International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 3 Mar. 2021, pp: 810-817 www.ijaem.net ISSN: 2395-5252

Defect Detection in Printed Circuit Board Using Image Subtraction

Pushpam Prakash

Submitted: 05-03-2021 Revised: 18-03-2021 Accepted: 20-03-2021

ABSTRACT: This research paper is about to detect the defects in PCB . This is to be done with the help of image subtraction algorithm in which pixel by pixel matching of test image and defected image is carried out.With the help of MATLAB software image subtraction is applied in the form of source code .The Ultimate goal is to make PCB error free in fabrication process.

Keywords: PCB, RBG, MATLAB

I. INTRODUCTION:

Printed circuit board (PCB) fabrication process is a multi disciplinary process, and etching is the most critical part in the PCB manufacturing process. Etching is the process, where the copper board will undergo 'peeling' process, where the circuit layout will be preserved while the rest of the copper background will be washed out. In order to minimise scrap caused by the wrongly etched PCB panel, inspection has to be done in early stage.

The traditional way to inspect any defect is visually employing human operators. This process is not only time consuming but also highly prone to errors due to humans factors. The next trend then is to use machine vision inspection system. However, all of the inspections are done after the etching process where any defective PCB found is no longer useful and is simply thrown away. Since etching process costs 70% of the entire PCB fabrication, it is uneconomical to simply discard the defective PCBs. Hence, this research paper proposes an automatic visual inspection on the PCB before the etching process so that any defect that could be found on a PCB would be able to be reprocessed. Although many algorithms are available in defect detection, both contact and noncontact methods, none is able to classify these defects. Contact method tests the connectivity of circuits but unable to detect major flaws in cosmetic defects. Non-contact uses methods such as ultrasonic and x-ray imaging to detect anomalies in the circuit design, both cosmetic and functional. The use of manual labour to visually inspect each PCB is no longer viable since it is prone to human errors, time consuming, requires large overhead costs and results in high wastage. Types of defects on single layer bare PCBs are Breakout, Pin-hole, Open Circuit, Mouse-bite, Missing Conductor, Spur, Short, Wrong Size Hole, Conductor Too Close, Spurious Copper, Excessive Short, Missing Hole. So thus we are using simple image subtraction algorithm in which XOR operation is applied between sample image and test image.This whole process is represented in the form of block diagram.

1.1 Block diagram and its description:

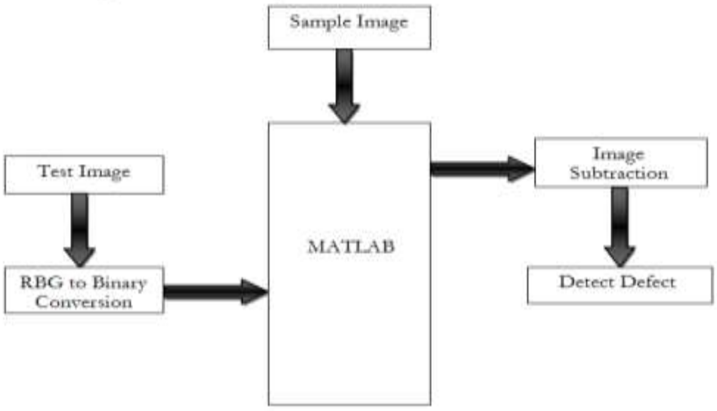


Fig 1. Block Diagram of Defect Detection

Sample and test image

In this system configuration sample image is compared with test image (image with defects). These both images should be converted into .jpg image format and from RGB to grayscale image due to the fact that digital system algorithm can be applied only to grayscale image. Low frequency components are filtered by using wavelet decomposition. Thresholding is being used for noise free image. And at last using image comparison operation on approximation coefficient will result in resultant defect image.

RBG to binary conversion

It is required to covert RGB image into Grayscale image for further converting into binary image. If image is not in a grayscale form then it is important to converts image into grayscale form. In greyscale conversion the image will be comprised as black at weakest intensity and white at strongest intensity and there will be many shades in between. It replace every pixel of image after calculation of gray conversion into new required grayscale pixel value. If grayscale level is done at 8 bit then it will give 256 shades. Here, grayscale image is having value from 0 to 255 Pixel value.

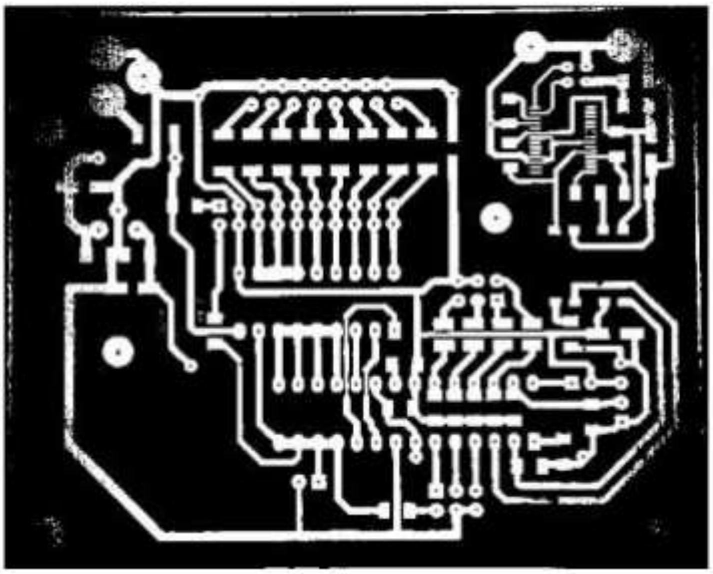


Fig 2. Image after Binary conversion

Binarization is process which converts grayscale image into binary image. Here threshold value is found using Otsu's method. It consists of computation histogram and probability of each intensity level of image. Desired threshold value is correspondence to the between class variance. The histogram of grayscale values of a document image typically consists of two peaks: A high peak corresponding to the white background and a smaller peak corresponding to the foreground. Hence, threshold grayscale value can be determined by an optimal value in the valley between the two peaks.

Images do have some stray pixels and some unwanted marks. By using Median filter noise can be filtered from the image. Smoothing operation in grayscale image is used for noise reduction and filtering is used for noise removal. Median filter which is nonlinear filter is more popular because it has excellent noise removal capabilities.

II. IMAGE SUBTRACTION PRINCIPLE

Image subtraction method used the concept of simple subtraction and rule. In this work, both images of sample image and test/defective image are compared pixel by pixel. The subtraction operation produces either negative or positive pixel value. Therefore, the outcome of this operation is divided into negative image and positive image.

The subtraction operation will produce either negative or positive image, '1' represents white pixel and '0' represents black pixel in a binary image.

Four rules exists for image subtraction operation
Rule 1: If 1-0 = 1 then it gives positive pixel image.
Rule 2: If 0-1 = -1 then it gives negative pixel image.
Rule 3: If 0-0 = 0 then it gives no change pixel image.

Rule 4: If 1-1 = 0 then it gives no change pixel image.

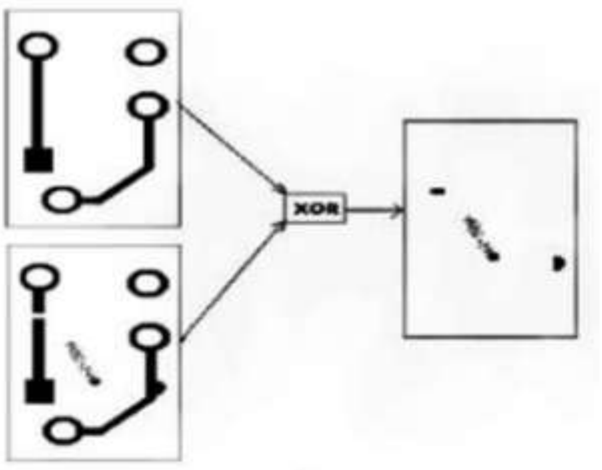


Fig 3. Image Subtraction Algorithm

III. WORKING

The working is divided into four different parts:
1. Loading the correct and incorrect image
2. Image subtraction
3. Analysing error
4. Resultant image

3.1 Loading the Correct and Incorrect Image

In this step we load the sample image (correct) into the GUI. Under this correct image is converted from the jpg to the MATLAB suitable format. The image which is loaded in the GUI is in the RBG image type and the image subtraction works on the binary form of the image so thus the binarization to both the PCB image is carried out. After the binarization of the image it will have two values 0 or 1. But sometimes direct RBG to binary conversion is not possible due to the light intensity problem, so thus MATLAB has the in-built function IMadjust which eliminates the problem of the light intensity. Lastly we will convert the result of IMadjust into the binary form.

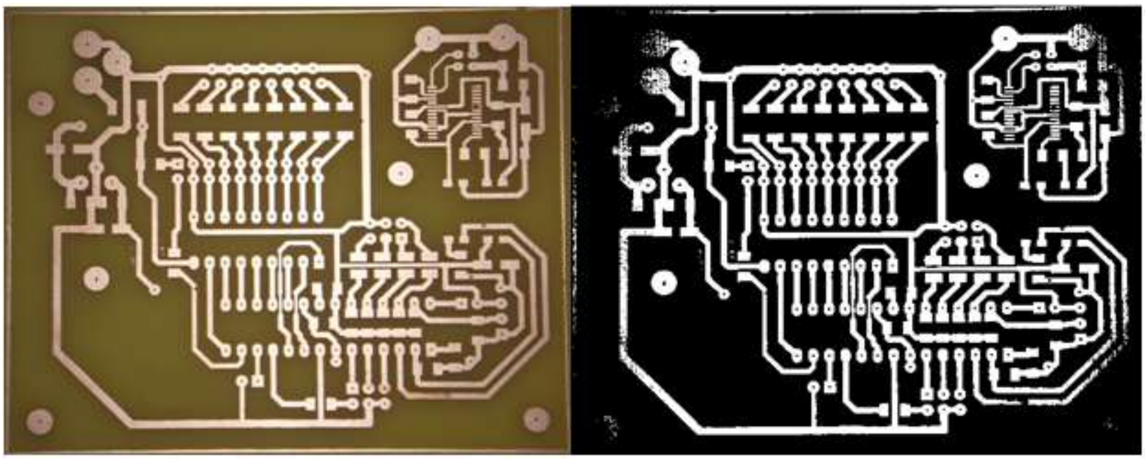


Fig 4. Correct test image and Binary image(without defect)

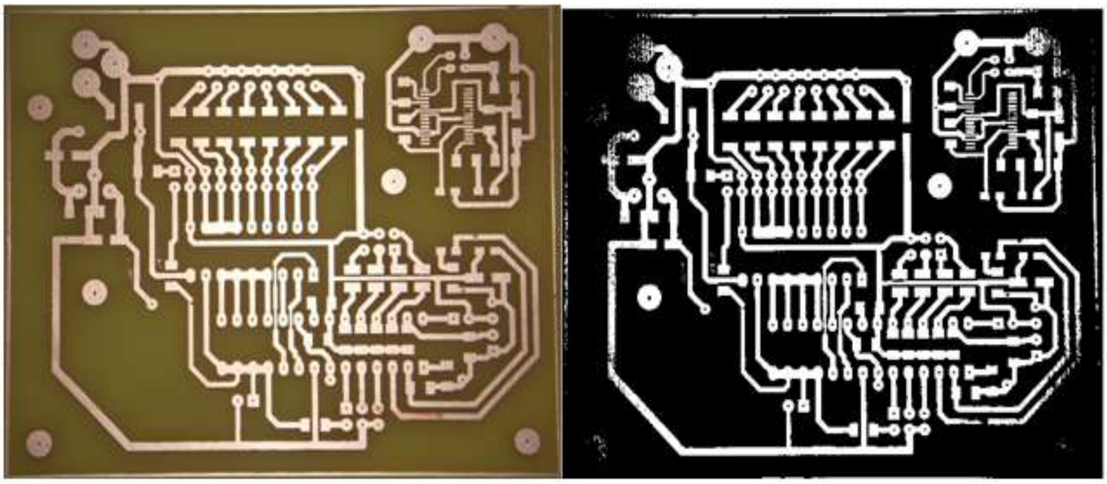


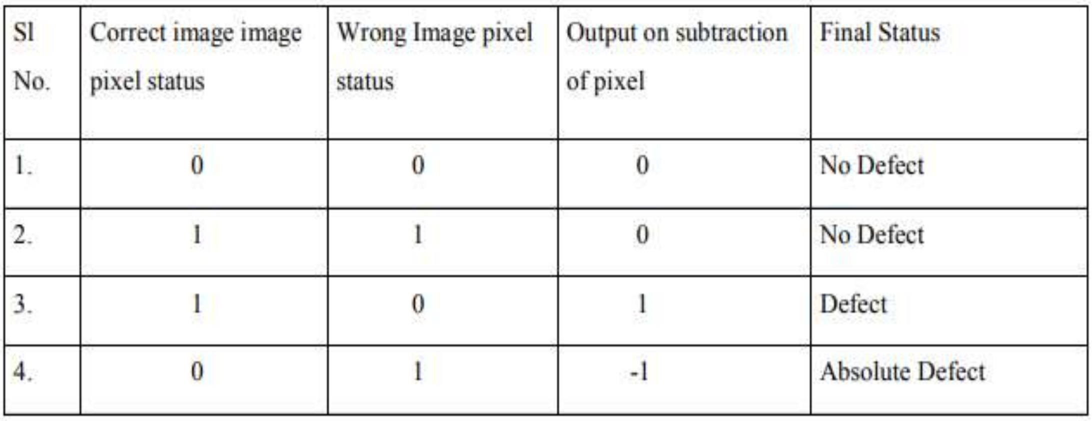
Fig 5. Loading of incorrect image and binary image

3.2 Image Subtraction

In this step one PCB is fully overlapped over another and then pixel by pixel subtraction is carried out .On the PCB artwork (Test image) all connections which are represented by white colour represents 1 (Binary format) and all black colours represent 0 .

Then the subtraction is carried out as shown in table 1

Table 1-Pixel Subtraction Algorithm



Sl No.	Correct image image pixel status	Wrong Image pixel status	Output on subtraction of pixel	Final Status
1.	0	0	0	No Defect
2.	1	1	0	No Defect
3.	1	0	1	Defect
4.	0	1	-1	Absolute Defect

3.3 Analysing Error

Matlab takes 1 as a defect and in the resultant image it is shown by white colour.Whenever the case of absolute defect comes then in that case we use a special built-in function abs which converts the -1 to 1 because Matlab doesn't understand the -1 .

3.4 Resultant Image

After the subtracting and analysing the type of error it will show the resultant image with the percentage
of error.

Source Code for GUI of defect detection

```
function original_Callback(hObject, eventdata, handles)
[A B]=uigetfile('*.jpg');
I=imread([B A]);
axes(handles.axes1)
imshow(I);
handles.I_original=I;
guidata(hObject,handles)
function defected_Callback(hObject, eventdata, handles)
I=handles.original;
bw=im2bw(I,graythresh(rgb2gray(I)));
axes(handles.axes2)
imshow(bw)
function pushbutton3_Callback(hObject, eventdata, handles)
I=handles.original;
gray=rgb2gray(I);
```

```
I_imadjust=imadjust(gray);
axes(handles.axes3)
imshow(I_imadjust)
handles.imadjust=I_imadjust;
guidata(hObject, handles)
function pushbutton4_Callback(hObject, eventdata, handles)
I=handles.imadjust;
bw_imadjust=im2bw(I,graythresh(I));
axes(handles.axes4)
imshow(bw_imadjust)
handles.bw_imadjust=bw_imadjust;
guidata(hObject, handles)
function pushbutton5_Callback(hObject, eventdata, handles)
[A B]=uigetfile('*.jpg');
I_defected=imread([B A]);
axes(handles.axes5)
imshow(I_defected);
handles.I_defected=I_defected;
guidata(hObject,handles)
function pushbutton6_Callback(hObject, eventdata, handles)
I_defected=handles.I_defected;
bwn=im2bw(I_defected,graythresh(rgb2gray(I_defected)));
axes(handles.axes6)
imshow(bwn)
function pushbutton7_Callback(hObject, eventdata, handles)
I_defected=handles.I_defected;
grayy=rgb2gray(I_defected);
I_imadjust1=imadjust(grayy);
axes(handles.axes7)
imshow(I_imadjust1)
handles.imadjust1=I_imadjust1;
guidata(hObject, handles)
function pushbutton8_Callback(hObject, eventdata, handles)
I_imadjust1=handles.imadjust1;
bw_imadjust1=im2bw(I_imadjust1,graythresh(I_imadjust1));
axes(handles.axes8)
imshow(bw_imadjust1)
handles.bw_imadjust1=bw_imadjust1;
guidata(hObject, handles)
function pushbutton9_Callback(hObject, eventdata, handles)
I=handles.bw_imadjust;
I_defected=handles.bw_imadjust1;
I_sub=im2bw(I)-im2bw(I_defected);
axes(handles.axes9)
imshow(abs(I_sub));
handles.I_sub=abs(I_sub);
guidata(hObject,handles)
bw = bwareaopen(abs(I_sub),200);
imshow(bw);
axes(handles.axes5);
[rows cols]=size(bw);
[B,L] = bwboundaries(bw,'noholes');
stats = regionprops(L,'Centroid');
hold on
r=rows/10;
theta=0:0.01:2*pi;
for i=1:length(B)
centroid = stats(i).Centroid;
x=centroid(1) + r*cos(theta);
y=centroid(2) + r*sin(theta);
plot(x,y,'r');
end
hold off
handles.I_sub=abs(I_sub);
guidata(hObject,handles)
I_sub=handles.I_sub;
[rows cols z]=size(I_sub);
error=100*sum(I_sub(:))/(rows*cols);
set(handles.text_error,'string',error)
if error>0
set(handles.text_defected,'string','Defect');
else
set(handles.text_defected,'string','No Defect');
end
```

IV. SIMULATION AND TESTING

The simulation and testing of research project has been done. The PCB Defect Detection giving results as per the desired expectations. Simulation is the imitation of the operation of a real-world process or system over time. Good coding software is one which can show the simulation results of the coding also in it. Here, we have used MATLAB software for testing and simulation purpose. To start, calling the GUI is done after that the correct and incorrect PCB is loaded in the GUI and binarization is performed for both the images. At last, image subtraction algorithm is finally applied and the detect is detected with the error percentage.

In MATLAB, following steps are initialized Now after opening the MATLAB the command window will appear. In the command window we have to enter the .m file name “subtract_gui” so that the GUI will open

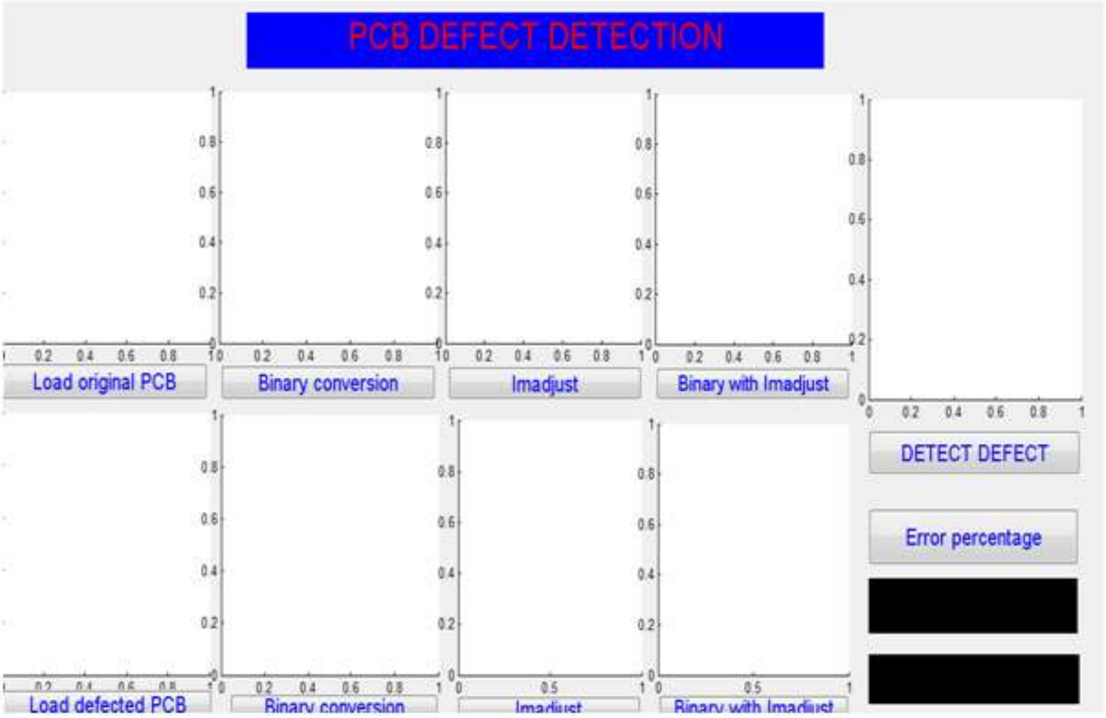
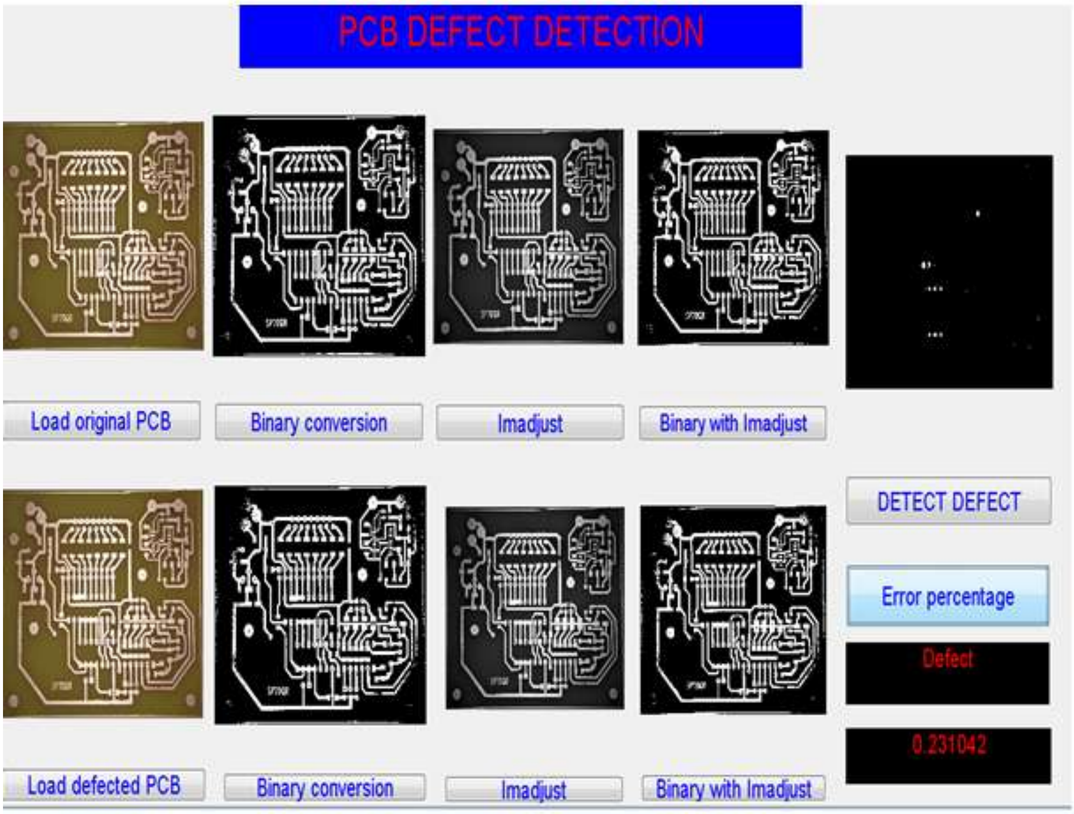


Fig 6. PCB Defect Detection GUI

After clicking all the push buttons respective images is loaded in GUI



Defect

0.231042

Fig 7. GUI showing error percentage in PCB

V. RESULTS

Based on the algorithms shown above, these algorithms need two images, namely sample image and test/defective image. These algorithms use as template image and as defective image. Then for both images are subjected to image subtraction operation to produce the resultant image. From the figure 6 we can see that defect in the PCB is found out by the image subtraction technique and with this much of defect the percentage defect is also defected.

We plot the resultant graph by studying and detecting defect of 5 defected PCBs having various defects, PCB number and its related error percentage are shown in the graph. Error line is the threshold line and error percentage above this line shows that the relevant PCB is defected. Thus PCB numbered 3 and 5 are correct and 1,2,4 are defected. So thus the Defect detect is above the threshold line and if the error is below the threshold then it is considered as the defect.

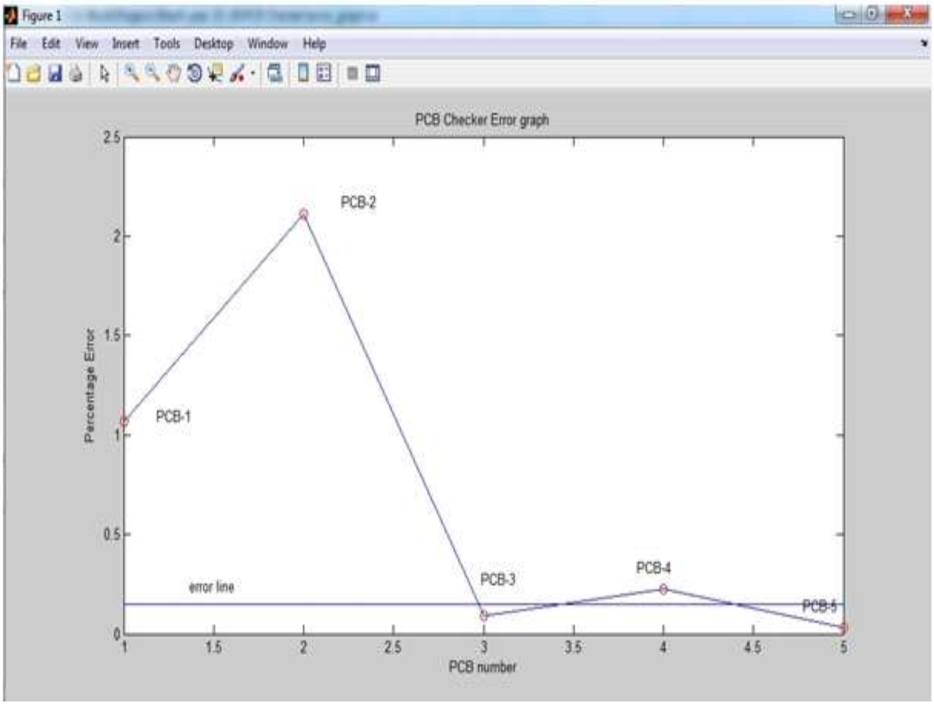


Fig 8. Resultant Graph

VI. CONCLUSION

PCB quality testing is very important from the point of view of sales and ultimately success of the product. Our simulated work in this project gave rise to lots of useful insights.

Especially, it is very clear now that using machine vision many of the defects on the PCB can be detected with good accuracy. Various advances take place in PCB manufacturing industry over the last decade. Machine vision may answer the manufacturing industry's need to improve product quality and increase productivity. Sometimes due to the light intensity problem the captured image from the camera is not properly converted to the binary image which causes the error in defect detection, but later with the use of IMadjust function we have overcome with this difficulty.

We see that our simulation works very well for different types of template and not just for a single type of template. With power of tools such as MATLAB, we were able to capture useful information, and process the same to detect the defects.

Advantage from the above result

If we see in most of the industry, there is only repetitive process. On the basis of this, we could say that the PCB Defect Detection has a huge future scope. It could have a large market scope in the PCB printing small and large industry.

The application of this project named programmable sequential load operation controller are as follows:

The application of this project is in the printing industry where in the bunches the PCB is printed and if one PCB has error then all the other PCB will have the same error resulting the wasting of the PCB. So by the use of this software testing of the PCB can carried out and if there is any defect then it should be cor- rected.

REFERENCES

[1]. + S. H. Indera Putera, Syahrul Fahmi Dzafaruddin, Maziah Mohamad, "MATLAB Based Defect Detection and Classification of Printed Circuit Board" Digital Information and Communication Technology and it's Applications (DICTAP), IEEE 2012 Second International Conference. Bangkok.

[2]. + S. H. Indera Putera and Z. Ibrahim, "Printed Circuit Board Defect Detection Using Mathematical Morphology and MATLAB Image Processing Tools", ICINT 2010, Shanghai, China, 2010.

[3]. Syamsiah Mashohor, Jonathan R. Evans and Ahmet T. Erdogan (2006)-" Automatic Hybrid Genetic Algorithm Based Printed Circuit Board Inspection". Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)0-7695-2614-4/06 IEEE.

[4]. Shih-Chieh Lin and Chia-Hsin Su (2006), "A Visual Inspection System for Surface Mounted Devices on Printed Circuit Board IEEE".

[5]. Ismail Ibrahim (March 2012), A printed circuit board inspection system with defect classification capability International Journal of Innovative Management, Information & Production ISME 2011 ISSN 2185-5439 Volume 3, Number 1.

[6]. Syamsiah Mashohor, Jonathan R. Evans and Ahmet T. Erdogan (2006) Automatic Hybrid Genetic Algorithm Based Printed Circuit Board Inspection Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)0-7695-2614-4/06 IEEE.

International Journal of Advances in Engineering and Management
ISSN: 2395-5252

IJAEM

Volume: 03 Issue: 03 DOI: 10.35629/5252

www.ijaem.net
Email id: ijaem.paper@gmail.com