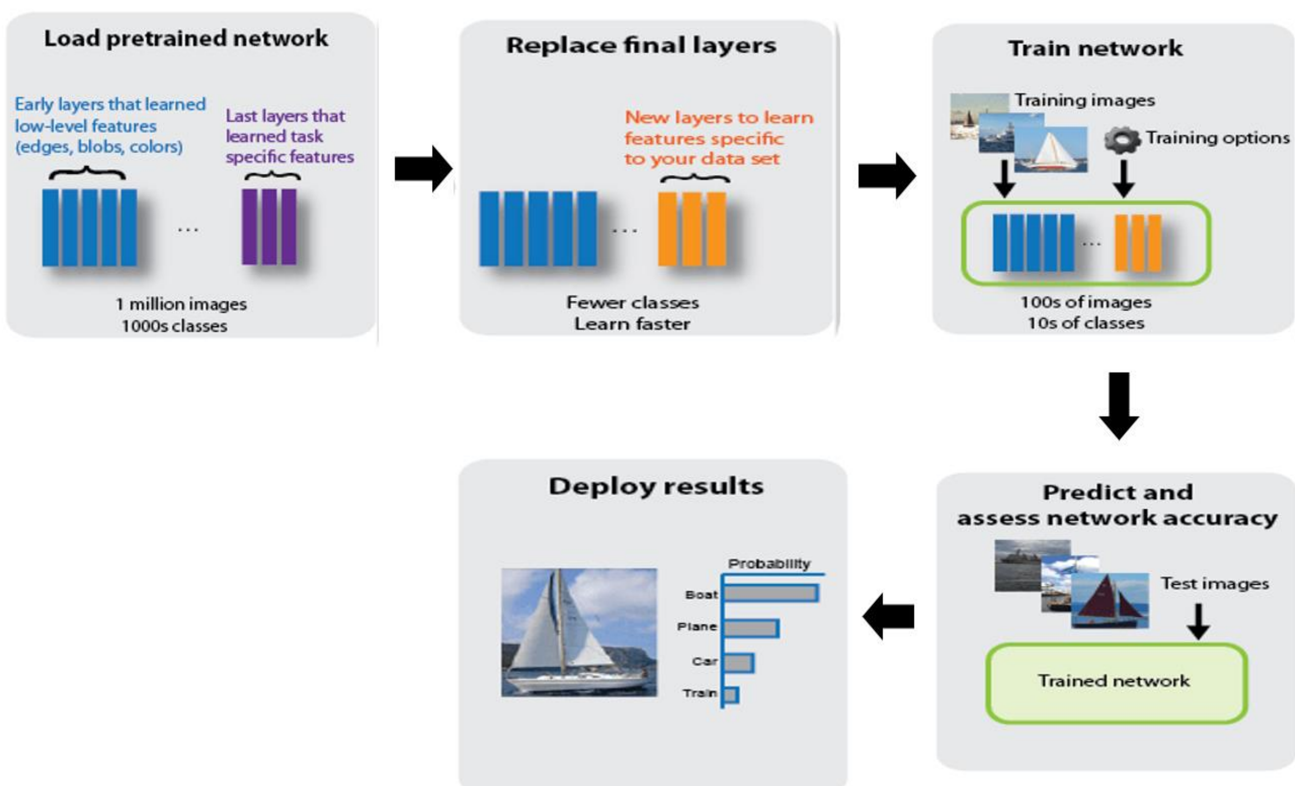

Fig4.1 Reuse Pre-trained Network

Code-

```
camera = webcam; % Connect to the camera
net = alexnet;   % Load the neural network
while true
    im = snapshot(camera);        % Take a picture
    image(im);                    % Show the picture
    im = imresize(im,[227 227]); % Resize the picture for alexnet
    label = classify(net,im);     % Classify the picture
    title(char(label));           % Show the class label
    drawnow
end
```

Image of detected object-

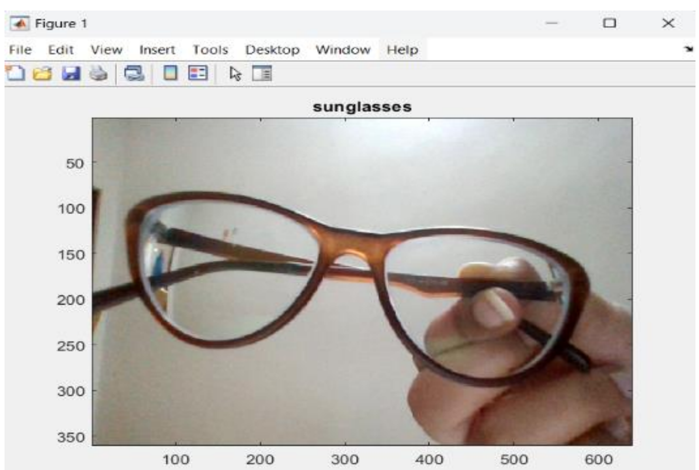

Fig 4.2: Sunglasses

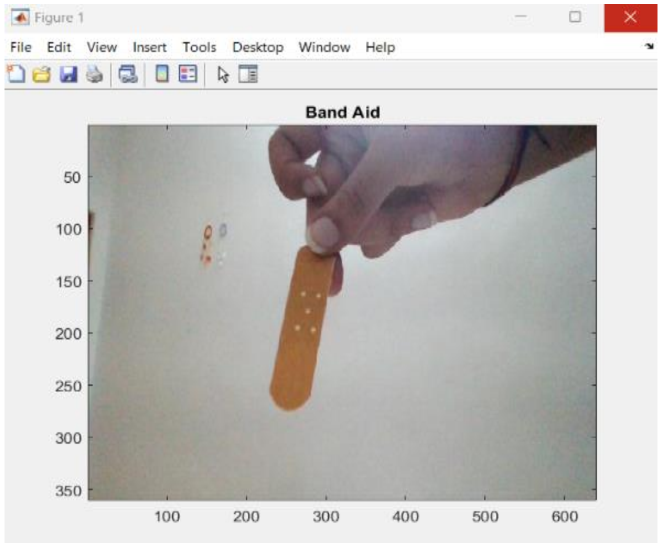

Fig4.3: Band Aid

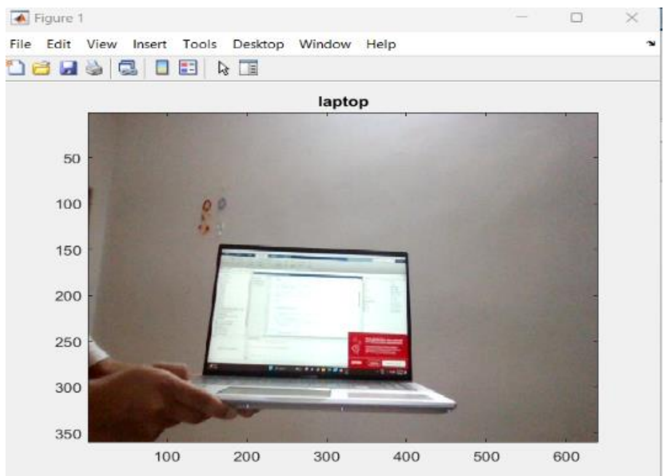

Fig4.4: Laptop

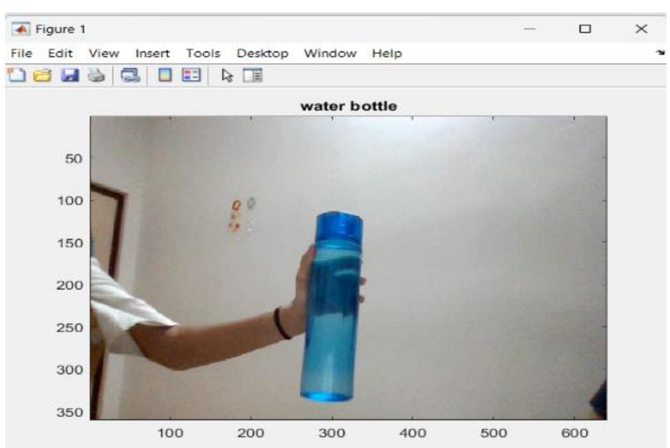

Fig 4.5: Water Bottle

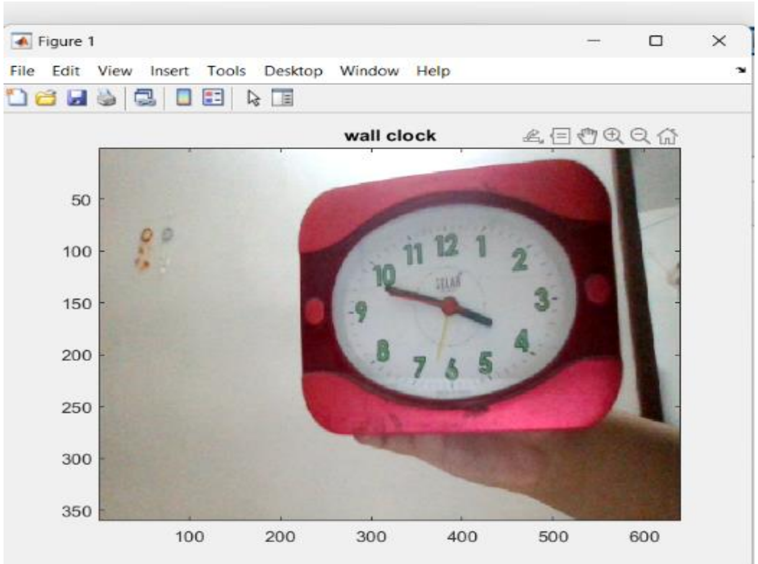

Fig4.6: Wall clock

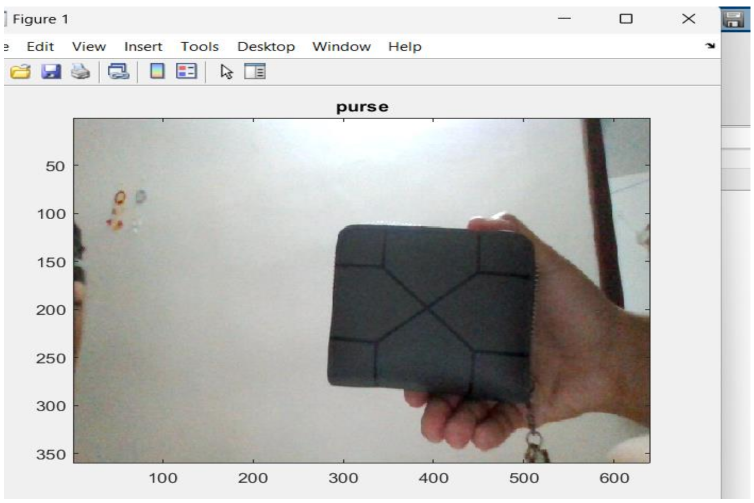

Fig 4.7:Purse

CHAPTER-5

RESULTS AND DISCUSSION

We have used a variety of MATLAB in built functions like Object Detection for cluttered image, Real time object detection and Object Detection using AlexNetbut their accuracy is not that strong. Rather than detecting the objects in every possible environment, they are only being detected in a particular environment. To overcome this problem, we are trying to develop an object detection program with better accuracy which will be able to detect in every possible environment.

This project is yet not developed completely due to many constraints as its being developed at a small level. But while developing this program at a large level we will be able to add a huge variety of databases which will make this a great success.

Here are a few glimpses of what this small setup looks like:

Fig 5: Setup

Moreover, we have tried to develop a feature in which the detected items will be stored and displayed in an Excel sheet along with their prices. However, due to a few constrains this system is not fully developed but in future, we will develop this feature completely.

This is how the excel file currently looks like:

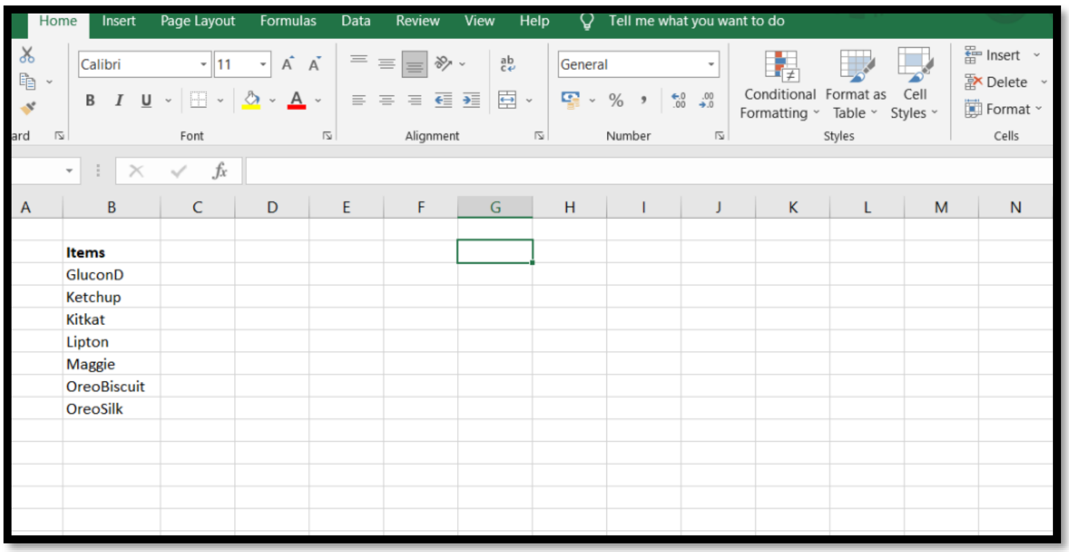

Fig 6: Excel sheet

This is how it should look like:

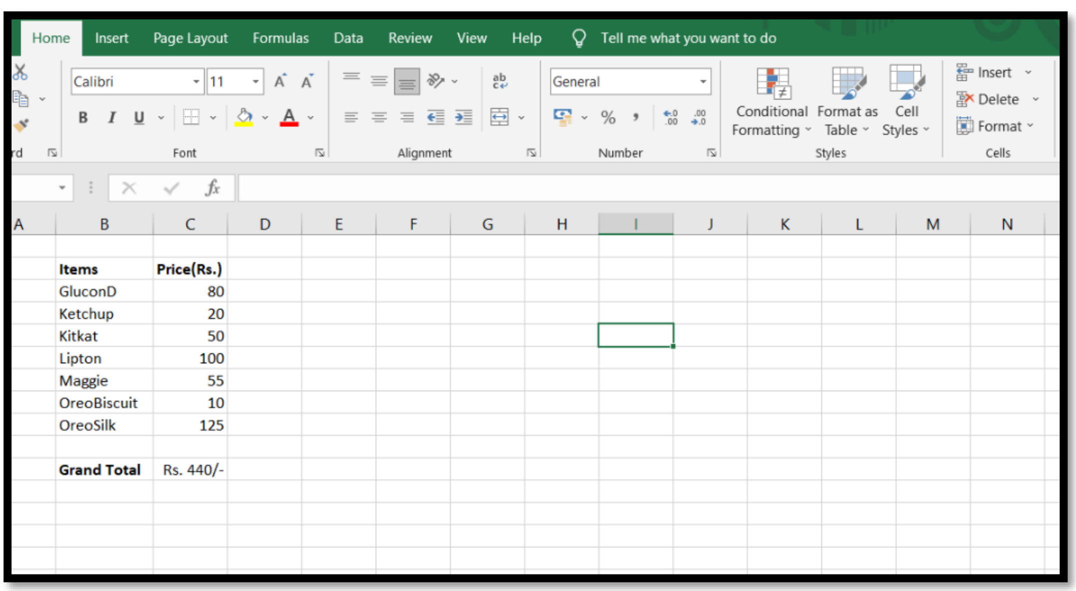

Fig 7: Excel sheet with price

CHAPTER-06

CONCLUSIONS AND FUTURE SCOPE

4.1: CONCLUSION

The improved recognition rate is a good balance for an application like billing system where accuracy of recognition is very crucial for correct billing. The real time automated billing system is advantageous over a barcode-based billing system as multiple objects can be detected simultaneously rather than laser scanning of individual objects. It also reduces human intervention. With further improvements in the system and additional features, object recognition-based billing systems can make barcodes obsolete thereby saving cost, time and human capital which is a precious resource.

In this work, we presented a review study on object detection. We have surveyed about different object detection techniques. This survey provides theoretical knowledge about different object detection approaches. Based on this study, Feature extraction using SURF has been found to be effective and is used for the detection and recognition of the glossary items.

The progression in science & technology development is an unstoppable process. Now & then evolution changing technologies are being invented. We can't imagine the upcoming future in which technology may occupy each & every place. This innovative project idea can be used is places like shopping complexes, supermarkets & malls to purchase the products.

4.2: FUTURE SCOPES:

This fascinating computer technology is related to computer vision and image processing that detects and defines objects such as grocery items, vehicles and animals from digital images and videos. Herewith are some of the main useful applications of object detection: Vehicle's Plates recognition, self-driving cars, Tracking objects, face recognition, medical imaging, object counting, object extraction from an image or video, person detection.

Shopping is easy, but waiting at the bill counter can be very boring & laborious. Rush plus cashiers who prepare a bill with a barcode scanner take longer & have longer-lasting results. This innovative project includes an automated billing system that can be placed in a shopping trolley. It provides an option to the shoppers to priorly create a shopping list. The same system can be used in various places.

The future of object detection technology is in the process of proving itself, and much like the original Industrial Revolution, it has the potential, at the very least, to free people from tedious jobs that will be done more efficiently and effectively by machines. It will also open up new avenues of research and operations that will reap additional benefits in the future. Thus, these challenges circumvent the need to a lot of training requiring a massive number of datasets to serve more nuanced tasks, with its continued evolution, along with the devices and techniques that make it possible, it could soon become the next big thing in the future.

REFERENCES

[1] E. Alionte and C. Lazar, "A practical implementation of face detection by using Matlab cascade object detector," 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 2015, pp. 785-790, doi: 10.1109/ICSTCC.2015.7321390.

[2] A. M. Chowdhury, J. Jabin, E. T. Efaz, M. Ehtesham Adnan and A. B. Habib, "Object detection and classification by cascade object training," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216377

[3] S. Mane and S. Mangale, "Moving Object Detection and Tracking Using Convolutional Neural Networks," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 1809-1813, doi: 10.1109/ICCONS.2018.8662921.

[4] M. Vashisht and B. Kumar, "A Survey Paper on Object Detection Methods in Image Processing," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-4, doi: 10.1109/ICCSEA49143.2020.9132871

[5] X. Chen, "Application of Matlab in Moving Object Detecting Algorithm," 2008 International Seminar on Future BioMedical Information Engineering, Wuhan, China, 2008, pp. 114-117, doi: 10.1109/FBIE.2008.108.

[6] https://www.isahit.com/blog/why-to-use-grayscale-conversion-during-image-processing

[7] .https://www.simplilearn.com/tutorials/matlab-tutorial/what-is-matlab-introduction-for-beginners