Islamic University of Technology

Boardbazar, Gazipur

MATH 4422

Project On

Fault Analysis of Matter Using Real Time Image Processing

Project By

Md. Ashaduzzaman Niloy

160021002

Md. Sayed Hasan Rifat

160021024

October 2018

Project Name:

Fault analysis of matters using real time image processing.

Project Description:

The project is mainly based on basic Image processing to find the accuracy of the objects of a production. The program will analyze the acquired images from live feed to detect the objects of unusual shapes amongst the desired output products.

Project Layout in Subdivisions:

- **Live feed acquisition:** Firstly we will connect a camera enabled device such as webcam, mobile camera or any Signal Lens Reflect (SLR) camera to acquire the live images of the object or the conveyer-belt. In this case we used a mobile camera and an android application, IP Webcam to broadcast the recordings to the MATLAB program.

- **Receiving the input:** The live feed can be acquired by various functions like webcam(‘ ’), ipcam(‘ ’), dCAM(‘ ’). In this case we used ipcam(‘feed address’) in the program. With this command, the recording will be available in MATLAB.

- **Processing the image:** After getting the feed, the program will snap a particular frame and will start processing the snapshot. Throughout the several steps of processing the program will simultaneously deliver particular step outputs in figures.

- **Project outcome:** After all the processing, the program will compose a figure where the faulty shaped objects will be marked precisely and accurately as well as publishing the number of existing faulty objects.

Project Codes: (as in .m file)

1) MainFile.m Code:

```
close all; clear all;
%% Access image acquisition device
vidobj=ipcam('http://192.168.0.102:8080/video');



%% Preview image
preview(vidobj);
```

```
%% Capture the image
rgb=snapshot(vidobj);
figure, imshow(rgb);


%% Background removal & object segmentation
I=rgb2gray(rgb);
background=imclose(I,strel('disk',15));
I2=imsubtract(background,I);
BW=im2bw(I2,graythresh(I2));
BW=bwareaopen(BW, 20);
BW=imclearborder(BW);
fill=imfill(BW,'holes');
figure, imshow(fill);


%% Post processing
level=thresh_tool(I2);
BW=im2bw(I2,level/255);
BW=bwareaopen(BW,20);
BW=imclearborder(BW);
fill=imfill(BW,'holes');
figure, imshow(fill);


%% Broken object identification function
[statsDefects,stats]=getStatsOnDefects(BW);


%% Labeling broken objects
figure; imshow(rgb);
hold on;
for idx=1:length(statsDefects)
    h=rectangle('Position', statsDefects(idx).BoundingBox, ...
        'LineWidth', 2);
    set(h,'EdgeColor',[0.75 0 0]);
    hold on;
end
hold off;
      qew=sprintf('Faulty Objects: %d',idx);
      msgbox(qew)


%%Clean Up
% delete(vidobj);
% clear vidobj;
% clear all;
% close all;
```

2) getStatsOnDefects.m File Code:

```
function [statsDefects,stats]=getStatsOnDefects(BW)


%%Use feature analysis to identify broken objects
[labeled, numObjects]=bwlabel(BW,8);
stats=regionprops(labeled,'Eccentricity','Area','BoundingBox');
areas=[stats.Area];
minSize=mean(areas)-0.05*std(areas);
eccentricities=[stats.Eccentricity];
idxOfDefects=find(areas<minSize & eccentricities>0.05);
statsDefects=stats(idxOfDefects);
```

3) Thresh_tool.m File Code: (A function developed by MATWORKS)

```
function [level,bw] = thresh_tool(im,cmap,defaultLevel)

.  .  .
.  .  .
.  .  .


function movex_text(h_txt,x_pos) %subfunction
FmtSpec=getappdata(get(get(h_txt,'parent'),'parent'),'FmtSpec');
msg=sprintf(FmtSpec,x_pos);
pos=get(h_txt,'position');
pos(1)=x_pos;
set(h_txt,'Position',pos,'String',msg)
end %movex_text

.  .  .
.  .  .
.  .  .

if nargout
    hhh=h;
end

end %vline (subfunction)
```

Project Breakdown:

Figure 1: Getting real time live feed

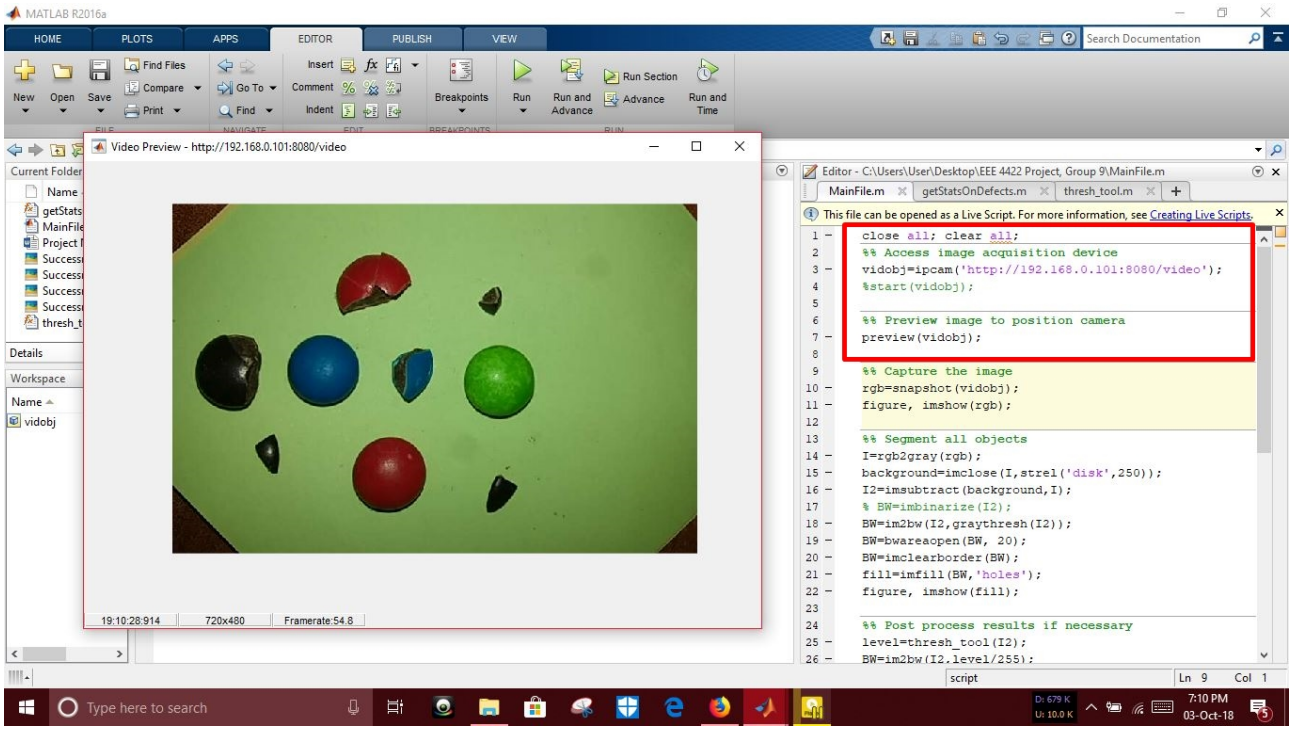


Figure 2: Capturing an instant frame from the stream

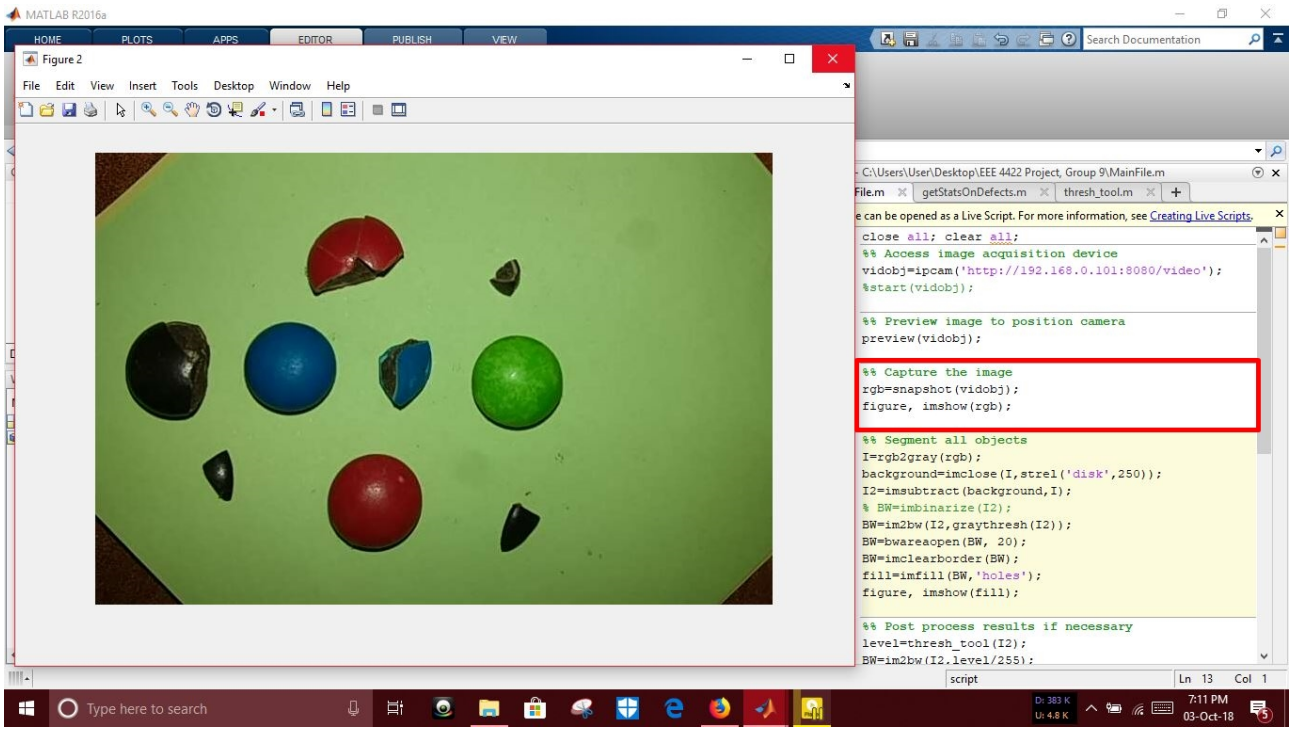


Figure 3: RGB to Grayscale Conversion

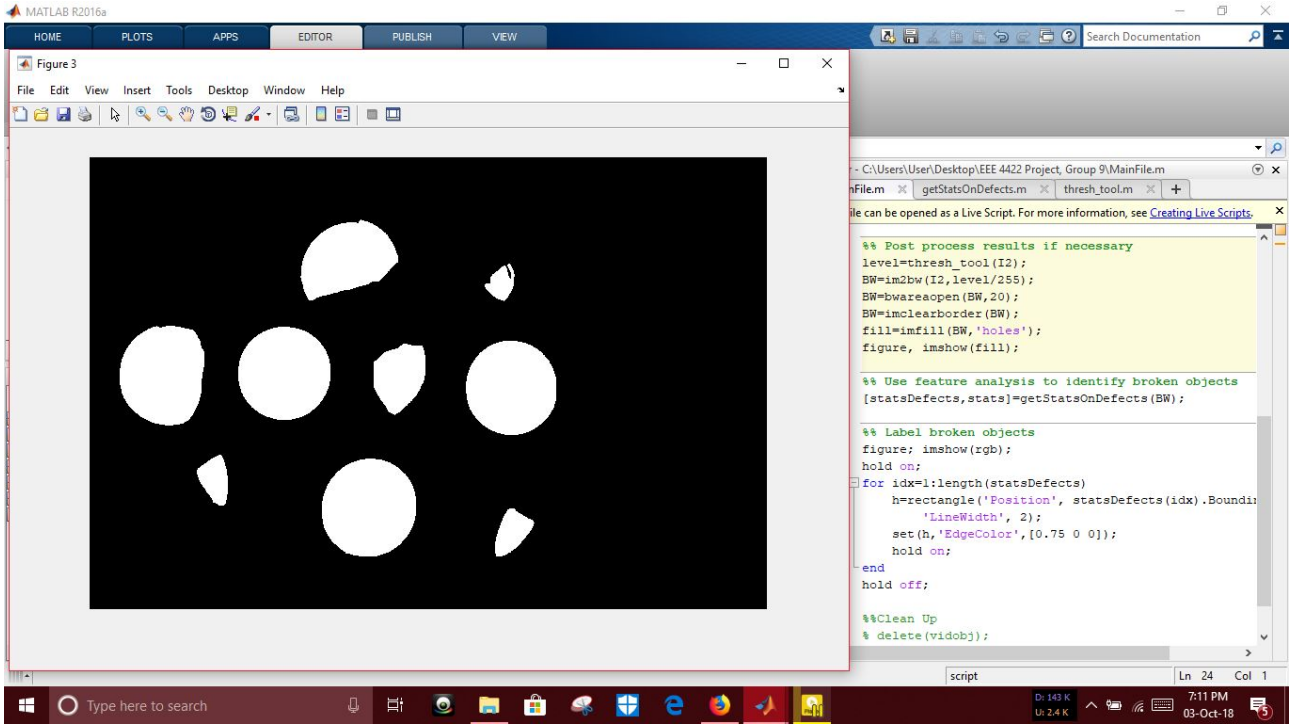


Figure 4: Using thresh tool & BW image conversion completion

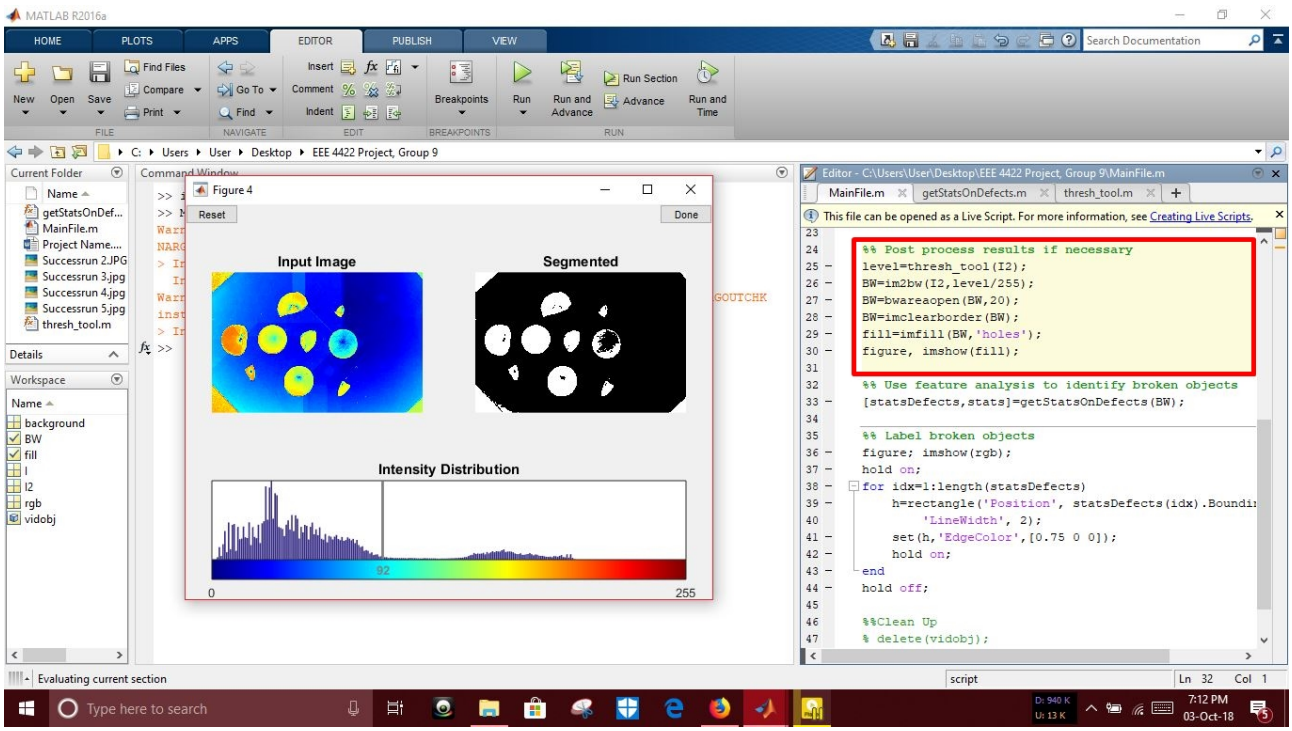


Figure 5: Getting object count

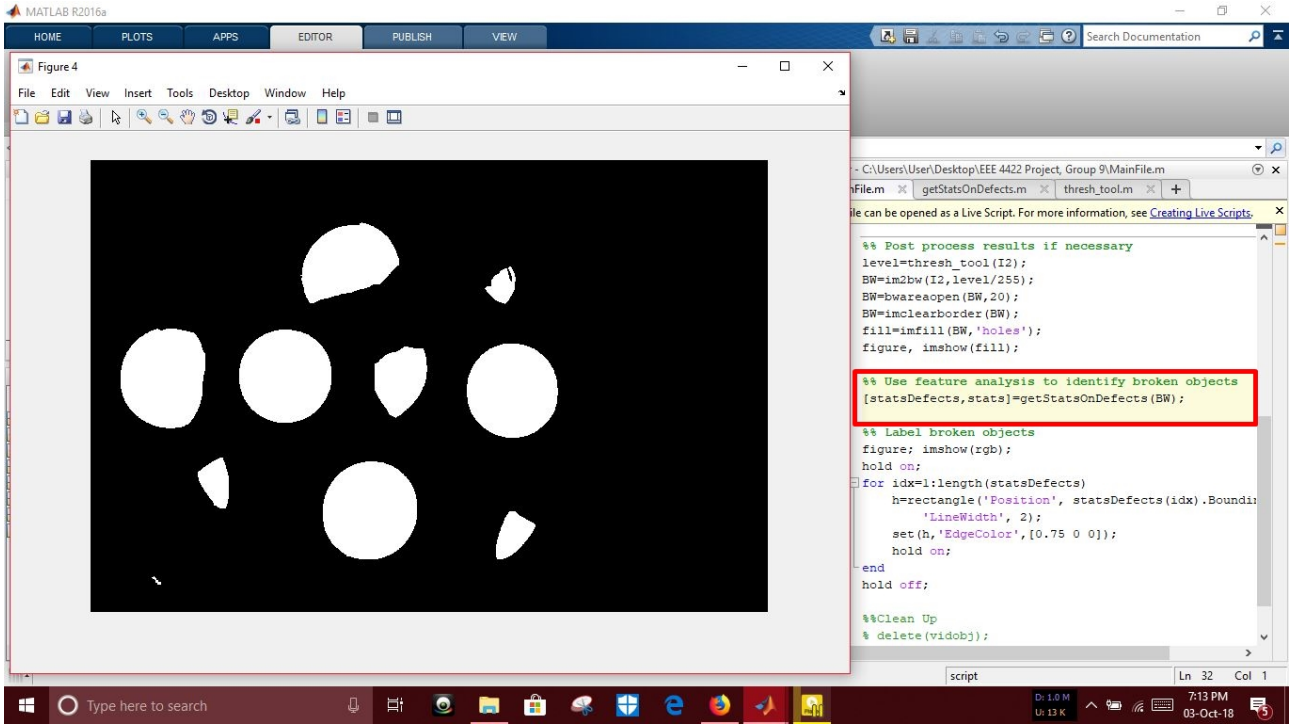


Figure 6: Marking Broken and faulty objects

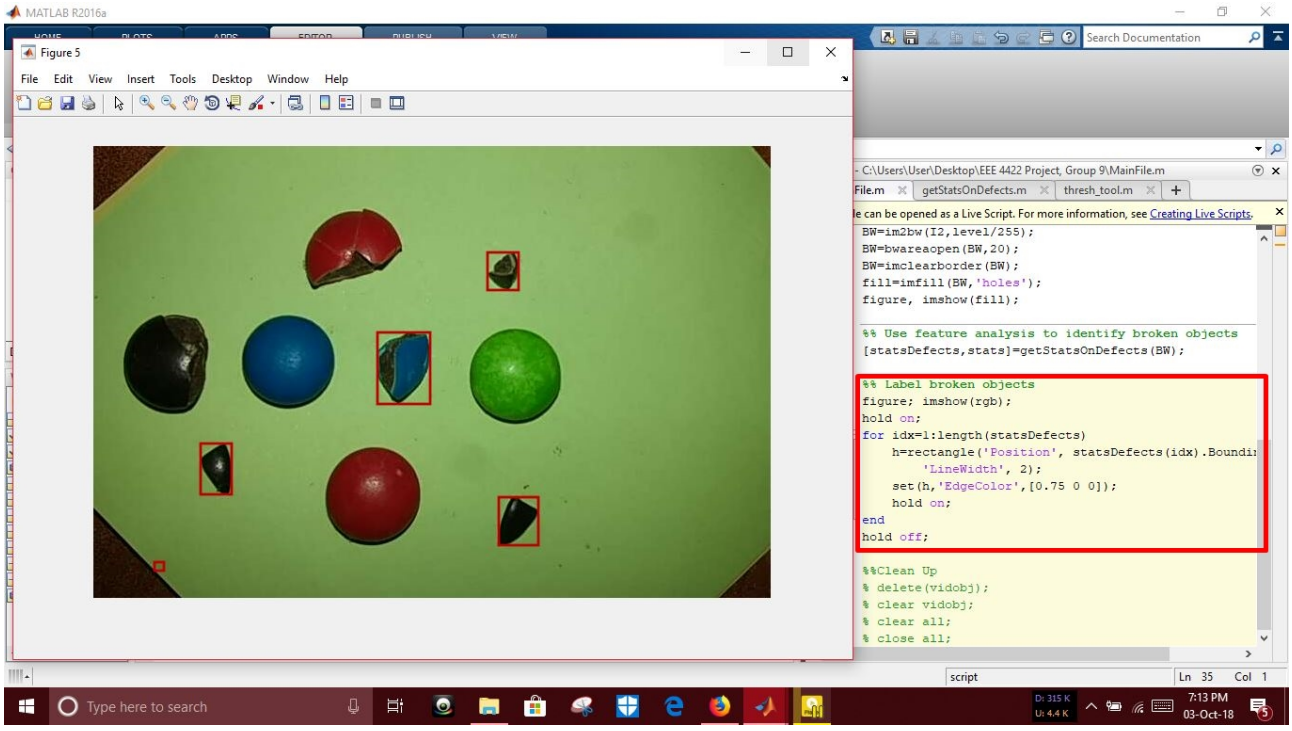
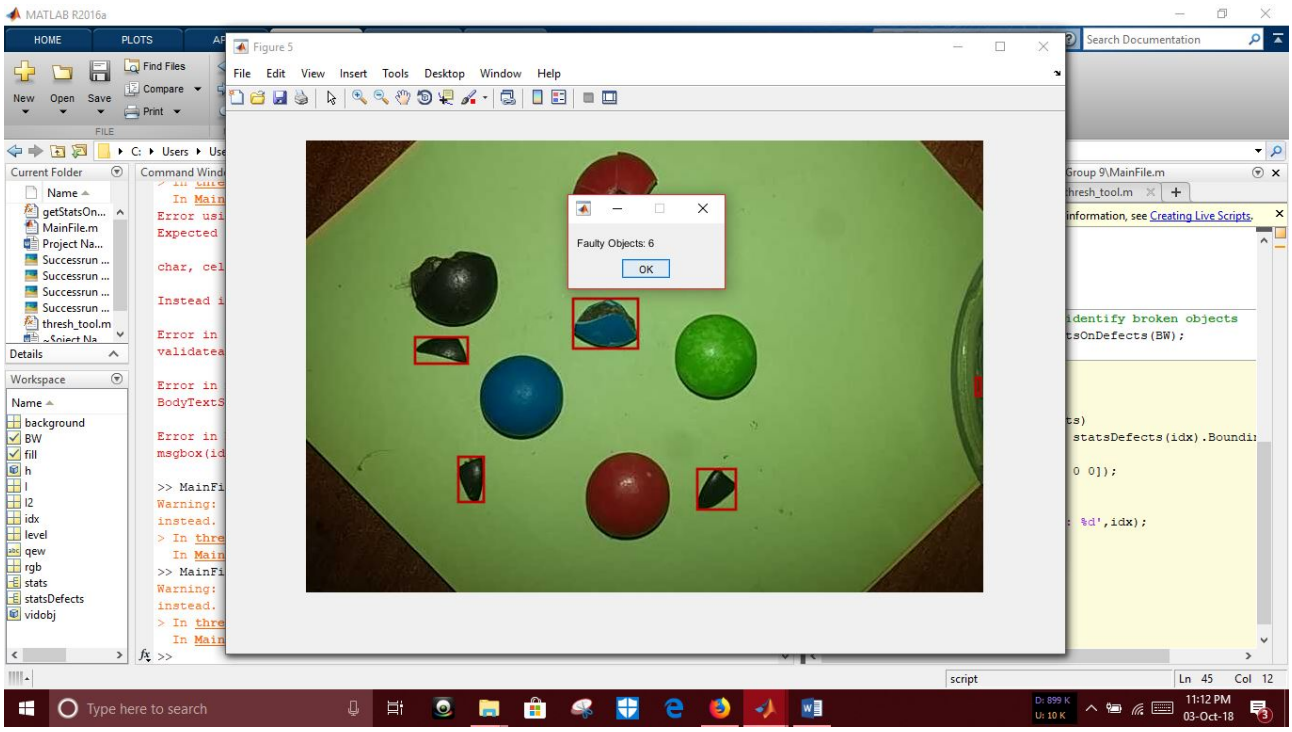


Figure 7: Finding the number of objects with fault



Project Drawbacks:

Thing can go wrong in various ways and for various variables. As the processing mainly depends on how the image is taken, the precise recording is needed. The code has to be modified depending upon the distance between the object and the lens. Also there's an issue about the lighting where too much focused lighting on a particular area can result in various gradient of colors of different intensity and eccentricity. So in short, the code has to be modified after the implementation of the camera according to the surroundings & has to position further if needed.

Real Life Implementation:

Fault analysis of the objects is of very significant use in mass production. To ensure the accuracy of the production and maintain a standard quality, production companies have to control quality & rigidness of the object. Doing this task manually is very much time worthy and costly. Instead, using automation is very much efficient, cheap & swift. There are many specialized techniques to do this particular job. But being able to do it in MATLAB is a massive scope for further use and analysis.

The End.