CHAPTER 1

INTRODUCTION

A printed circuit board (PCB) mechanically supports and electrically connects electronic components or electrical components using conductive tracks, pads and other features etched from one or more sheet layers of copper laminated onto and/or between sheet layers of a non-conductive substrate. Components are generally soldered onto the PCB to both electrically connect and mechanically fasten them to it.

A basic PCB consists of a flat sheet of insulating material and a layer of copper foil, laminated to the substrate. Chemical etching divides the copper into separate conducting lines called tracks or circuit traces, pads for connections, vias to pass connections between layers of copper, and features such as solid conductive areas for EM shielding or other purposes. The tracks function as wires fixed in place, and are insulated from each other by air and the board substrate material. The surface of a PCB may have a coating that protects the copper from corrosion and reduces the chances of solder shorts between traces or undesired electrical contact with stray bare wires. For its function in helping to prevent solder shorts, the coating is called solder resist.

Alternatives to PCBs include wire wrap and point-to-point construction, both once popular but now rarely used. PCBs require additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Specialized CAD software is available to do much of the work of layout. Mass-producing circuits with PCBs is cheaper and faster than with other wiring methods, as components are mounted and wired in one operation. Large numbers of PCBs can be fabricated at the same time, and the layout only has to be done once. PCBs can also be made manually in small quantities, with reduced benefits. PCBs can be single-sided (one copper layer), double-sided (two copper layers on both sides of one substrate layer), or multi-layer (outer and inner layers of copper, alternating with layers of substrate). Multi-layer PCBs allow for much higher component density, because circuit traces on the inner layers would otherwise take up surface space between components.

1.1 Historical Background

Before the development of printed circuit boards electrical and electronic circuits were wired point-to-point on a chassis. Typically, the chassis was a sheet metal frame or pan, sometimes with a wooden bottom. Components were attached to the chassis, usually by insulators when the connecting point on the chassis was metal, and then their leads were connected directly or with jumper wires by soldering, or sometimes using crimp connectors, wire connector lugs on screw terminals, or other methods. Circuits were large, bulky, heavy, and relatively fragile (even discounting the breakable glass envelopes of the vacuum tubes that were often included in the circuits), and production was labour-intensive, so the products were expensive.

Development of the methods used in modern printed circuit boards started early in the 20th century. In 1903, a German inventor, Albert Hanson, described flat foil conductors laminated to an insulating board, in multiple layers. Thomas Edison experimented with chemical methods of plating conductors onto linen paper in 1904. Arthur Berry in 1913 patented a print-and-etch method in the UK, and in the United States Max Schoop obtained a patent to flame-spray metal onto a board through a patterned mask. Charles Ducas in 1927 patented a method of electroplating circuit patterns.

The Austrian engineer Paul Eisler invented the printed circuit as part of a radio set while working in the UK around 1936. In 1941 a multi-layer printed circuit was used in German magnetic influence naval mines. Around 1943 the USA began to use the technology on a large scale to make proximity fuses for use in World War II. After the war, in 1948, the USA released the invention for commercial use. Printed circuits did not become commonplace in consumer electronics until the mid-1950s, after the Auto-Sembly process was developed by the United States Army. At around the same time in the UK work along similar lines was carried out by Geoffrey Dummer, then at the RRDE. During World War II, the development of the anti-aircraft proximity fuse required an electronic circuit that could withstand being fired from a gun, and could be produced in quantity. The Centralab Division of Globe Union submitted a proposal which met the requirements: a ceramic plate would be screen printed with metallic paint for conductors and carbon material for resistors, with ceramic disc capacitors and subminiature vacuum tubes soldered in place.

The technique proved viable, and the resulting patent on the process, which was classified by the U.S. Army, was assigned to Globe Union. It was not until 1984 that the Institute of Electrical and Electronics Engineers (IEEE)awarded Harry W. Rubinstein the Cledo Brunetti Award for early key contributions to the development of printed components and conductors on a common insulating substrate. Rubinstein was honored in 1984 by his alma mater, the University of Wisconsin-Madison, for his innovations in the technology of printed electronic circuits and the fabrication of capacitors. This invention also represents a step in the development of integrated circuit technology, as not only wiring but also passive components were fabricated on the ceramic substrate.

Originally, every electronic component had wire leads, and a PCB had holes drilled for each wire of each component. The component leads were then inserted through the holes and soldered to the copper PCB traces. This method of assembly is called through-hole construction. In 1949, Moe Abramson and Stanislaus F. Danko of the United States Army Signal Corps developed the Auto-Sembly process in which component leads were inserted into a copper foil interconnection pattern and dip soldered. The patent they obtained in 1956 was assigned to the U.S. Army. With the development of board lamination and etching techniques, this concept evolved into the standard printed circuit board fabrication process in use today. Soldering could be done automatically by passing the board over a ripple, or wave, of molten solder in a wave-soldering machine. However, the wires and holes are inefficient since drilling holes is expensive and consumes drill bits and the protruding wires are cut off and discarded.

From the 1980s onward, small surface mount parts have been used increasingly instead of through-hole components; this has led to smaller boards for a given functionality and lower production costs, but with some additional difficulty in servicing faulty boards.

In the 1990s the use of multilayer surface boards became more frequent. As a result, size was further minimized and both flexible and rigid PCBs were incorporated in different devices. In 1995 PCB manufacturers began using microvia technology to produce High-Density Interconnect (HDI) PCBs.

1.2 Manufacturing Process

PCB manufacturing consists of many steps.

PCB CAM

Manufacturing starts from the fabrication data generated by computer aided design, and component information. The fabrication data is read into the CAM (Computer Aided Manufacturing) software. CAM performs the following functions:

1. Input of the fabrication data.
2. Verification of the data
3. Compensation for deviations in the manufacturing processes (e.g. scaling to compensate for distortions during lamination)
4. Panelization
5. Output of the digital tools (copper patterns, drill files, inspection, and others)

Copper patterning

The first step is to replicate the pattern in the fabricator's CAM system on a protective mask on the copper foil PCB layers. Subsequent etching removes the unwanted copper.

1. Silk screen printing uses etch-resistant inks to create the protective mask.

2. Photoengraving uses a photo mask and developer to selectively remove a UV-sensitive photoresist coating and thus create a photoresist mask. Direct imaging techniques are sometimes used for high-resolution requirements. Experiments were made with thermal resist.

3. PCB milling uses a two or three-axis mechanical milling system to mill away the copper foil from the substrate. A PCB milling machine (referred to as a 'PCB Prototyper') operates in a similar way to a plotter, receiving commands from the host software that control the position of the milling head in the x, y, and (if relevant) z axis.

4. Laser resist ablation Spray black paint onto copper clad laminate, place into CNC laser plotter. The laser raster-scans the PCB and ablates (vaporizes) the paint where no resist is wanted.

5. Laser etching The copper may be removed directly by a CNC laser. Like PCB milling above this is used mainly for prototyping

Subtractive, additive and semi-additive processes

Subtractive methods remove copper from an entirely copper-coated board to leave only the desired copper pattern. In additive methods the pattern is electroplated onto a bare substrate using a complex process. The advantage of the additive method is that less material is needed and less waste is produced. Semi-additive is the most common process: The unpattern board has a thin layer of copper already on it. A reverse mask is then applied. Additional copper is then plated onto the board in the unmasked areas.

Chemical etching

Chemical etching is usually done with ammonium persulfate or ferric chloride. For PTH (plated-through holes), additional steps of electroless deposition are done after the holes are drilled, then copper is electroplated to build up the thickness, the boards are screened, and plated with tin/lead. The tin/lead becomes the resist leaving the bare copper to be etched away.

Lamination

Multi-layer printed circuit boards have trace layers inside the board. This is achieved by laminating a stack of materials in a press by applying pressure and heat for a period of time. This results in an inseparable one piece product. For example, a four-layer PCB can be fabricated by starting from a two-sided copper-clad laminate, etch the circuitry on both sides, then laminate to the top and bottom pre-preg and copper foil. It is then drilled, plated, and etched again to get traces on top and bottom layers.

Drilling

Holes through a PCB are typically drilled with drill bits made of solid coated tungsten carbide. Coated tungsten carbide is used because board materials are abrasive. High-speed-steel bits would dull quickly, tearing the copper and ruining the board. Drilling is done by computer-controlled drilling machines, using a drill file or Excellon file that describes the location and size of each drilled hole.

Holes may be made conductive, by electroplating or inserting hollow metal eyelets, to connect board layers. Some conductive holes are intended for the insertion of through-hole-component leads. Others used to connect board layers, are called vias.

Bare-board test

Boards with no components installed are usually bare-board tested by humans for "shorts" and "opens". A short is a connection between two points that should not be connected. An open is a missing connection between points that should be connected. Along with short and open circuit other defects like, extra holes, missing holes, etc. are checked.

The PCBs which clear these testing process are sent to next process and others which fail in testing process are scraped. These testing are done by human vision or by applying voltage to the PCB.

Protection and packaging

PCBs intended for extreme environments often have a conformal coating, which is applied by dipping or spraying after the components have been soldered. The coat prevents corrosion and leakage currents or shorting due to condensation. The earliest conformal coats were wax; modern conformal coats are usually dips of dilute solutions of silicone rubber, polyurethane, acrylic, or epoxy. Another technique for applying a conformal coating is for plastic to be sputtered onto the PCB in a vacuum chamber. The chief disadvantage of conformal coatings is that servicing of the board is rendered extremely difficult.

Many assembled PCBs are static sensitive, and therefore they must be placed in antistatic bags during transport. When handling these boards, the user must be grounded (earthed). Improper handling techniques might transmit an accumulated static charge through the board, damaging or destroying components.

The damage might not immediately affect function but might lead to early failure later on, cause intermittent operating faults, or cause a narrowing of the range of environmental and electrical conditions under which the board functions properly. Even bare boards are sometimes static sensitive: traces have become so fine that it's quite possible to blow an etch off the board (or change its characteristics) with a static charge. This is especially true on non-traditional PCBs such as MCMs and microwave PCBs.

1.3 Motivation

In any manufacturing process one of the most important stages is the testing stage. It is no different in PCB manufacturing process. Nowadays PCBs are most vital part of electronic industry. It is present in almost all the electronic products ranging from, a kid’s toy to safety equipment in a car to a satellite. Hence lot of care should be taken from PCB manufacturing companies to provide good quality products so, at most importance should be given in the testing stage by the manufacturer.

If the testing stage is not done properly the industry can loose millions of dollars when they retrieve their product from the market which also effects their brand name in the market reducing their sales.

1.4 Problem statement

In today’s manufacturing process testing is done in two methods Human testing, Automated testing. In human testing, a person is appointed for testing process , they check each and every PCB for the defects manually. This is very time consuming and the efficiency might not be that high, as they have to check each and every part of the PCB and detect the defect. It is not so easy to detect the all the defects by human eye as some of them might be very small and go un-noticed from a human eye. In automated testing, a system is used for testing the bare PCB. It applies a small voltage to the bare PCB and checks for proper conduction of the current through the PCB. Sometimes this can be dangerous because of the short and open circuits in the PCB. These problems can be solved using digital image processing for detecting defects and defects can be classified using machine learning process by training the system.

1.5 Objectives

The main objectives of this project is to detect the defects like open circuit, short circuit, extra holes, missing holes, all defects. This can be done by following these steps,

1. Creation of data base and models.
2. Defect detection.
3. Defect classification.

These steps can be achieved using digital image processing through machine learning. In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

1.6 Organization of the report

Chapter 1:

This chapter discuss about the PCB and their types. An overview of the current PCB manufacturing processes and their problems from which the motivation towards this project raised.

Chapter 2:

This chapter discusses about the various papers and journals referred on the field of PCB defect detection and classification based on various algorithms and methodology and the best among them are chosen to execute this project.

Chapter 3:

The information about the hardware and software requirements along with the brief description about the software and equipment used are given in this chapter.

Chapter 4:

In this chapter the methodology that best suits the project is described thoroughly in detail with all the steps included and their implementation is discussed along with required block diagrams and flow charts.

Chapter 5:

In this chapter the project results are described in detail along with the respective screenshots. The efficiency of the project for different defect types are calculated and tabulated accordingly and the future works of the project is also discussed in the chapter.

CHAPTER 2

LITERATURE SURVEY

This chapter discusses about the research done regarding PCB manufacturing, machine learning, different algorithms which can be used to obtain the required output and finally selecting the best ones among them.

2.1 Existing methodology

There are typically three methods applied during the AOI process: a comparison reference (CR), non-reference verification (NV), and a hybrid approach (HA). The CR method compares an inspected image with a reference image. It measures any existing dissimilarities between the reference image and the inspected image. Thus, this method requires a reference image, and inspection is difficult without such an image. The NV approach tests the design rule of the PCB for detecting faults. It essentially verifies the widths of the insulators and conductors. However, this approach makes it difficult for users to design the rules as restrictions on the image features, and it cannot detect faults without such rules [1]. The HA approach combines various types of CR and NV methods. That is, the approach utilizes a reference image or design rules to inspect a PCB image. However, it still does not solve the limitation of being unable to conduct an inspection without reference information. Because the circuits used in a PCB are diverse and complex, the reference information increases significantly, thereby decreasing the inspection's efficiency. Therefore, additional studies are required to detect faults without reference information.

Numerous algorithms have been proposed to improve the accuracy of a PCB inspection. Such algorithms can be categorized into two approaches: referential and non-referential methods.

A referential method is based on a comparison between the inspected image and a reference image. To measure the dissimilarity between these two images, image subtraction and template matching techniques are used. Wu et al proposed an inspection method based on a subtraction method [2]. The image subtraction method compares both images [3]. The resulting image, which is obtained after this Operation, contains only some portions of a fault [4].Therefore, the reference image and the inspected image should be placed in a fixed position to compare both images. In addition, a reference image of the same size is required. Template matching is a technique for identifying the parts of the image that match the reference image.

This method extracts the features of both the reference and inspected images. It then calculates the similarities of these features. One of the major disadvantages of template matching is that a large amount of information regarding the reference image must be used. Therefore, this method requires a mass storage device that can store all of the information. Moreover, the inspected images also have to be precisely matched for a comparison with the reference image [5]. Acciani et al. suggested an inspection algorithm that extracts the wavelet and geometric features, and then detects a defect after learning the fault pattern using a neural network and k-nearest neighbours [6].

When extracting the features of a PCB image, this method uses the maximum value of the correlation coefficients between the features of the reference image and the inspected image [7]. But these methods are not very efficient, hence consider kurtosis which is a very basic feature extraction algorithm present in MATLAB software [8]. This considers colour, texture and distance as features to be extracted to train the system for creation of database [9]. This requires very less time compared to other algorithms to both training and executing. Because of its simple nature the efficiency of kurtosis is also low [10]. One algorithm which gives very high efficiency with reasonable computational time is Histogram Of Gradients (HOG). This method divides the image given to the system in pixels. Then it measures the intensity value of each pixels and differentiate the images accordingly [11].

There are two types of machine learning, supervised and unsupervised. Supervised machine learning is used to classify the input based on the database and the unsupervised is used to clustering and modelling the inputs [12] . For the application of PCB defect detection supervised machine learning should be used. In supervised machine learning Support Vector Machine (SVM) is usually used to classify the inputs with high efficiency [13]. SVM machine learning consists of two types, binary SVM and multiclass SVM. Where binary SVM has only two outputs either 1(true) or 0(false). This method works on the principle that the algorithm considers one threshold value and all the inputs whose values are above it are considered true and rest are considered as false. This is helpful in applications which are primitive and which want the output to be either true or false. But the multicast SVM considers the features of the input with the features of the database and then classify the input accordingly [14].

2.2 Outcome of literature survey

From the literature survey it is clear that which process is best suited for the project to give required output. In this case according to Malge P S and Nadaf R S non referential approach for detecting the defects in test PCB by comparing it with the inputted images. Based on Kelly c, Siddiqui F M, Bardak B and Woods R Histogram of Gradients is the best method for feature extraction of images. Supervised method is the more compatible method to classify the test images by comparing it with the database according to Mehryar Mohri, Afshin rotmizadeh and Ameet Talwalkar. Finally in this project SVM algorithm is used for defect classification as it fulfils the need to classify the images to more than 2 types.

CHAPTER 3

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software requirement specification provides the information like hardware and software necessary to carry out project. The requirements desired from the software are set during the implementation stage where the software is designed to fulfil the requirements.

3.1 Hardware requirements

The hardware and software components of a system are required to install and use software efficiently. System requirements for operating system will be hardware components, while other applications software will list both hardware and operating system requirement. System requirements are most commonly listed as minimum and recommended requirements. The minimum system requirements need to be met for the software to run on the system.

3.1.1 Mobile camera

In this project the test image can be given in two types, first method is by directly from the images already in the system and the second method is for giving real time images by a mobile camera using a third party application like IP webcam etc. This application can run in mobiles with android 4.1 and above. In this method the output efficiency depends on the quality of the mobile camera output. If the mobile camera is of less than 8MP quality the output efficiency is not up to the mark. Whereas good efficiency can be achieved with cameras above 12MP sensors.

3.1.2 Bare PCBs

Bare PCBs are the PCBs with no electronic components soldered on them. These PCB images are used as test images to be tested and correct images are used to train the system using machine learning algorithms.

3.1.3 Processor i3 and above

Systems with only Intel processors above i3 are compatible to the MATLAB R2015a version 8.5.0.197613. Hence such system which are compatible with such requirements should be used.

3.2 Software Requirements

- Software: MATLAB (using MATLAB R2015a version 8.5.0.197613) .
- Operating System: Microsoft window 7 or above.
- Database consisting of jpg files.

3.2.1 MATLAB software description

MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran. It provides an interactive environment with hundreds of built - in functions for its own high - level programming language.

The name MATLAB stands for MATrix LABoratory. MATLAB's built - in function provide excellent tools for linear algebra computations, data analysis, image processing, signal processing, optimization, numerical solution of ordinary differential equation(ODEs), quadrature, and many other types of scientific computations. Most of these functions use state of the art algorithms. There are numerous functions for 2-D and 3-D graphics, as well as for animation. Also, for those who cannot do without their FORTRAN or C codes, MATLAB even provides an external interface to run those program from within MATLAB. The user, however, is not limited to the built - in function; he can write his own functions in the MATLAB language. Once written, these functions behave just like the built - in functions. MATLAB's language is easy to interpret and is user-friendly.

Along with these specifications there are functional and non functional requirements which play a major role in the software performance.

Functional Requirements

Functional requirement defines a function of a software system. A function is described as a set of inputs and outputs. The functional requirements are a kind of prerequisite that identifies with the conduct of the framework to certain inputs ought to fulfil.

The main functional unit in the system is database of PCBs that is stored in a folder used to train the system and to detect the defects in PCBs. The efficiency of the system increases as the number of images stored in the database are increased.

Non Functional Requirements

Non Functional Requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non Functional Requirements are often called "quality attributes" of a system. Usability, performance comes under the non functional requirements. They cannot be represented by the series of code but they emerge as property from entire work. The important non functional requirements explained as follows.

- Modifiability: Irrespective of any changes which is made to database the system must be able to produce proper results.
- Reliability: Reliability defines how the system works consistently and properly according to the specifications.
- Accuracy: The system must be able to provide accurate or precise results.
- Usability: Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

This chapter discusses about two parts: Methodology, Implementation of the project. In methodology part the general working and important terms are discussed and in implementation part the complete working with implementation is explained.

4.1 Methodology

The Figure 4.1 shows the selection of basic outline of methodology in different stages for the classification. Each block explained in detail later from section 4.1.4.

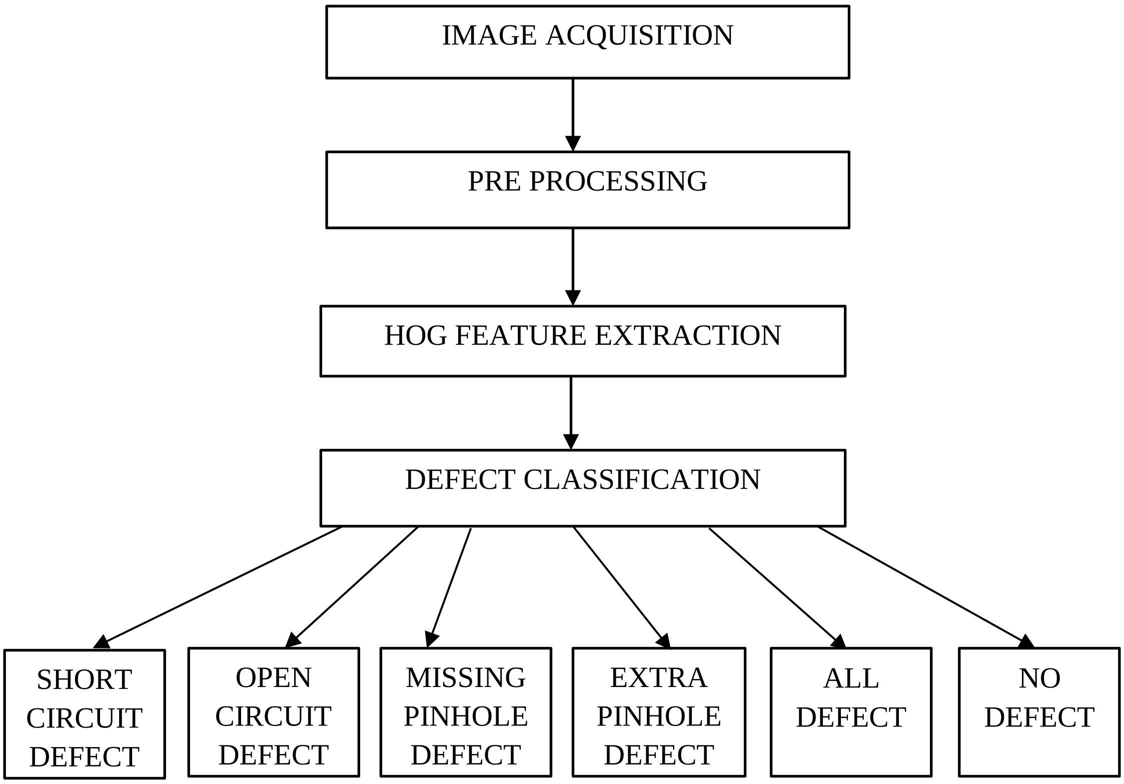


Figure 4.1 Block diagram which describes basic outline of methodology

The implementation of PCB defect detection is done using following concepts.

1. Machine learning
2. Support vector machine(SVM)
3. Histogram of oriented gradient(HOG).

4.1.1 MACHINE LEARNING

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false.

In unsupervised learning, the algorithm builds a mathematical model from a set of data which contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget, and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment, and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans.

4.1.2 Support vector machines

Support vector machines (SVMs), also known as support vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

SVM method is chosen to implement the PCB defect detection, Now let's see SVM classification in detail. Support Vector Machines (SVMs) are a set of supervised learning methods which have been used for classification, regression and outlier's detection. There are number of benefits for using SVM such as: i) It is effective is high dimensional space, ii) Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient, iii) It is versatile because holds different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels. Most real-world problems involve non-separable data for which no hyper plane exists that successfully separates the positive from negative instances in the training set. One good solution to this inseparability problem is to map the data onto a higher dimensional space and define a In order to get better results the selection of an appropriate kernel function is important, since the kernel function defines the transformed feature space in which the training set instances separating hyper plane there. This higher-dimensional space is called the transformed feature space, as opposed to the input space occupied by the training instances will be classified. There are two type of svm classification here Multiclass SVM aims to assign labels to instances by using support-vector machines, where the labels are drawn from a finite set of several elements.

The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction includes these concepts specified:

- Building binary classifiers that distinguish between one of the labels and the rest (one-versus-all) or between every pair of classes (one-versus-one). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest-output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification.

4.1.3 HOG Feature extraction

In this section the histogram of oriented gradient is described with the flow of the feature extracting process in detail.In Preprocessing HOG feature descriptor used for PCB defect detection is calculated on a 250×250 patch of an image. Of course, an image may be of any size. Typically patches at multiple scales are analyzed at many image locations. The only constraint is that the patches being analyzed have a fixed aspect ratio. This have to be selected a patch of size 250×250 for calculating our HOG feature descriptor. This patch is cropped out of an image and resized to 250×250. Now we are ready to calculate the HOG descriptor for this image patch.Gradient of image To calculate a HOG descriptor, first calculate the horizontal and vertical gradients; after all, calculate the histogram of gradients. This is easily achieved by filtering the image with the kernels. The gradient image removed a lot of non-essential information (e.g. constant color background), but highlighted outlines. In other words, you can look at the gradient image and still easily say there is a PCB in the picture. At every pixel, the gradient has a magnitude and a direction. For color images, the gradients of the three channels are evaluated. The magnitude of gradient at a pixel is the maximum of the magnitude of gradients of the three channels, and the angle is the angle corresponding to the maximum gradient.In this step, the image is divided into cells and a histogram of gradients is calculated for each cells. We will learn about the histograms in a moment, but before we go there let us first understand why we have divided the image into cells. One of the important reasons to use a feature descriptor to describe a patch of an image is that it provides a compact representation. Now let us consider that An 8×8 image patch contains 8x8x3 = 192 pixel values. The gradient of this patch contains 2 values (magnitude and direction) per pixel which adds up to 8x8x2 = 128 numbers. By the end of this section we will see how these 128 numbers are represented using a 9-bin histogram which can be stored as an array of 9 numbers. Not only is the representation more compact, calculating a histogram over a patch makes this representation more robust to noise. Individual gradients may have noise, but a histogram over 8×8 patch makes the representation much less sensitive to noise. But why 8×8 patch , Why not 32×32 ! It is a design choice informed by the scale of features we are looking for. HOG was used for PCB defect detection initially. 8×8 cells in a photo of a PCB scaled to 250×250 are big enough to capture interesting features (e.g. the holes, the top of the circuit tracks etc.).The histogram is essentially a vector (or an array) of 9 bins (numbers) corresponding to angles 0, 20, 40, 60 … 160. we see the raw numbers representing the gradients in the 8×8 cells with one minor difference — the angles are between 0 and 180 degrees instead of 0 to 360 degrees. These are called “unsigned” gradients because a gradient and it’s negative are represented by the same numbers. In other words, a gradient arrow and the one 180 degrees opposite to it are considered the same. But, why not use the 0 – 360 degrees , Empirically it has been shown that unsigned gradients work better than signed gradients for PCB defect detection. Some implementations of HOG will allow you to specify if you want to use signed gradients. The next step is to create a histogram of gradients in these 8×8 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40 … 160.. We are looking at magnitude and direction of the gradient of the same 8×8 patch . A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude. Normalization Let’s say we have an RGB colour vector [128, 64, 32]. The length of this vector is sqrt(sqr(128)+ sqr(64)+ sqr(32))=146.64. This is also called the L2 norm of the vector. Dividing each element of this vector by 146.64 gives us a normalized vector [0.87, 0.43, 0.22]. Now consider another vector in which the elements are twice the value of the first vector 2 x [128, 64, 32] = [256, 128, 64]. You can work it out yourself to see that normalizing [256, 128, 64] will result in [0.87, 0.43, 0.22], which is the same as the normalized version of the original RGB vector. You can see that normalizing a vector removes the scale. Now that we know how to normalize a vector, you may be tempted to think that while calculating HOG you can simply normalize the histogram the same way we normalized the vector above. It is not a bad idea, but a better idea is to normalize over a bigger sized block. Now consider a 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector and it can be normalized just the way a 3×1 vector is normalized. The window is then moved by 8 pixels and a normalized 36×1 vector is calculated over this window and the process is repeated which is illustrated in Figure 4.2.

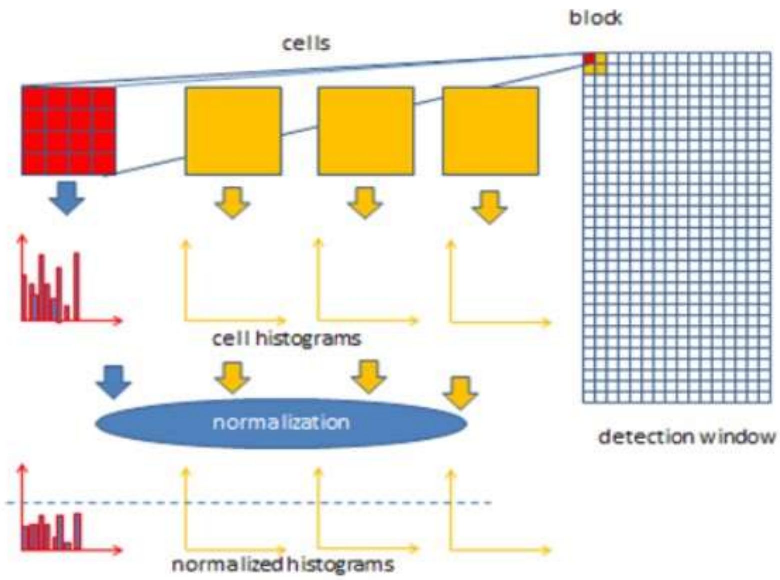


Figure 4.2 HOG feature extracting procedure

4.2 Implementation of PCB Defect Classification

The procedure of the creation of database model as follows: now to train the image first take images with different defects and segregate the images into positive and negative images, in the positive images set images of only one kind of defect in the other hand in negative image set we store other defective PCB images than given in the positive images . now after creating image dataset use HOG feature extracting algorithm and extract the feature like color ,texture and distance from the defective PCB images and store the feature in the form of models by creating models using keyword train images. Train the image features to the system. Thus storing the models hence creation of database is completed.

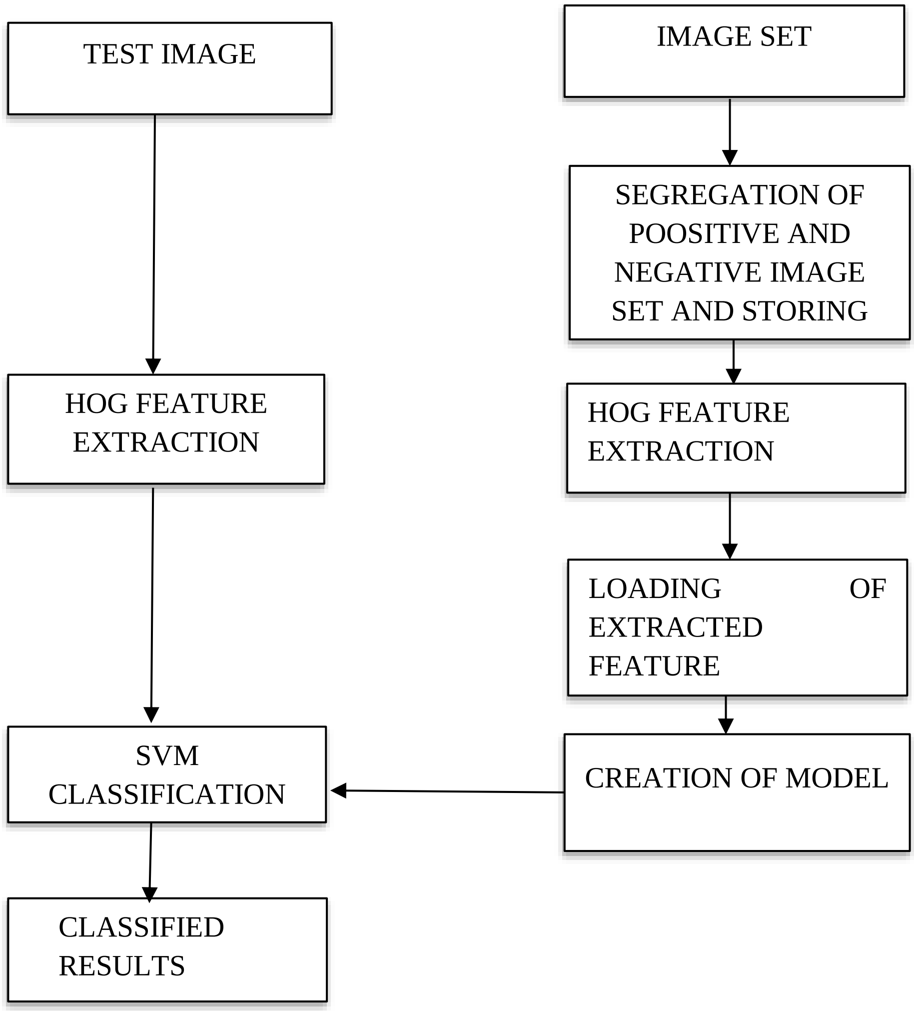


Figure 4.3 Block Diagram of Methods Involved in Defect Classification

Now the test sample image is fed to the system and HOG feature extraction algorithm is applied to extract the features hence storing the extracted values and is compared with each models stored in the database ,if any of the feature are matched then it classify the image and gives the result i.e It names the defect, This complete process is depicted in Figure 4.4.

The methodology depends upon mainly three steps:

1. Algorithm for HOG Feature extraction

2. Algorithm for loading images

3.Creation of model

4.2.1 Algorithm for HOG Feature extraction:

Feature extraction function is called by the train image code. Feature extraction function is used for training the system for recognizing the defect in the PCB. Training is done by loading all the files model1, model1, model2, model3, model4, model5 where model1 consist of the short circuit defect positive and negative files in which positive file consists all different short circuit defect images and negative file consists of all other defect images except short circuit defect images in it.

Feature extraction function will perform following operations as shown in Figure 4.7

Step 1: firstly listing of all positive files with ‘jpg’ file format from the specified path from the user is done .The command used to implement is ‘dir’. Likewise for each and every defect same procedure is followed. The specified path should be to the folder which consists of positive and negative folder of each defect.

Step 2: Once listing of all images done using ‘imread’ command reading of each and every image is done then resizing of image will occur for every image with command ’imresize’ the standard size 250x250 has adopted.

Step 3: Colour image is converted to grey scale image using ‘rgb2grey’command for every image. Grey scale image will well differentiate between defect region and no defect region in the image.

Step 4: After this from the grey scale image extraction of H.O.G. features is done by using the inbuilt function ‘extractHOGFeatures’.

At last these features are saved in the variable specified in the form of cell same procedure is followed or extracting the feature for the negative folder of a particular defect which consists of all other PCB defect images except that. Above algorithm is applied for short circuit defect, open circuit defect, missing hole defect, extra hole defect, all defect. Feature extraction is very step in process of determining the defect in the test PCB as depends on accuracy of the code.

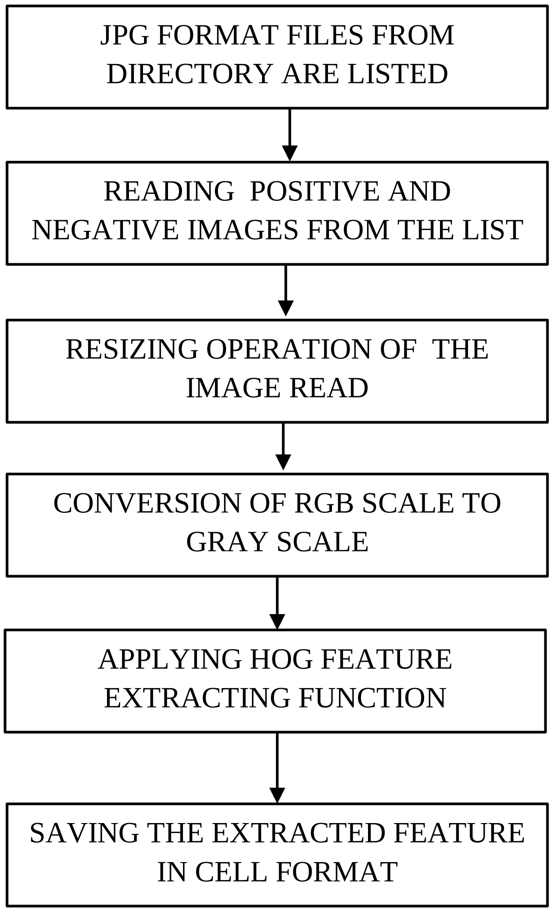


Figure 4.4 Block diagram of algorithm for HOG feature extraction process

4.2.2 Algorithm for Loading images

In the loading of a images in this method plays a major role in constructing a model .For loading the images first assigning the path to dataset to a variable used as reading part of the program is necessary , which in turn checks for the matrix format file in the assigned directory then if yes then it will directly load the file to output else it will load images of positive and negative separately to a variable ,the next step is to find the size of the image which is necessary to build the cell file for defining the size to store the values for getting the size of the image length is used ,then the storing operation takes place here after getting the size of the cell, using the images loaded the cell is constructed by image values and storing it in the variable .This variable is then passed on to the main program function for the svm training operation of the images takes place and convert to matrix format file for the creation of model . this complete process is illustrated in Figure 4.8.

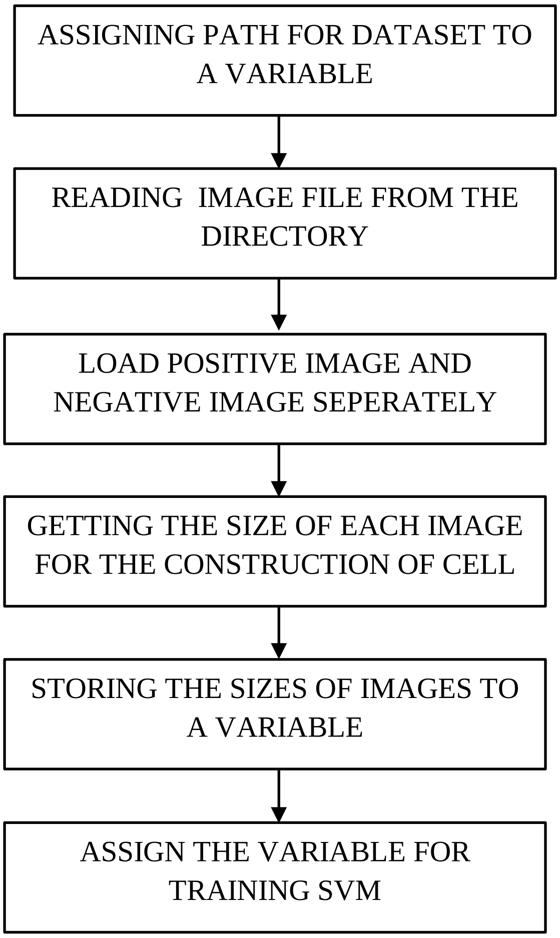


Figure 4.5 Block diagram of algorithm for loading process

4.2.3 Algorithm Creation of model

In the Figure 4.6 shows the process involved in creation of model starting from assigning the path since there are two types of image set one is positive(file consist of one type of defect) and another is negative(this file consist of all other defect except the defect in positive file) to read the images first path should be assigned to respective directory next the path is assigned to the variable similarly the process is repeated to all defects and its path is assigned to respective variables. These variables are then applied to feature extracting function which will be discussed in detail later after applying the feature extracting function, loading functions are being used to load since the images should be loaded separately one by one and there should be segregation between positive and negative images.

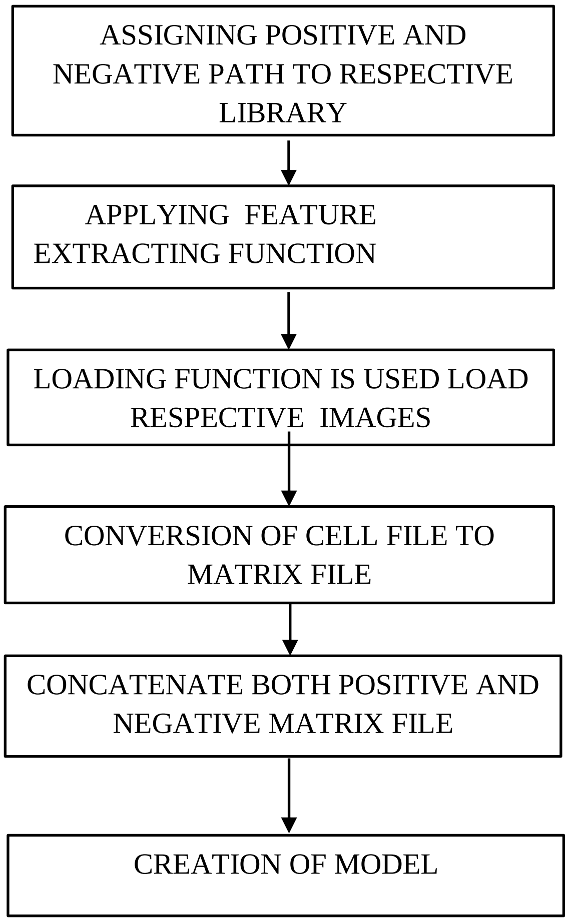


Figure 4.6 Block diagram of algorithm for creation of model

After completion of all these operation a cell format feature extracted file is generated to both positive and negative which should be converted to matrix format file using built in function cell2mat then two matrix format file is generated which should be concatenated to form a model, thus creation of model is done similarly for each defect each model is created so model 1,model 2,model 3,model 4,model 5(M1,M2,M3,M4,M5 respectively).

4.3 Flow of proposed system

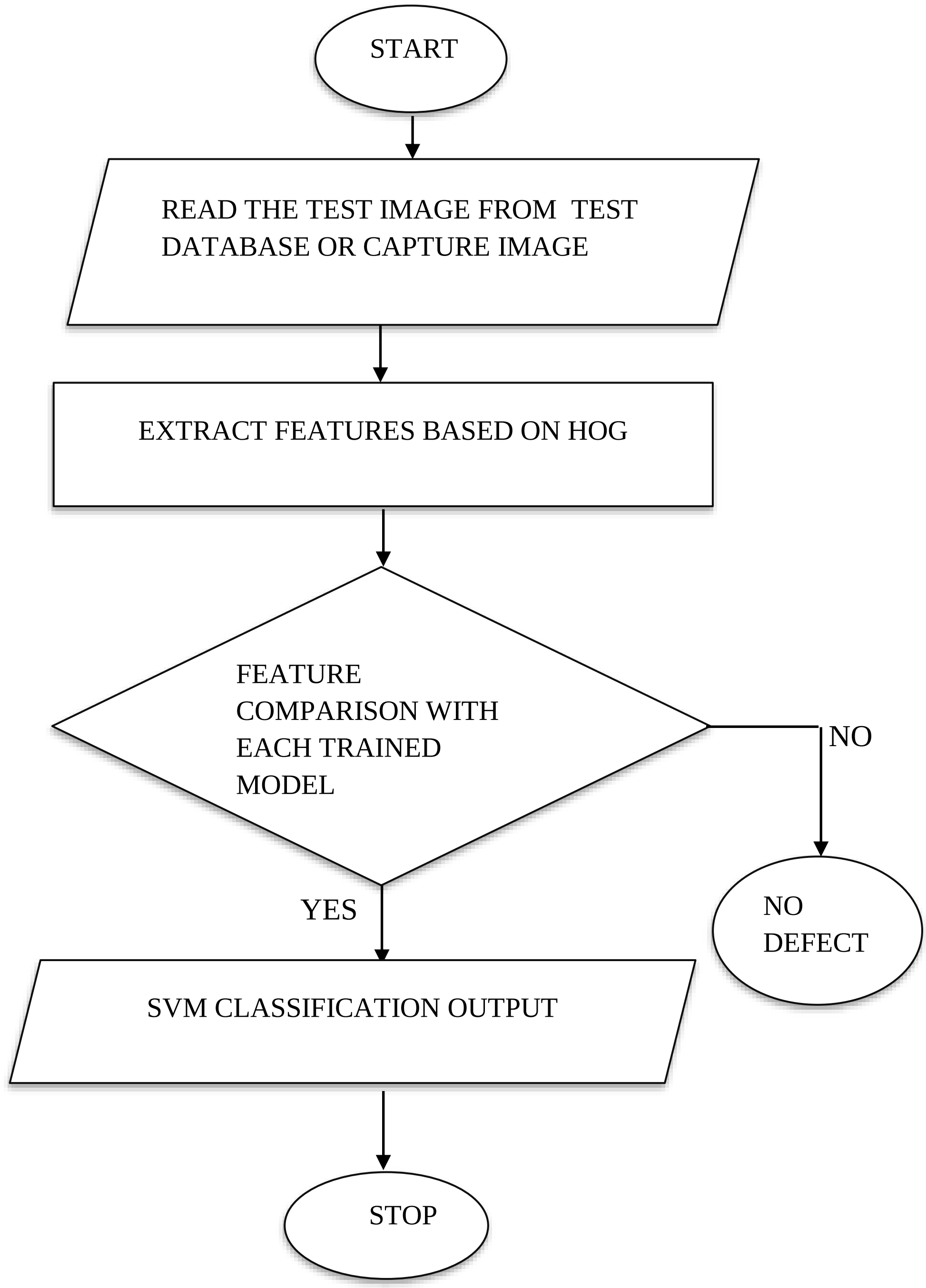


Figure 4.7 Flow chart for proposed system

Flow chart depicted in Figure 4.7 show the flow of the proposed system , step wise as follows:

Step1: Read the test image from the PCB test image data base or capture an test PCB image.

Step2: Extract the features using histogram of gradients algorithms .

Step3: Compare the extracted feature with each trained model i.e extracted feature is compared with model 1 ,model 2,model 3,model 4,model 5 one by one step wise(M1,M2,M3,M4,M5)

Step4: If the extracted feature does not match with the model it will directly assign no defect.

Step5: If extracted Features matches, the model SVM classification is applied .

Step6: Classified defect types like open circuit, short circuit, extra pinhole, missing pinhole or all defect results are displayed in the GUI .

CHAPTER 5

RESULTS AND DISCUSSION

Test PCB image is necessary to check the output for our proposed method of checking the defect .In order to detect the defect in the PCB image training of the system is done using SVM classification of machine learning. Given test PCB image is compared with the each defect database which was created at the backend and classification is done accordingly. In order to check the defect in the PCB image the user can manually give the file location of the test image by clicking on the Browse option on the output window or they can use camera to capture realistic test PCB image directly. In the output window selected PCB image is displayed and after compilation with the database models of each defect one of the following result is displayed on the screen.

- Short circuit
- Open circuit
- Missing pin hole
- Extra hole
- All defects
- No error

To check the accuracy, time, effectiveness of the proposed methods, several experiments were conducted for each defects over total of 480 PCB boards. Accuracy will depend on the clarity of the entered PCB test image and brightness levels are optimistic. Effectiveness of the code will depend on the number of different PCB images used for training the system. If there are more PCB images in each model of the respective defect types. Accuracy and efficiency will increase.

If there are shades on the defect region test while taking photo of the PCB images then the proper output is not shown and user should make in to consideration that if there exist dust, extra gum on the PCB images are considered. The table shows the accuracy of this method.

To compare and evaluate the performance of the SVM classification methods, we used four evaluation criteria: (1) accuracy, (2) true positive rate (TPR), (3) true negative rate (TNR), and (4) precision.

Table 5.1 Performance evaluation table

Defect	Dataset	Positive	Negative	Accuracy(%)	TPR	TNR
Short circuit	50	44	5	88	0.897	0.103
Open circuit	50	47	3	94	0.940	0.060
Missing pin hole	48	48	0	100	1.000	0.000
Extra hole	49	49	0	100	1.000	0.000
All defects	19	17	2	89.4	0.894	0.105
No error	10	10	0	100	1.000	0.000
Total	226	215	11	95.13	0.955	0.045

1. Accuracy= $\frac{\text{Number of correct output}}{\text{Number of correct output}+\text{Number of false output}} * 100$ ….. 5.1

   for short circuit defect: Accuracy= $\frac{44}{50} * 100$= 88%

2. TPR (true positive rate) = $\frac{\text{positive}}{\text{positive}+\text{negative}}$ ….. 5.2

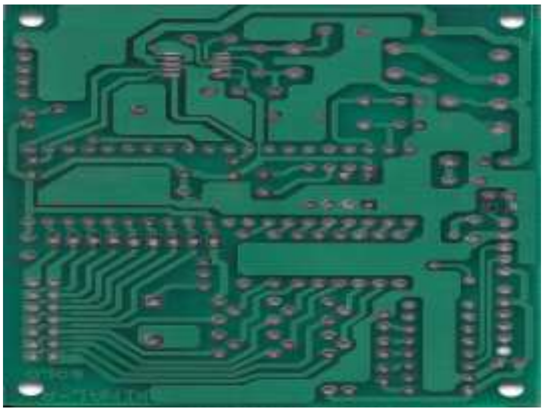
   for short circuit defect TPR= $\frac{44}{49}$ = 0.897

3. TNR(true negative rate)= 1-TPR ..… 5.3

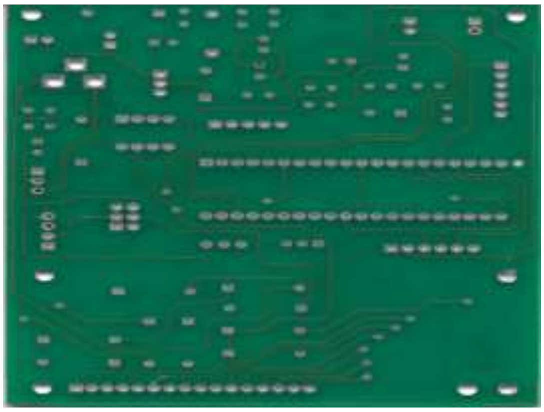
   =1-0.897 =0.103

Input Dataset images

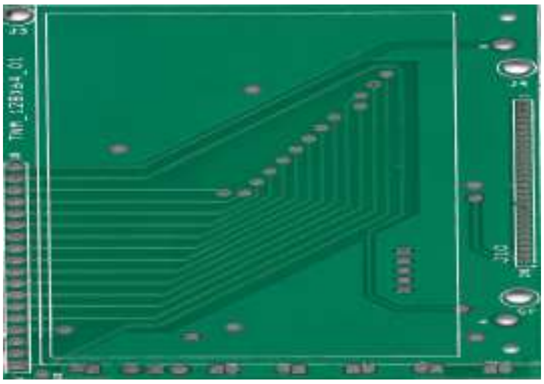
There are total 480 PCB images considered for training the system the number of images of particular defect in training is as shown in table 5.1.In Short circuit defect 50 PCB images which are short circuited are considered for the training the system. Similarly for open circuit defect there are 50 images which are open circuited. Likewise for extra hole and missing hole 49 and 48 images where considered Which are mainly made from the following 10 PCB images i.e, Short circuit, Open circuit , extra hole, missing hole and All type of defect were created to this following images using Photoshop(CS6) and used for training. Along with in order to check No error in PCB following images were used



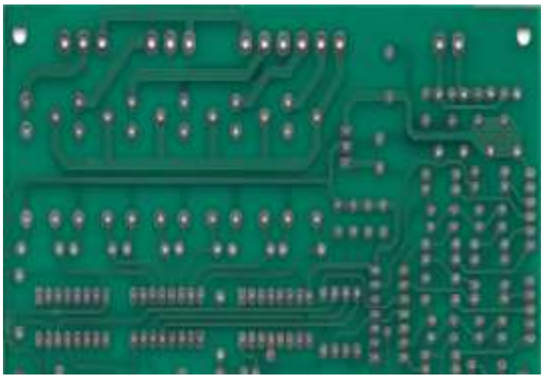
Original PCB 1



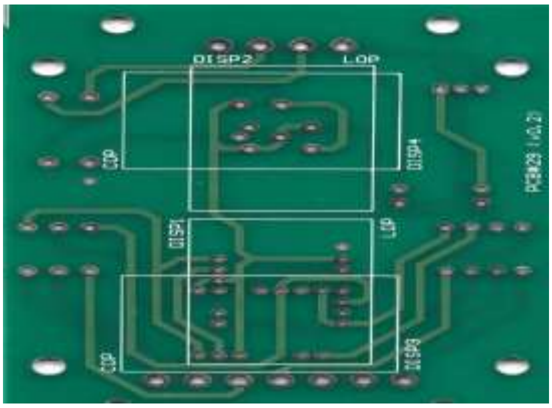
Original PCB 2



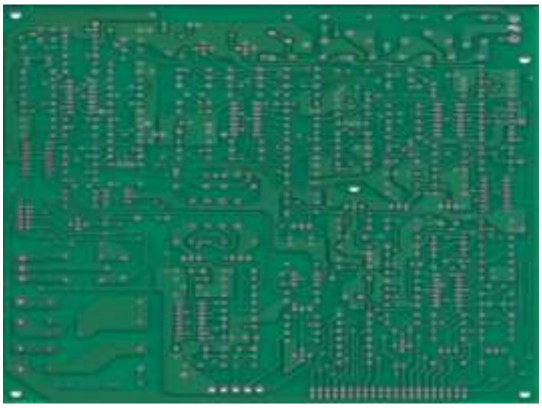
Original PCB 3



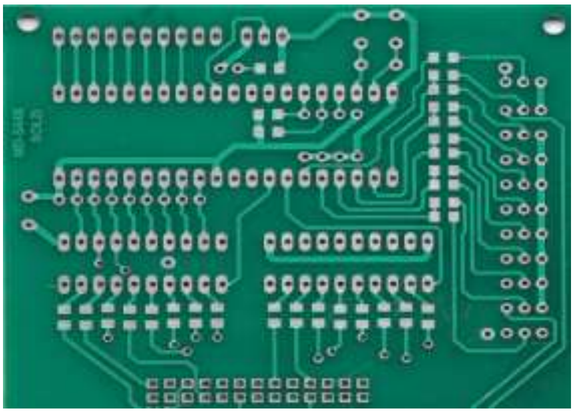
Original PCB 4



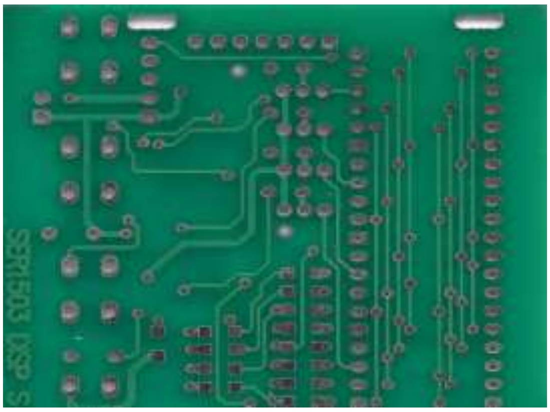
Original PCB 5



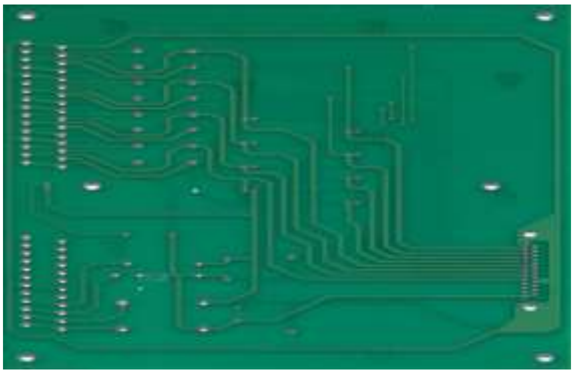
Original PCB 6



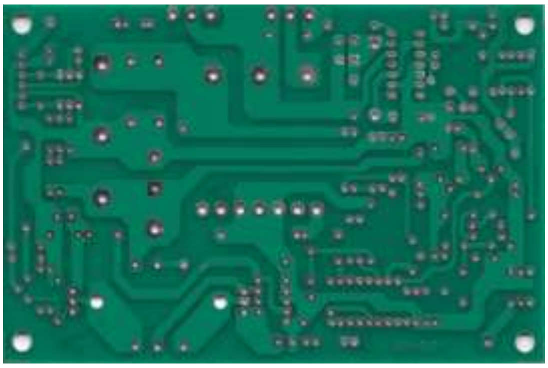
Original PCB 7



Original PCB 8



Original PCB 9



Original PCB 10

5.1 Output images

Output window images of some main kind of defects are shown in this section. Out of 6 outputs short, all defects and No error given. The Algorithms contain different Image Operations each Image Operation gives out a result. This chapter focuses on results of the operations done on different defect PCB image and the final result obtained from these operations. The final result is displayed in the GUI(Graphic User Interface).

5.1.1 Output for short circuit

Short circuit defect will be recognized if the test PCB image is short circuited and the result will be displayed on the GUI at the front end as shown in figure 5.1. The user can select the test PCB image with the short circuit defect on the front end by selecting the PCB image file location the image is displayed on the GUI for user convenience. After this user should click on "**Detect**" button for checking the defect. In the analysis of the correctness of the result there are few experiment were conducted. Total 50 test PCB image with the short circuit defect were tested out of which 44 were shown correct.

Hence the accuracy, TPR, TNR, precision were shown in Table 5.1. The total time taken to show the result is calculated with the function "**tic**" i.e, time in count function is used to calculate.

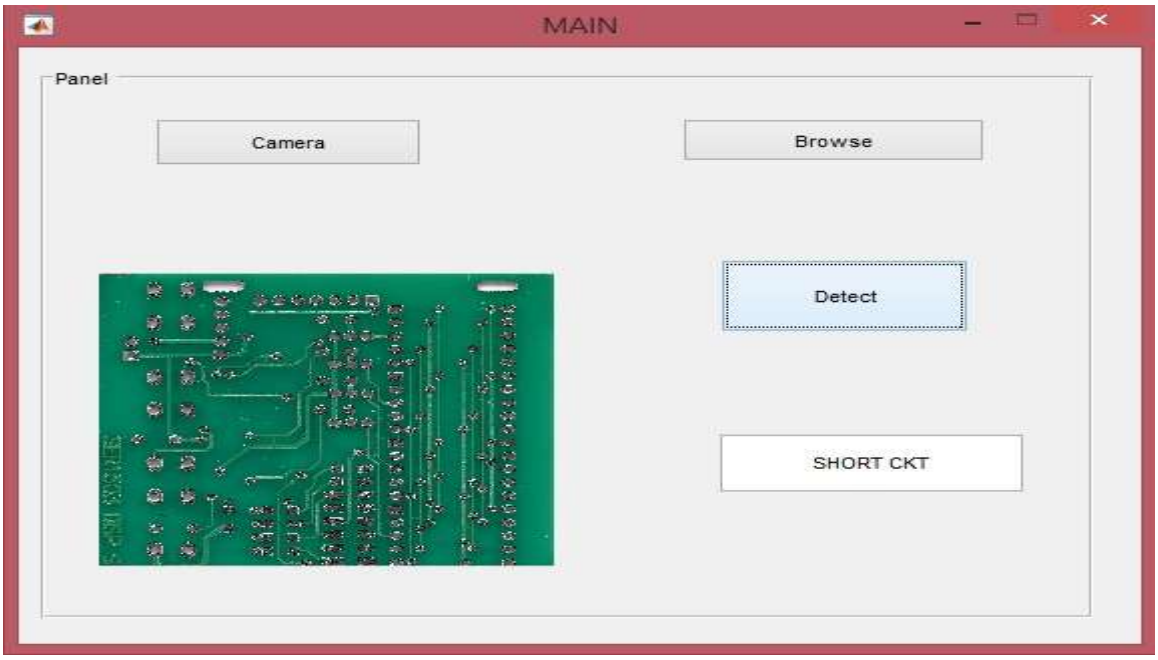


Figure 5.1 Short circuit defect output

5.1.2 Output for All defect

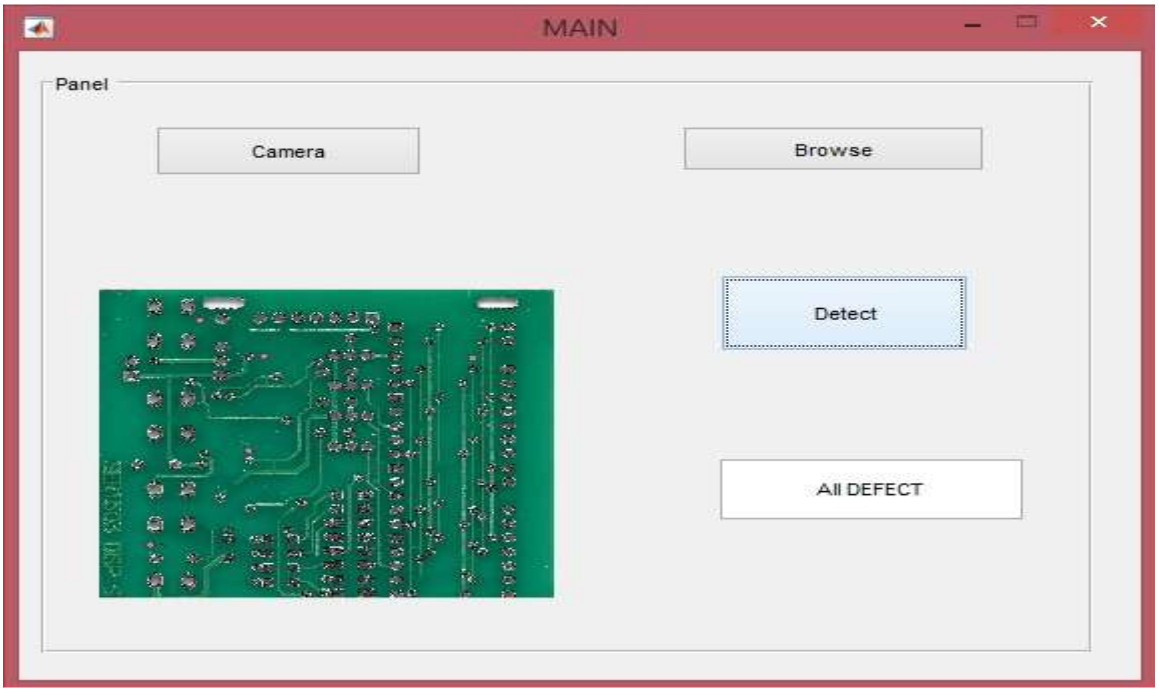


Figure 5.2 All defect output

All type of defect will be recognized if the test PCB image has multiple type of defect in only one PCB and the result will be displayed on the GUI at the front end as shown in figure 5.2. The user can select the test PCB image with the multiple type of defect in only one PCB on the front end by selecting the PCB image file location the image is displayed on the GUI for user convenience. After this user should click on "**Detect**" button for checking the defect. In the analysis of the correctness of the result there are few experiment were conducted. Total 19 test PCB image with the multiple type of defect in only one PCB were tested out of which 17 were shown correct. Hence the accuracy, TPR, TNR, precision were shown in Table 5.1. The total time taken to show the result is calculated with the function "**tic**" i.e, time in count function is used to calculate.

5.1.3 Output for No error

PCB with No defect will be recognized if the test PCB image has no defect in it and the result will be displayed on the GUI at the front end as shown in figure 5.3. The user can select the test PCB image with no defect on the front end by selecting the PCB image file location the image is displayed on the GUI for user convenience. After this user should click on "**Detect**" button for checking the defect. In the analysis of the correctness of the result there are few experiment were conducted. Total 10 test PCB image with no defect were tested out of which 10 were shown correct.

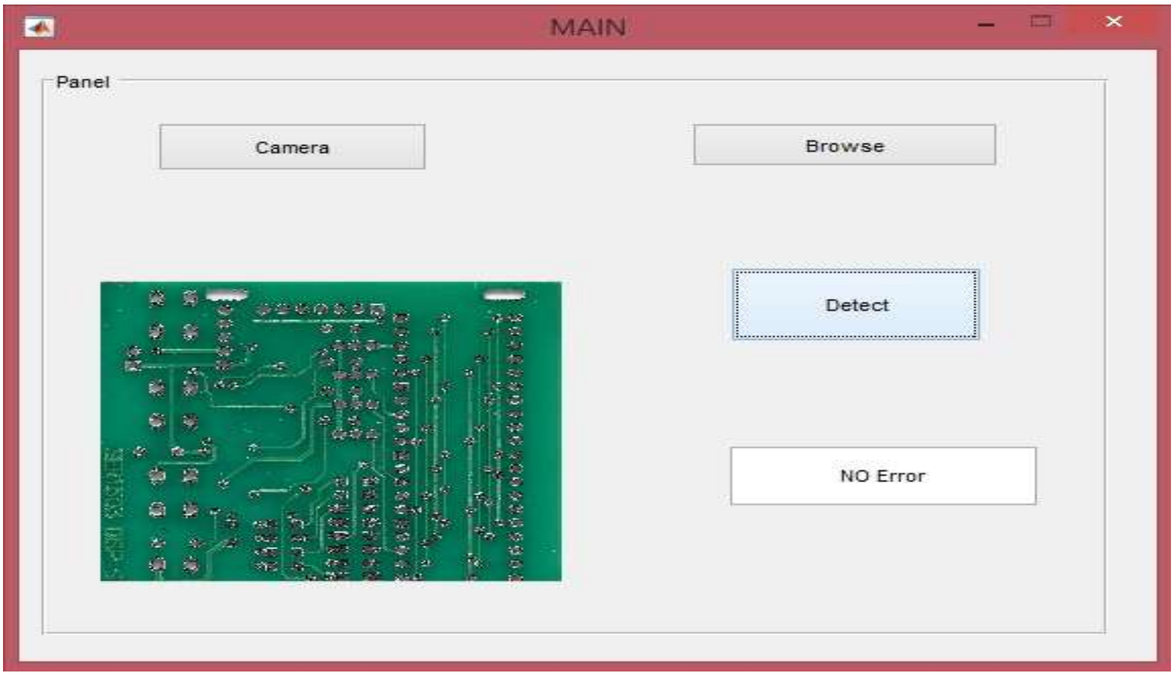


Figure 5.3 PCB with no error output

5.2 Simulation time

Time taken for simulation of defect PCB is recorded as shown in Table 5.2.Analysis is also made by considering the the number of defect in one PCB along with kind of defct it has. Classification is done with respect to PCB with one, two, three and greater than three defect in single PCB. Specified method is followed for all kind of defects. From the table short circuit PCB is having minimum simulation time whereas all defect PCB take maximum simulation time these variation is mainly due to HOG feature extracted values from the given test PCB. Hence HOG value of short circuit is minimum since it contain only values of its defect whereas all defect contain HOG values all type of defects.

Missing hole defect will be recognized if the test PCB image will not have the hole in the specified location in the standard template and the result will be displayed on the GUI at the front end. Open circuit defect will be recognized if the test PCB image is open circuited and the result will be displayed on the GUI at the front end. The user can select the test PCB image with the open circuit defect on the front end by selecting the PCB image file location the image is displayed on the GUI for user convenience. Extra hole defect will be recognized if the test PCB image has extra hole drilled during manufacturing at unexpected spots and the result will be displayed on the GUI at the front end. Hence in this chapter proposed methodology's efficiency and accuracy rate are given along with brief output description.

Table 5.2 Simulation time for number of defects in seconds.

No. of defects in one PCB	Short circuit	Open circuit	Extra hole	Missing hole	All defects
1	0.729	0.732	0.821	0.789	1.129
2	0.792	0.799	0.896	0.855	1.269
3	0.826	0.891	0.915	0.965	1.394
4	0.925	1.093	1.093	1.075	1.593

CONCLUSIONS AND FUTURE WORK

This method presented the implementation of a technique to Detect PCB errors and classify them via MATLAB. Our technique shows that it is feasible to use the MATLAB R2015a version 8.5.0.197613 software for simulation as it has a feature of H.O.G.(Histogram of gradient) feature extraction from a given image and provides a platform for kernel level computation with speed and accuracy compare to earlier versions. This version is best suited for machine learning codes. . Detecting the errors present in PCB so that further manufacturing cost associated with this error PCB can be avoided. Its objective is to detect the errors that are present in PCB during Mass Production. It can also be used in college labs to detect the errors in PCBs present in electronic Instruments or while mounting electronic components over it. We believe that PCB production can be increased efficiently and error rates can be reduced significantly by using this method.

- Reduction of testing time.
- Reduction of human error.
- Automation.
- Reduction of human resource
- It can be used in PCB manufacturing industry to detect the defects in a PCB.
- By counting the changing times of the peripheral boundary's gray value, we can recognize the defect type easily.
- The method in the paper can be also applied to the defect detection and recognition of other flat surface materials.

Defect in the PCB images were detected by applying supervised machine learning method.

In which support vector machine (SVM) classification is applied which extracts H.O.G(Histogram of gradients) features from the image and classify based on it by creating positive and negative models for each defect. At final test image H.O.G. feature are compared with database and the result is showed on the frontend GUI. This method is

user friendly since there is presence of front end where user can access image stored in folders directly and fast there is no need of running the program again.
There is no need of original PCB image for this method to detect the defect in test image. Any PCB board image can be processed without original PCB image of it given for training the system.

PCB defect classification can be done through different methodology. Since SVM classification method works based on extracting the HOG features it is having more efficiency compared to other methodology by this method up to 90% efficiency can be obtained provided more PCB image with best pixel quality has to be used for training the system using machine learning method. If the number of PCB images with respective defect in each defect type is more then efficiency obtained is also more when we test the code if the test PCB image clarity is less then the efficiency may decrease or if there is any dust ,extra gum, extra etching in the PCB in that case also efficiency will decrease.

The region of the defect in test PCB image can also been know through image subtraction, image registration method after knowing which type of defect it has by using above specified methodology. For that user should have original or parent correct PCB image of the test PCB image then only algorithm will work.
In future the two separate algorithm here used to firstly to know which type of defect it has and secondly to find the spot of defect in that particular test PCB image can be interfaced singly using GUI application. Where the two separate results window and can be combined in one single output window to display type and spot of defect.

REFERENCES

[1] Malge, P.S.; Nadaf, R.S. “A survey: Automated visual PCB inspection algorithm. Int. J. Eng. Res.” Technol. 2014, 3, 1

[2] Wu, W.Y.; Wang, M.J.J.; Liu,” C.M. Automated inspected of printed circuit boards through machine vision.” Comput. Ind. 1996, 28, 103–111.

[3] Sundaraj, K.” PCB inspection for missing or misaligned components using background subtraction.” WSEAS Trans. Inf. Sci. Appl. 2009, 6, 778–787.

[4] Chauhan, A.P.S.; Bhardwaj, S.C.” Detection of bare PCB defects by image subtraction method using machine vision.” In Proceedings of the World Congress on Engineering, London, UK, 6–8 July 2011; Volume 2, pp. 103–111.

[5] Moganti, M.; Ercal, F.; Dagli, C.H.; Tsunekawa, S.” Automatic PCB inspection algorithms: A survey.” Comput. Vis. Image Understand. 1996, 63, 287–313

[6] Acciani,G.;Brunetti,G.;Fornarelli,G “.Application of neural networks in optical inspection and classification of solder joints in surface mount technology. Int. IEEE Trans. Ind. Inform. 2006, 2, 200–209.

[7] Huang, S.H.; Pan, Y.C.” Automated visual inspection in the semiconductor industry: A survey.” Comput. Ind. 2015, 66, 1–10.

[8] J. Veraart, D. H. J. Poot, W. V. Hecke, I.Blockx, A. V. der Linden, M. Verhoye, and J. Sijbers, “More accurate estimation of diffusion tensor parameters using diffusion kurtosis imaging,” Magn. Res. Med., vol. 65, no. 1, pp. 138–145, Jan. 2011.

[9] M. Lazar, J. H. Jensen, L. Xuan, and J. A. Helpern, “Estimation of the orientation distribution function from diffusional kurtosis imaging,” Magn. Res. Med., vol. 60, no. 4, pp. 774–781, Oct. 2008.

[10] D.H.J.Poot, A.J.denDekker, E.Achten, M.Verhoyed and J.Sijbers, “Optimal experimental design for diffusion kurtosis imaging,” IEEE Trans. Med. Imag., vol. 29, pp. 819–829, Mar. 2010.

[11] Kelly, C., Siddiqui, F. M., Bardak, B., & Woods, R. (2014). Histogram of oriented gradients front end processing: An FPGA based processor approach. In Proceedings of the 2014 IEEE workshop on Signal Processing Systems Institute of Electrical and Electronics Engineers Inc

[12] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar, “Foundations of Machine Learning”, One Rogers Street Cambridge MA: The MIT Press, 2012.

[13] James Cussens, “Machine Learning”, IEEE Journal of Computing and Control, Vol. 7, No. 4, pp 164-168, 1996.

[14] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica, Vol. 31, No. 3, pp. 249-268, 2013

[15] T. Joachims. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. The MIT Press, Cambridge,MA,USA,1999.

[16] C.J.C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” submitted to Data Mining and Knowledge Discovery, 1998

[17] Rizzi, S.A., Przekop, A., and Turner, T.L., "On the response of a nonlinear structure to high kurtosis non-Gaussian random loadings," EURODYN 2011, 8th International Conference on Structural Dynamics, Paper 41, Leuven, Belgium, July 4-6, 2011.

[18] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, “Online sequential fuzzy extreme learning machine for function approximation and classification problems,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 4, pp. 1067–1072, Aug. 2009.

[19] E. Osuna, R. Freund, and F. Girosi, “An Improved Training Algorithm for Support Vector Machines,” Proc. IEEE Neural Networks for Signal Processing VII Workshop, IEEE Press, Piscataway, N.J., 1997, pp. 276–285.