Universidad Autónoma de Baja California

Instituto de Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería

Contributions to OCR for Unreadable Characters in Printed Circuit Boards by means of Pattern Matching and Machine Learning Techniques

This dissertation is submitted for the degree of

DOCTOR OF PHILOSOPHY

BY

CARLOS FABIAN NAVA DUEÑAS

THESIS DIRECTOR

Dr. FELIX FERNANDO GONZALEZ NAVARRO

THESIS CO-DIRECTOR

Dr. ROBERTO LUIS IBARRA WILEY

Mexicali, B. C. August 2019

Dedication

To Sofia, my life...

Acknowledgemets

Thanks to God for all blessings that He gives me every day of my life. Thanks for the people that loves me and for the ones that I love.
Thanks to Dr. Felix Fernando Gonzalez Navarro, for all the support all these years, I'm so grateful for having learned so much from you.
Thanks to Dr. Roberto Luis Ibarra Wiley, for your wonderful support to start this journey years ago when I was not prepared to do this.
Thanks to my mother, for feeding me and giving me courage to finish what I started.
Thanks to my love Lourdes for your lovely support in the last two years : −)

Abstract

In the last few decades, new computer vision technologies and image processing techniques have been very important in the improvement and automation of manual processes in many technical areas, e.g., in the semiconductor industry. In this thesis, we propose and compare several techniques in the areas of pattern matching and machine learning to have optical character recognition (OCR) of damaged or unreadable numerical digit characters from images on printed circuit boards (PCBs). We describe how the best machine learning algorithms are applied to extract the principal characteristics and features to compute, classify and find the correct numerical character that corresponds to those features. We also present our work in the improvement of the image quality in the pre-processing stage to make pattern matching a good option over some specific conditions of PCBs damage.

Keywords: Machine Learning, Image Processing, OCR, PCA, KNN, Neural Networks, Optics.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Optical character recognition is the recognition and conversion of characters in a digital image to a digital textual. The first beginnings of optical character recognition in a commercial form come from 1950s. After many decades of development, the Optical character recognition is not trivial, since are expecting to have high levels of precision becomes a challenging problem. An optical character recognition engine may fail due to degradation in the image. In this work, a methodology will be carried out to be able to read characters which are subject to surface noise which causes the characters to be incomplete, missing sections for a complete reading, so that algorithms in classification theory and artificial intelligence will be of great help to be able to get to the end of creating a mathematical model for the resolution of the problem.

1.2 Motivation

Since the beginning of the human race thousand of years ago, people have wondered how human vision works to recognize in a fraction of second patterns, shapes, forms, colors, etc. From the middle of last century with the birth of electronics and computers, a mixture between optics and computers have been developing. Now, computer vision and machine learning techniques are walking together in the same path to develop and apply intelligent algorithms to recognize patterns artificially. These improvements bring us one step closer to solving vision problems as optical character recognition of damage or incomplete characters. Every day we are closer to develop algorithms close to the human cognitive and use cameras and process that images like our brain and eyes can do it.

1.3 Problem Specifications

In the semiconductor industry, there are many automatic systems, some of them will need vision systems to improve their processes. Some of these processes could be printed circuits boards (PCBs). They will carry special identification characters, which they are used to keep track of the material over the entire production line and validate in a machine of what is going to be processed a that specific time. These processes need to discard all the material that is damaged, devices that have not being approved for such processes and avoid any damaged material that could be mixed and generated huge production losses.

For this, optical character recognition in production floor is a must. These printed circuit boards could have good characters quality —see Figures 1.1 and 1.2.

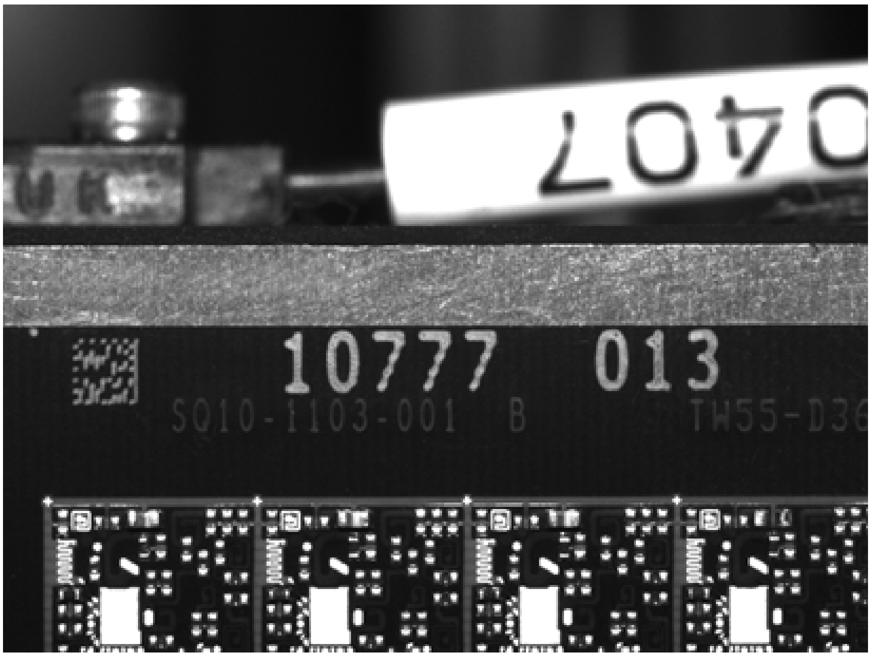


Figure 1.1: PCB with a clean view of characters.

Figure 1.2: Close view of a clean characters zone.

Our developed software can do an optimal optical character recognition using pattern matching, with any problems, when characters are not subject to any superficial damage over the PCB characters zone —see Figure 1.3:

Figure 1.3: Optical Character Recognition in clean characters zone.

There may even be some degree of surface damage on the characters and our systems and still be capable to read them —see Figures 1.4 and 1.5.

Figure 1.4: Some degree of superficial noise is fine.

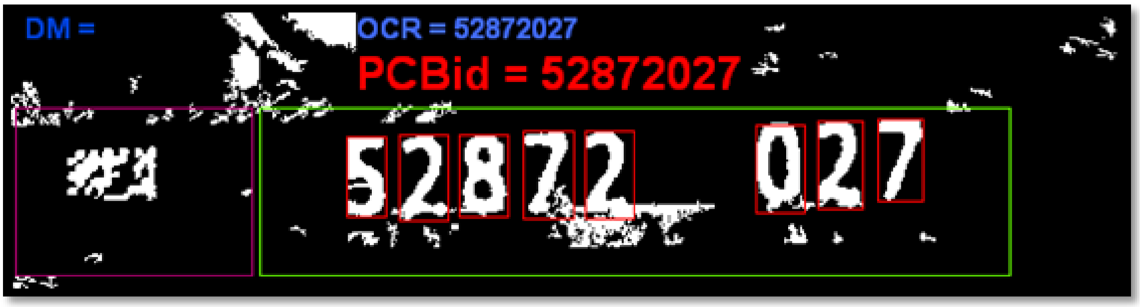


Figure 1.5: Optical Character Recognition with some noise in characters zone.

However, the daily production for PCBs show us that superficial damage is constant, and image quality can be not good —see Figure 1.6.

Figure 1.6: Extreme PCB Dirt Cases

In the following Figure 1.7 we show some early software development for OCR using basic pattern matching in a PCB with extreme dirt.

Figure 1.7: Optical Character Recognition with extreme noise in characters zone.

Our initial research work, has been carried out with the implementation of a character recognition method using matrix comparator. We had some positive results when instead of having a full matrix comparison of images with an specific alphabet we used fragmented and extended alphabets. Our results increase detectability of characters with superficial damage in some degree. In the following Figures 1.8 and 1.9 we show this work.

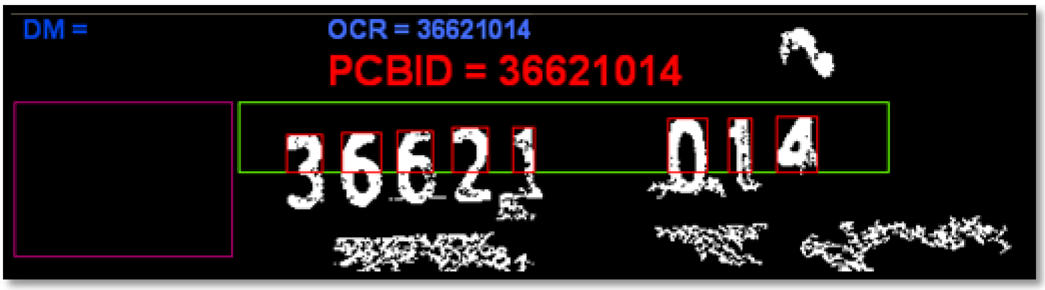


Figure 1.8: OCR Pattern Improvements case 1.

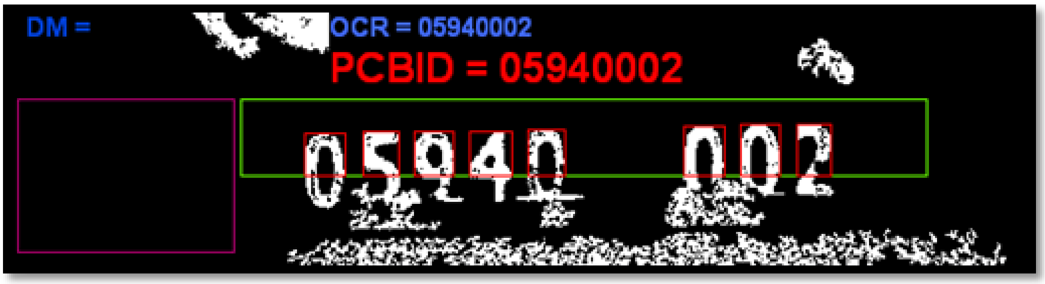


Figure 1.9: OCR Pattern Improvements case 2.

Even though we had good results with initial research, those are not the best we can get for any factory requirements of production line. The purpose of this research is to use different machine learning algorithms, pattern matching techniques and image processing improvements to reach out the best conditions for optical character recognition when characters show high level of superficial damage.

1.4 Objectives and Goals

1.4.1 General Objective

Development and implementation of the best techniques for image processing, pattern matching and machine learning to have the best recognition for damage or incomplete characters in printed circuits boards.

1.4.2 Specific Objectives

1. Characterization of the best optical setups for different machine process that will use applications for optical character recognition.

2. Image Database Bank for training and test machine learning algorithms.

3. Development and evaluation of different machine learning algorithms for optical character recognition.

4. Development and implementation of image processing improvements for better image quality for optical character recognition.

5. Conference presentations, publications, patent protection.

1.5 Methodology

1. Review, analysis and characterization of the main optical setups:
    (a) Cameras.
    (b) Lenses.
    (c) Illumination.
    (d) Frame Grabbers.
        i. GigE.
        ii. IEEE 1394b.
        iii. Smart Cameras.
2. Development and implementation of vision setup.
3. Development and implementation of pre-processing and processing imaging techniques.
4. Development of an image database for training, validation and tests.
5. Development and implementation of pattern matching techniques for optical character recognition.
6. Evaluation of different machine learning algorithms for optical character recognition.
    (a) PCA.
    (b) KNN classifier.
    (c) Bayesian Discriminant LDA and QDA classifiers.
    (d) GSC classifier.
    (e) Naive Bayes classifier.
    (f) Neural Networks.

1.6 Thesis Outline

The content of this thesis is organized in five chapters. Some of them have more details due its complexity, but not less important one to another.

1. Chapter 1. This chapter gives an introduction to the problem, objectives and methodology that will be used to complete this work.

2. Chapter 2. This chapter gives a full State of the Art theory of computer vision, image theory, acquisition, features extraction, best optical hardware and lighting.

3. Chapter 3. This chapter show us the material and methods needed to build an vision setup for any machine process, digit images database and machine learning techniques for the improvement of the OCR for damaged characters.

4. Chapter 4. In this chapter we present the development and results of the machine learning techniques and processing improvements.

5. Chapter 5. This chapter shows the conclusions and future work.

Chapter 2

State of the Art

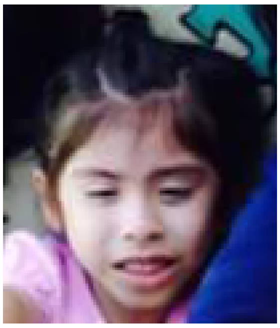
2.1 Computer vision history

Computer vision is the automatic analysis of images and videos by computers in order to obtain information of the real world. Computer vision is based by the capabilities of the human vision system and, when initially started around the 1960s and 1970s, we thought that this problem would be straightforward to solve. However, that was not completely true. The reason is that we think that computer vision is as easy as our own human vision system which makes the visual task seem intuitive to our conscious minds. In fact, the human visual system is very complex and even the estimates of how much of the brain is involved with visual processing vary from 25% up to more than 50%.

Also, we perceive the three-dimensional structure of the world around us with apparent ease but this is not the case with computer vision. Think of how vivid the three-dimensional percept is when you look at a picture of a child, —see Figure 2.1. You can see her beautiful smile shape, the yellowish wall at the background or the detail rainbow patterns of light that play between the wall and the child.

Figure 2.1: The natural human vision system has no problem to identify a child from the background or a rainbow from the foreground. Our human vision can manage color differences, variations in shading or focus and do a correct segmentation of all elements from background

Looking at a framed group portrait —see Figure 2.2, you can easily identify the child from the people at the picture, and even you can guess emotions from their facial appearance. Perceptual psychologists have spent decades trying to understand how the human visual system works and, even though they can devise optical illusions to tease apart some of its principles —see Figure 2.3. (Szeliski, 2010)



(a) Child Template

(b) Group of People

(c) Child Recognition

Figure 2.2: Child Face Recognition

Researchers in computer vision have been developing, in parallel, mathematical techniques for recovering the three-dimensional shape and appearance of objects in imagery. Now, there are many techniques for accurately computing a partial 3D model of an environment from thousands of overlapping photographs. Also, we can track a person or car moving against a complex background. With all of these advances, having a computer recognition of an image at the same level as a toddler child (for example, counting all of the animals in a picture) remains elusive. This is difficult because vision is an inverse problem, in which we seek to recover some unknowns given insufficient information to fully specify the solution. We must therefore resort to physics-based and probabilistic models to disambiguate between potential solutions.

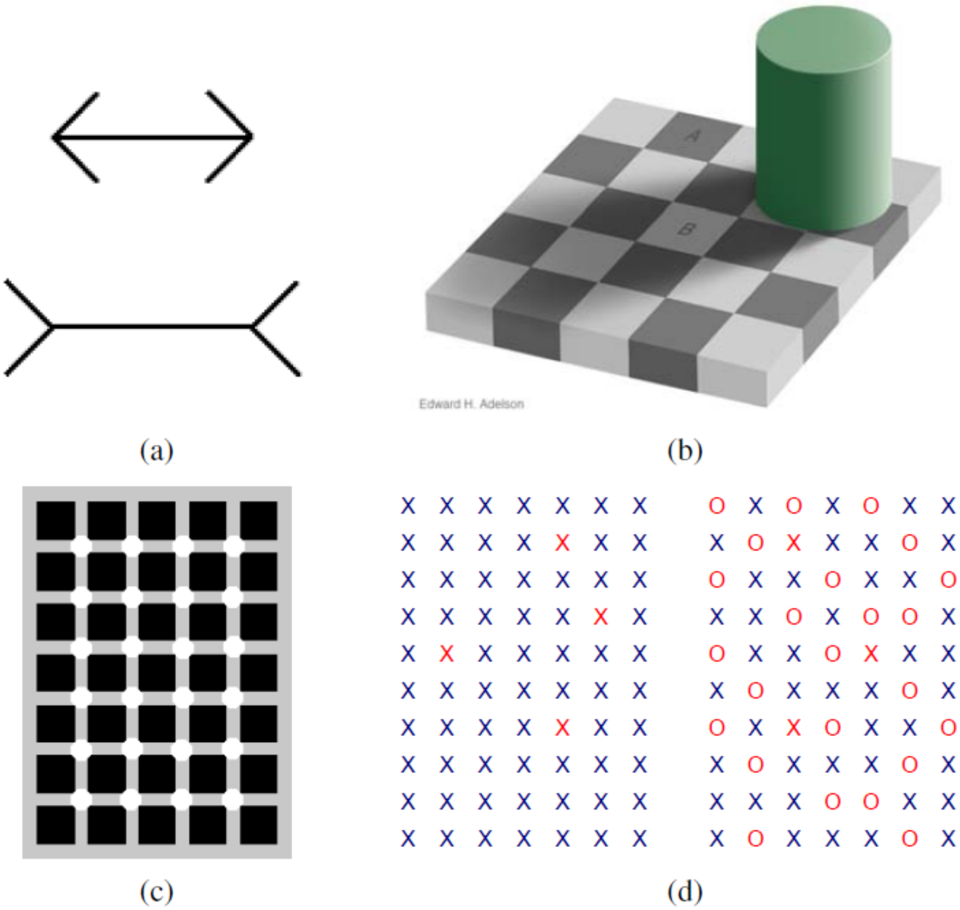


Figure 2.3: Common optical illusions. (a) Classic Muller-Lyer Illusion. (b) White square B in the shadow and the Black A in the light. (c) A variation of the Hermann grid illusion. (d) Counting red X's

In computer vision, we are trying to do the inverse, to describe the world that we see in one or more images and to reconstruct its properties, such as shape, illumination, contrast and color distributions. Humans can do this task so effortlessly, while computer vision algorithms are so error prone, so difficult to develop and apply. The perception that vision should be easy dates back to the early days of artificial intelligence, when it was initially believed that the cognitive (logic proving and planning) parts of intelligence were intrinsically more difficult than the perceptual components. (Boden, 2006)

The first challenge facing anyone working with computer vision is that the problem is difficult. To illustrate this difficulty, we will show three different versions of the same image in Figure 1.1. For a computer, an image is just an array of values, such as the array shown. For us, using our complex human vision system, we can perceive this as a face image in colors. If we apply a color plane extraction we can turn color image to a grayscale or 8-bit image, where we can obtain an array of 8-bit values —see Figure 2.4. Computer vision is like understanding this array, but is more complicated as the array is really much bigger and more complex —see Figure 2.5.

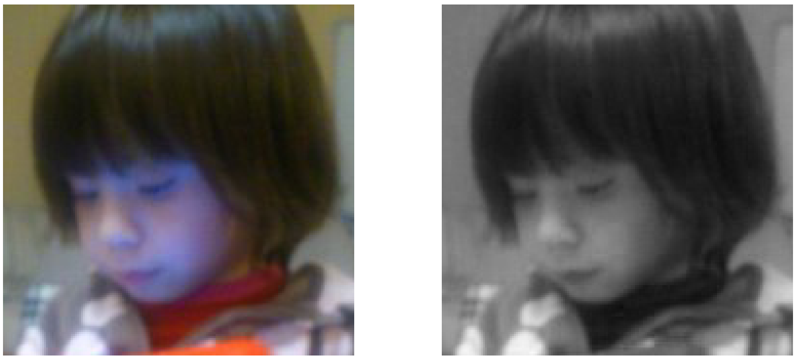


Figure 2.4: Child Face Color-Gray-scale

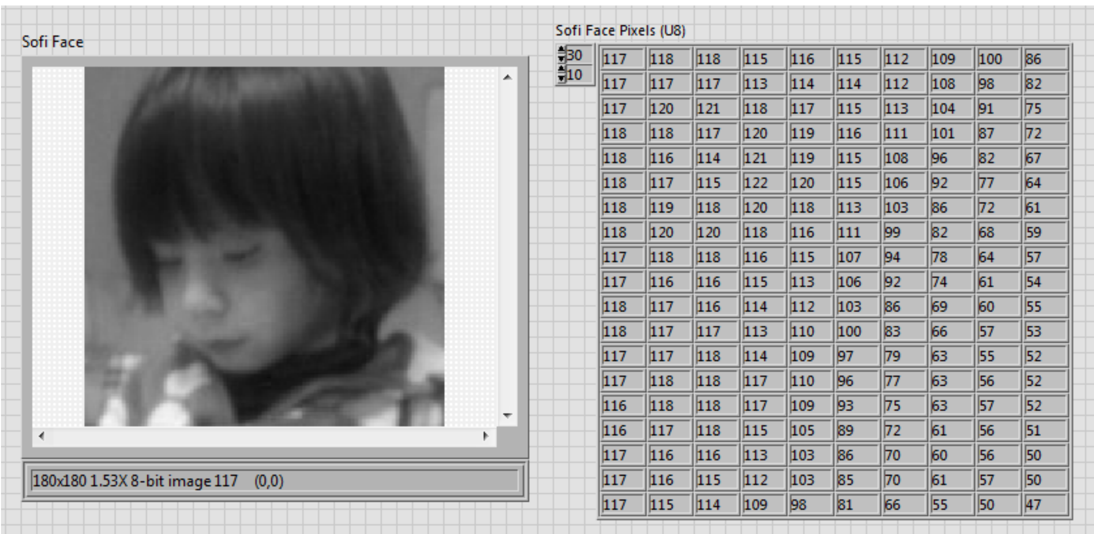


Figure 2.5: Face Gray-scale Array

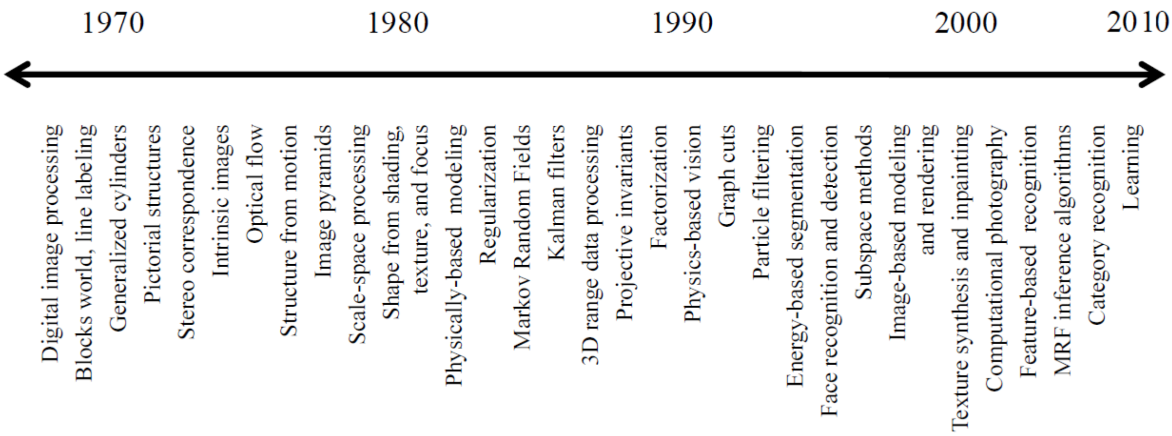


Figure 2.6: Computer Vision Timeline

Constructing computer vision systems seemed to be an unrealistic task for scientists at the middle of the 20th century. Neither engineers nor data analysts were fully equipped to extract information from images. As a result, the idea of information being transformed into editable and systematized data seemed impossible or unrealistic. In practice, it meant that all image recognition services, including space imagery and X-rays analysis, demanded a manual tagging at some point. A basic computer vision timeline is described —see Figure 2.6, from the beginning of digital image processing around 60s to after 2000 first machine learning techniques applications.

- At the beginning of 1960s, scientists were interested in how the connection between an electronic device and data analysis can be achieved. Researchers managed to come up with the first AI programs. These programs were used to teach computers to play checkers, speak English, and solve simple mathematical tasks. Inspired by the achieved results, the scholars got over-enthusiastic about the potential of computers in vision area. The famous Summer Vision Project is an example of how unrealistic most expectations about computer vision can be achieved —see Figure 2.7, (Papert, 1966)
- Another important name in the history of computer vision is Larry Roberts. In his PhD thesis, Machine perception of three-dimensional solids, (Roberts,

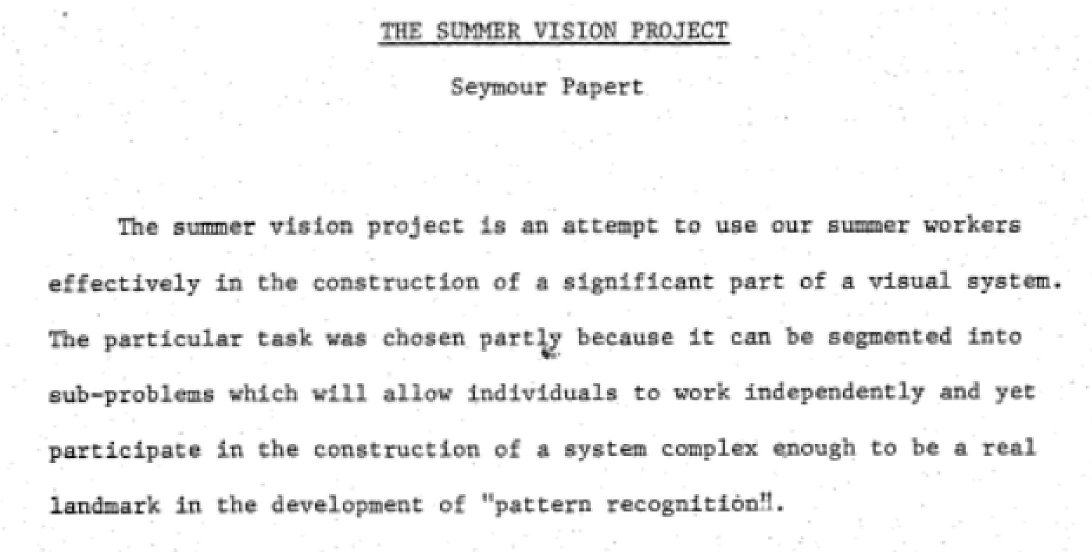
1963), Roberts outlined basic ideas about how one could extract 3D information from 2D images. Roberts laid down the foundations for further research on computer vision technology —see Figure 2.8.

- In the early 1970s, computer vision was viewed as the visual perception component of an ambitious agenda to mimic human intelligence and make robots have intelligent behavior. At the time, it was believed by some of the early pioneers of artificial intelligence and robotics at places mentioned before like (MIT, Stanford), that solving the visual input data problem would be an easy step along the path to solving more difficult problems such as higher-level reasoning and planning —see Figure 2.9, (Szeliski, 2010).

- At the 1980s, research was focused on more sophisticated mathematical techniques for performing quantitative image and scene analysis. Image pyramids started being widely used to perform tasks such as image blending. Continuous versions of pyramids using the concept of scale-space processing were also developed. In the late 1980s, wavelets started displacing or augmenting regular image pyramids in some applications. The use of stereo as a quantitative shape was extended by a wide variety pf techniques as shape from shading, shape from texture, and shape from focus. During this period improvements in edge and contour detection were developed and all these new techniques were unified as a mathematical framework like discrete Markov Random Field (MRF) model, which use a better global search and optimization algorithms. Three dimensional data processing (acquisition, merging, modeling and recognition) was developed at this time also. Researchers started working with optical character recognition (OCR) in different industrial applications to read and verify letters, numbers and symbols. In late 80s, smart cameras emerged with very limited computational ability —see Figure 2.10, (Szeliski, 2010), (Dawson, 2014).

- At the 1990s, computer vision start becoming more common in manufacturing environments leading to the creation of the machine vision industry, where hundreds of companies begin selling machine vision systems. LED lights for the machine vision industry are developed, and advances are made in sensor function and control architecture, furthering advancing the abilities of machine vision systems. Costs of machine vision systems begin dropping. (Systems, 2019). At this time started research in projective reconstructions, factorization techniques (solve efficiently problems for which orthographic camera approximations were applicable), multiview stereo algorithms (produce 3D surfaces) were also an active topic of research. Tracking algorithms also improved a lot, including contour tracking using active contours such particle filters, level sets, as well as intensity-based techniques, often applied to tracking faces and whole bodies. Image segmentation, a topic which has been active since the earliest days of computer vision was also an active topic of research. At this time started appearing statistical learning techniques first in the application of principal component eigenface analysis to face recognition and linear dynamical systems for curve tracking. Perhaps the most notable development in computer vision during the 1990s was the increased interaction with computer graphics, especially in the area of image modeling. The idea of manipulating real world images directly to create new animations with image morphing techniques. At the same time, image based modeling techniques for automatically creating realistic 3D models from collections of images were also being introduced —see Figures 2.11 and 2.12, (Szeliski, 2010).

- After 2000, computer vision has continued to see a deepening interplay between vision and graphics fields. In particular, many of the topics were introduced under the rubric of image based rendering, image stitching, light field capture and high dynamic range (HDR). Image capture through exposure bracketing were developed as computational photography to acknowledge the increased use of such techniques in everyday digital photography. For example, the rapid adoption of exposure bracketing to create high dynamic range images needed the development of tone mapping algorithms to convert such images back to displayable results. In addition to merging multiple exposures, techniques were developed to merge flash images with non-flash images and to interactively or automatically select different regions from overlapping images. Research in texture synthesis can be classified as computational photography technique, since they re-combine input image samples to produce new photographs. Another research trend during the last few years has been the emergence of feature based techniques (combined with learning) for object recognition. Feature based techniques also dominate other recognition tasks, such as scene, panorama and location recognition. While interest point features tend to dominate current research, some groups are pursuing recognition based on contours and region segmentation. Another significant trend has been the development of more efficient algorithms for complex global optimization problems. While this trend began with work on graph cuts, a lot of progress has also been made in message passing algorithms, such as loopy belief propagation (LBP). The final trend, which now dominates the visual recognition research, it is the application of sophisticated machine learning techniques to computer vision problems. This trend coincides with the increased availability of immense quantities of partially labeled data on Databases or the Internet, which makes it more feasible to learn object categories without the use of careful human supervision —see Figure 2.12, (Szeliski, 2010). Researchers at AT&T Bell Laboratories developed convolutional neural networks that were successfully used for zip code recognition by the postal service. This was the beginning of the convolutional neural networks that are widely used in computer vision today —see Figure 2.13, (Lecun, 1989; Li, 2019). Another milestone in the computer vision is the normalized cuts algorithm, where segmentation has the ability to separate an image into distinct and sensible parts. With this, a computer is able to recognize and label segmented parts of an image before combining them and analyzing the picture —see Figure 2.14, (Shi, 2000). Standford researchers created IMAGENET (Li, 2009), putting more than 14 million images classified in more than 22 thousands categories to provide to the public access to data that can be used in object recognition —see Figure 2.15.


THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Figure 2.7: MIT Summer Vision Project

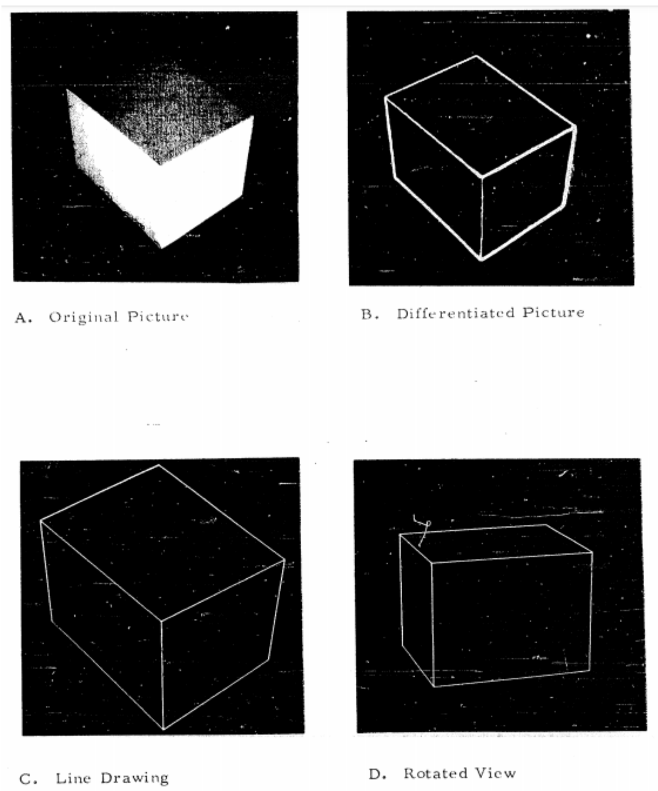


Figure 2.8: Machine Perception 3D Construction

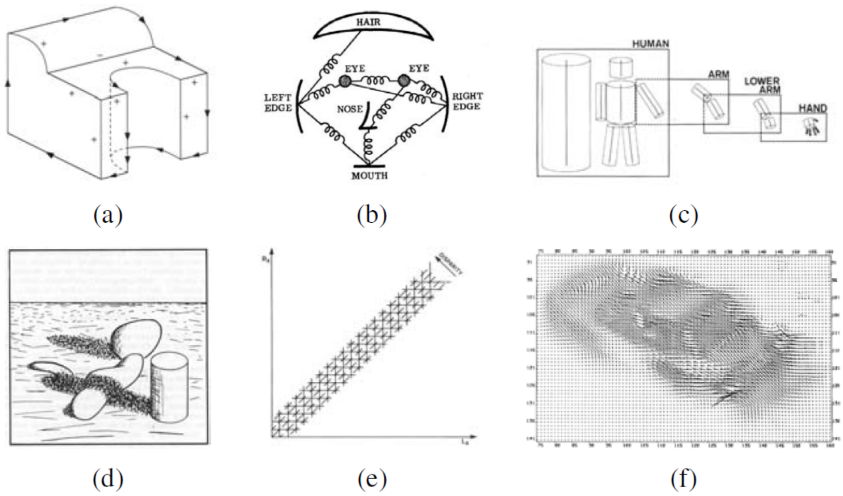


Figure 2.9: 1970s Examples. (a) Line labeling. (b) Pictorial structures. (c) Articulated body model. (d) Intrinsic images. (e) Stereo correspondence. (f) Optical flow

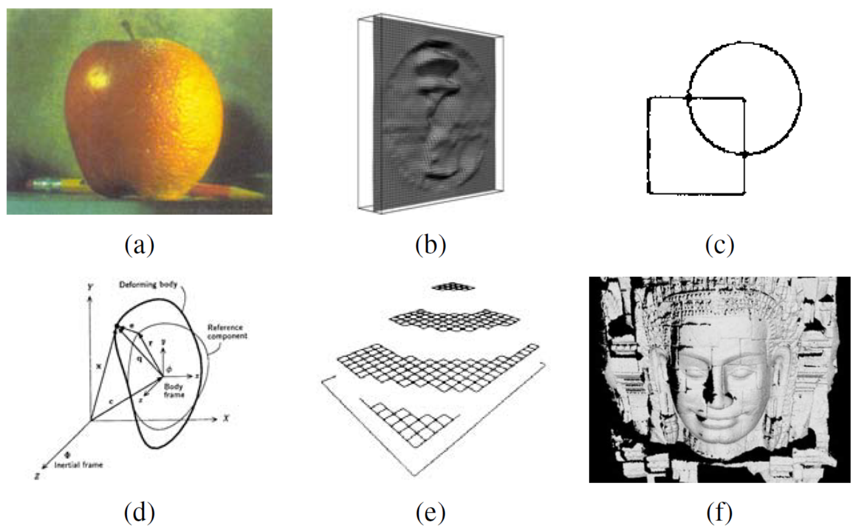


Figure 2.10: 1980s Examples. (a) Orange-Apple Pyramid Blending. (b) Image Shape from Shading. (c) Figures Edge Detection. (d) Physical Based Models. (e) Surface Reconstruction. (f) Data Acquisition and Merging.

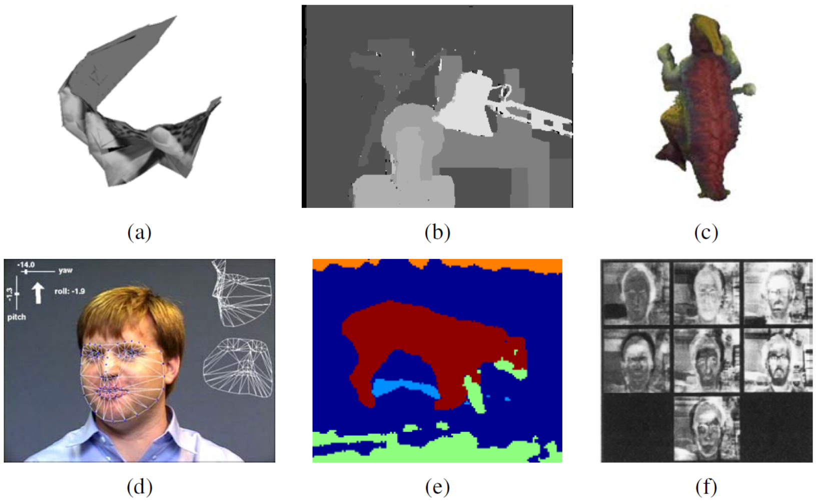


Figure 2.11: 1990s Examples. (a) Structure from motion. (b) Dense stereo matching. (c) Multiview reconstruction. (d) Face tracking. (e) Image segmentation. (f) Face recognition.

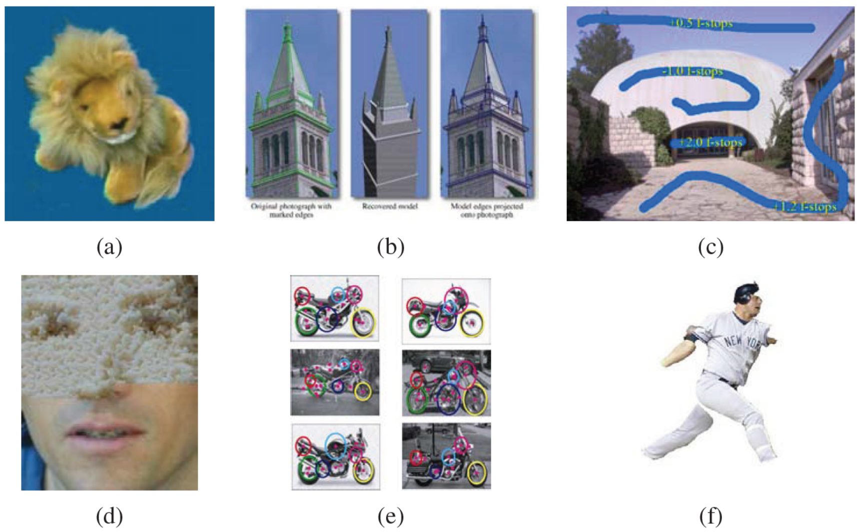


Figure 2.12: 1990s Latest Examples. (a) Image based rendering. (b) Image based modeling. (c) Interactive tone mapping. (d) Texture synthesis. (e) Feature based recognition. (f) Region based recognition.

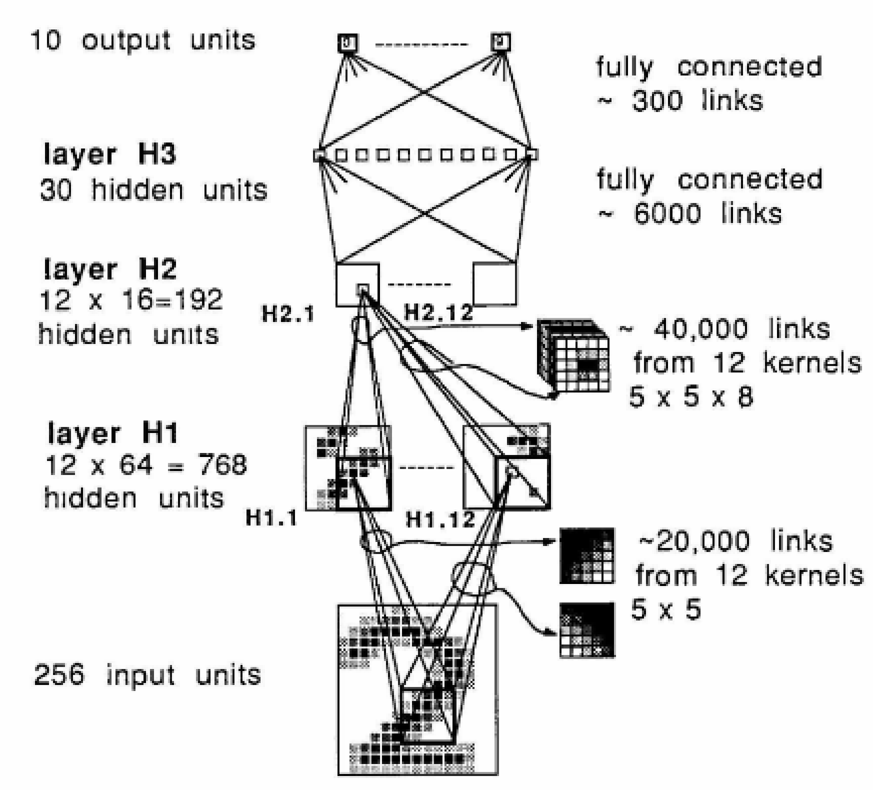


Figure 2.13: Postal Service Convolutional Neural Network

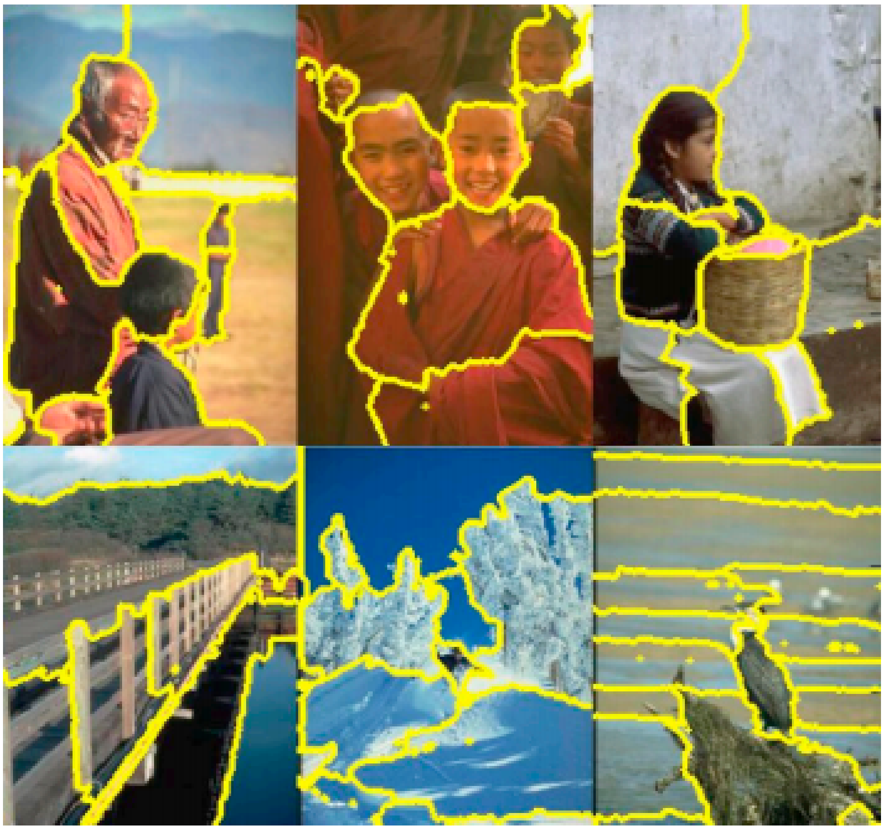


Figure 2.14: Normalized Segmentaion Cut Images

Figure 2.15: IMAGENET. Database with more than 14 million images classified in more than 22 thousands categories

2.2 Image fundamentals, sensing and acquisition

2.2.1 Image Theory

A real image can be defined as a 2 dimensional function $f(x, y)$, where x and y are spatial plane coordinates, and the amplitude of f at any pair of coordinates (x, y) is proportional to the brightness or gray level of the image at that point —see Figure 2.16, (Gonzalez, 2008).

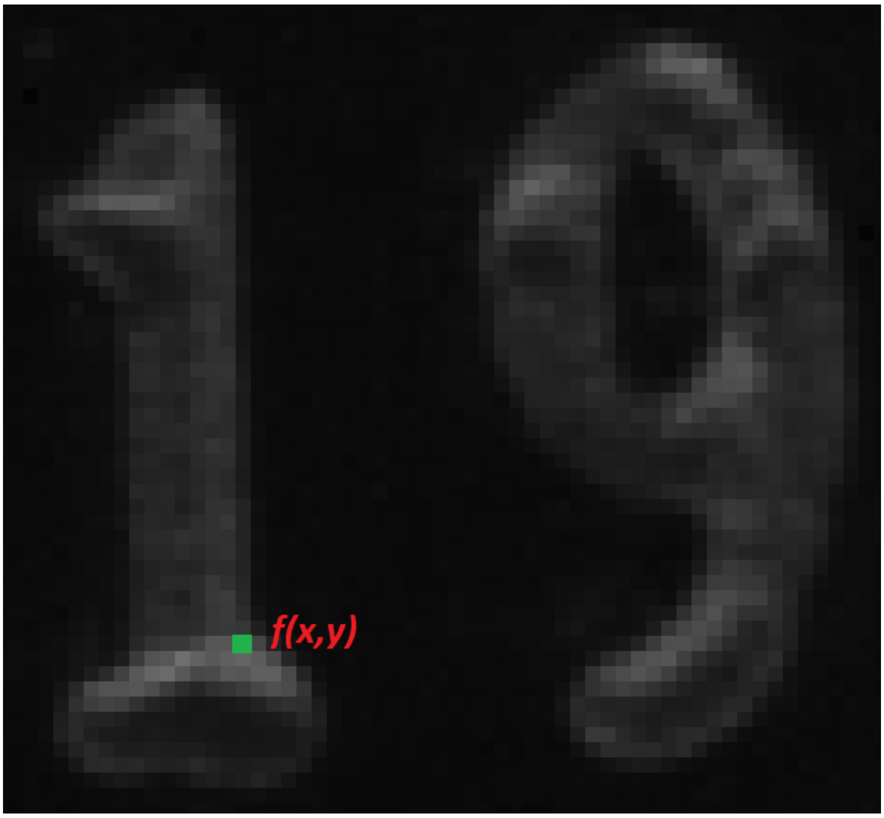


Figure 2.16: Real Grey Image

A digital image is the sampling and quantization of a 2 dimensional real image both in spatial coordinates and brightness. This digital image can be represented as $I(m, n)$ = samples of $f(x, y)$ where m, n are locations integers and I is the intensity values of f where they are all finite, discrete quantities at that location m, n. Here we can note that a digital image is composed of a finite number of elements, each of which has a particular location and a value. These elements are called pixels —see Figure 2.17, (Gonzalez, 2008).

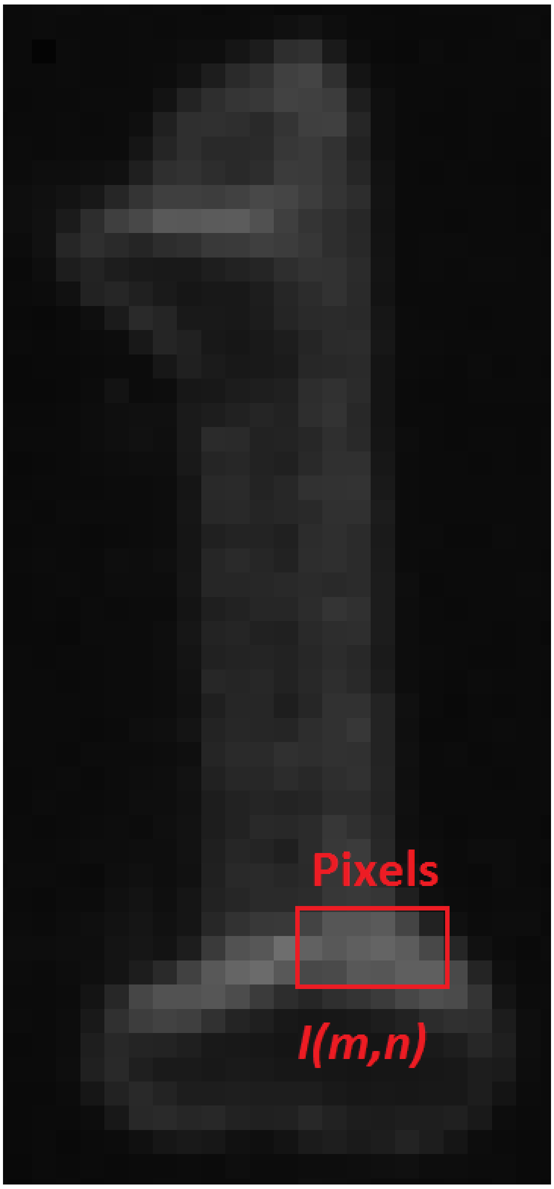


Figure 2.17: Group of Pixels

Digital images are formed by energy interacting with a device that responds in a way that can be measured. These measurements are taken at various points across a two-dimensional grid in the world in order to create the image. These measuring devices are called sensors, and many different types are in use. Sensors may respond to various parts of the electromagnetic (EM) spectrum, acoustical energy, electron beams, lasers, or any other signal that can be measured. The EM spectrum consists of visible light, infrared (IR), ultraviolet (UV), x-rays, microwaves, radio waves, or gamma waves —see Figure 2.18, (Umbaugh, 2010).

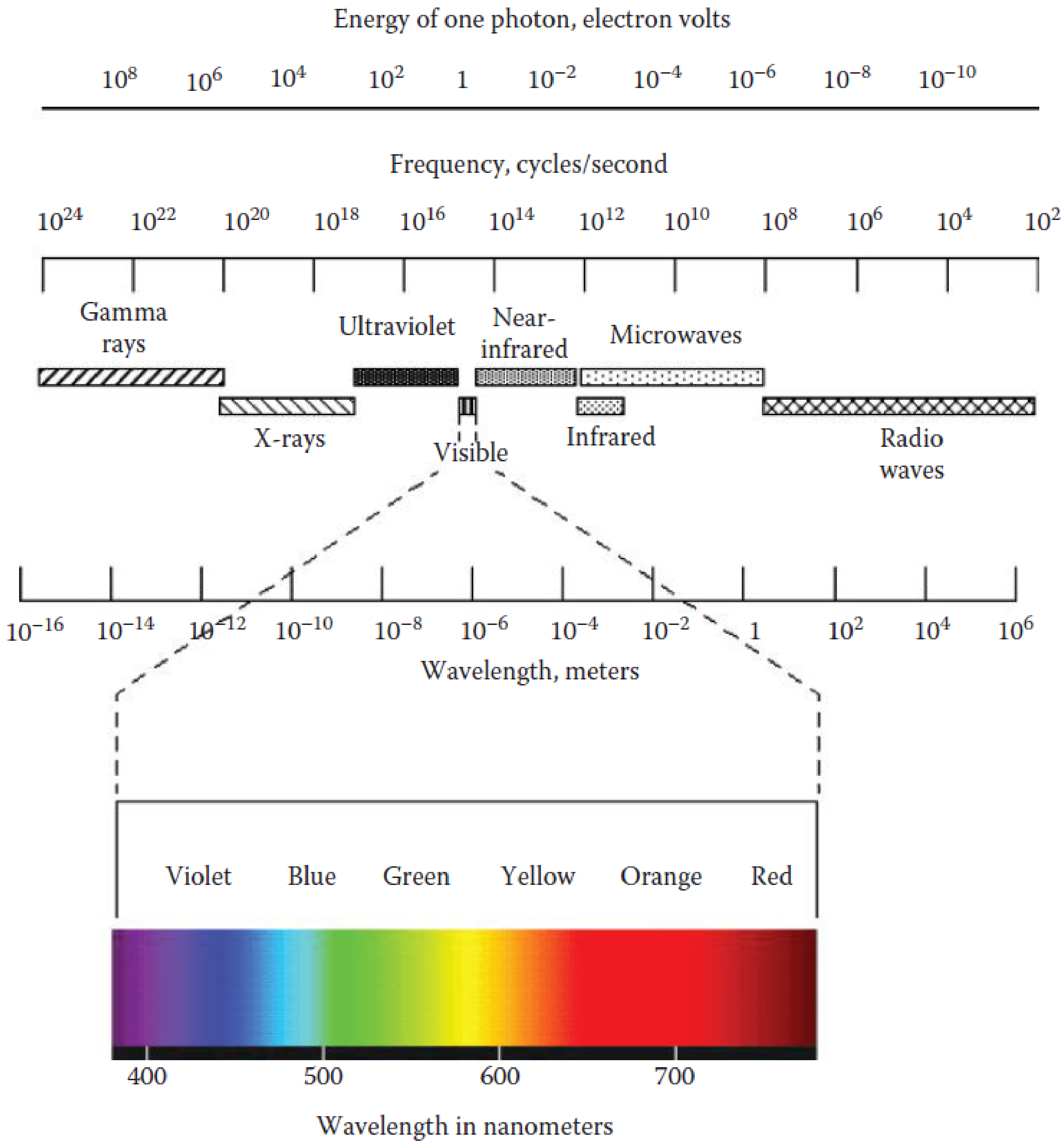


Figure 2.18: Electromagnetic Spectrum

Electromagnetic radiation is formed of electric and magnetic fields that are alternating in a sinusoidal wave traveling at speed of light in free space. Additionally, electromagnetic radiation can be modeled as a massless particles called photons, where they have the minimum amount of energy (quantum)that can be measured in the electromagnetic signal. The energy of a photon is measured in electron-volts, which is the kinetic energy that an electron acquires in being accelerated through an electronic potential of one volt. As we can see at Figure 2.18, when frequency decreases, the energy contained in a photon decreases. For the visible range, wavelengths go from 400 nm (violet) to 700 nm (red) —see Figure 2.19, (Umbaugh, 2010).

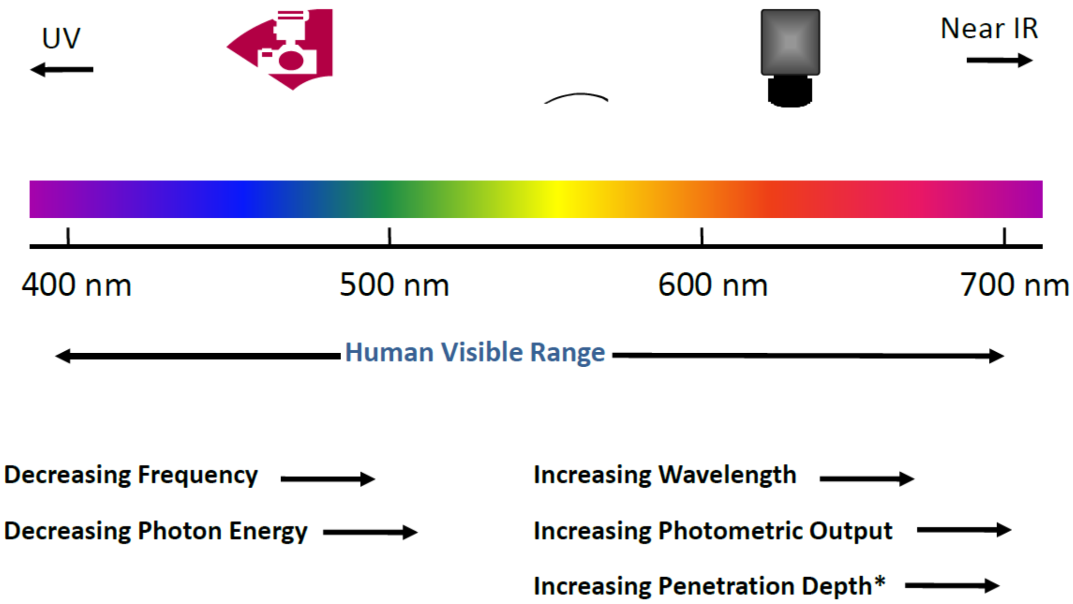


Figure 2.19: Visible Light Spectrum

2.2.2 Visible Light Model

The basic model for visible light imaging is shown at Figure 2.20. Light source emits light that is reflected from the object, and it is focused by the lens onto image sensor. Sensor responds to the light energy by converting it into electrical energy that is then measured. This measurement is proportional to the incident energy, as the brightness of the image at that point. The way an object appears in an image is due the way it reflects light, this is called the reflectance function of the object and is related to the color and texture. The color determines those wavelengths of light that are absorbed and those that are reflected, and the texture determines the angle at which the light is reflected. This is particularly very important in our vision systems that will be explained later on. This indicates how the quality of the image will be used to detect the correct features —see Figure 2.21.

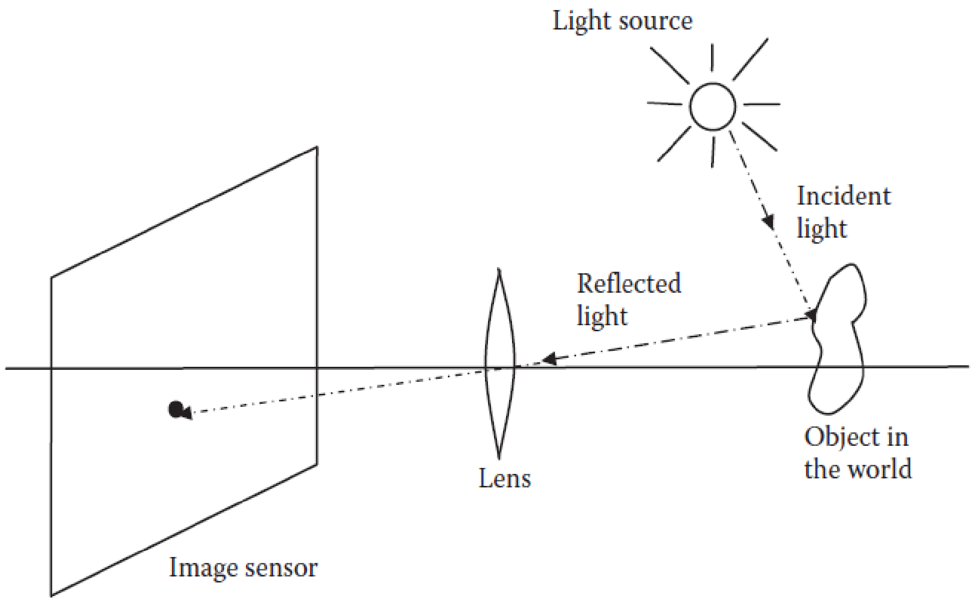
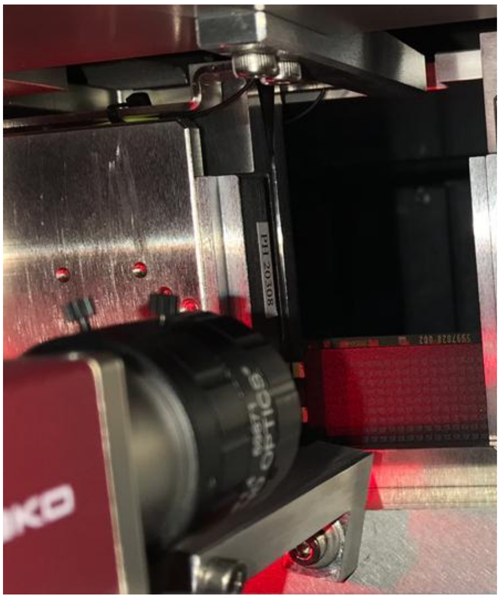


Figure 2.20: Visible Light Imaging Model



(a) Camera-Object

(b) Light-Object

(c) Image-Object

Figure 2.21: Visible Light Model Camera

In imaging, two terms are necessary to define brightness, one is called irradiance, while the light reflected from an object is referred to as radiance. Irradiance is the amount of light falling on a surface, such as an image sensor, while radiance is the amount of light emitted from a surface into a solid unit angle —see Figure 2.22.

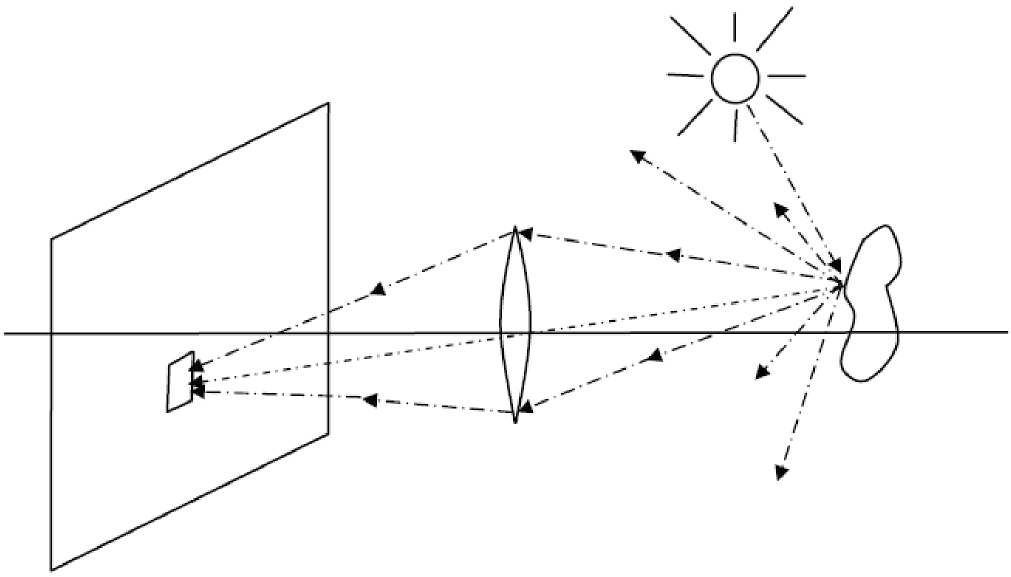


Figure 2.22: Irradiance Radiance Concepts

Most of the images are generated by an illumination source and reflection or absorption of energy from that source by the elements of the imaged scene. The incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor, and a digital quantity is obtained from each sensor by digitizing its response (Gonzalez, 2008). There are three principal imaging sensor arrangements. Single imaging sensors —see Figure 2.23, are typically arranged in lines or in two-dimensional arrays. The line sensor —see Figure 2.24, is typically used in imaging applications that require a single line scan at a time, such as in many manufacturing applications that need high resolution images. With a line scanner speed and resolution can be increased, while cost is minimized. The array (Matrix) sensor —see Figure 2.25, is the primary type used in almost all cameras, and the sensing element is typically a charge-coupled device (CCD) or a complementary metaloxide-semiconductor (CMOS) device (Umbaugh, 2010).

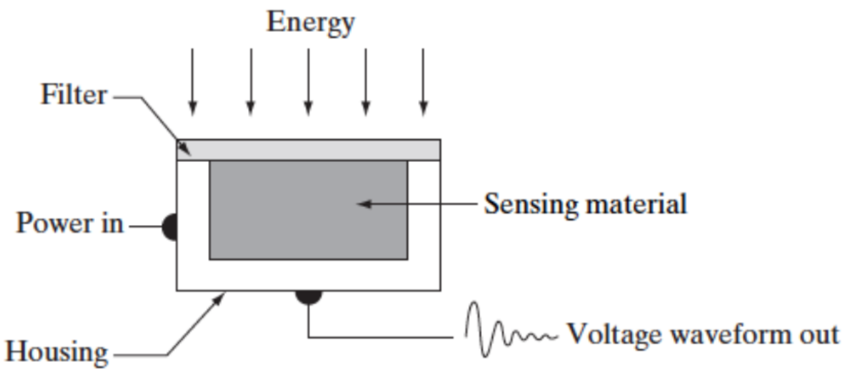


Figure 2.23: Single Imaging Sensor

Figure 2.24: Line Imaging Sensor

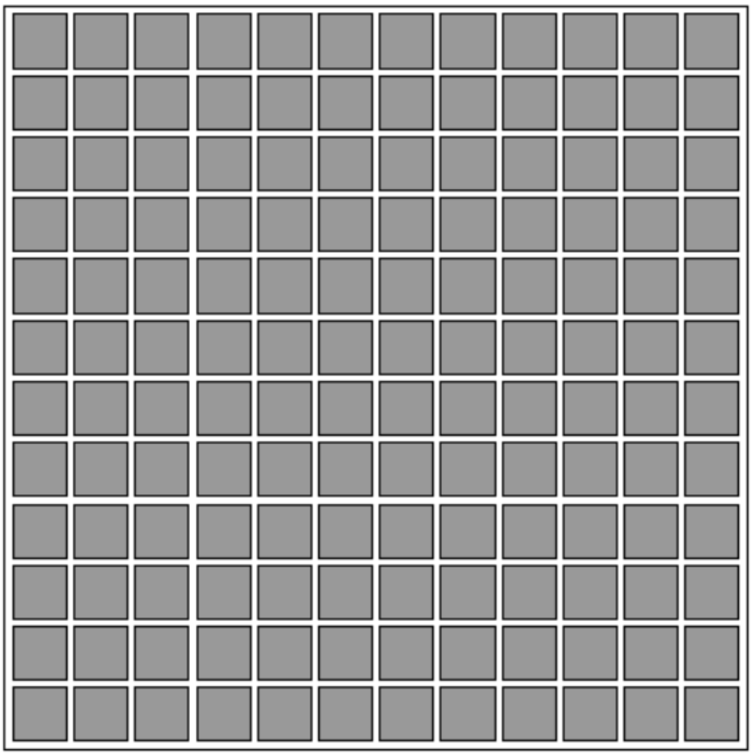


Figure 2.25: Array (Matrix) Imaging Sensor

The first function performed by the imaging system is to collect the incoming energy and focus it onto an image plane. If the illumination is light, the front end of the imaging system is a lens, which projects the viewed scene onto the lens focal plane. The sensor imaging array, which is coincident with the focal plane, produces outputs that are proportional to the integral of the light received at each sensor. Digital and analog circuitry convert these outputs to a video signal, which is then processed by another section of the imaging system. The output is a digital image. This image is form when photons strike the capture sensor where electron-hole pairs are generated on sensor sites. Those electrons that were generated are collected over a certain period of time. The number of electrons are converted to pixel values —see Figure 2.26. (Gonzalez, 2008).

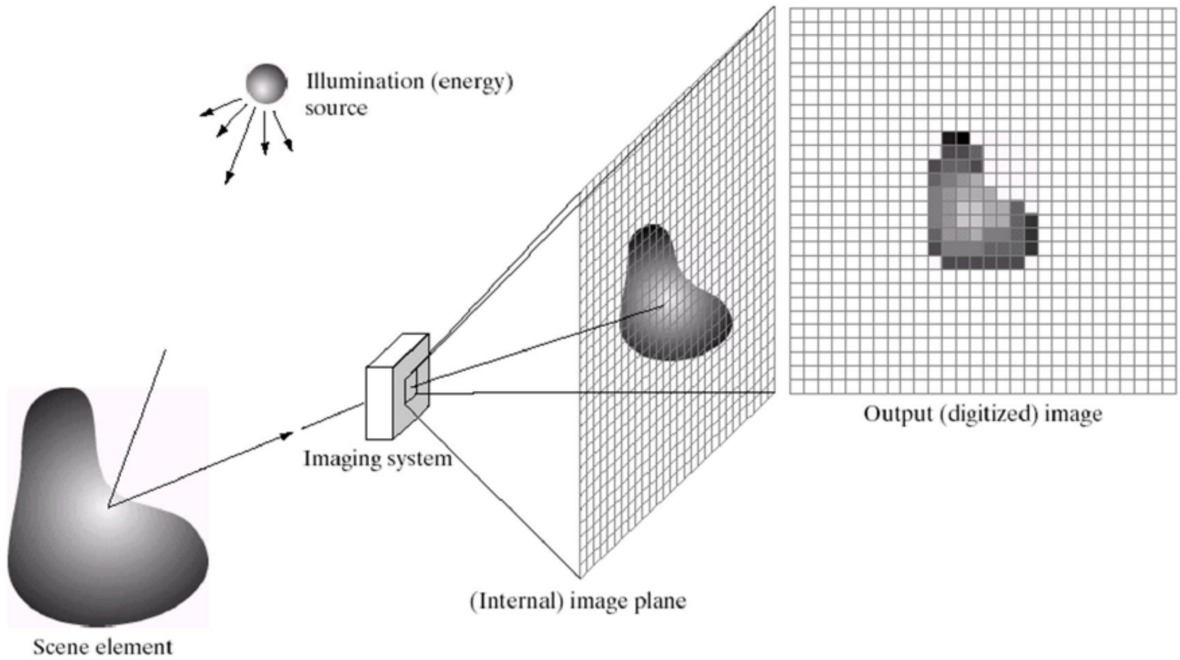


Figure 2.26: Image acquisition process

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two types of discretization processes: sampling and quantization. Digitizing a finite number of pixels, the coordinate values, is called sampling (Spatial Resolution) and digitizing the amplitude values, by a finite number of bits, is called quantization (Gray-scale resolution). The quality of the digital image is determined to a large degree by the number of samples, spacial coordinates (x, y), and discrete intensity I levels used in sampling and quantization —see Figures 2.27 and 2.28.

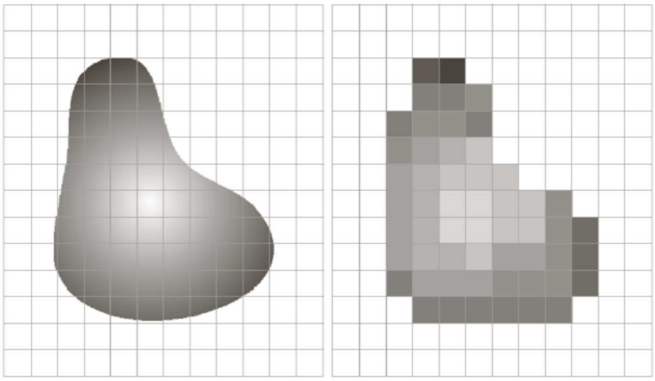


Figure 2.27: Image projected onto a sensor array and the sampling-quantization image.

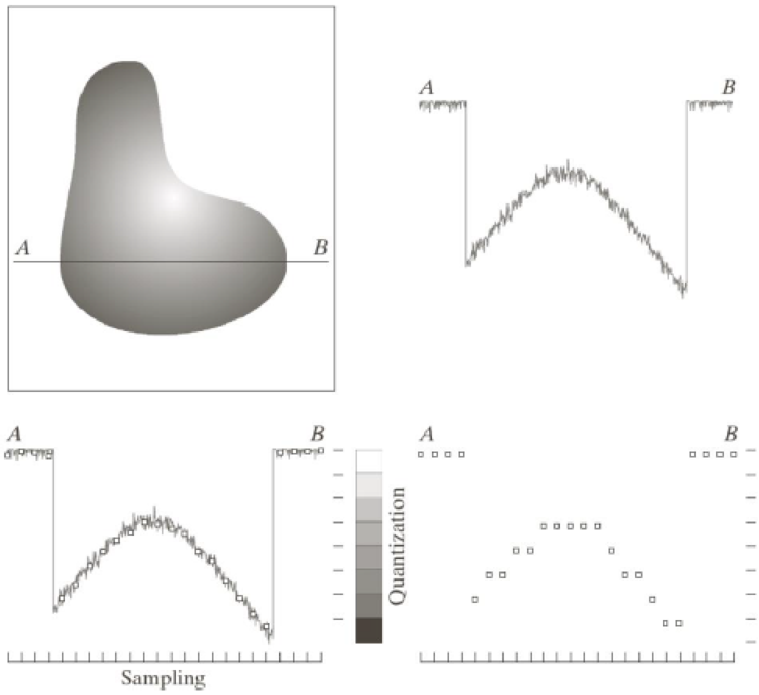


Figure 2.28: Process to generate a digital image. A continuous image and a scan line from A to B in the Figure upper level. Sampling and quantization and the final digital scan line at the Figure bottom.

2.3 Image Pre-processing and processing techniques

We define image pre-processing to be any transformation of the pixel data before to build the model that relates the data to the world. Such transformations are often ad-hoc heuristics: their parameters are not learned from training data, but they are chosen based on experience of what works well. The image data may be formed with some aspects of the real world that do not necessary are the best data to be analyzed . For example, in an object detection task, the RGB values will change depending on the camera gain, illumination, object pose and particular instance of the object. The goal of image pre-processing is to remove as much of this unwanted variation as possible, while retaining the aspects of the image that are critical to for the final decision. In a sense, the need for pre-processing represents some degree of failure. We are admitting that we cannot directly model the relationship between the RGB values of an image and the world state. Inevitably, we must pay a price for this (resources, processing time). It is very probable that some of the task-related information is also discarded. Fortunately, in these recent years of computer vision, this rarely seems to be the limiting factor that governs the overall performance. It should be emphasized that image pre-processing is very important technique that can influence the performance of vision systems at least as much as the choice of model.(Prince, 2012)

Pre-processing an image so that the resulting image is more suitable than the original for a specific application. An image pre-processing method that works well for one application may not be the best method for another application —see Figure 2.29.

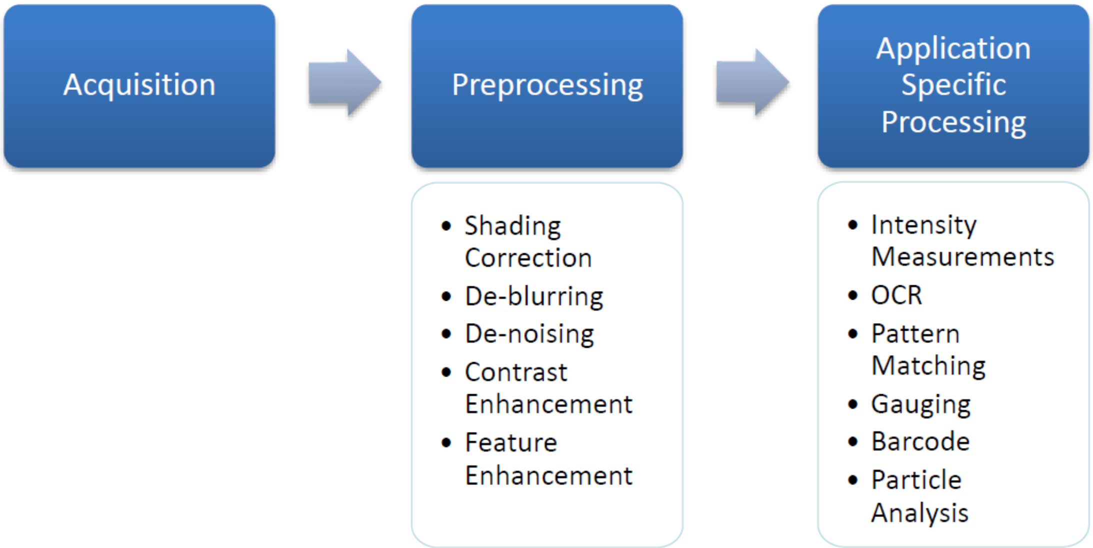


Figure 2.29: Image Pre-procesig Stages.

In a more detail image analysis process, illustrated in Figure 2.30, can be broken down into three primary stages: (1) Preprocessing, (2) Data Reduction, and (3) Feature Analysis. Preprocessing is used to remove noise, and eliminate irrelevant, visually unnecessary information as we explained before. Noise is unwanted information that can result from the image acquisition process. Other preprocessing steps might include gray level or spatial quantization (reducing the number of bits per pixel or the image size), or finding regions of interest for further processing. The second stage, data reduction, involves either reducing the data in the spatial domain and/or transforming it into another domain called the frequency domain —see Figure 2.31 and then extracting features for the analysis process. In the third stage, feature analysis, the features extracted by the data reduction process are examined and evaluated for their use in the application. (Umbaugh, 2010).

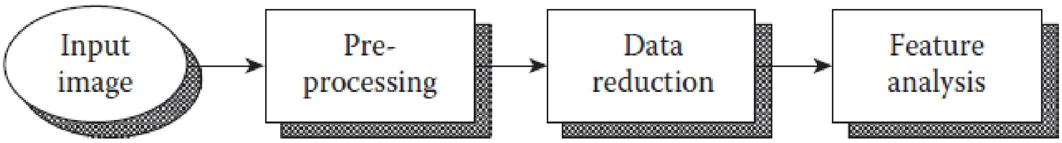


Figure 2.30: Image analysis.

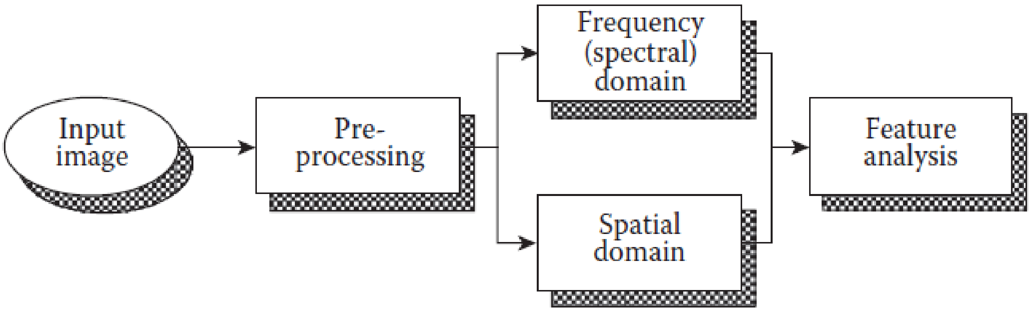


Figure 2.31: Image analysis domains.

A more detailed diagram of this process is shown in Figure 2.32. After pre-processing we can perform segmentation on the image in the spatial domain or convert it into the frequency domain via a mathematical transform. Between segmentation and the transform block we can extract spectral features on segmented parts of the image. These processes we may choose to filter the image. This filtering process further reduces the data and allows us to extract the features that may be required for analysis. After the analysis, we have a feedback loop that provides for an application review of the analysis results. This approach often leads to an iterative process that is not complete until satisfactory results are achieved. The application feedback loop is a key aspect of the entire process.

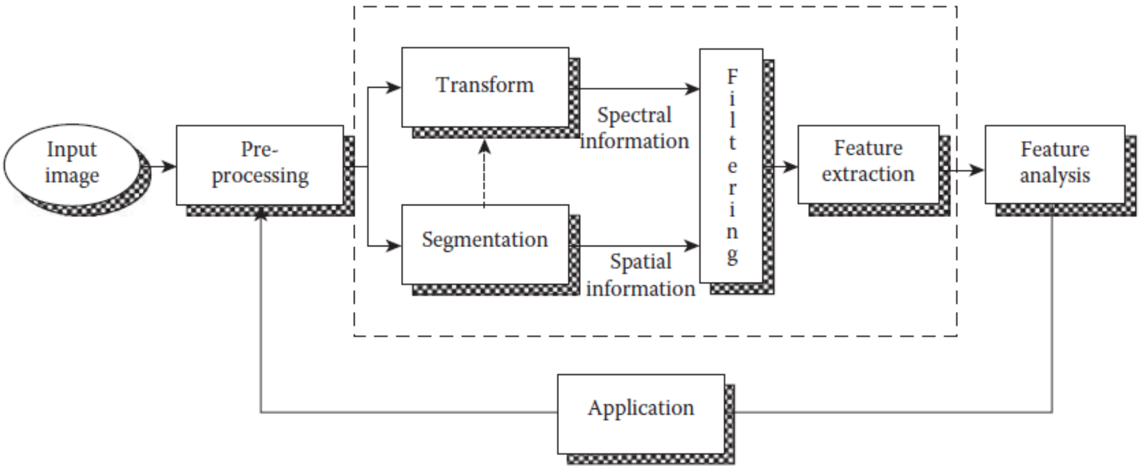


Figure 2.32: Image analysis detail.

Image pre-processing may be a manipulation of an image to isolate features or improve feature contrast for more reliable and faster image analysis. Some general examples:

- Eliminate noise.
- Change contrast or brightness.
- Extract edges or colors.
- Reduce background content.
- Shifting.
- Rotating.
- Mirroring.
- Inverting.
- Unwrapping.
- Sampling.
- Binarization.
- Warping-Shifting.
- Morphology.
- Spatial Filtering.
- Whitening.
- Histogram equalization.
- Linear filtering.
- Local binary patterns.

We can show some examples in the following Figures 2.33, 2.34 and 2.35.

Figure 2.33: Whitening and Histogram equalization. These transformations reduce variation due to contrast and intensity changes

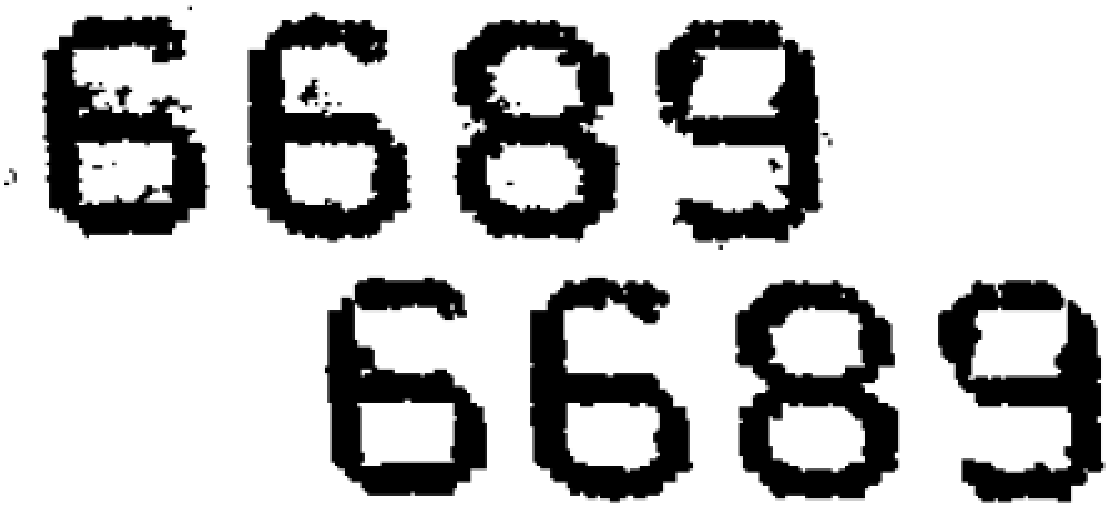
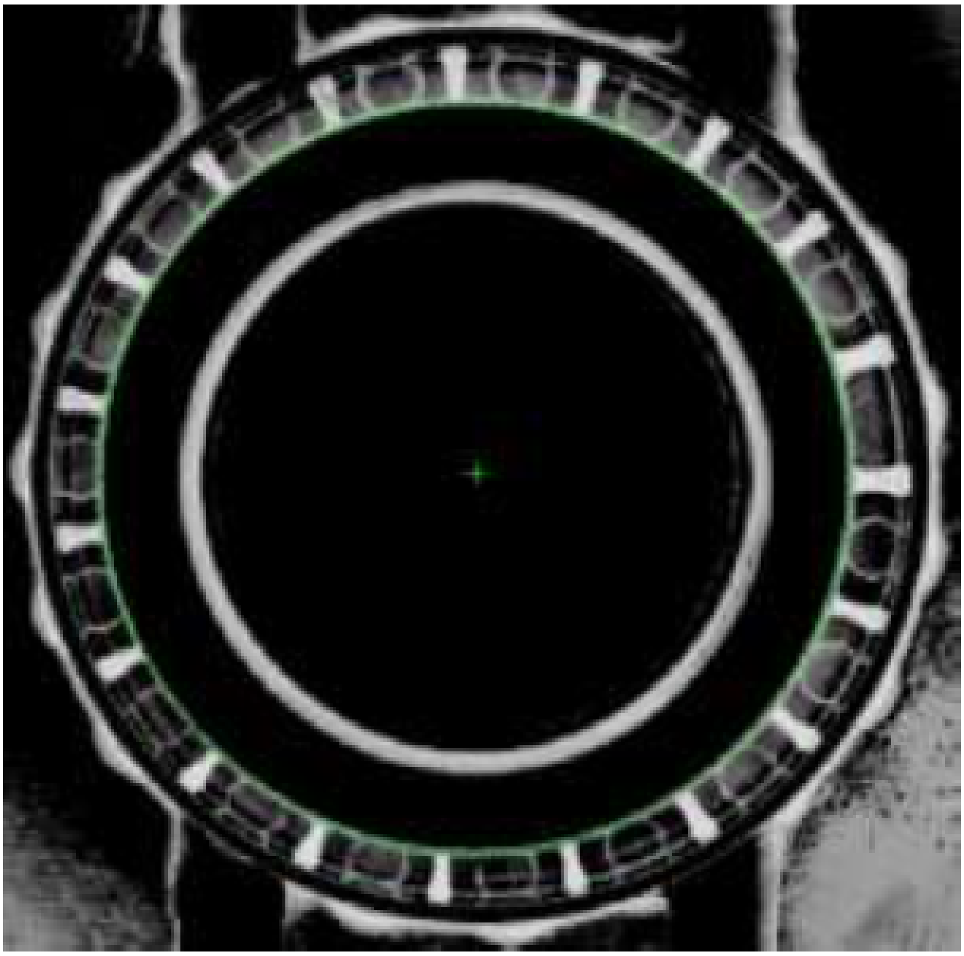


Figure 2.34: Binary image noise reduction



(a) Circular-Object

(b) Unwrapping

Figure 2.35: Circular Wrapping

2.4 Feature extraction

Feature analysis and pattern classification are often the final steps in the image analysis process. Feature analysis involves examining the features extracted from the images and determining if and how they can be used to solve the imaging problem under consideration. In some cases the extracted features may not solve the problem and the information gained by analyzing the features can be used to determine further analysis methods that may prove helpful, including additional features that may be needed. Pattern classification, often called pattern recognition, involves the classification of objects into categories. For many imaging applications this classification needs to be done automatically, via computer analysis. The patterns to be classified consist of the extracted feature information, which are associated with image objects and the classes or categories will be application dependent. We consider extraction of features that can be useful for solving computer imaging problems. Image segmentation allows us to look at object features, and the image transforms provide us with features based on spatial frequency information spectral features. The object features of interest include the geometric properties of binary objects, histogram features, spectral features, texture features, and color features. Once we have extracted the features of interest, we can analyze the image.

Exactly what we do with the features will be application-dependent. If we are working on a computer vision problem, the end goal may be the generation of a classification rule in order to identify objects. If we are working to develop a new image compression algorithm, we may want to determine what image data is important where the insignificant information can be compressed or eliminated completely. For image restoration we may want to determine the type of noise that exists in the image, or how the image has been degraded. Image analysis may help us to solve an image enhancement problem by allowing us to determine exactly what it is that makes images visually pleasing. As was shown in Figure 2.36, feature extraction is part of the data reduction process and is followed by feature analysis. One of the important aspects of feature analysis is to determine exactly what features are important, so the analysis is not complete until we incorporate application feedback into the system —see Figure 2.37. (Umbaugh, 2010).

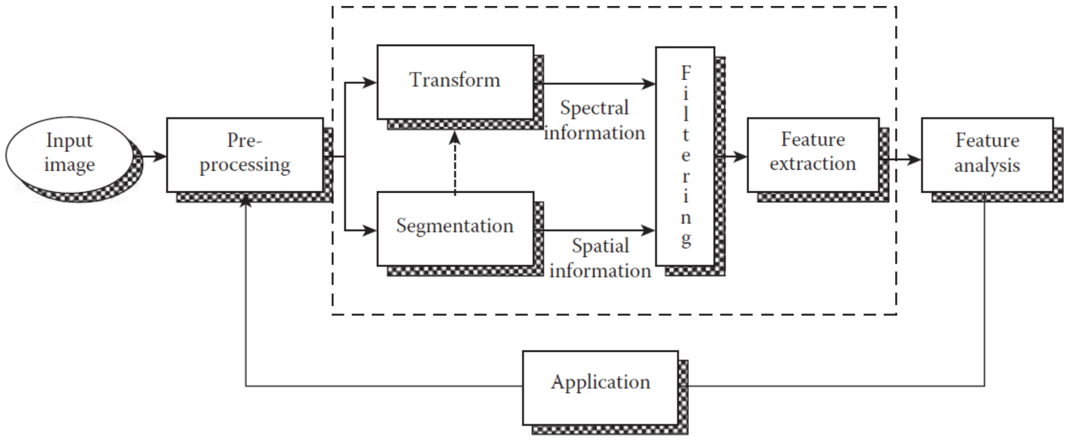


Figure 2.36: Image complete analysis process.

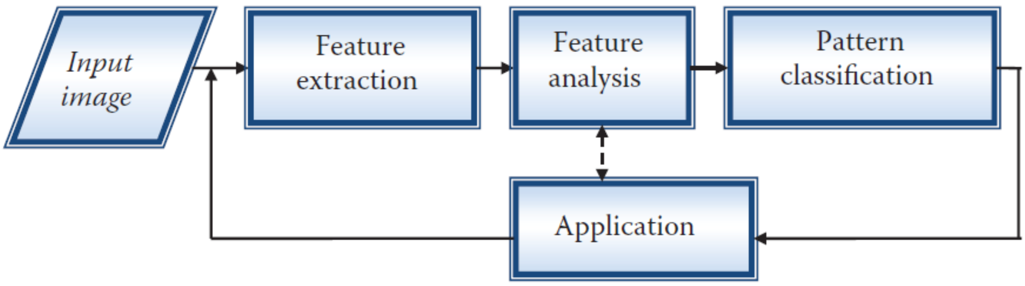


Figure 2.37: Feature extraction.

Feature extraction is a process that begins with feature selection. The selected features will be the major factor that determines the complexity and success of the analysis and pattern classification process. Initially, the features are selected based on the application requirements and the developer experience. After the features have been analyzed, with attention to the application, the developer may gain insight into the applications needs that will lead to another iteration of feature selection extraction and analysis. The overall process shown in Figure 2.37 will continue until an acceptable success rate is achieved for the application. When selecting features for use in a computer imaging application, we want to consider the following desirable attributes.

- Robust. It will have similar results under various conditions, such as lighting, cameras, lenses, and so on.
- Discriminating. It is useful for differentiation of classes (object types) of interest.
- Reliable. It provides consistent measurements for similar classes (objects).
- Independent. It is not correlated to other features.

We show the most important feature extractions in the following list.

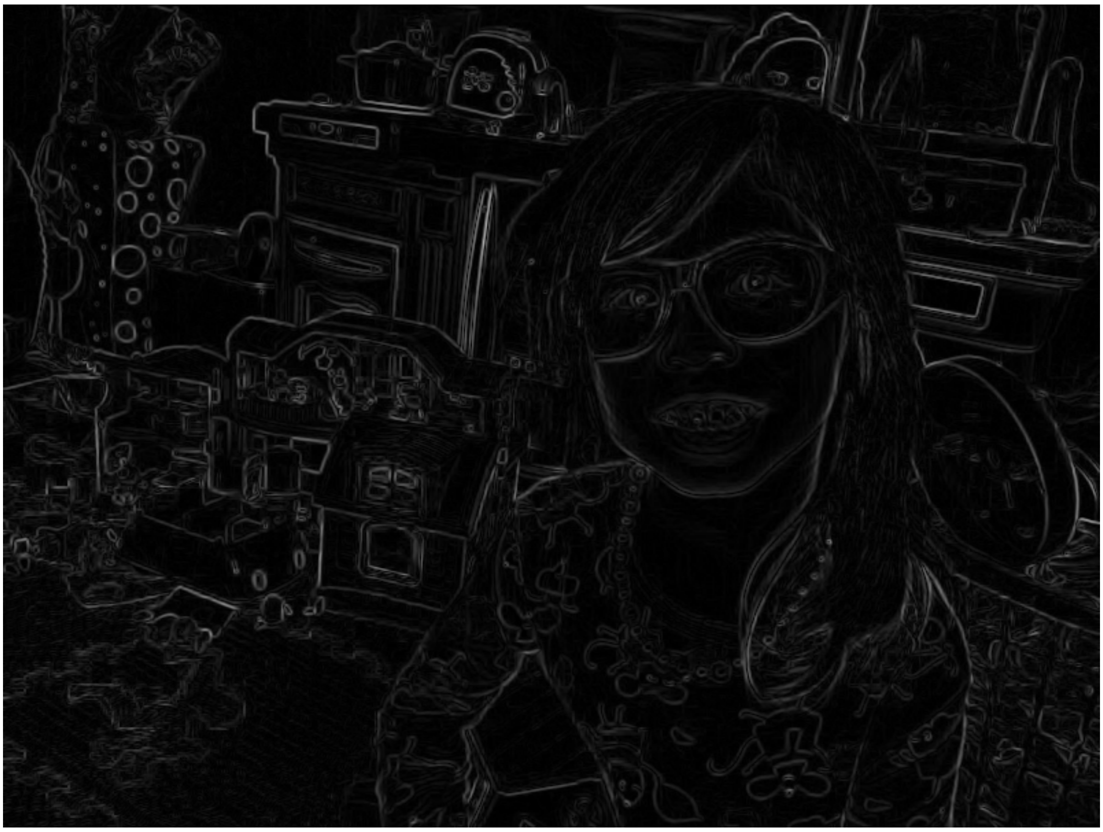
- Shape features. This depend on a silhouette of the image object under consideration, so all that is needed is a binary image. We can think of this binary image as a mask of the image object, as shown in Figure 2.38. The basic binary object features are area, center of area, axis of least second moment, projections, Euler number, perimeter, thinness ratio, irregularity, aspect ratio and moments.
- Histogram features. It is a histogram of an image is a plot of the gray level values versus the number of pixels at that value. The shape of the histogram provides us with information about the nature of the image, or subimage if we are considering an object within the image. A very narrow histogram implies a low contrast image, a histogram skewed toward the high end implies a bright image, and a histogram with two major peaks, called bimodal, implies an object that is in contrast with the background. The histogram features that we use are statistical based features, where the histogram is used as a model of the probability distribution of the gray levels. These statistical features provide us with information about the characteristics of the gray level distribution for the image or subimage —see Figure 2.39.
- Color features. Color is useful in many applications. Typical color images consisting of three color planes, red, green, and blue, can be treated as three separate gray scale images when you remove one of those planes. This approach allows us to use any of the gray level features, but with three times

as many, one for each of the three color bands. By using this approach we may be able to determine that information useful for the application is contained in one, two, or all three of the color bands. Often, when interested in color features, we want to incorporate information into the feature vector related to the relationship between the color bands. These relationships are found by considering normalized color, or color differences. The color features chosen will be primarily application specific. Some form of relative color is best, because most absolute color measures are not very robust. In many applications the environment is not carefully controlled, so a system developed under specific color conditions using absolute color may not function properly in a different environment. We need to remember all the factors that contribute to the color, like the lighting, the sensors, any optical filtering, and any print or photographic process in the system model. If any of these factors change then any absolute color measures, such as red, green, or blue, will change. An application specific relative color measure can be defined, or a known color standard can be used for comparison. When using a known color standard, the system can be calibrated if the conditions change —see Figure 2.40.

- Texture features. Texture is related to properties such as smoothness, coarseness, roughness, and regular patterns. Texture is a very useful characterization for a wide range of image. It is generally believed that human visual systems use texture for recognition and interpretation. In general, color is usually a pixel property while texture can only be measured from a group of pixels. A large number of techniques have been proposed to extract texture features. Based on the domain from which the texture feature is extracted, they can be broadly classified into spatial texture feature extraction methods and spectral texture feature extraction methods. Texture features are extracted by computing the pixel statistics or finding the local pixel structures in original image domain, whereas the latter transforms an image into frequency domain and then calculates feature from the transformed image, (Ping, 2013).

(a) Original color child room.

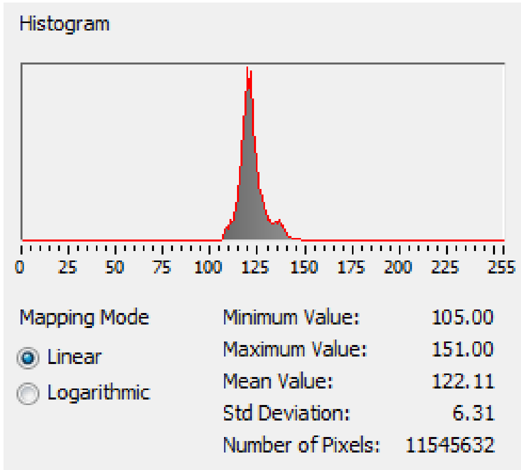
(b) Gray color child room



(c) Child Shape Perimeter Objects

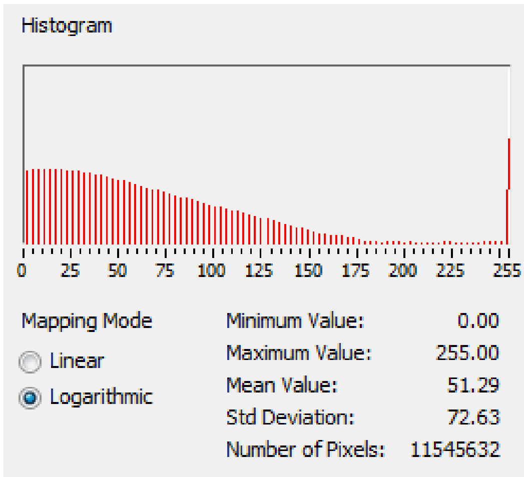
Figure 2.38: Shape Features Child Room

(a) Low contrast



(b) Low contrast histogram

(c) High contrast



(d) High contrast histogram

Figure 2.39: Histogram Features Child

(a) Child Full Color

(b) Child red color remove

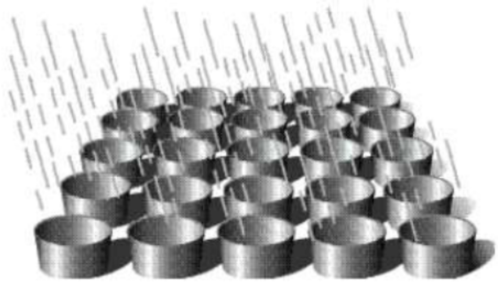
(c) Child green color remove

(d) Child blue color remove

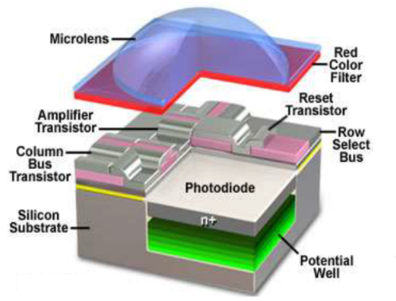
Figure 2.40: Color Features

2.5 Image acquisition devices

As we mentioned before, all machine vision cameras create an image by exposing arrays of photosensitive material to light energy, where we can call them as photon buckets. Exposure duration is time limited and typically adjustable. The energy in a bucket captured during an exposure period becomes a micro-voltage for that bucket —see Figure 2.41.



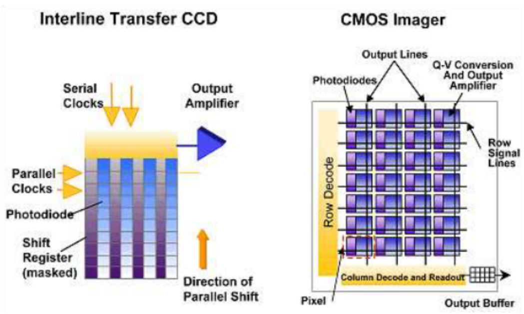
(a) Bucket Sensor Example



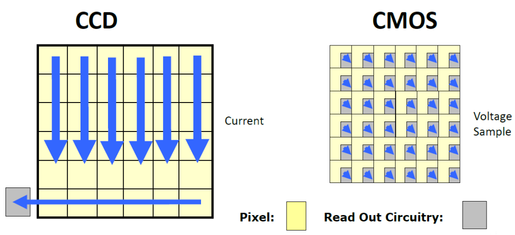
(b) Sensor microlens

Figure 2.41: Photodiodes

Each element in a camera sensor array is called a pixel (picture element). The energy value for each individual pixel is output as a micro-voltage upon acquisition of each image where the voltage determines the intensity level for that pixel. Pixel data transfer architecture varies by sensor type. Most widely used are CCD and CMOS —see Figure 2.42.



(a) CCD CMOS Arquitectures Photodiodes



(b) CCD CMOS Arquitectures Output

Figure 2.42: CCD CMOS Arquitectures. Main difference is how they transfer the charge out the pixel into the camera electronics (Read out)

Now we present the latest camera interface standards.

2.5.1 GigE Vision (Gigabit Ethernet - GigE)

- Created to mold Gigabit Ethernet to the needs of Machine Vision.
- Designed to increase stability and determinism, while reducing CPU load.
- Built in error checking and packet resend features.
- Contains two protocols.

  GVCP (GigE Vision Control Protocol) for establishing a constant link to camera for settings, configuration, etc.

  GVSP (GigE Vision Streaming Protocol) for streaming images.
- The Main Advantages of GigE Vision. Cable Length and Cost Effective Components —see Figure 2.43.
- Open framework for transferring imaging data and control signals between cameras and PCs over standard Network connections like GigE, 10 GigE, WiFi —see Figure 2.44.

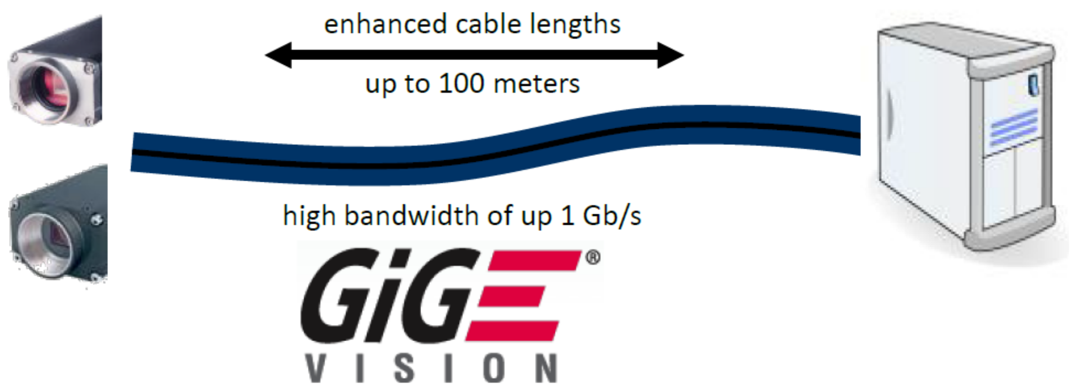
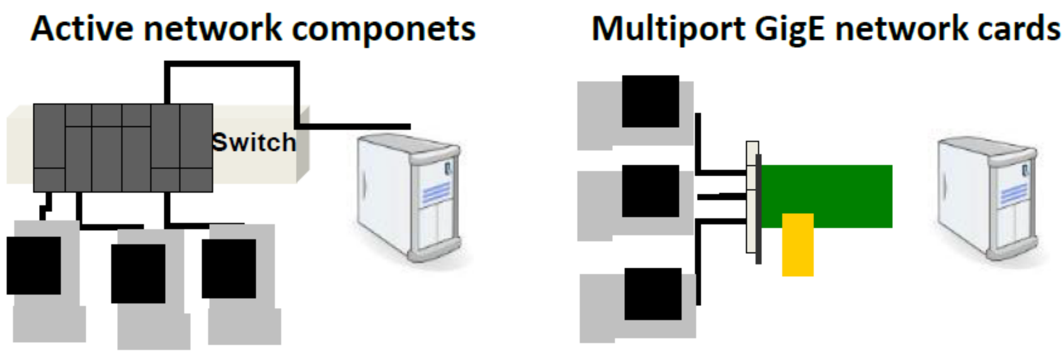


Figure 2.43: GigE Interface.



(a) Active Network GigE

(b) Multiport Network GigE

Figure 2.44: Multicameras Applications Design

2.5.2 DCAM (FireWire - IEEE1394)

- Industry standard for 1394a/b digital cameras created in 1996 by Apple (and TI) in the late 1980s.
- Specifies video formats, registers, features.
- Older interface, gradually dying out in the consumer market.
- There are 2 versions, 1394a and 1394b. The only difference is the bandwidth and connector type.
- Some of our research use cameras with that interface —see Figure 2.45.

(a) Firewire Logo

(b) Multiport card with 1394a and 1394b

Figure 2.45: Multicameras Applications Design

2.5.3 Camera Link

- Camera Link was launched in 1999 by key players in vision and formally adopted in 2000 by the AIA. Camera Link establishes a point to point dedicated link between camera and frame grabber for high speed and easy cable concept.
- Ground up designed for connection of cameras + frame grabbers —see Figure 2.46
- Standardized for 8 / 10 / 12 bit, single, multi-tap, RGB.
- The serial communication is routed through a separate channel than image capture. The frame grabber provides a special DLL file for software to access through.
- Frame grabbers are usually supply with processing software. They can be PCI, PCIe, PCIx. Some of them can have additional IO. They usually require a camera file which tells the grabber what camera is attached and how to handle it.
- Cables are very robust and designed for industry. They Can be found in various lengths up to 10m. Right angle and high flex available. MiniCLreduces connector size. PoCLPower over Camera Link offersone cable solution.
- Camera Link configurations: 2.0 Gbit/s, 4.1 Gbit/s and 6.8 Gbit/sec.

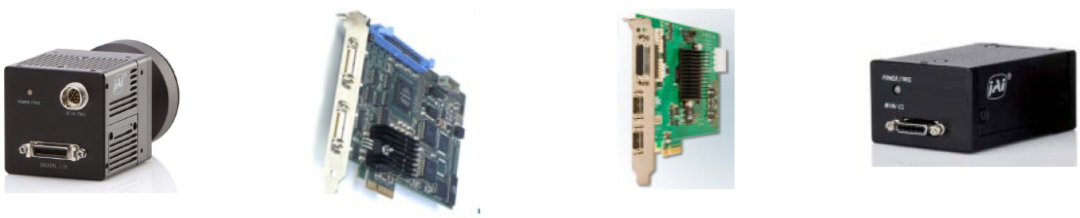
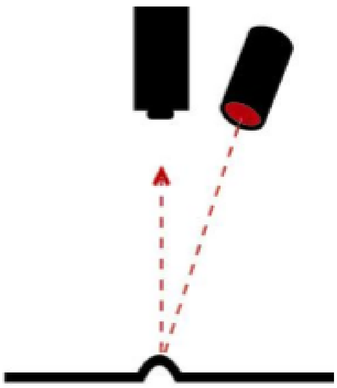


Figure 2.46: Camera Link Interface.

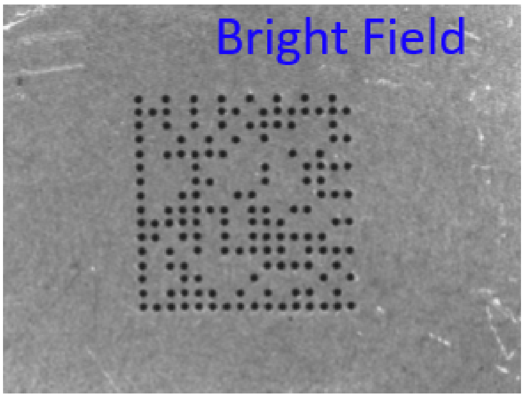
2.6 Illumination techniques

In this section we present the most common illumination techniques that are used.

- Partial Bright Field. Lighting type can be ring, spot or bar. They can be used over non specular materials, area and dark field light. If light is high angle, specular surfaces might reflect glare. If we have diffuse, flat or smooth surfaces, reflection will be even distributed. No working distance (WD) limit. This is only limited by device light intensity. We can see at the following Figure 2.47 this technique.



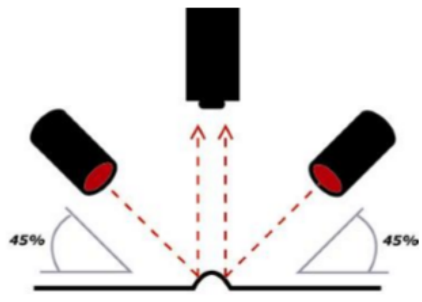
(a) Partial Bright Field.



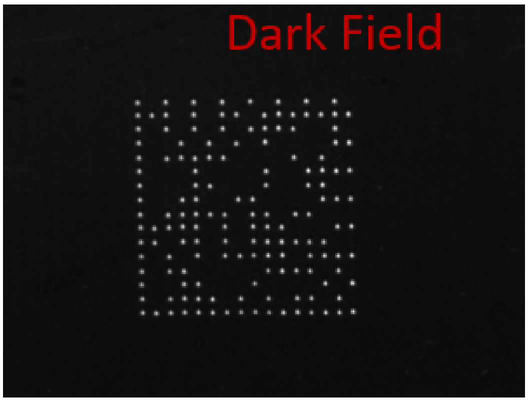
(b) Bright Field over surface

Figure 2.47: Partial Bright Field Lightning Technique

- Dark Field. Lighting type can be angled ring, spot or bar. They can be used over non specular materials, surface, edges and look through transparent parts. Shape and Contour Enhanced, Flat Polished Surfaces Dark, Diffuse Surfaces Bright and check edges. Light must be very close to part, Large footprint, Limited spot size, Ambient light may interference. We can see at the following Figure 2.48 this technique.



(a) Dark filed.



(b) Dark Field Example

Figure 2.48: Dark Field Lighting Technique

- Back Lighting. Used for Edge or whole detection. Useful on translucent materials, Liquid fill levels, Glass or plastic cracks, Vision Guided, Robotics Pick and Place, Gauging (Including high accuracy measurements). Multiple angle light from back light diffuser and Bending around objects. Technique shown at following Figure 2.49.

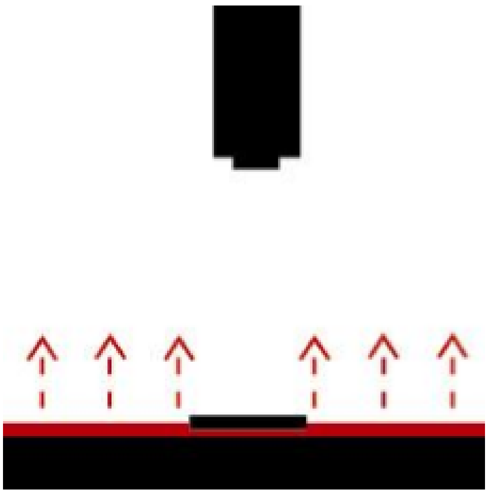


Figure 2.49: Back Lightning Techniques.

Chapter 3

Materials and methods

3.1 Vision Setup

We built a generic optical hardware setup as shown in the following Figure 3.1.

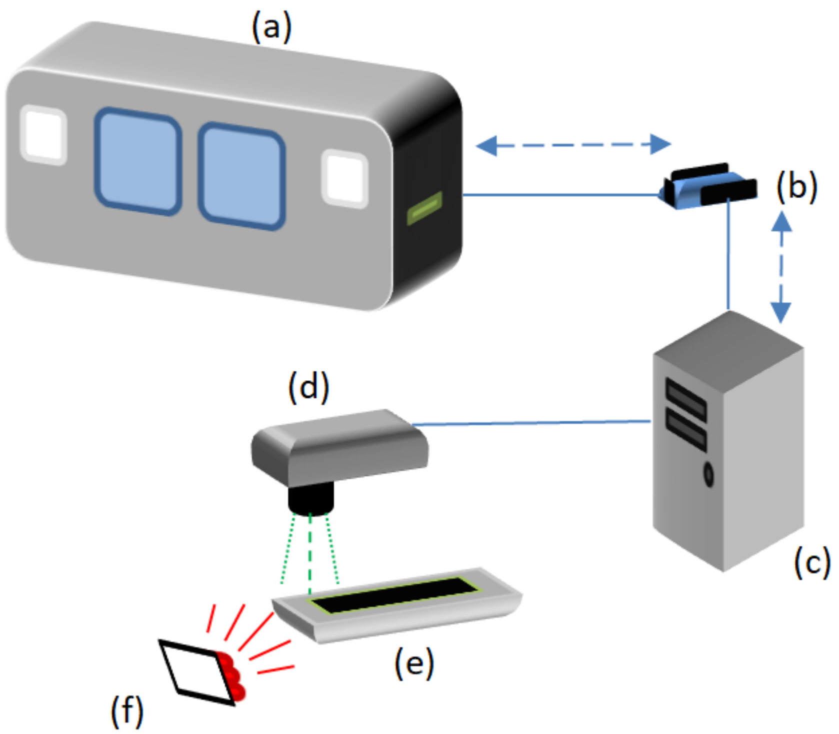


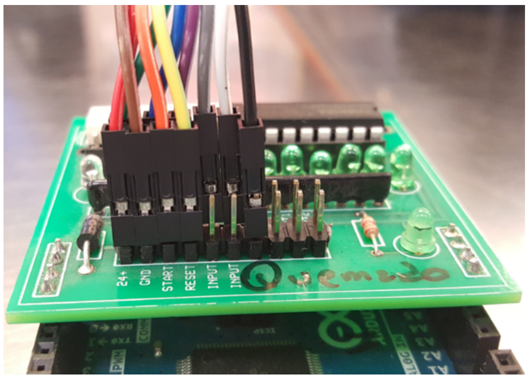
Figure 3.1: Generic Optical Setup. (a) Any process machine, (b) Communication Control Board, (c) Image Processing Computer, (d) Camera-Lens, (e) PCB and (f) Lighting.

Components of our vision setup:

- Process Machine. In any semiconductors factory there are different processes in production line. These processes can be manual, semiautomatic or full automatic. For those that are not full automatic, do not have any vision system or if they do have, maybe they do not meet the goals for vision requirements our solution may be adaptable as a full automatic vision system for reading OCR characters and complete the requirements for that process.

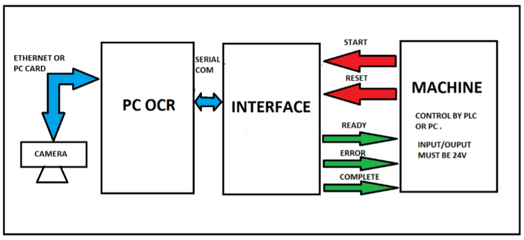
- Communication Control Board. We developed a communication control board based on Arduino Mega 2560 Rev3 and a shield for signals management for handshake between our vision system and process machine —see Figure 3.2. Our system is capable of connection to any kind of machine process if those systems have any available signals for communications (Handshake), Figure 3.3. We show the timing chart for communications between systems. Every signal will follow the following chart —see Figure 3.4.

(a) Arduino Mega 2560 Rev3

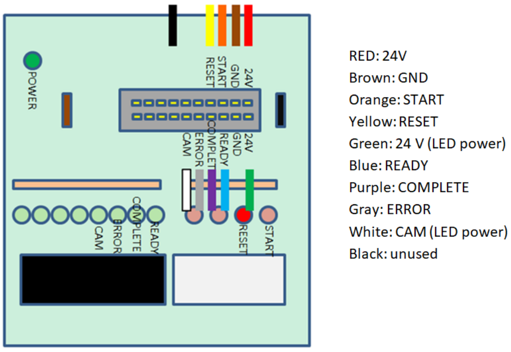


(b) Shield sideview

Figure 3.2: Arduino and Shield Design



(a) Signals Block Diagram OCR System



(b) Shield

Figure 3.3: Shield Design

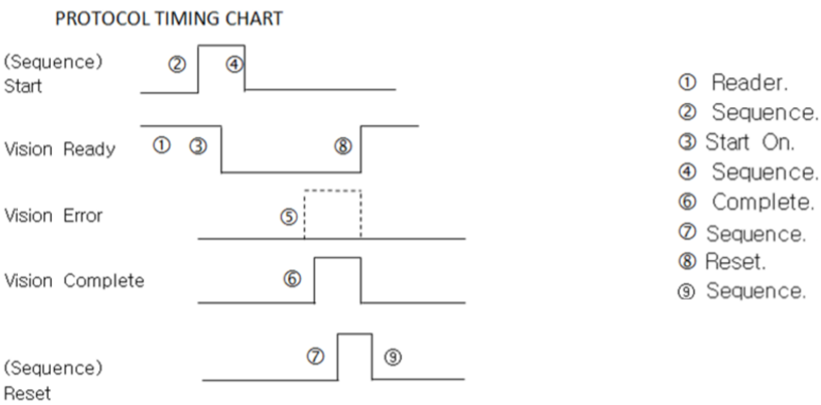
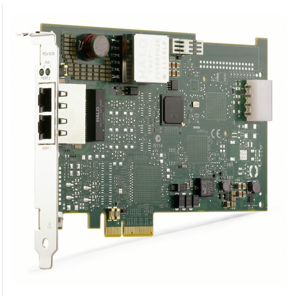


Figure 3.4: Protocol Timing Chart

- Image Processing Computer. This computer will have all the libraries, software developed for image processing, frame grabbers devices and connectivity toolkits to the camera and servers. Depending of type of camera interface, we use different frame grabbers from National Instruments like NI PCIe−8236 GigE vision frame grabber with Power over Ethernet (PoE) and NI PCIe−8255 vision frame grabber for Firewire 1394 —see Figure 3.5.



(a) PCIe−8236 GigE Frame Grabber

(b) PCIe−8255R Firewire 1394 Frame Grabber

Figure 3.5: Frame Grabbers

- Camera−Lens. We developed our vision systems using different technologies. Those technologies were chosen due available working area basically.

  - NI−1752 Smart camera. The most important feature about this system is that it does not need a frame grabber to operate. It can be plug directly to factory network. In our vision software we can use network connectivity toolkits to connect directly and take images for processing. This is very helpful when image processing computer does not have available PCIe slots for GigE frame grabbers —see Figure 3.6.

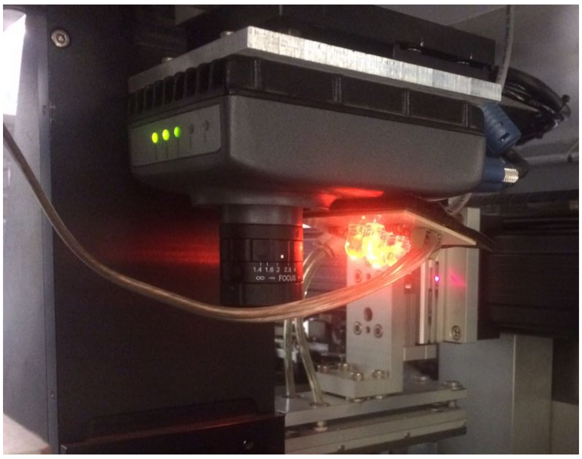


Figure 3.6: NI−1752 SmartCamera System

– AVT Mako GigE and Stingray Firewire cameras. These cameras do need frame grabbers for connecting to image processing computer. They connect directly to the computer. Image acquisition is done with frame grabber toolkits. Software is full developed in LabVIEW for any type of camera technology, Smart camera, GigE or Firewire 1394 —see Figure 3.7.

Figure 3.7: AVT Cameras

- Lens were approximately calculated with the following equation and Figure 3.8. In our case, mechanical spaces are critical because we are setting up a vision system that was not part of the initial design of the process machine. Working distance and horizontal field of view are the most critical variables to have the correct final image. We are using fixed local length lenses for focal lengths from 4.5 mm, 25 mm and 35 mm. They typically have a minimum working distance with no maximum, Figure

$$\text{Focal Length} = \frac{\text{Pixel Pitch} \times \text{Active Pixels} \times \text{Working Distance}}{\text{Field of View}} \tag{3.1}$$

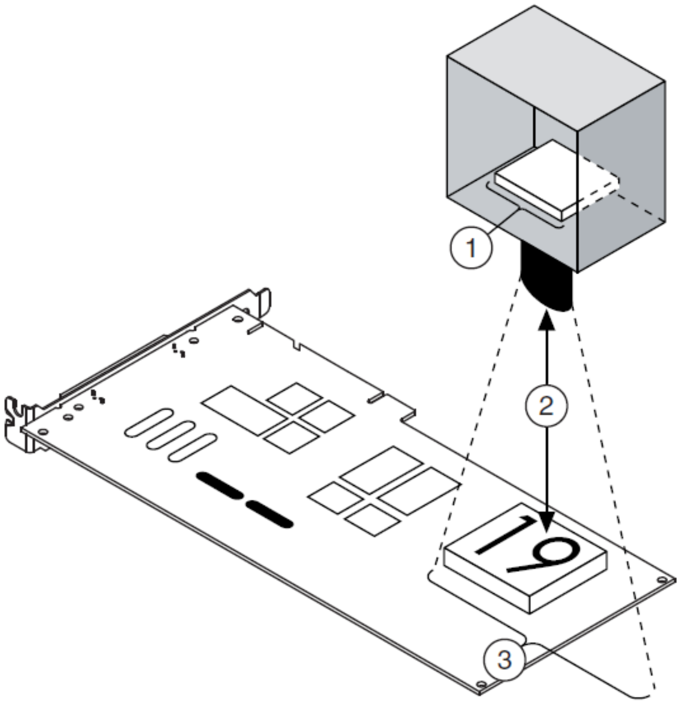
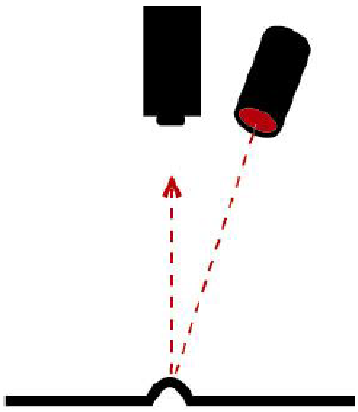


Figure 3.8: Vision Fundamental Optics. (1) Horizontal Image Width, (2) Working Distance, (3) Horizontal Field of View

Figure 3.9: Fixed local length lenses from 4.5mm, 25 mm and 35 mm in our vision systems

- Lighting. Our vision systems is designed for any machine process. We explain that due a mechanical issues we can not have enough free space to install the best lighting devices or any other camera that requires more space. For this, the best solution due limited space is to have a partial bright field or also called directional lighting because is designed for only one lightning device. This is the most common vision lighting technique for generating contrast and enhancing topographic detail —see Figure 3.10. But definitely this can generate a problem if we have specular surfaces, that will generate not uniform light distribution over PCB surface and bright hotspot reflections —see Figure 3.11.

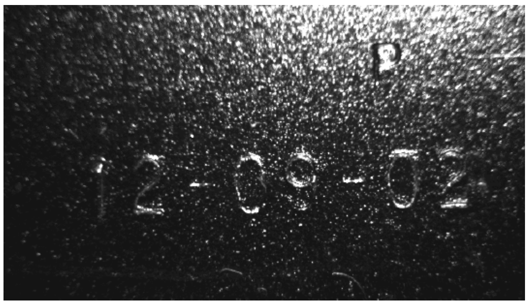


(a) Partial Bright Field Lighting

(b) Bright Field Light Device

Figure 3.10: Bright Field Lighting Technique

(a) Partial Bright Field Lighting closeup. (a) Light, (b) Camera, (c) PCB



(b) Bright Hospot

Figure 3.11: Bright Hotspots

3.2 Image Databases

We have developed and installed more than one hundred vision systems for different process machines. Each of them save data directly to a FTP server all the images that for some reason could not be read one or many of their characters. There are thousands of theses images like shown in the following Figure 3.12.

Figure 3.12: OCR Saved Images for future re-inspection.

We developed two software for extracting digits in automatic form:

- Auto-extracting Digits Software. Developed in LabVIEW. It detects in automatic mode possible blobs that can be digits —see Figure 3.13.

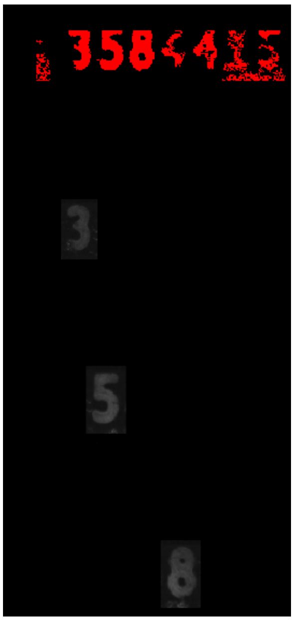


Figure 3.13: Software for Extracting Digits from PCB cuts.

- Software for manual sorting of digits. We developed a software that moves unrecognized digits. This classify those digits in easy way and automatically move to its digit folder —see Figure 3.14.

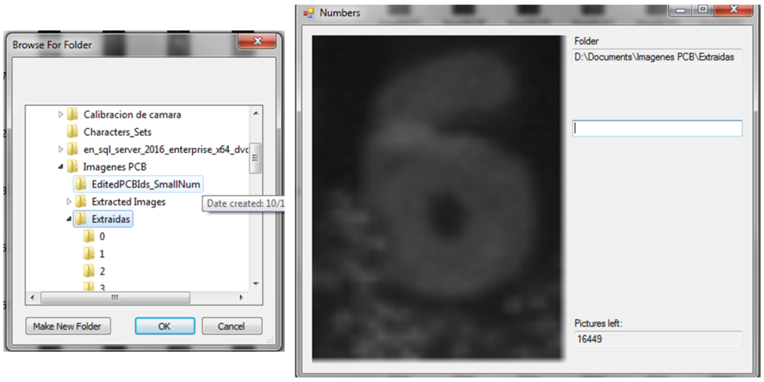


Figure 3.14: Software for manual Digits Classification and Sorting.

Finally, we could create an image digit database of 15,000 samples, distributed in 1,500 samples for each digit from 0 to 9.

3.3 Template Patten Matching

We have developed two types of template pattern matching for OCR.

3.3.1 Full Template Pattern Matching

1. Alphabet – Training Characters. We have created an alphabet with hundreds of complete images for each digit —see Figure 3.15.

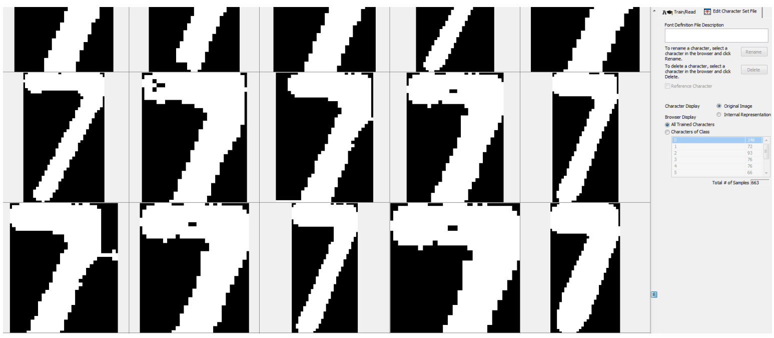
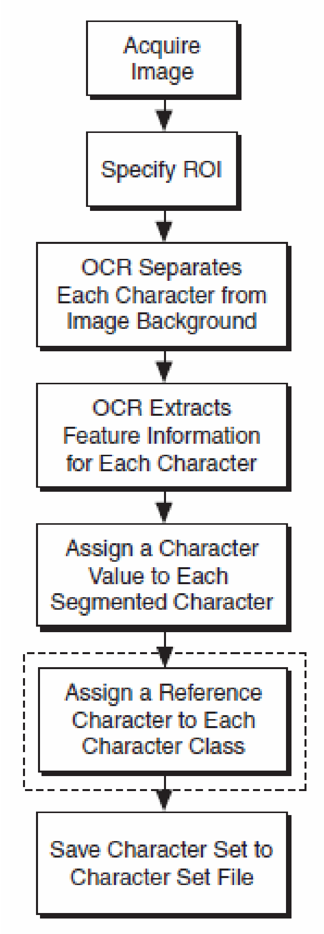


Figure 3.15: OCR Digits Alphabet.

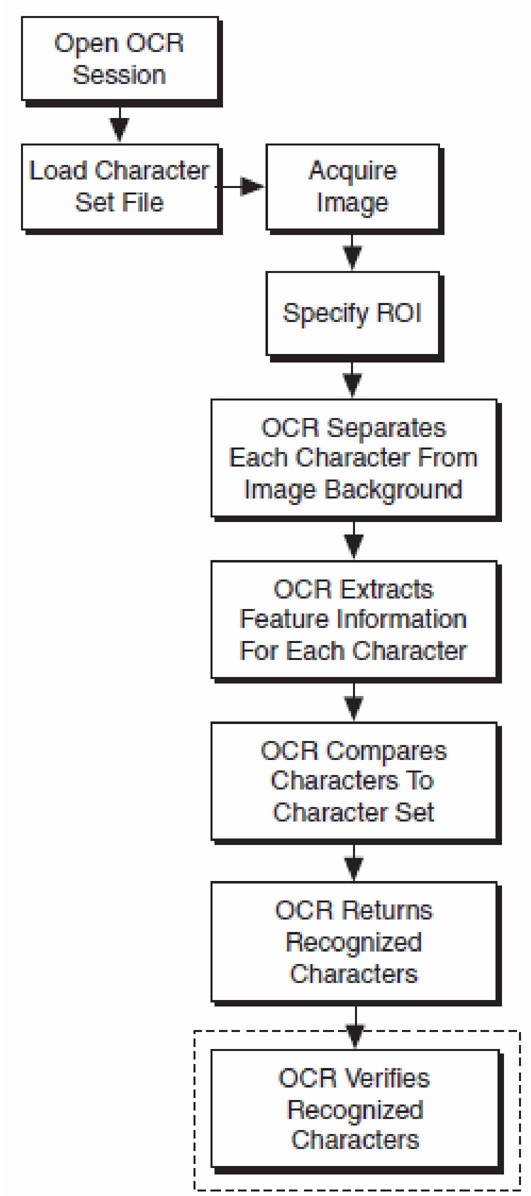
2. When a blob for possible character area is extracted, this is compare for each alphabet digit. Bigger the alphabet, more processing time will be required.

3. If those characters that will be read have some lighting variations, blur, noise, geometric transformations like shifting, rotation or scaling for those in the template, it will be difficult to match. It will take a long time for image processing with no chances of success. Some machines process damage identification character on PCB with dirt on it.

4. if those characters that will be read are clean, complete with no apparent superficial damaged, then they will be read very quickly using the alphabet —see Figure 3.16



(a) Training Characters Process



(b) OCR Compare

Figure 3.16: OCR Compare Process

3.3.2 Adaptive Block Search and Alphabet Extended

Some of the machine process damaged the PCB OCR section with superficial dirt over it —see Figure 3.17.

Figure 3.17: OCR Digits Damage with superficial dirt in lower locations.

- We designed a model that automatically create virtual adaptive blocks that change in the vertical direction its height.
- It start with an initial value for height. Then in a loop is decreasing a some constant factor.
- Then software looks for any possible pattern that could be a digit.
- If it finds a candidate, then it will compare it with the extended alphabet.
- Extended alphabet has many samples that will be exactly if some parts of the digits are missing.
- With this pattern matching is faster and the compare is close to full match —see Figure 3.18

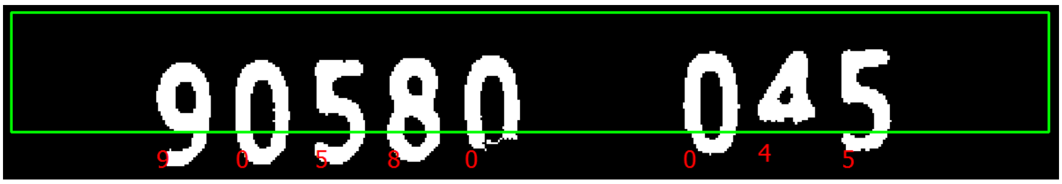


Figure 3.18: Adaptive OCR Damaged Read Block.

3.4 Image Processing Cleaning

- We apply many processing techniques for remove unwanted data just before OCR pattern matching.

- Particle filtering use removal of: Area, Center of Mass X and Y, Bounding Rectangle (Left, Top, Right, Bottom), Bounding Rectangular Width and Height, Horizontal and Vertical Segment Length. We also use erode and dilate for removing lines that are not part of the digits —see Figure 3.19.

Figure 3.19: Noisy Image Cleaning Processing.

- Also if we use the Adaptive Block Search, it helps to see only some sector and omit those that it is possible to have dirt over the lower part of the digits —see Figure 3.20

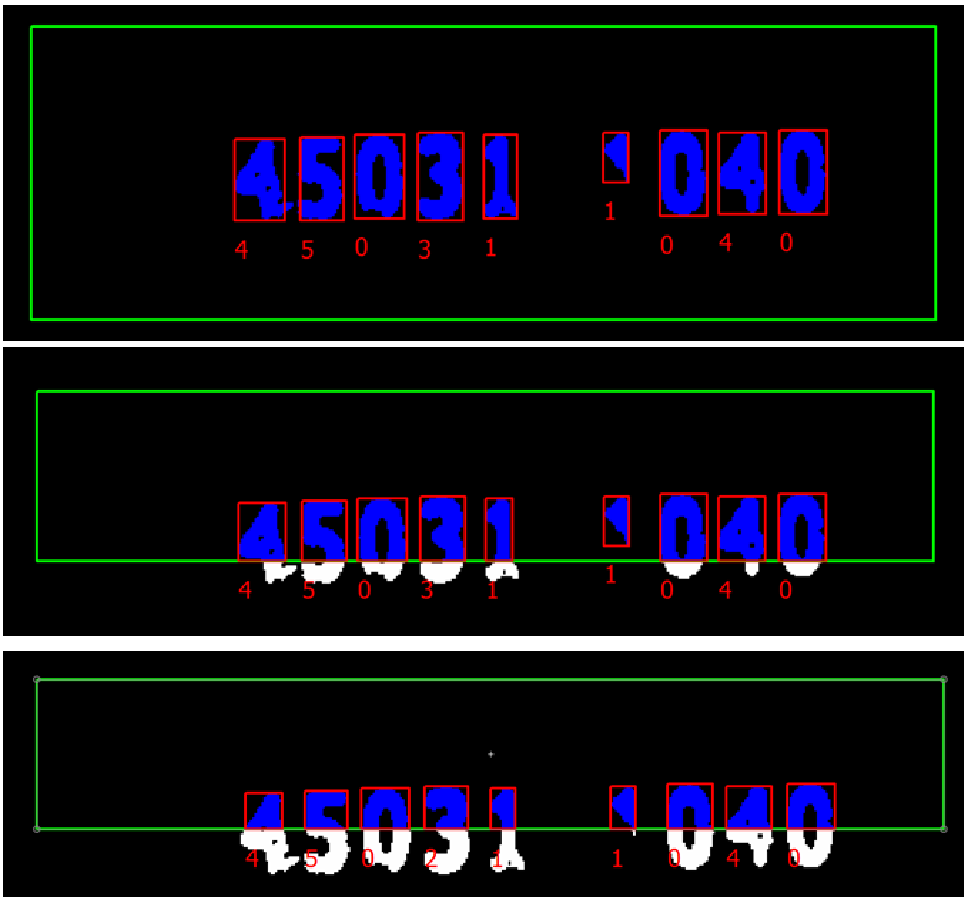


Figure 3.20: Image Cleaning Processing with Adaptive Block Search.

3.5 Thresholding Preprocessing

1. Our vision systems use a Global Gray scale Thresholding for the binarization of a gray image.

2. Our new model consist in a mix of Global Threshold with Local Threshold.

3. The first step is to use local threshold where we analyze each pixel based in the intensities statistics of the surround pixels. We use local thresholding to isolate objects of interest from the background in images that exhibit nonuniform lighting changes. Non uniform lighting changes, such as those resulting from a strong illumination gradient or shadows, often make global

thresholding ineffective. In the following Figures 3.21 and 3.22 we can see the differences.

Figure 3.21: Original PCB with superficial damage.

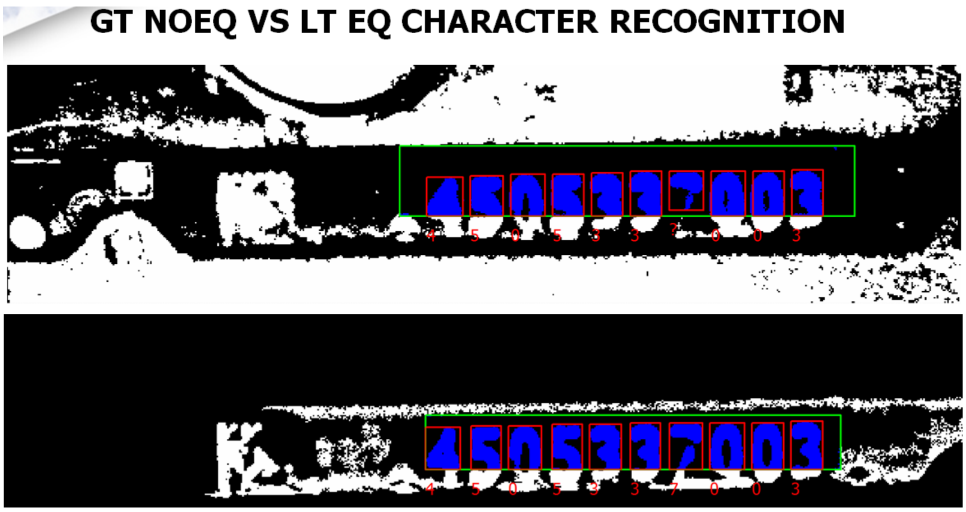


Figure 3.22: Global Threshold vs Local Threshold OCR.

4. Those difference came from the partial bright field lighting. This is because it creates a non uniform distribution of light over the surface and if the surface is specular then hotspot and not equal distribution will be over the PCB.

5. The local thresholding algorithm calculates local pixel intensity statistics like range, variance, surface fitting parameters, or their logical combinations for each pixel in an inspection image. The result of this calculation is the local threshold value for the pixel under consideration. The algorithm compares the original intensity value of the pixel under consideration to its

local threshold value and determines whether the pixel belongs to a particle or the background. A user defined window specifies which neighboring pixels are considered in the statistical calculation. The following Figure 3.23 shows a simplified local thresholding window.

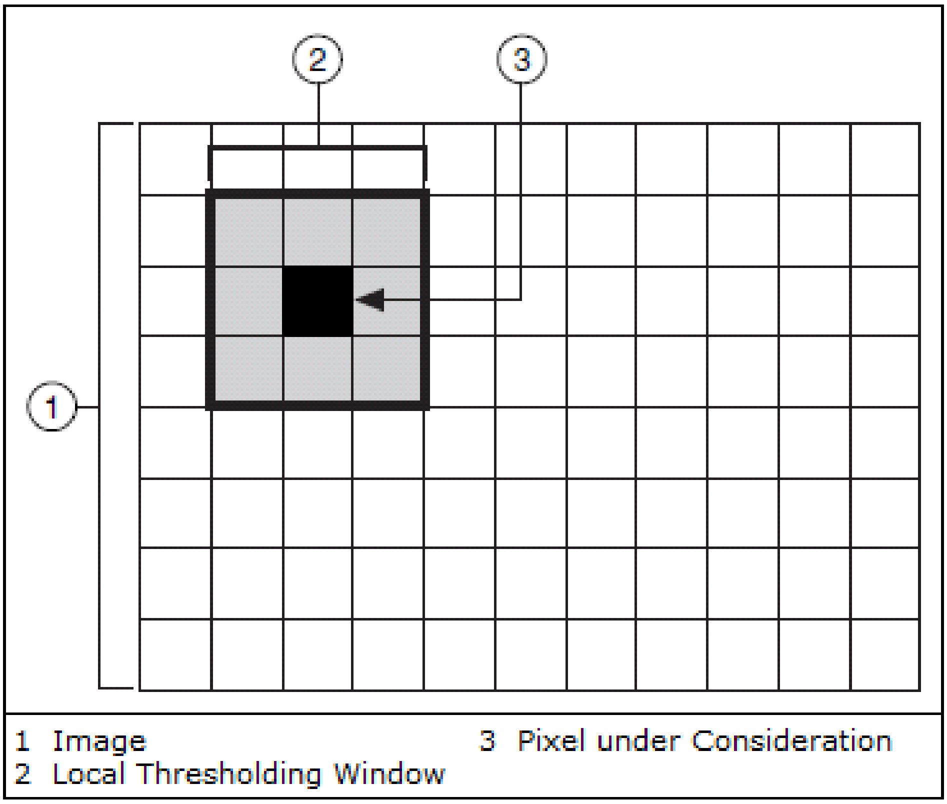


Figure 3.23: Local Threshold Window.

6. We compute the OCR pattern matching using the local threshold. If we can read the characters we stop the inspection process with a successful approach. If we can not read it, then we discard local threshold approach and we move to global threshold where we need to use a loop to take different values from 255 to 0 and make the pattern matching for OCR. This take longer computing time instead of just one value from local threshold.

7. Our prototypes reduce OCR time from 3 seconds to less than 100 ms using local threshold when is successful.

3.6 Principal Component Analysis (PCA) Dimensional Reduction

PCA is a standard technique for visualizing high dimensional data and for data preprocessing. PCA reduces the dimensionality (the number of variables) of a data set by maintaining as much variance as possible. Principal component analysis (PCA) rotates the original data space such that the axes of the new coordinate system point into the directions of highest variance of the data. The axes or new variables are called principal components (PCs) and are ordered by variance: The first component, PC 1, represents the direction of the highest variance of the data. The direction of the second component, PC 2, represents the highest of the remaining variance orthogonal to the first component and so on. This can be naturally extended to obtain the required number of components which together span a component space covering the desired amount of variance.

Since components describe specific directions in the data space, each component depends by certain amounts on each of the original variables: Each component is a linear combination of all original variables. Low variance can often be assumed to represent undesired background noise. The dimensionality of the data can therefore be reduced, without loss of relevant information, by extracting a lower dimensional component space covering the highest variance. Using a lower number of principal components instead of the high-dimensional original data is a common preprocessing step that often improves results of subsequent analyses such as classification, which this is what we are using in our research. For visualization, the first and second component can be plotted against each other to obtain a two dimensional representation of the data that captures most of the variance (assumed to be most of the relevant information), useful to analyze and interpret the structure of a data set.

For our research, we use our previous explained single image database —see Figure 3.24.

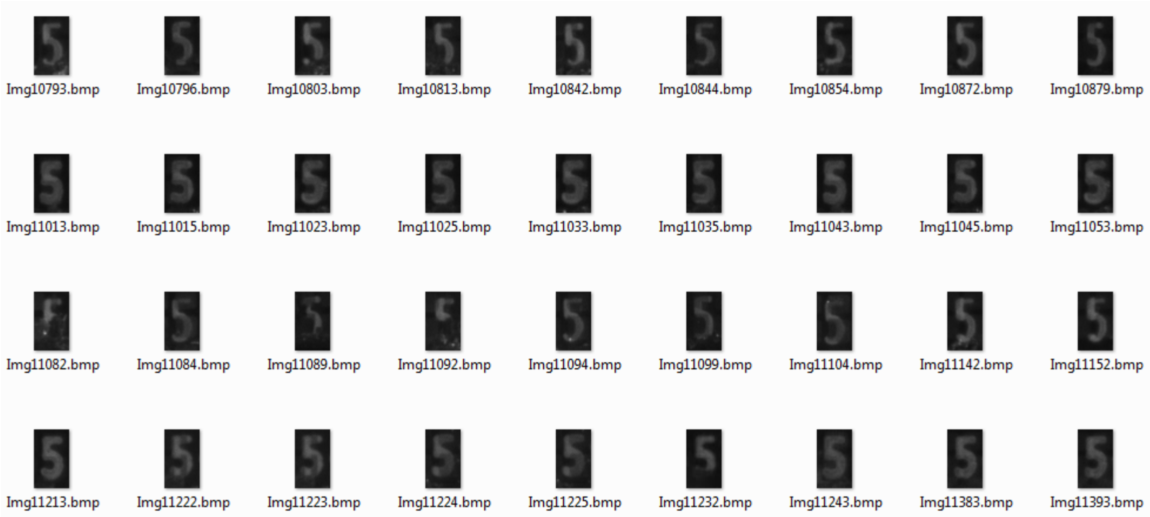


Figure 3.24: Single digits images.

We let I_i any (k, l) digit image —see Figure 3.25, from the original dataset. $\forall I_i$:

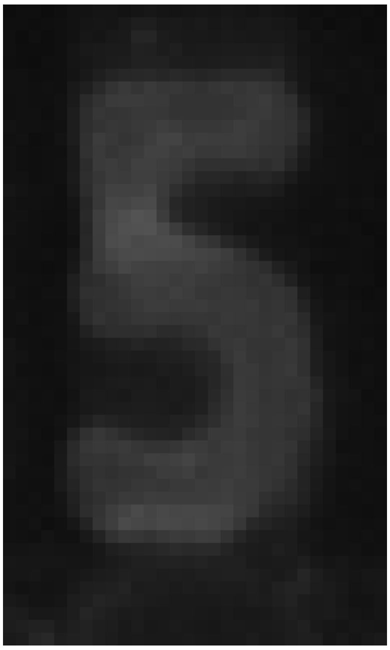


Figure 3.25: I_i example 5 digit image matrix with size $(k = 50, l = 30)$ pixels.

1. Convert I_i to gray-scale from previous RGB images.

2. Transform matrix I_i to a row-vector of size $(1,\ k \times l)$.

3. Create a matrix M of size $(n, k \times l)$ where n is the number of samples for each digit image, $M \leftarrow M \cup I_i$

4. Extract the mean for each column:

   $\Phi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} m_{ij}$

   $\hat{M} \leftarrow M_j - \Phi_j$

5. Compute the covariance matrix γ of $\hat{M}$

6. Compute eigenvectors and eigenvalues (PC, V) of γ

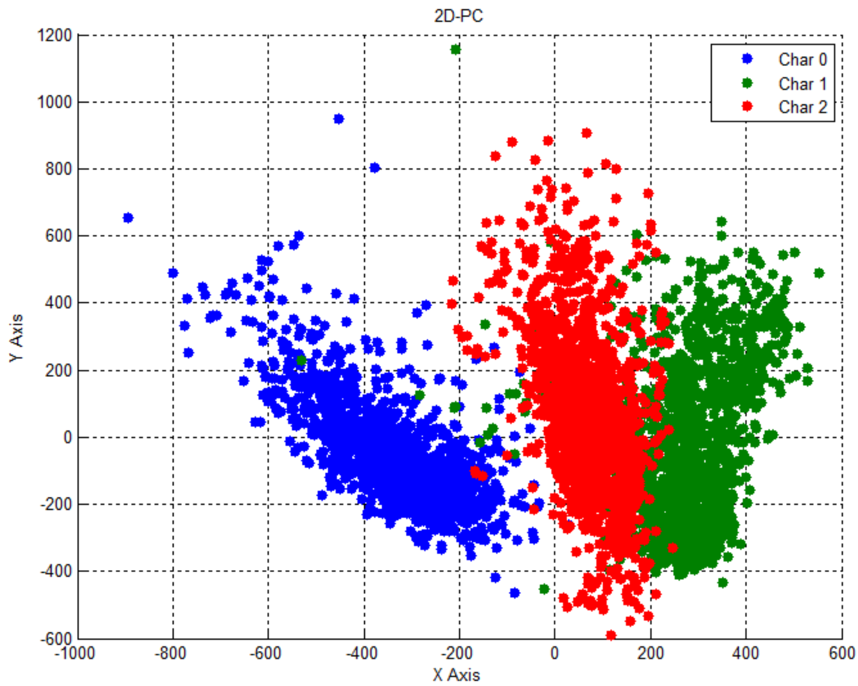
7. Sort matrix PC by columns in descend order ruled by vector V

8. Project $\hat{M}$ into the first P principal components:

$$X \leftarrow \hat{M} \times PC \tag{3.2}$$

9. A new digit dataset is now assembled, X

The principle component analysis (PCA) digit image test dataset must be processed by applying the first 5 steps, but using ϕ_j and $\hat{M}$ calculated from the training set. With this, we can apply a classification model to identify tests images dataset. At the following Figure 3.26 we show the principal 2 and 3 components of the PCA for only 3 classes $(0, 1, 3)$ of our dataset of $15,000$ images.



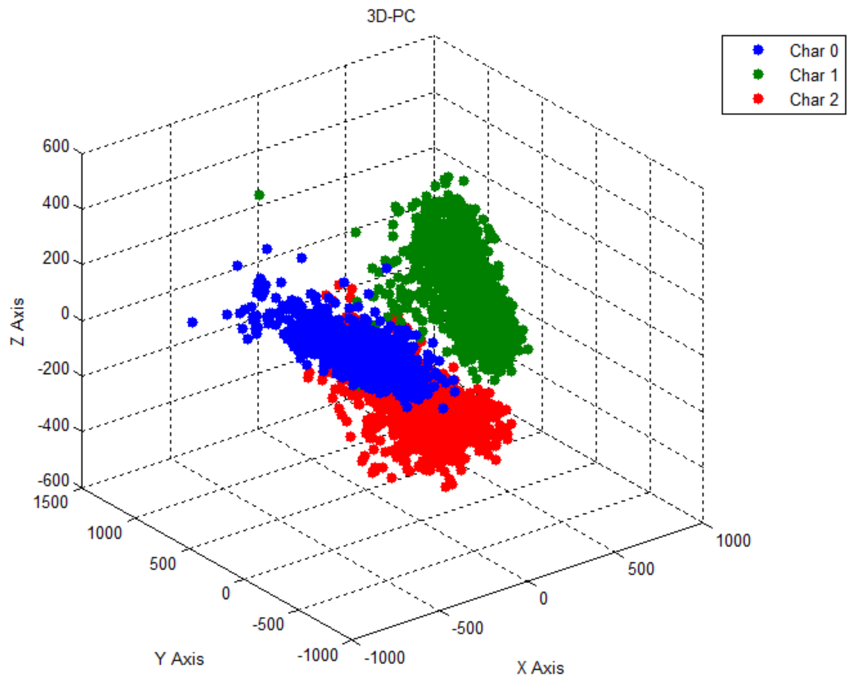


Figure 3.26: PCA First 2 and 3 Components for 3 classes 0, 1 and 2

3.7 Bayesian Discriminants

3.7.1 Linear Discriminant Analysis - LDC Classifier

A discriminant function is associated with a class C_i for $i = 1...l$. This function assign a numerical value to an instance for each class. A classifier based in discriminant functions assigns an object X to the class j if

$$d_j(X) > d_i(X) \forall i = 1...l; i \neq j \tag{3.3}$$

Following the classification rule

$$C_j \mid d_j(X) = max[d_i(X)] \forall i = 1...l \tag{3.4}$$

if we consider Bayesian theory where

$$P(C_i \mid X) = \frac{P(X \mid C_i) \bullet P(C_i)}{P(X)} \tag{3.5}$$

We have the probability discriminant function.

$$d_i(X) = ln[P(X \mid C_i)] + ln[P(C_i)] \tag{3.6}$$

Assuming Gaussian behavior we have the linear discriminant classifier:

$$d_i(X) = \ln P(C_i) + \mu_i \sum{}^{-1} x^T - \frac{1}{2} \mu_i \sum{}^{-1} \mu_i^T \tag{3.7}$$

3.7.2 Quadratic Discriminant Analysis - QDA Classifier

And we can show the quadratic discriminant classifier:

$$d_i(X) = \ln P(C_i) - \frac{1}{2} (\ln | \sum{}_i | + (X - \mu_i)^t \sum{}_i^{-1} (X - \mu_i)) \tag{3.8}$$

We can show an example of the hyperplane for linear and quadratic data distributions at the following Figure 3.27.

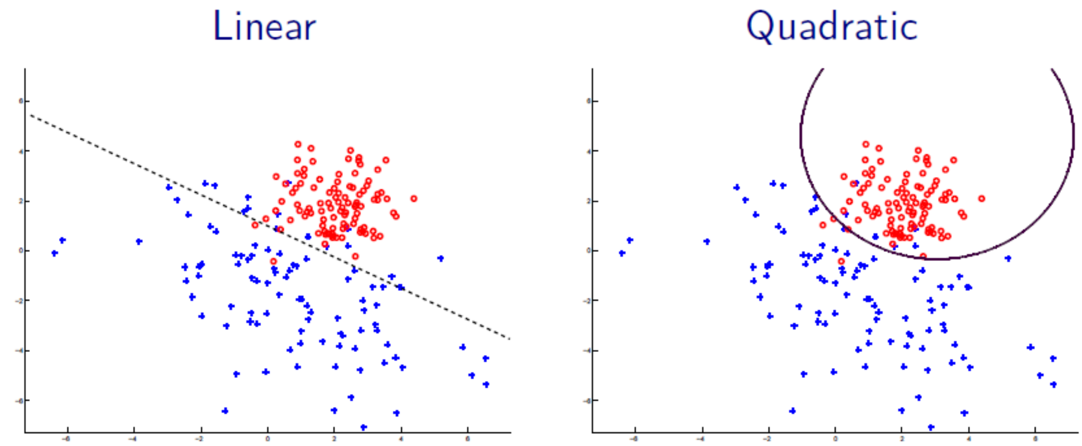


Figure 3.27: Linear and Quadratic Discriminants.

3.8 KNN Classifier

The K−Nearest Neighbor Algorithm for classification is the simplest of all machine learning algorithms. It is based on the principle that the samples that are similar, generally lies in close vicinity. K−Nearest Neighbor is instance based learning method. Instance based classifiers are also called lazy learners as they store all of the training samples and do not build a classifier until a new, unlabeled sample needs to be classified. Lazy learning algorithms require less computation time during the training phase than other learning algorithms (such as decision trees, neural networks and Bayes algorithm) but more computation time during the classification process. Nearest−neighbor classifiers are based on learning by resemblance, by comparing a given test sample with the available training samples which are similar to it. For a data sample X to be classified, its K−nearest neighbors are searched and then X is assigned to a class label to which majority of its neighbors belongs to it. The choice of k also affects the performance of k−nearest neighbor algorithm. If the value of k is too small, then K−NN classifier may be vulnerable to over−fitting because of noise present in the training dataset. On the other hand, if k is too large, the nearest−neighbor classifier may mis−classify the test sample because its list of nearest neighbors may contain some data points that are located far away from its neighborhood. K−NN fun-

damentally works on the belief that the data is connected in a feature space. For this, all the points are considered in order, to find out the distance among the data points. We can select of many distances types, but for our research we use the Euclidean distance. The distance selected is due to the data type of data classes used. In this a single value of K is given which is used to find the total number of nearest neighbors that determine the class label for unknown sample. If the value of K=1, then it is called as nearest neighbor classification.
The K-NN classifier works as follows.

1. Initialize value of K.
2. Calculate the Euclidean distance between input sample and training samples.

$$D_{Euclidian} = \sqrt{\sum_{i=1}^{l}(Test_i - Training_i)^2} \tag{3.9}$$

3. Sort the distances.
4. Take top K−nearest neighbors.
5. Apply simple majority.
6. Predict class label with more neighbors for input sample.

In the following example, we have three classes (X, Y, Z) where there is a sample P that is required to find its class label. If we chose K=5 and calculate the Euclidean distance for each sample pair, then is found that four nearest neighbor samples are falling in the class label X, while a single pair distance belongs to the class label Z. That means sample P is assigned to class X as it is the principal class for that sample —see Figure 3.28. In our research work, we are using K=3 for the 10 digits (0 to 9). (Cover, 1967; Jadhav, 2016).

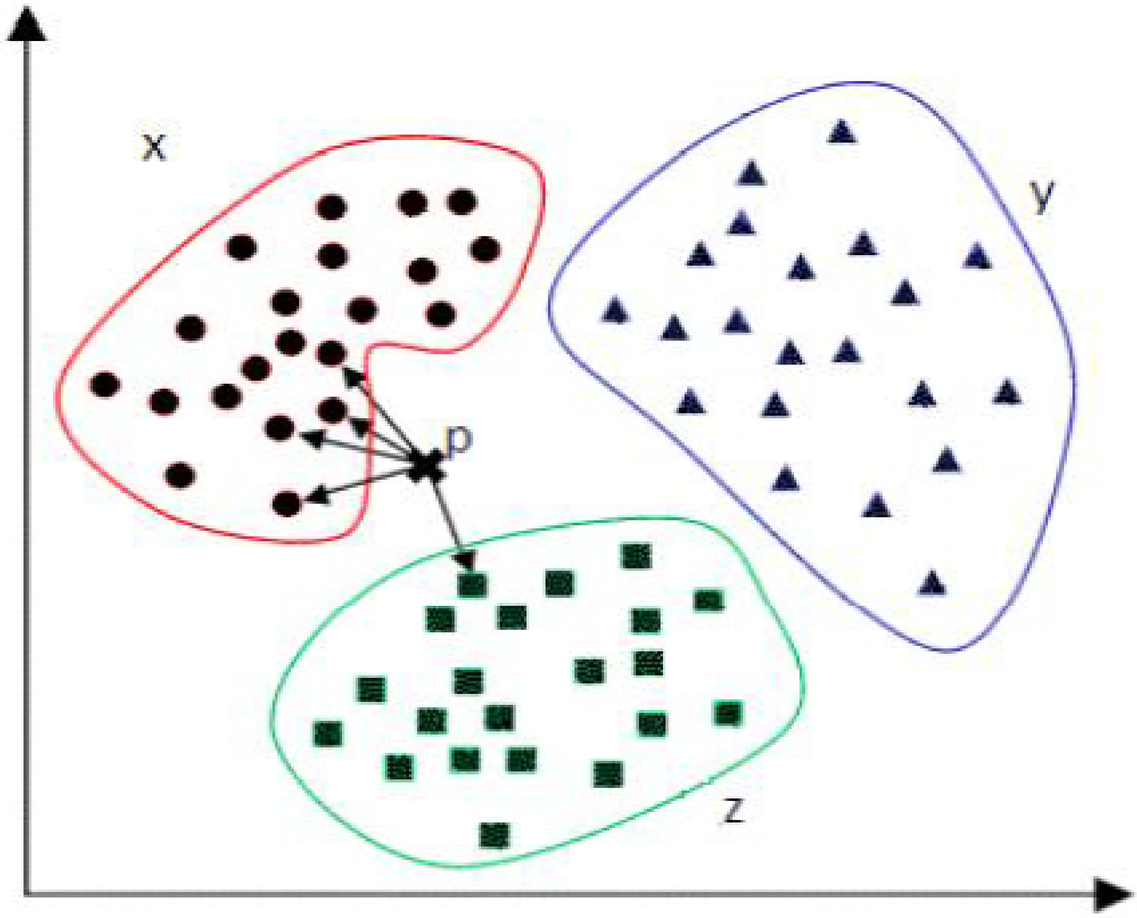


Figure 3.28: KNN example for 3 classes and a chosen K=5 for a sample P.

Some advantages to use KNN classifier.

- Easy to understand and implement.
- Training is very fast.
- It is robust to noisy training data.
- It performs well on applications in which a sample can have many class labels.

Some disadvantages to use KNN classifier.

- Lazy learners incur expensive computational costs when the number of potential neighbors which to compare a given unlabeled sample is large.
- It is sensitive to the local structure of the data.
- Memory limitation.
- As it is supervised lazy learner, it runs slowly.

3.9 Naive Bayes

Naive Bayes Classifier is a simple Statistical Bayesian Classifier. It is called Naive as it assumes that all variables contribute towards classification and are mutually correlated. This assumption is called class conditional independence. It is also called Simple Bayes or Independence Bayes.They can predict class membership probabilities, such as the probability that a given data item belongs to a particular class label. A Naive Bayes classifier considers that the presence (or absence) of a particular feature (attribute) of a class is unrelated to the presence or absence of any other feature when the class variable is given. The Naive Bayes Classifier technique is based on Bayesian Theorem and it is used when the dimensionality of the inputs is high. In our research is well implemented for inputs dimensionality of $1,500$ features and $15,000$ samples. (Jadhav, 2016). Bayesian classification is based on Bayes Theorem. This state that: Let X is a data sample whose class label is not known and let H be some hypothesis, such that the data sample X may belong to a specified class C. Bayes theorem is used for calculating the posterior probability $P(C \mid X)$, from $P(C)$, $P(X)$ and $P(X \mid C)$, where:

- $P(C \mid X)$ is the posterior probability of target class.
- $P(C)$ is called the prior probability of class.
- $P(X \mid C)$ is the likelihood which is the probability of predictor of given class.
- $P(X)$ is the prior probability of predictor of class.

$$P(C \mid X) = \frac{P(X \mid C) \bullet P(C)}{P(X)} \tag{3.10}$$

The Naive Bayes classifier works as follows.

1. Let D be the training dataset associated with class labels. Each sequence is represented by n−dimensional element vector, $X = (x_1, x_2, x_3, ..., x_n)$

2. Consider that there are m classes $C_1, C_2, C_3, ..., C_m$. Suppose that we want to classify an unknown sequence X, then the classifier will predict that X belongs to the class with higher posterior probability, conditioned on X. The Naive Bayesian classifier assigns an unknown sequence X to the class C_i if and only if $P(C_i \mid X)$ ¿ $P(C_j \mid X)$ for $1 \leq j \leq m$, and $i \neq j$, above posterior probabilities are computed using Bayes Theorem.

Some advantages to use Naive Bayes classifier.

- It requires short computational time for training.

- It improves the classification performance by removing the irrelevant features.

- It has good performance.

Some disadvantages to use Naive Bayes classifier.

- The Naive Bayes classifier requires a very large number of samples to obtain good results.

- Less accurate as compared to other classifiers on some datasets.

3.10 Neural Network

Artificial Neural Network is computing system inspired by biological neural network that constitute animal brain. Such systems learn to perform tasks by considering examples, generally without being programmed with any specific rules. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non linear function of the sum of its inputs. The connections between artificial neurons are called edges. Artificial neurons and edges typically have a weight (W) that adjusts as learning proceeds. The weight W increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times —see Figure 3.29 The Neural Network is constructed from 3 type of layers. (Ognjanovski, 2014).

1. Input layer X. Initial data for the neural network. In our research, we are using PCA data from $15,000$ samples with the first most 50 principal components.

2. Hidden layers. Intermediate layer between input and output layer and it is the place where all the computation is done. In this section we have the activation nodes and usually are noted as W. We are using 50 neurons in this section.

3. Output layer. This produce the result for given inputs.

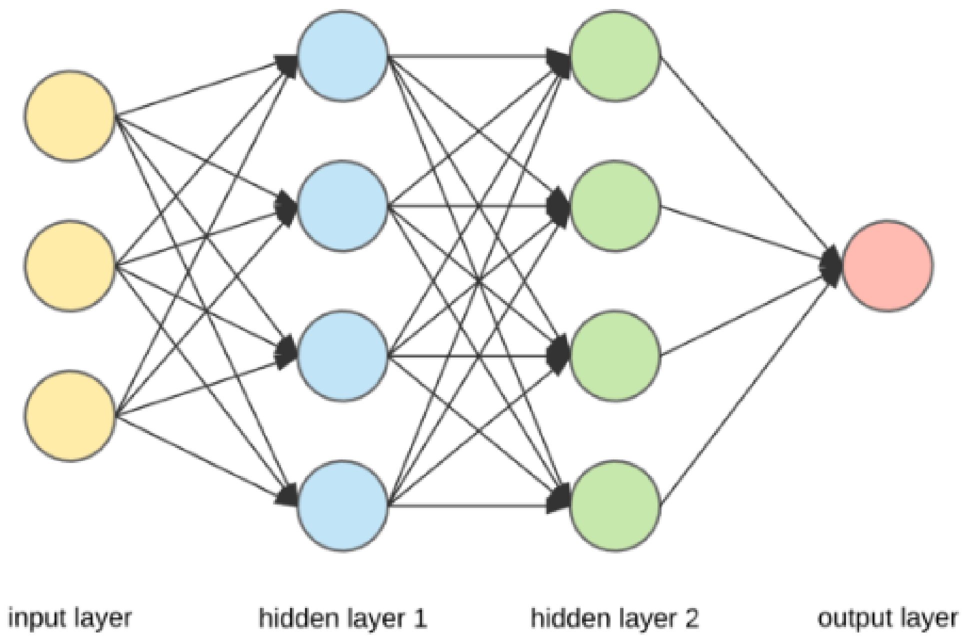


Figure 3.29: Artificial Neural Network (ANN) Layers

- Activation Function. In neural networks the activation function defines if given node should be activated or not based on the weight sum. In our research case we use the Sigmoid Function as the activation function.

$$g(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \tag{3.11}$$

- MSE. We use the mean square error to reach the minimum performance gradient or performance goal.

$$E^n = \frac{1}{2} \times \sum_{k=1}^{c} (Y_k - t_k)^2 \tag{3.12}$$

- Back-propagation. This is a method to calculate the gradient of the loss function $J(W(k))$ (produces the cost associated with a given state) with respect to the weights in an ANN. What we want to do is minimize the cost function $J(W)$ using the optimal set of values for W. Back-propagation is a method we use in order to compute the partial derivative of $J(W)$. Back-propagation is about determining how changing the weights impact the overall cost in the neural network. The derivative of a function, in our case $J(W)$ on each variable weight W tells us the sensitivity of the function with respect to that variable or how changing the variable impacts the function value. (Gour, 2019)

$$W(k+1) = w(k) - \eta \times \nabla J(W(k)) \tag{3.13}$$

Chapter 4

Results

4.1 OCR for Unreadable Damaged Characters on PCBs using PCA, Neural Networks, Naive Bayes, KNN, Discriminats LDC-QDC as classifiers and using a Database of 15,000 image samples

4.1.1 Introduction to all classifiers 15,000 samples

As we mentioned in the chapter before, we have created an image database of $15,000$ samples images. Each image has a matrix size of (50×30) having 1500 features if we convert matrix to row form. We compute the PCA for a matrix of $15,000 \times 1,500 + 1$ column for classes —see Figure 4.1.

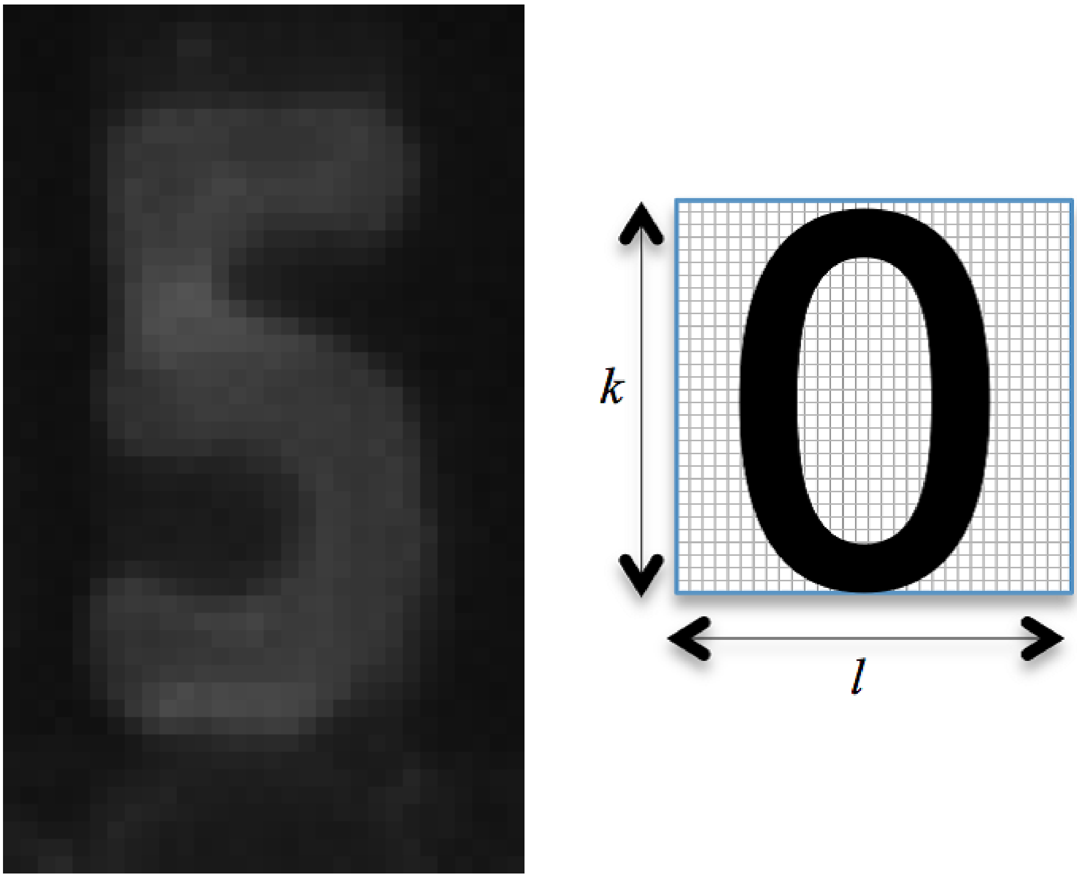


Figure 4.1: Real digit image and size structure $k = 50$ and $l = 30$

We take the first 50 PCA components in a matrix of $15,000$ samples rows —see Figure 4.2 and $1,500$ samples features plus one column for theirs classes that will be used in all algorithms for training, test and validation. We use a 70% Training, 15% Validation and 15% Test.

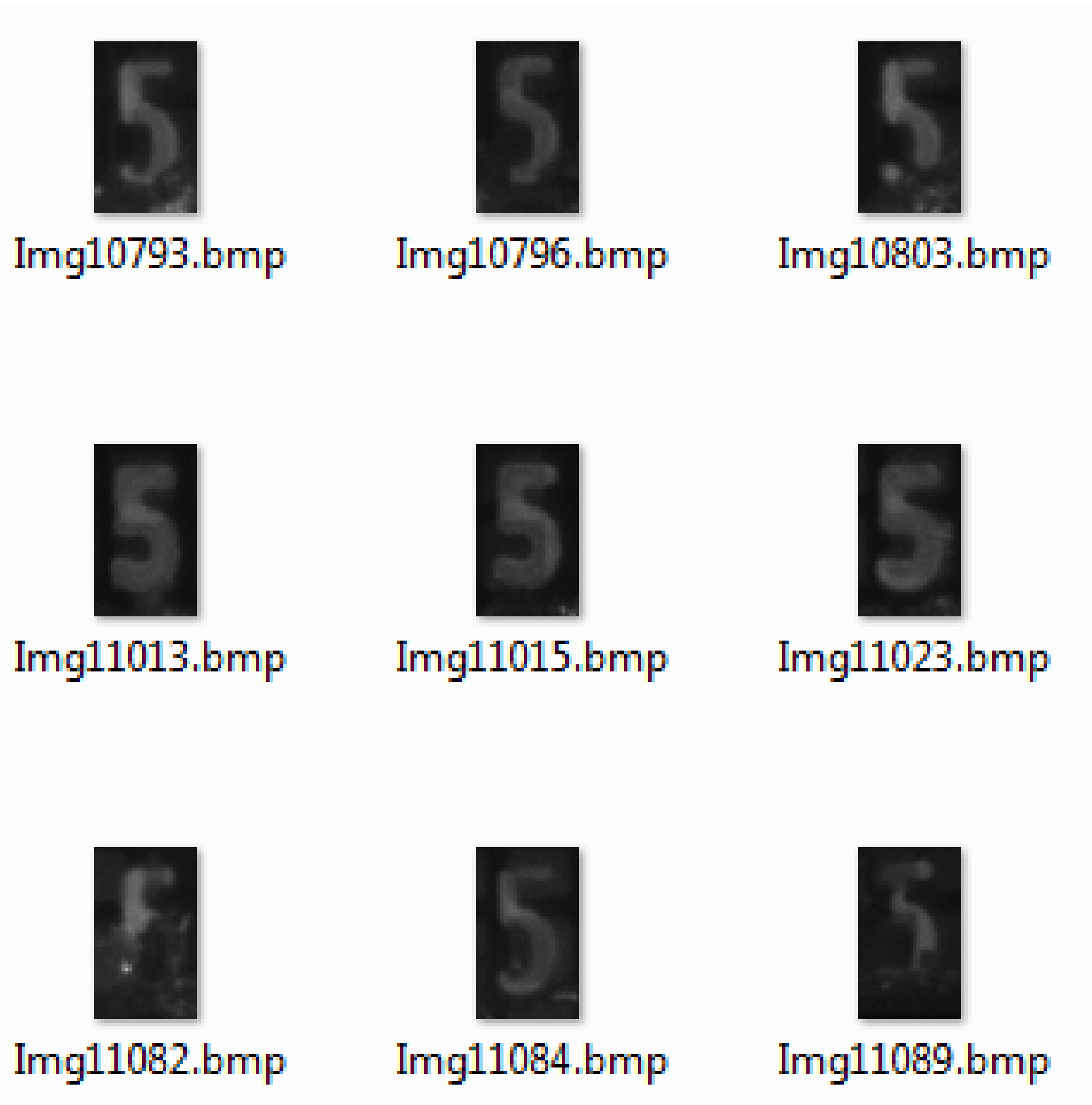


Figure 4.2: Thousand Digits for PCA.

4.1.2 PCA Neural Networks Results.

1. We construct a Neural Network with 50 neurons, 50 PCA components, 10 classes and 15,000 samples —see Figure 4.3.

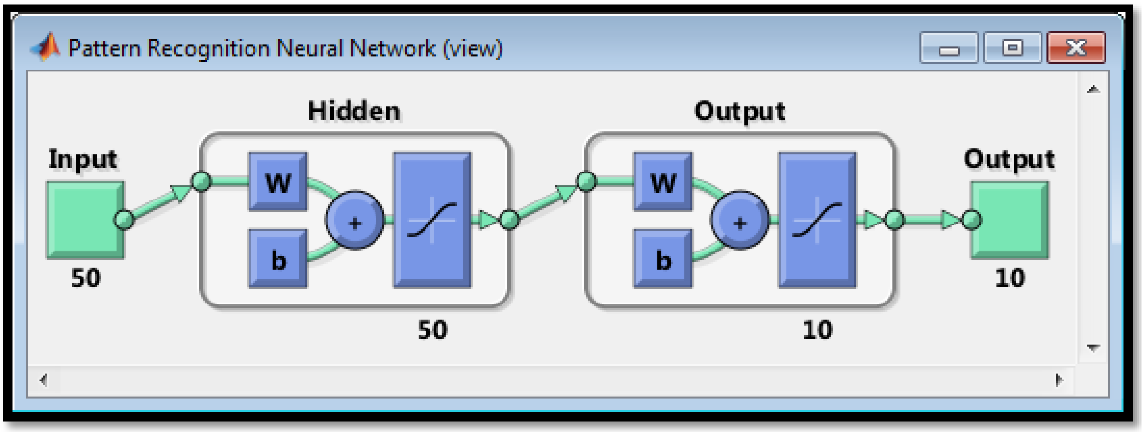


Figure 4.3: Neural Network.

2. Valiudation and Test Data setup —see Figure 4.4.

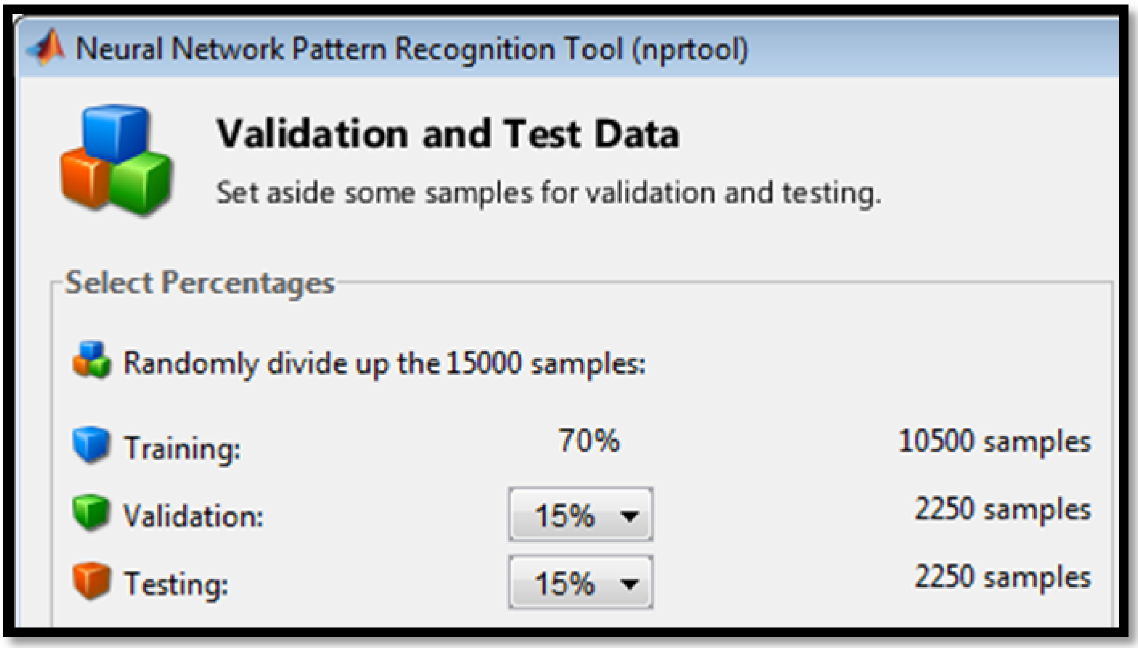


Figure 4.4: Validation and Test Data.

3. Neural Network Results —see Figure 4.5.

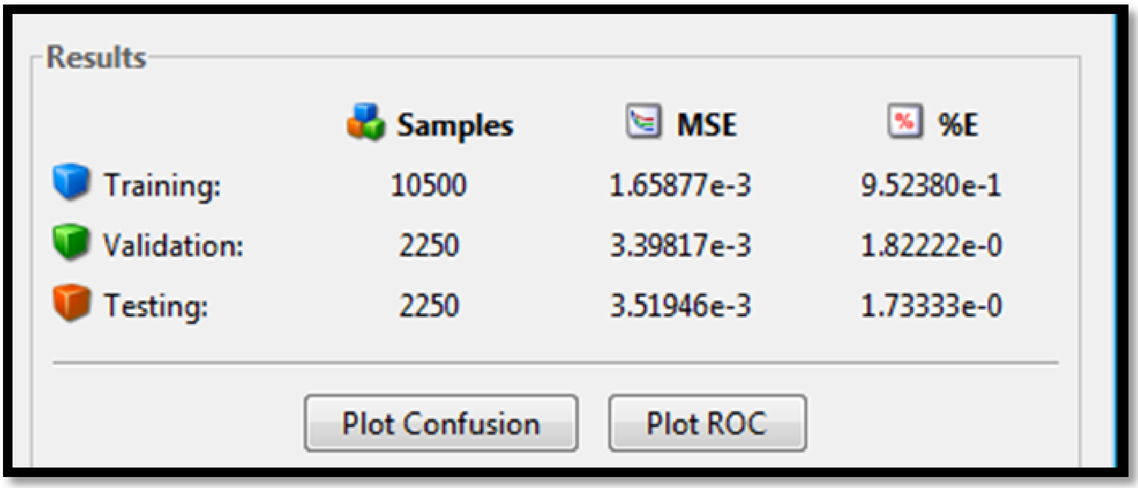


Figure 4.5: Neural Network Results.

4. Best Validation Performance at Epoch 183 —see Figure 4.6.

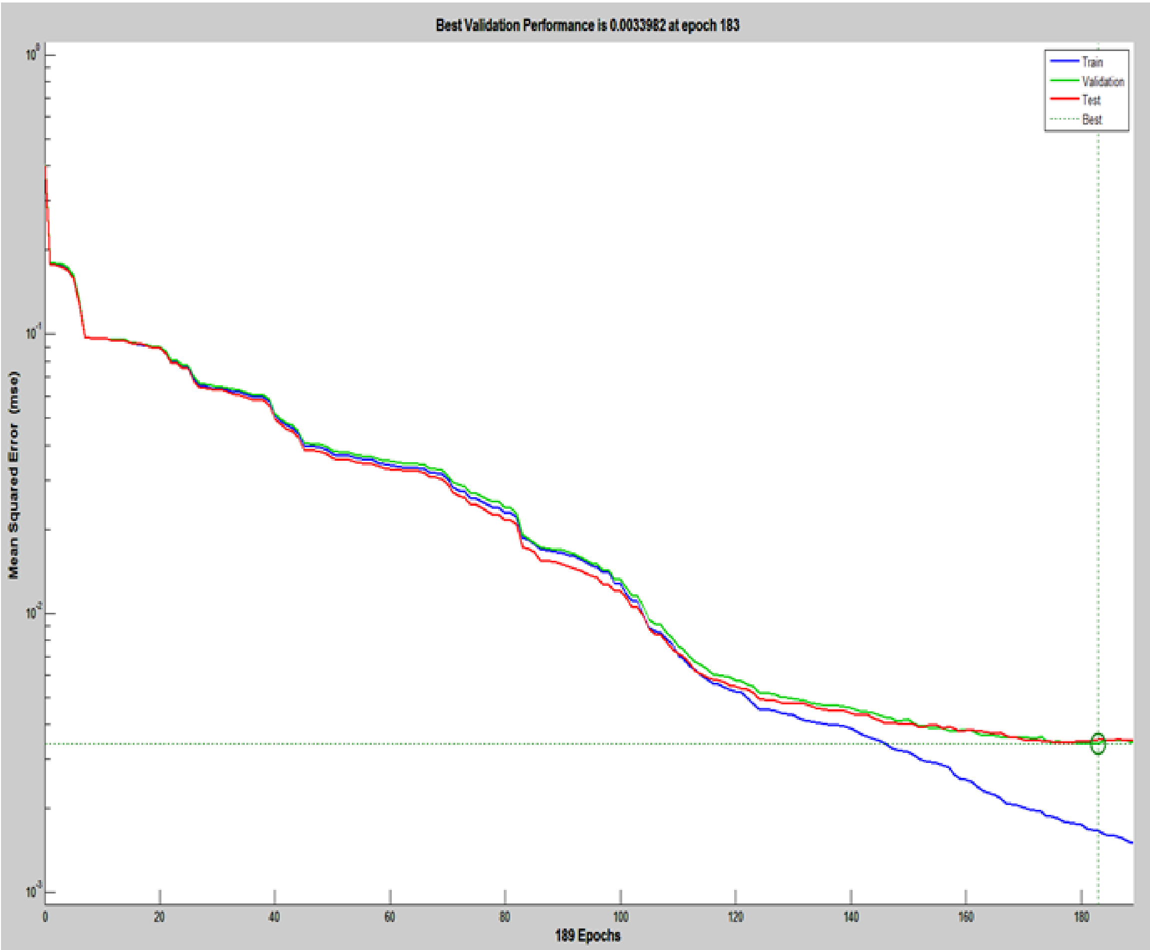


Figure 4.6: Neural Network Results Plot.

5. Test Confusion Matrix —see Figure 4.7.

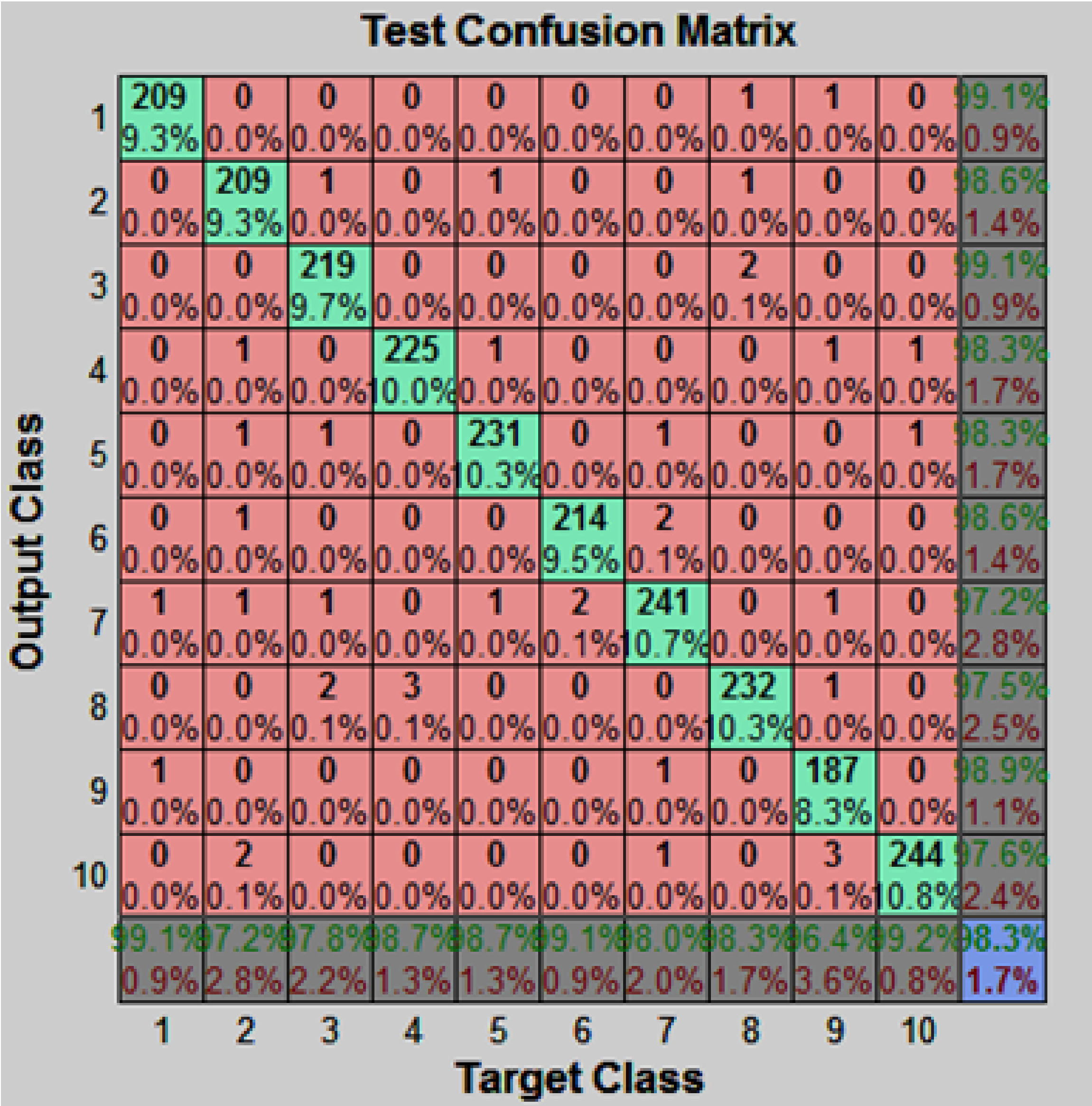


Figure 4.7: Neural Network Results Test Confusion Matrix.

4.1.3 PCA Naive Bayes Results

We compute first a training procedure to know the best PCA components that we use —see Figure 4.8.

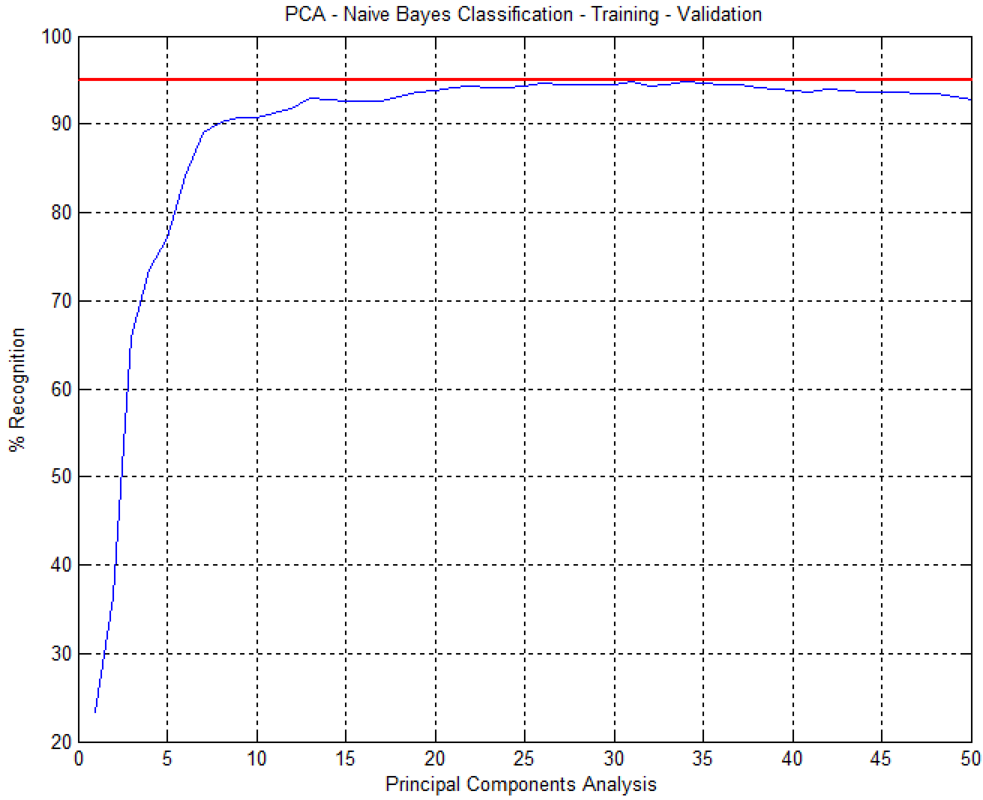


Figure 4.8: PCA Naive Bayes Recognition Training

We apply Naive Bayes Algorithm using the maximum recognition at PCA 34 —see Figure 4.9.

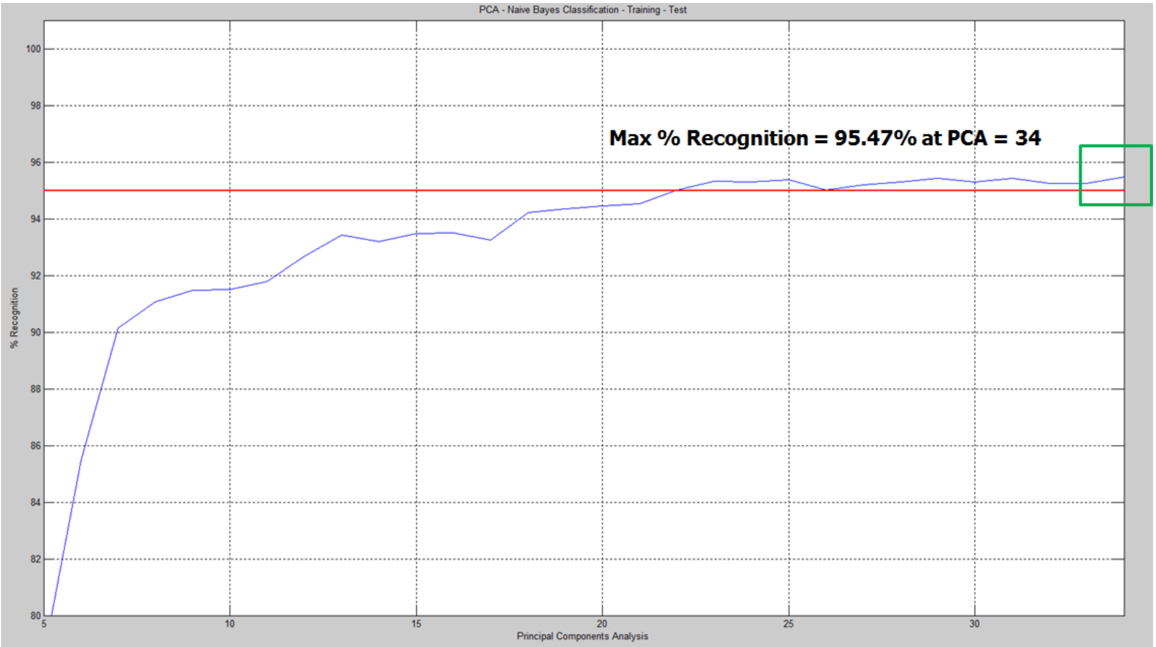


Figure 4.9: Naive Bayes Max %Recognition = 95.47%

4.1.4 PCA KNN Results

We compute first a training procedure using $k = [1, 3, 5]$ to know the best PCA components that we use —see Figure 4.10.

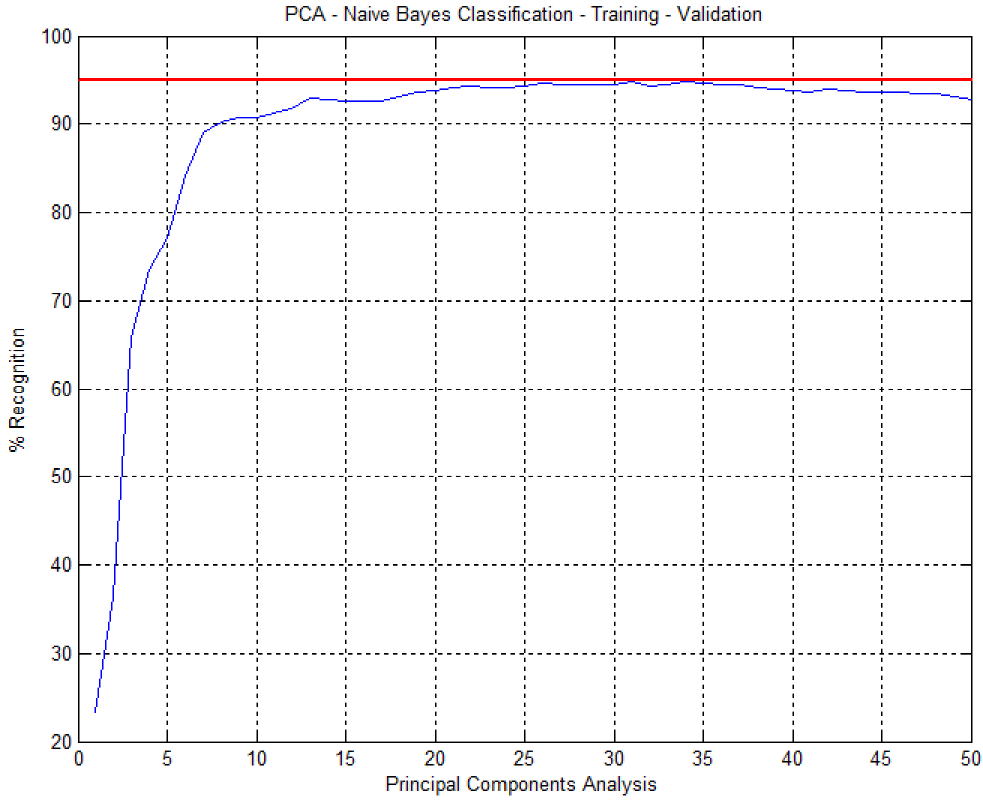


Figure 4.10: PCA KNN Recognition Training with $k = [1, 3, 5]$

We apply KNN Algorithm and have maximum recognition at PCA 31 and using $k = 5$ —see Figure 4.11.

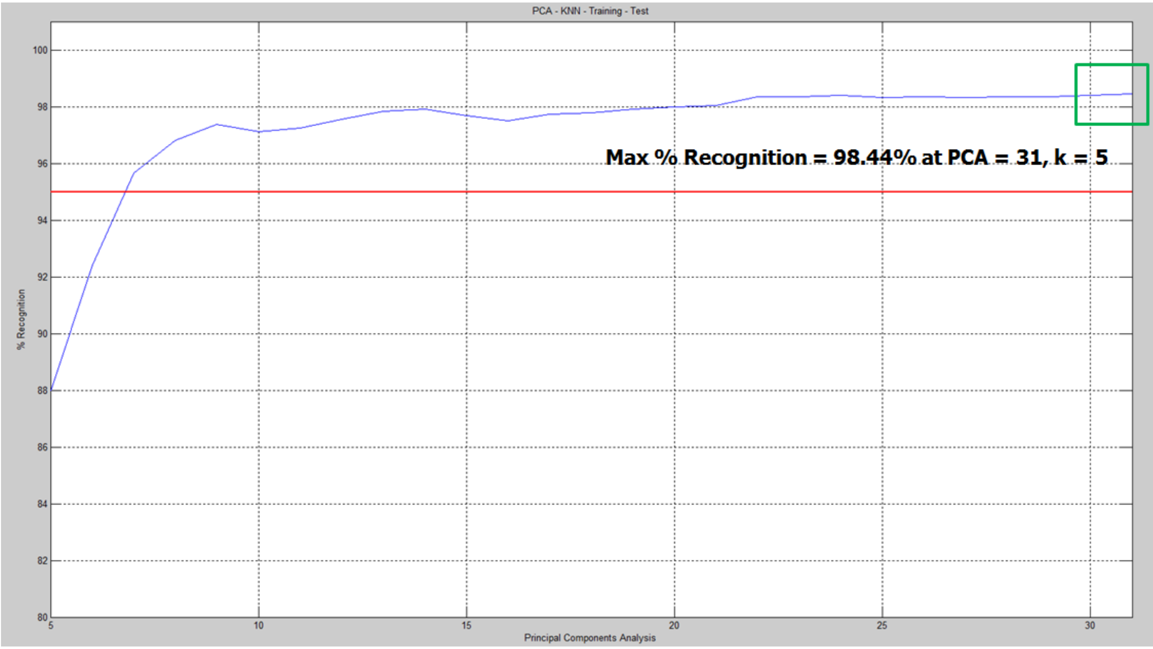


Figure 4.11: KNN Max %Recognition = 98.44%

4.1.5 PCA Discriminants QDC Results

We use compute LDC and QDC algorithms for the first 50 PCA components. It's clearly the QDC has a better improvement as we can see at the recognition graphic —see Figure 4.12. We chose QDC for the best discriminant and apply QDC Algorithm and with maximum recognition at PCA 42 from previous Figure —see Figure 4.13.

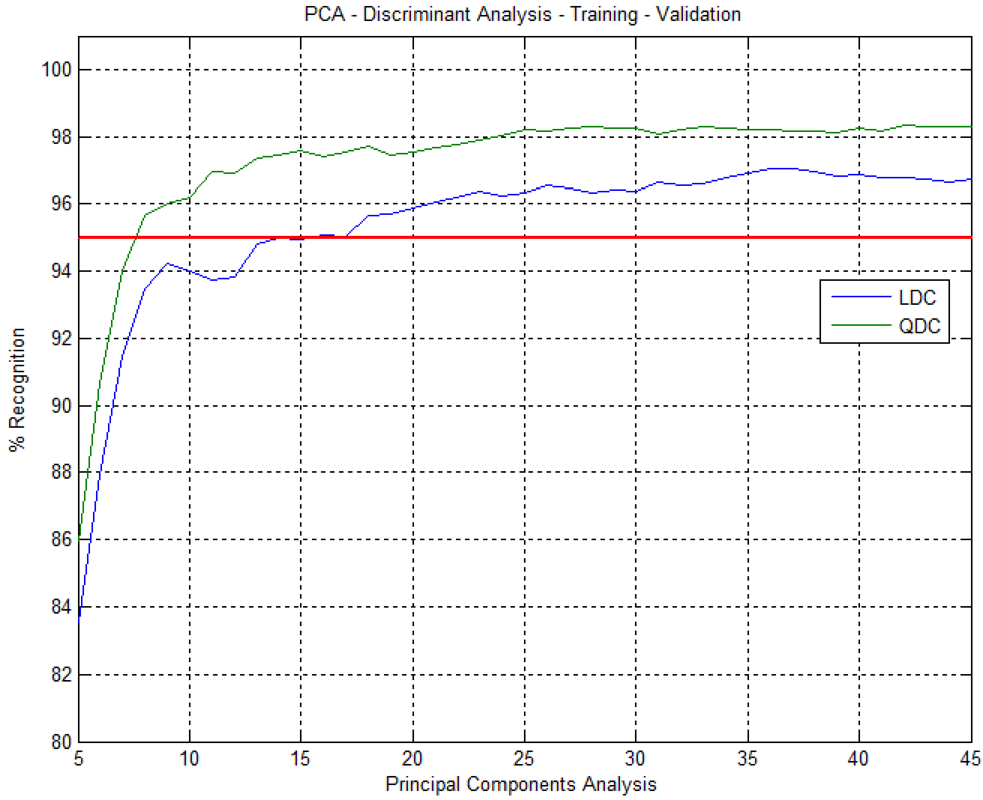


Figure 4.12: PCA LDC QDC First 50 PCA Components Zoom

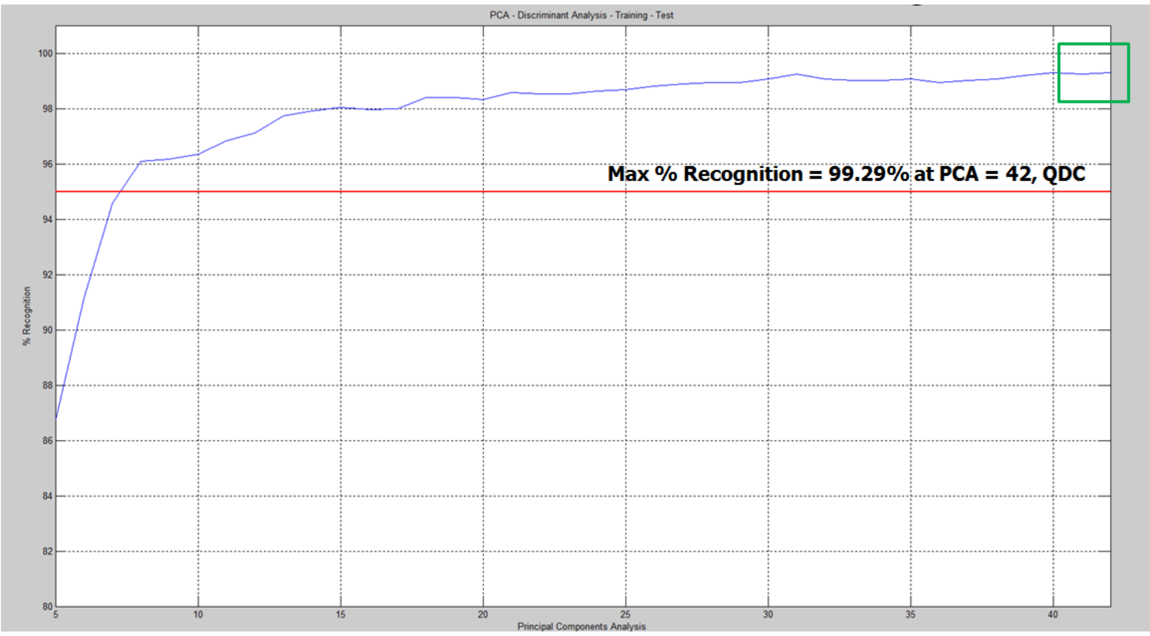


Figure 4.13: QDC Max %Recognition = 99.29%

4.1.6 Machine Learning Algorithms Results

Finally we can show a comparative using the most common machine learning algorithms —see Figure 4.14.

CLASSIFIER	% RECOGNITION
NAIVE BAYES, PCA = 34	95.47%
GSC, KNN, k = 5	96.76%
NN, PCA = 50	98.30%
KNN, k = 5, PCA = 31	98.44%
DISCRIMINANT, QDC, PCA = 42	99.29%

Figure 4.14: All classifiers. Max %Recognition = 99.29% using QDC Discriminant

4.2 OCR for Unreadable Damaged Characters on PCBs using Principal Component Analysis and Bayesian Discriminant Functions using initial dataset of 500 samples

4.2.1 Introduction Bayesian

Optical character recognition (OCR) has been an important technology used to convert characters from a digital image to a digital text. There are basically two types of OCR algorithms: the first technique is related with the matching of matrix images, where an alphabet of stored character images is used to compare with an input image (Mori, 1992), (Casey, 1996). This pattern matching does not work well when new fonts are encountered or input character images are unreadable. The second technique decomposes an input image to extract the principal features (Misiak, 2008), (Jolliffe, 2002), (Nedevschi, 2012). Then, classifiers are used to compare the input image features with some stored image features and choose the best match.

Our actual system implemented uses the traditional OCR technique, pattern matching. Our implemented vision system reads identification characters on printed circuits boards (PCBs) for lot integrity and machine control. This commonly used technique is not robust enough because many of the images of PCBs shown some damage on the characters due dirt or the results of bad previous processes (Mori, 1992), (Desrochers, 2001). Our actual OCR detectability is around 97% at best. Our system starts with a monochrome VGA image acquisition of the upper left section of a PCB, using a NI-1752 smart camera, with a full resolution of 640x480 pixels with a maximum data transfer of 60 fps using a GigE port. The selected resolution and data transfer speed parameters meet the factory production schedule of inspected PCBs. The camera has a gray-scale output image type with a maximum character resolution to cover the entire PCB characters positions —see Figure 4.15.

Figure 4.15: PCB with no damaged characters Bayesians.

As mentioned before, due to problems with previous processes in the production line, some PCBs present some residual dirt over the characters, making some characters unreadable for the pattern matching technique, as shown in the following Figure 4.16.

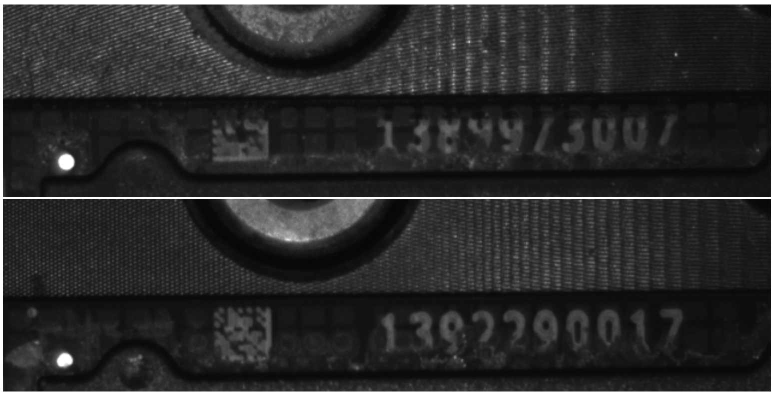


Figure 4.16: PCBs with evident residual dirt over characters.

The principal problem is that operators have lower throughput than automatic OCR software, and this leads to manually writing down the information from the screen when the actual recognition software fails, increasing the process time, making possible errors from wrong readings, inducing higher production costs. Taking into consideration these facts, a better approach has to be considered (Misiak, 2008). This paper presents the proposal for implementing a character recognition technique for unreadable characters using extraction features and Bayesian classifiers.

4.2.2 Data Set Constructor Bayesian

Our implementation starts with an experimental dataset constructed of 500 character images. In this dataset, we have 50 images that correspond to each numerical digit image from 0 to 9. Next —see Figure 4.17 shows several digit image samples.

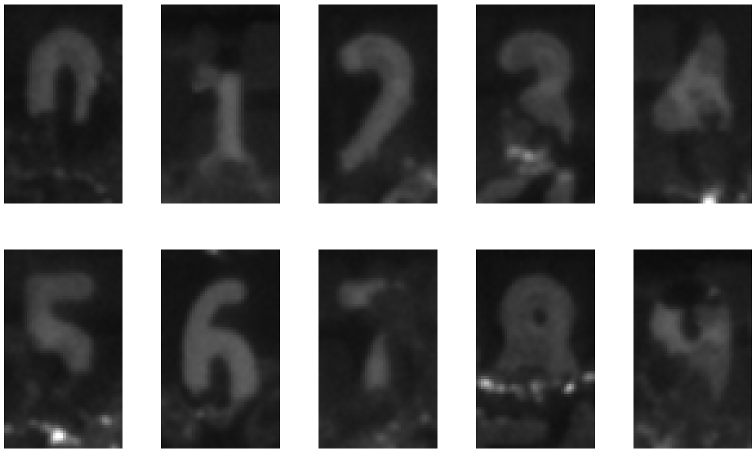


Figure 4.17: Some damaged digit images from dataset.

For our previous dataset, we let I_i any (k, l) digit image —see Figure 4.18, from the original dataset. $\forall I_i$:

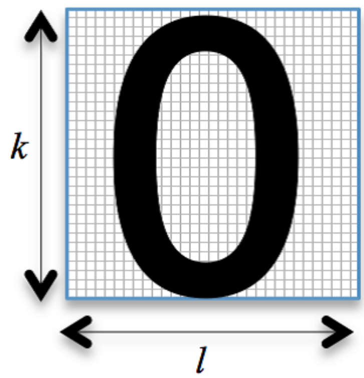


Figure 4.18: I_i digit image matrix with size (k, l).

- Convert I_i to gray-scale (if previous images are RGB type).
- Transform matrix I_i to a row-vector of size $(1,\ k \times l)$.
- Create a matrix M of size $(n, k \times l)$ where n is the number of samples for each digit image, $M \leftarrow M \cup I_i$

4.2.3 Principal Components Analysis and Bayesian Classification Bayesian

The purpose of this portion of the paper is to map the matrix M into the eigenspace by means of the the first P principal components. We follow the next steps:

1. Extract the mean for each column:

   $\Phi_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} m_{ij}$

   $\hat{M} \leftarrow M_j - \Phi_j$

2. Compute the covariance matrix γ of $\hat{M}$
3. Compute eigenvectors and eigenvalues $(PC, V$ of γ
4. Sort matrix PC by columns in descend order ruled by vector V
5. Project $\hat{M}$ into the first P principal components:

$$X \leftarrow \hat{M} \times PC \tag{4.1}$$

6. A new digit dataset is now assembled, X

The principle component analysis (PCA) digit image test dataset must be processed by applying the first 5 steps, but using ϕ_j and $\hat{M}$ calculated from the training set. Given the new Eigen-data set, two Bayesian algorithms, linear and quadratic discriminant classifiers must be trained and tested by means of 10x10 cross-validation method. These algorithms are widely used parametric methods, which assume that the class distributions are multivariate Gaussian (Duda, 2001; Ibraheem, 2011; Yang, 2005). With linear discriminant analysis (LDA), all classes are assumed to have the same covariance matrix, but quadratic discriminant analysis (QDA) does not need such an assumption; however, the number of parameters to be estimated from the data available for each class is much higher, entailing lower statistical significance. The discriminant functions associated to each classifier are defined as:

1. Linear Discriminant Classifier

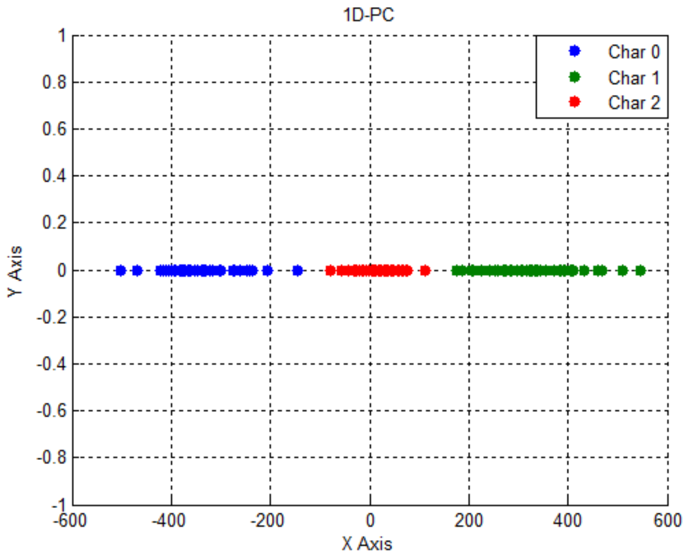
$$g_k(x) = \ln P(\omega_k) + \mu_k^t \sum^{-1} x - \frac{1}{2}\mu_k^t \sum^{-1} \mu_k \tag{4.2}$$

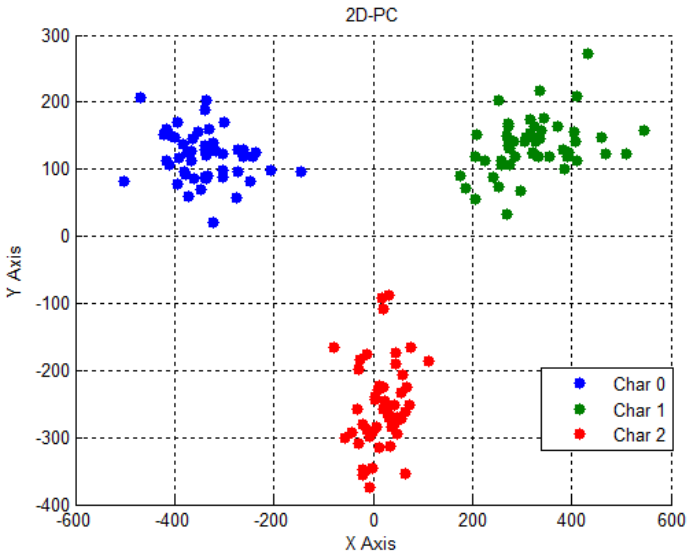
2. Quadratic Discriminant Classifier

$$g_k(x) = \ln P(\omega_k) - \frac{1}{2}(\ln |\sum{}_k| + (x-\mu_k)^t \sum{}_k^{-1} (x-\mu_k)) \tag{4.3}$$

4.2.4 Experimental Results Bayesian

In the following Figure 4.19 we show the first 3 principal components from matrix X with only 3 characters of data: 0, 1 and 2.


1D-PC
Char 0
Char 1
Char 2
Y Axis
X Axis
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-600
-400
-200
0
200
400
600


2D-PC
Char 0
Char 1
Char 2
Y Axis
X Axis
300
200
100
0
-100
-200
-300
-400
-600
-400
-200
0
200
400
600

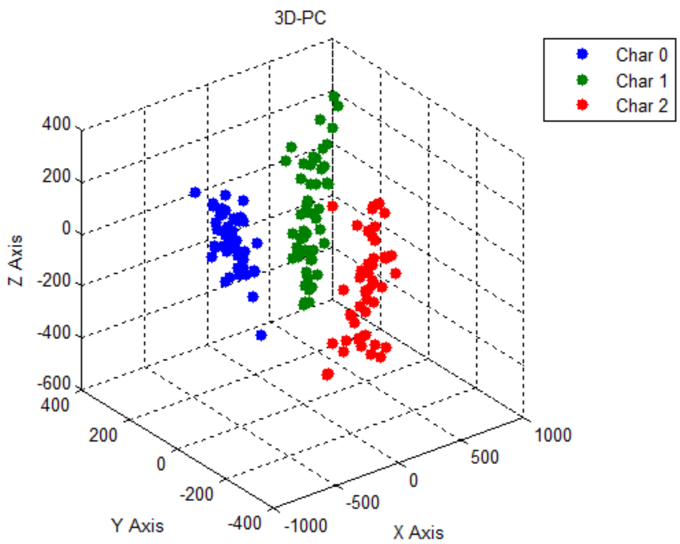


Figure 4.19: 1 PC, 2 PC and 3 PC for characters 0, 1 and 2.

For more digit characters, the principal components were not easy to visually classify—see Figure 4.20:

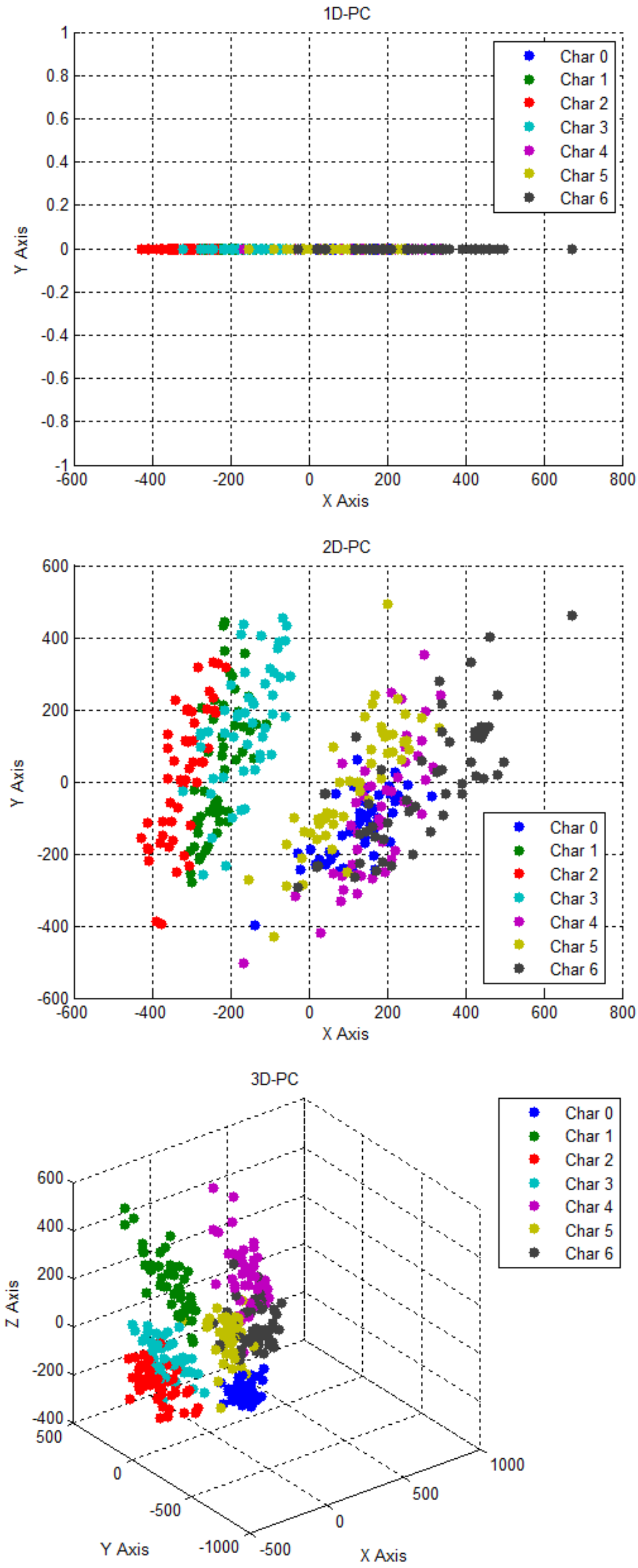


Figure 4.20: 1 PC, 2 PC and 3 PC for characters 0 to 6.

From previous simulations we can see that using 2 or 3 principal components is not enough to have a difference in the proximity of the characters groups. It is clear that groups for characters 0, 1 and 2 are close. The next step will use the linear and quadratic discriminant classifiers using more than 3 principal components from X matrix.

Our classification process, for linear and quadratic discriminants, was trained in the complete training data set and tests the performance in the test data set. A 10x10 cross-validation model validation technique was computed to estimate how our classification model was performed, (Kohavi, 1995) —see Figure 4.21 shows these experimental results:

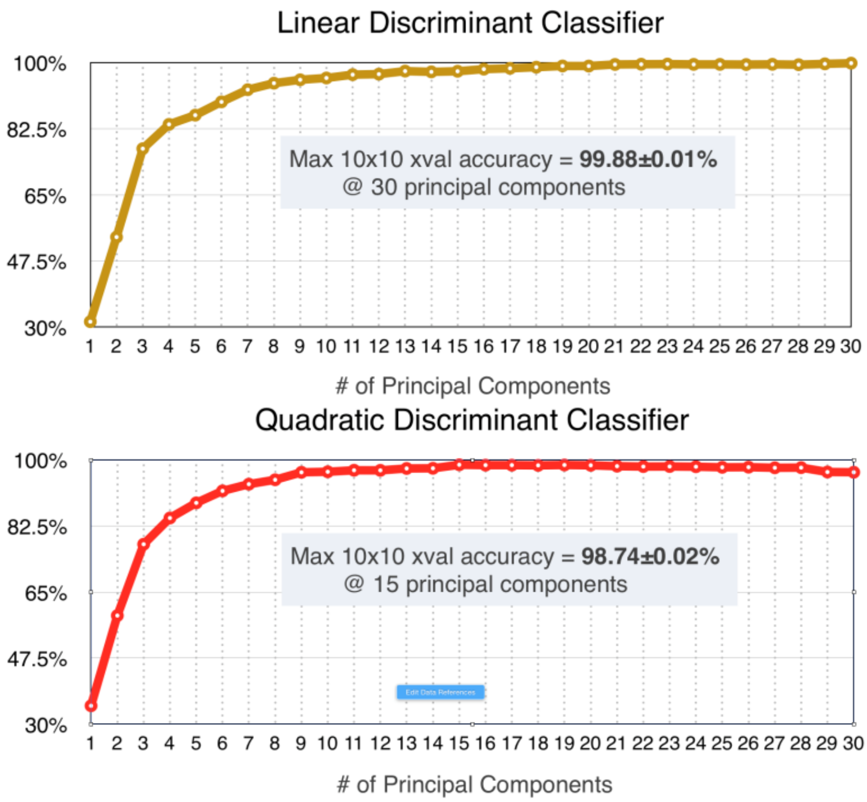


Figure 4.21: 10x10 cross-validation for LDC and QDC classifiers.

For previous simulations, it has seen that classification algorithms yield promising results. The 10x10 cross-validation recognition rate for the linear discriminant classifier shows an interesting 99.88%, by using the first 30 principal components.

To check the performance of the linear discriminant classifier we compute the confusion matrix, where it shows a 100% of recognition rate at characters 0, 1, 3, 6, 7, 8 and 9. Relative difficulties are seen in character 2, which was misclassified as character 7 in one case. Character 4 was classified as character 1 in just one case. Quadratic discriminant classifier performance was as follow: 10x10 cross-validation recognition rate of 98.74% by means of the first 15 principal components. The average confusion matrix shows perfect recognition in only one character, 9.

4.3 OCR for Unreadable Damaged Characters on PCBs Using GSC Algorithm and kNN Classifier using initial dataset of 500 samples

4.3.1 Introduction KNN

Optical character recognition (OCR) has been an important technology used to convert characters from a digital image to a digital text. There are basically two types of OCR algorithms: the first technique is related with the matching of matrix images, where an alphabet of stored character images is used to compare with an input image (Casey, 1996; Mori, 1992). This pattern matching does not work well when new fonts are encountered or input character images are unreadable. The second technique decomposes an input image to extract the principal features (Desrochers, 2001; Duda, 2001; Favata, 1996). Then, classifiers are used to compare the input image features with some stored image features and choose the best match. Our actual system implemented at the factory uses the traditional OCR technique i.e. pattern matching. Our implemented vision system reads identification characters on printed circuits boards (PCBs) for lot integrity and machine control. This commonly-used technique is not robust enough because many of the images on PCBs present some damaged characters due to dirt or as a result of bad previous processes (Mori, 1992). Actual OCR detectability is around 95% at best. It starts with a monochrome VGA image acquisition of the upper left section of a PCB, using a NI-1752 smart camera, with full resolution, 640x480 pixels and maximum data transfer @60 fps using a GigE port. The selected resolution and data transfer speed parameters meet the factory production schedule of inspected PCBs. The camera has a grayscale output image type with a maximum character resolution to cover the entire PCB characters positions —see Figure 4.22.

Figure 4.22: PCB with no damaged characters.

Due to some problems with previous processes in the production line, some PCBs will have some residual dirt over the characters, making some characters unreadable for the pattern matching technique, as shown in the following Figure 4.23.

Figure 4.23: PCBs with evident residual dirt over characters.

The principal problem is that operators have lower throughput than automatic OCR software, and this leads to manually writing down the information from the screen when the actual recognition software fails, increasing the process time, making possible errors from wrong readings, resulting in higher production costs. Taking into consideration these facts, a better approach has to be considered [3]. This paper presents a proposal for implementing a character recognition technique for unreadable characters using Gradient, Structural and Concavity (GSC) extraction features and K-Nearest Neighbor Classifier using Euclidian Distance (Ballerini, 2012; Hu, 2007; Rodriguez, 2002).

4.3.2 Data Set KNN

The experimental data set consists of 500 character images, 50 images correspond to each numerical digit from 0 to 9 —see Figure 4.24 shows some examples of damaged digit image samples.

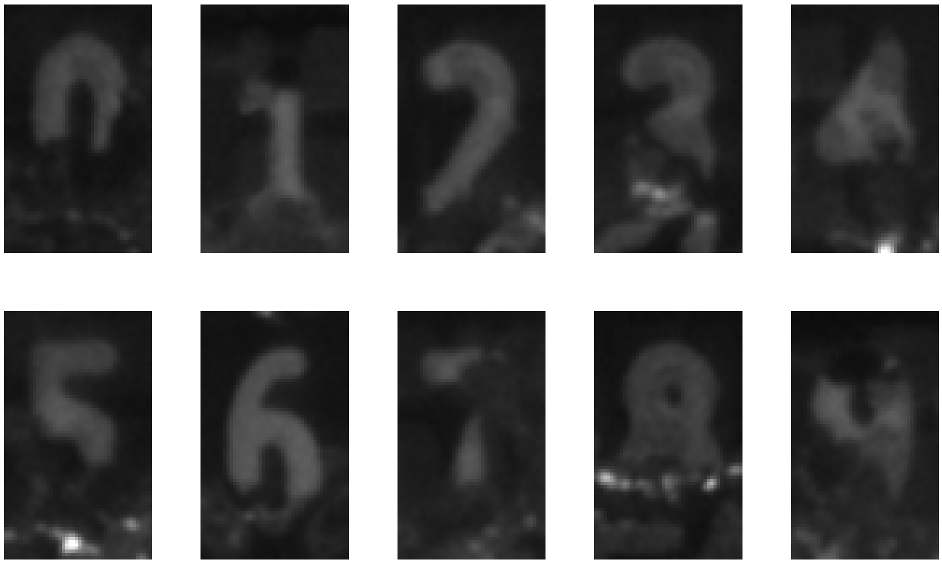


Figure 4.24: Some damaged digit images from dataset.

For our previous dataset, a pre-processing step is applied as follows: Let I_i any digit image of size (k, l) —see Figure 4.25, from the original dataset, $\forall I_i$:

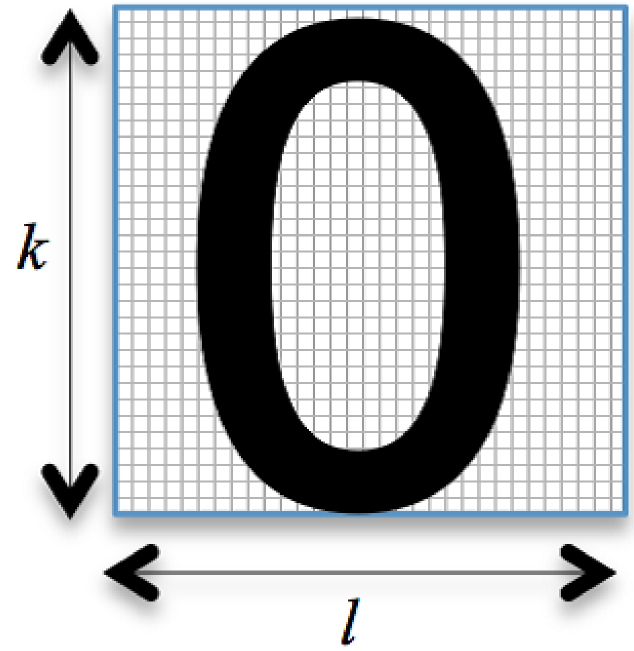


Figure 4.25: I_i digit image matrix with size (k, l). k=50 and l=30 in our experiments

- Convert to gray-scale (if previous images are color RGB type).
- A threshold is applied to binarize.
- I_i is split in 60 non-overlapping regions (10 x 6 grid), as shown is the Figure 4.26.

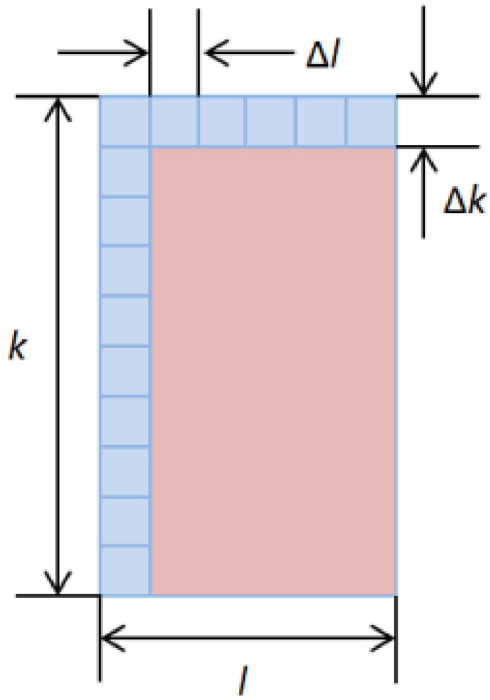


Figure 4.26: I_i is split in a 10 x 6 grid of $\delta k = \delta l = 5$ pixels

4.3.3 Gradient, Structural and Concavity (GSC) Recognition Algorithm KNN

The GSC algorithm to extract information from the image was implemented [3]. It constructs features of an image by applying a three-step feature extraction process: 1-Gradient step detect local features by analyzing the stroke shape on small distance; 2-Structural step, extract features from stroke trajectories by extending distances of gradient; 3-Concavity analysis detects stroke relationships across the image —see Figure 4.27 shows the final Total Feature Vector (TFV) constructed for each image.

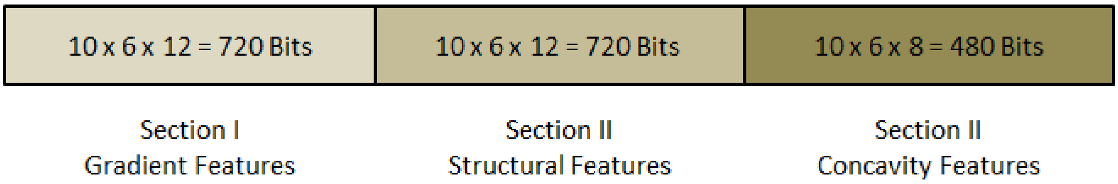


Figure 4.27: Total Feature Vector of each I_i digit image with size of 1920 pixels

Section I - Gradient Features. Two dimensional convolutions in the X and Y direction is applied to get the gradient features using an 3 x 3 Sobel operators on the original I_i binary image —see Figure 4.28. Gradients from an image representation of the Digit Character 9 are shown in Figure 4.29.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_x = K_x * M \qquad G_y = K_y * M$$

Figure 4.28: G_x and G_y are the 2D convolution of a 3 x 3 matrix for every pixel on original I_i digit image matrix. An extra zero padded border is added to I_i.

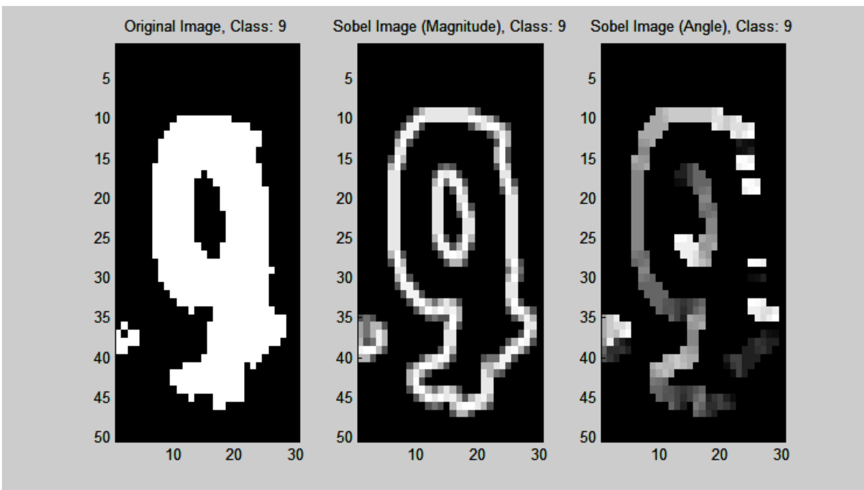


Figure 4.29: Gradient magnitude and direction are shown for Digit Image 9

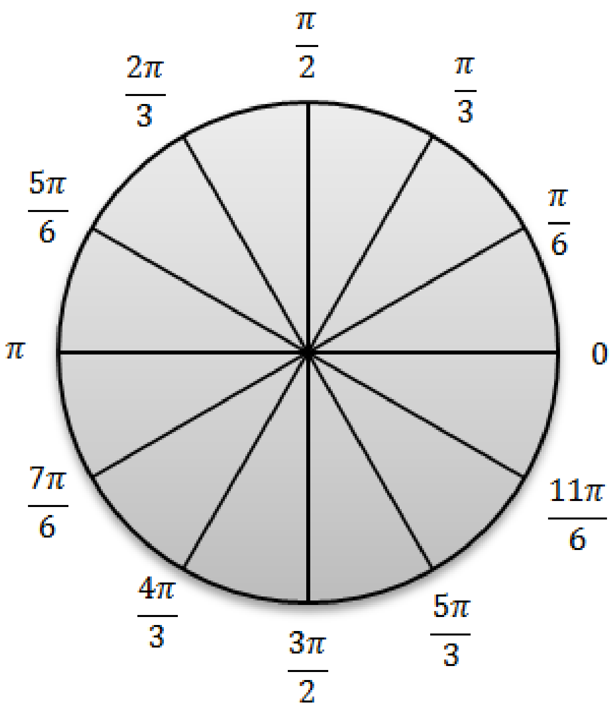


Figure 4.30: Gradient range from 0 to 2π in 12 equal space regions

A histogram is applied for each of the 60 non-overlapping regions of the complete image grid, incrementing the counter for every Gradient Angle that falls in each region —see Figure 4.30. A threshold is applied and a final 720 Bits is created for the first part of the TFV, as previously shown in Figure 4.27.

Section II - Structural Features. For each pixel of the expanded Ii digit image with zero padded border, a set of 12 rules is applied using 8 pixels around the main pixel. These rules look for specific gradient patterns form with the nearest pixels, like horizontal lines $(0, 4)$, vertical lines $(2, 6)$, diagonals $[(5, 1), (3, 7)]$ and corners $[(0, 2), (2, 4), (4, 6), (6, 0)]$ —see Figure 4.31 shows these rules in a graphical positioning:

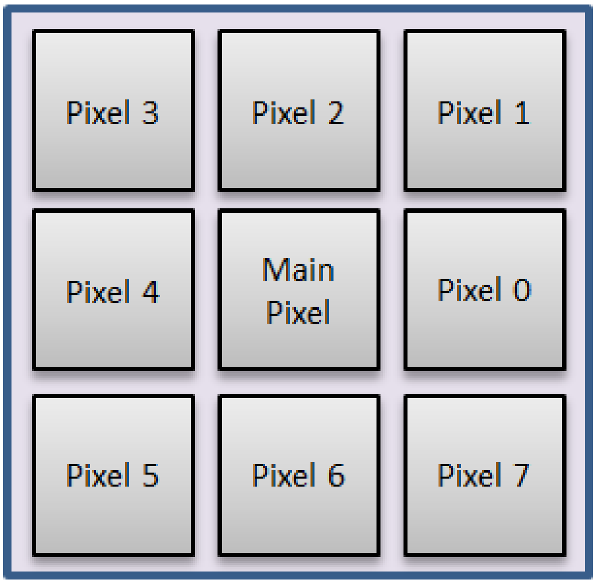


Figure 4.31: Eight nearest pixels around Main Pixel

A threshold is applied for each of the 12 rules result, for each of the 60 non-overlapping regions to binarize the complete set. A final 720 Bits set is created for the second part of the TFV, as previously shown in Figure 4.27.

Section III - Concavity Features. Three feature sections form the last part of the GSC algorithm:

1. **Coarse Pixel Density**: A histogram is applied to count all the character pixels at each of the 60 non-overlapping regions. Then, a threshold is applied to binarize the result. For this, a 60 Bits new set is included as the first part of the Concavity Features.

2. **Large-Stroke**: Like the previous section, two histograms are applied, one for the horizontal and one for the vertical pixels strokes in each direction. A threshold is applied to binarize the result. For this section, a 2 x 60 = 120 Bits new set is included as the second part.

3. **Upward, Downward, Left, Right and Holes**: In this last section of the concavity features, for every pixel in each of the 60 non-overlapping regions, rays are fired to hit character pixels, borders and check if we have holes or

character pixels in all directions. In this last section, a 5 x 60 = 300 Bits new set is included.

As shown in Figure 4.27, the TFV for the Section III has the following Bits set:

$$1x(10x6) + 2x(10x6) + 5x(10x6) = 480Bits \tag{4.4}$$

4.3.4 K-Nearest Neighbor Classifier

The k-Nearest Neighbor (kNN) Classifier Algorithm was chosen for the damaged character recognition step. It was selected because of its simplicity and fast performance, and the absence of prior assumptions about data set probability distributions. The classification occurs when a majority vote among kNNs with respect to any particular test set is given. In the complete experiments k parameter was fixed to 3. The resulting Total Feature Matrix has the form as shown in Figure 4.32. See Class is added at the end column:

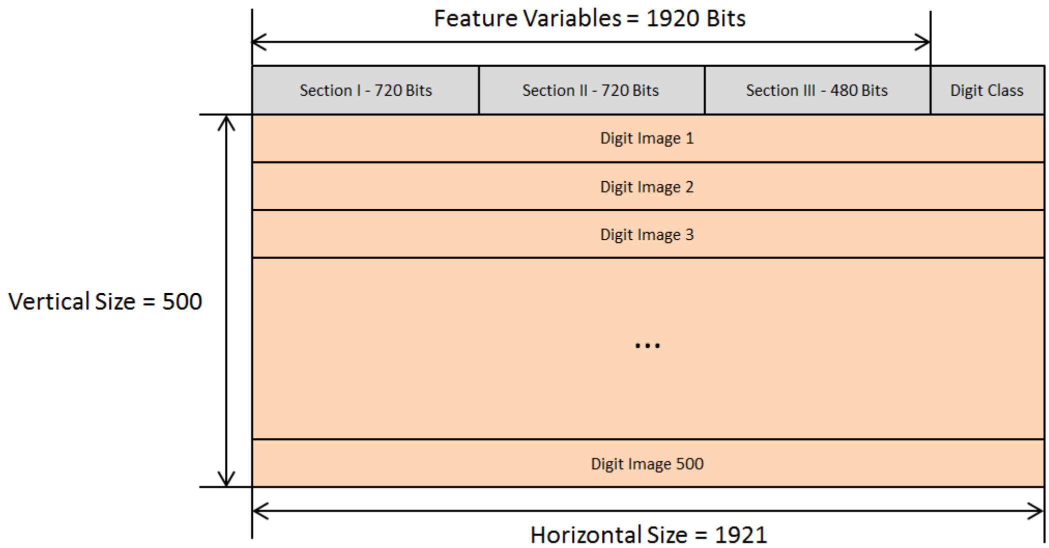


Figure 4.32: Final Total Feature Matrix for all Image Dataset with Digit Class included 0 to 9

The Euclidean Distance was chosen as the first approach to compute the distance metric as follows:

$$D_{Euclidian} = \sqrt{\sum_{i=1}^{l} (Test_i - Training_i)^2} \quad (4.5)$$

4.3.5 Experimental Results

A first-round experiments were conducted as follows: a kNN Euclidian Distance and a K = 3 was trained and tested varying the proportion of training-test samples. A range of 10% to 90% training and 90% to 10% test sets were analyzed. The following Figure 4.33 shows this experimental setting result —see Figure 4.34 shows a zoom of the 60% to 90% section.

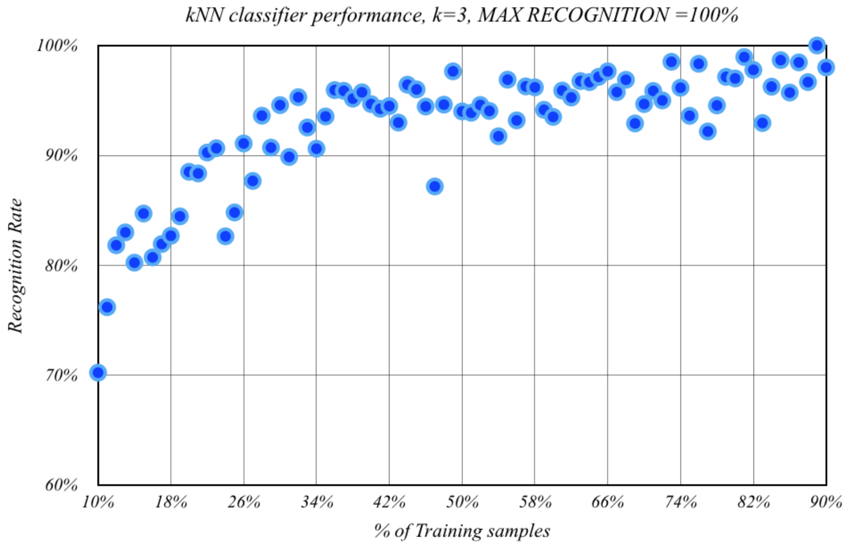


Figure 4.33: kNN Classifier performance for all image dataset

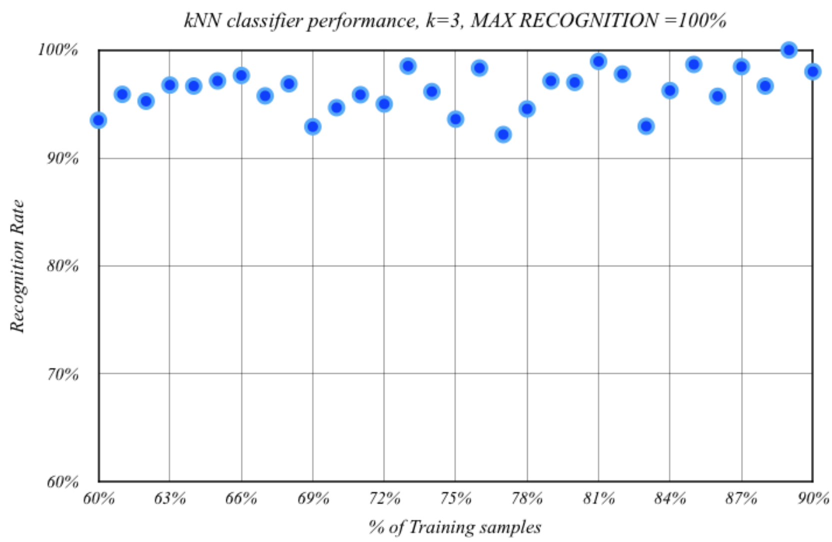


Figure 4.34: Zoom in the kNN Classifier performance.

It's seen that classification stage yields promising results. The kNN Classifier shows interesting peaks of 100% recognition rate at high training percentage by using the GSC algorithm. In order to assess more precisely the classification performance of the GSC+kNN proposal, Monte Carlo cross-validation strategy was selected (Qing-Song, 2001). A total of 100 random data splits of 90% training and 10% test samples were analyzed —see Figure 4.35 shows results and main statistics.

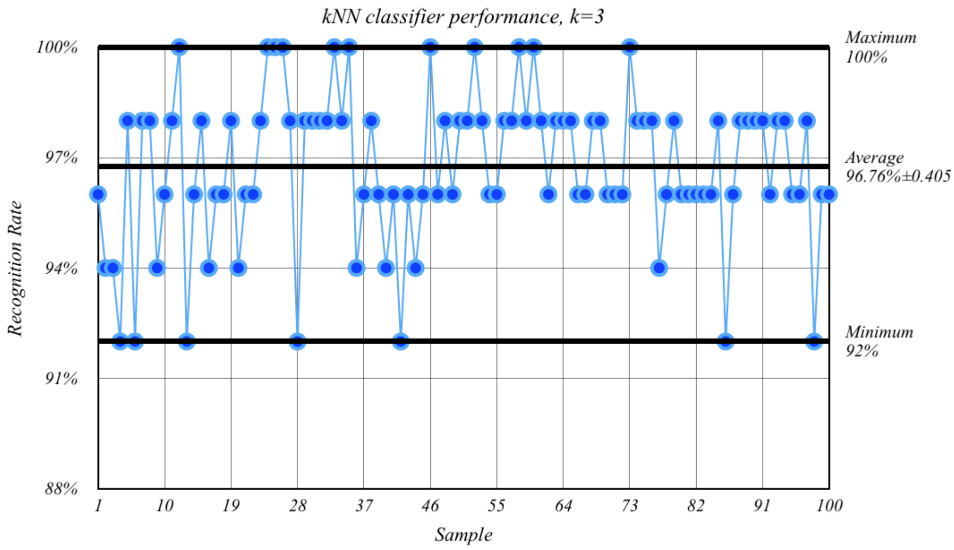


Figure 4.35: Monte Carlo 100 cross-validations with a 90%-10% training-test split.

Chapter 5

Conclusions

5.1 Conclusions

- We have described in this thesis the research and development that we have been doing in the last few years to make a robust vision system that could have a high recognition rate when the PCB identification numbers are damaged due environmental dust, molding plastic or any other particles that can cover some sections of the digits area on PCB.

- Our vision system is designed to be added to any process machine that have available input−output signals for using our communication control board. Our vision system is capable to be implemented in any process that request to inspect identification numbers on PCB. As we explained before, our vision system can work from a smart camera to a camera that use frame grabbers like GigE or 1394b Firewire.

- Optical hardware will be almost the same with minor changes in lens and working distance from camera to the objective. This is because processes machines they are different and most of them, they do not have available space to have other lighting devices or have the correct working distance to not have any lens aberrations for be very close to the objective.

- We described the basics of image theory, how important is to have the best image quality for the success of the information that we want to obtain. In our case, for the conditions that we can not control, we are getting images with not an even brightness distribution. This can make that an image has some parts more brighter than other, and in some cases, depending of the PCB surface we can have bright spots.

- To solve these problems, we can get support in hardware optics like filters, polarizers, special lights wavelengths or other equipment that can help us reducing or eliminating the bright spots. But we can use image pre−processing techniques also as contrast enhancements, de−blurring, shading correction, color extraction or any other that can help us to improve the quality of our images and eliminate irrelevant or unnecessary information that make hard to get the correct features or patterns that we will like to process.

- One of the most important steps in the image complete analysis process is the feature extraction. If we do success to get those features the complete image analysis will fail. That is the reason we need to get all the available tools to clean the image that we will use to process.

- We also check for the best acquisition technologies these days. In our developments, we specially use GigE Ethernet frame grabbers for GigE cameras. GigE technology is based in Ethernet communications. That means we can use normal Ethernet cables. We can put all those cameras in the internal network of a factory o company and control it from other computer systems. We can energize those cameras with PoE (Power Over Ethernet) if frame grabber can have the PoE option available.

- We worked in a sequence way to store all images from all distributed vision systems that for some reason they could not been read using pattern matching for OCR. Then an automated software that we developed can extract those parts of the images that can be digits. After this point, we need to classify those digit images and put them in automatic form in its digit folder, so we developed a software to manually identify what digit is and move it directly to its folder space, where we can classify more than $15,000$ digit images, $1,500$ for each digit.

- For pattern matching we are working with two ways. One is using a complete full ocr templates. This is the best scenario where images are almost clean with no apparent damage over the identification numbers. In this case correlation is fast for the complete alphabet that need to be checked. But normally, this does not happen in daily production line, where PCB identification numbers get damaged over the entire process line. To solve this, we developed a method to automatically a search box change its size to find pieces of digits and compare them with an extended alphabet. This is similar in how human can still detect a number or letter if it is not completely showing and our brain is capable to read it. With this method applied in the processes machines they improve detectability. It could be time consuming but its better than a process machine stops requiring manual input data.

- In the machine learning techniques we could exploit the advantages of having a huge set of images and compute the principal components analysis and take the most significant components and reduce a huge features samples matrix in a small one. This matrix is used in all the other classifiers.

- The implementation of principal components and Bayesian linear and quadratic discriminants demonstrates an improvement for the readings in optical character recognition from a previous detectability from pattern matching of 97%,at best, to close to 99 % for this paper technique. In this paper we can conclude that our data follows almost a linear nature. However, this assertion should be taken with caution, given that the data comes from one machine. Future work will increase the dataset to more machines and other classification techniques will be included, like artificial neural networks.

- The implementation of the GSC Algorithm and kNN Classifier with Euclidean Distance shows an improvement for the readings of damaged or incomplete characters using optical character recognition from a previous detectability strategy (Pattern Matching) from 95% to 96.76% ± 0.405. Future work will consider increasing the dataset samples and the use of other distance metrics, as well as other classification algorithms.

- For all the classifiers that we tested, QDA Discriminant has the best %recognition with more than 99.29%.
- Finally, we can say that Patten Matching can be as good as Machine Learning if we can have a good pre−processing cleaning and have a good template matching. Machine learning is the best solution for getting the correct answer for the digit that we are trying to get its class, but it can be time consuming and hard to implement in production processes machines. With all the embedded technologies improving everyday it will be easy to implement machine learning algorithms inside these devices and process OCR in damaged characters on PCB in real time.

5.2 Future Work

- Evaluation of more spaces. Kernel PCA, Non−Negative Matrix Factorization, Independent Component Analysis.
- Other classifiers like SVM for Multi-Class.
- Deep Learning.
- Increase image database to $30,000$ samples.
- More improvements in the image pre−processing.
- Embedded devices for neural networks, like Intel Neural Compute Stick 2, Google Coral Dev Board, LattePanda.

Appendix A

Appendix A

1. CSCI 2015: 2015 International Conference on Computational Science and Computational Intelligence —see Figure A.1.

Certificate of Presentation

This is to certify that

Carlos Fabian Nava Duenas

has presented the paper entitled

OCR for Unreadable Damaged Characters on PCBs using Principal Component

at The 2015 International Conference on Computational Science
&
Computational Intelligence (CSCI'15)

December 7-9, 2015 | Las Vegas, USA

Kaveh Arbtan on behalf of Steering Committee

Official Seal

CSCI'15
December 07-09, 2015
Monte Carlo Resort
Las Vegas Nevada, USA

Figure A.1: CSCI2015 Certificate.

2. CSCI 2015 Paper —see Figure A.2.

OCR for Unreadable Damaged Characters on PCBs Using Principal Component Analysis and Bayesian Discriminant Functions

Carlos F. Nava-Dueñas, *member IEEE*
Skyworks Solutions, Inc.; Engineering Institute
UABC, Mexicali, Mexico
nava.carlos@uabc.edu.mx

Felix F. Gonzalez-Navarro, *member SMIA*
Engineering Institute
UABC, Mexicali, Mexico
fernando.gonzalez@uabc.edu.mx

***Abstract*— In the last few decades, new computer vision technologies and image processing techniques have been very important in the improvement and automation of manual processes in many technical areas, e.g., in the semiconductor industry. In this paper, we propose to change the actual pattern matching methods implemented to have optical character recognition by the use of principal component analysis method to extract the principal characteristics and features of damaged or unreadable numerical digit characters from images on printed board circuits (PCBs) and compute linear and quadratic Bayesian discriminant functions to classify and find the correct numerical character that corresponds to those features. In the**

implemented vision system reads identification characters on printed circuits boards (PCBs) for lot integrity and machine control. This commonly used technique is not robust enough because many of the images of PCBs shown some damage on the characters due dirt or the results of bad previous processes [1], [6]. Our actual OCR detectability is around 97% at best. Our system starts with a monochrome VGA image acquisition of the upper left section of a PCB, using a NI-1752 smart camera, with a full resolution of 640x480 pixels with a maximum data transfer of 60 fps using a GigE port. The selected resolution and data transfer speed parameters meet the

Figure A.2: Paper OCR PCA Bayesian Discriminant Functions.

3. ICAI 2016: The 18th International Conference on Artificial Intelligence —see Figure A.3.

Certificate of Presentation

This is to certify that

Carlos F. Nava-Duenas

has presented the paper entitled

OCR for Unreadable Damaged Characters on PCBs Using GSC Algorithm and kNN Classifier

at *The 2016 International Conference on Artificial Intelligence (ICAI2016)*

held simultaneously with 19 other joint conferences as part of

The 2016 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'16) at the Monte Carlo Resort, Las Vegas Nevada, USA
July 25-28, 2016

Kaveh Arbtan on behalf of Steering Committee

WORLDCOMP'16
July 25-28, 2016
Monte Carlo Resort
Las Vegas Nevada, USA

Figure A.3: ICAI2016 Certificate.

4. ICAI 2016 Paper —see Figure A.4.

OCR for Unreadable Damaged Characters on PCBs Using GSC Algorithm and kNN Classifier

Carlos F. Nava-Dueñas, *member IEEE*
Skyworks Solutions, Inc.; Engineering Institute
UABC, Mexicali, Mexico
nava.carlos@uabc.edu.mx

Felix F. Gonzalez-Navarro, *member SMIA*
Engineering Institute
UABC, Mexicali, Mexico
fernando.gonzalez@uabc.edu.mx

Abstract— **In this paper, we propose to change the actual implemented pattern matching method to have optical character recognition by implementing the Gradient, Structural, Concavity (GSC) algorithm to extract the features of damaged, unreadable or incomplete numerical digit characters from images on printed board circuits (PCBs). Grayscale color images are acquired from a charge-coupled device (CCD) camera, assembling a dataset of 500 matrix images samples for the character digits from 0 to 9. The GSC feature extraction method is applied to get the characteristics that will be used in the character recognition step. Experimental results show that applying GSC algorithm to extract the features and using k-Nearest Neighbor (kNN) Classifier with the Euclidian Distance can improve optical character recognition (OCR) detectability of damaged characters from actual 95% to more than 97% in early tests.**

with a maximum character resolution to cover the entire PCB characters positions, as shown in Fig. 1.

Fig. 1. PCB with no damaged characters.

Due to some problems with previous processes in the production line, some PCBs will have some residual dirt over the characters, making some characters unreadable for the pattern matching technique, as shown in the following Fig. 2.

Figure A.4: Paper OCR Using GSC Algorithm and KNN Classifier.

5. USPATENT US 10,255,513 B2 —see Figure A.5.

US010255513B2

(12) **United States Patent**
Nava

(10) **Patent No.: US 10,255,513 B2**
(45) **Date of Patent: Apr. 9, 2019**

(54) **SYSTEMS AND METHODS FOR RECOGNITION OF UNREADABLE CHARACTERS ON PRINTED CIRCUIT BOARDS**

(71) Applicant: **SKYWORKS SOLUTIONS, INC.**, Woburn, MA (US)

(72) Inventor: **Carlos Fabian Nava**, Mexicali (MX)

(73) Assignee: **Skyworks Solutions, Inc.**, Woburn, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/612,730**

(22) Filed: **Jun. 2, 2017**

(65) **Prior Publication Data**

US 2017/0372158 A1 Dec. 28, 2017

Related U.S. Application Data

(60) Provisional application No. 62/344,624, filed on Jun. 2, 2016.

(51) **Int. Cl.**
G06K 9/32 (2006.01)
G06K 9/46 (2006.01)
G06K 9/18 (2006.01)
G06K 9/62 (2006.01)

(52) **U.S. Cl.**
CPC ***G06K 9/325*** (2013.01); ***G06K 9/18*** (2013.01); ***G06K 9/4638*** (2013.01); ***G06K 9/6276*** (2013.01); *G06K 2209/01* (2013.01)

(58) **Field of Classification Search**
CPC G06K 9/325; G06K 9/6276; G06K 9/18; G06K 9/4638; G06K 2209/01
USPC 382/147
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2010/0310172 A1 * 12/2010 Natarajan G06K 9/00865 382/187
2015/0130925 A1 * 5/2015 Park H05K 13/08 348/87

* cited by examiner

Primary Examiner — Amandeep Saini
(74) *Attorney, Agent, or Firm* — Chang & Hale LLP

(57) **ABSTRACT**

Systems and methods for recognition of unreadable characters on printed circuit boards. In some embodiments, a method for recognizing characters can be utilized for recognition of damaged characters on a printed circuit board. The method can include obtaining a digital image for each of a plurality of characters on the printed circuit board. The method can further include dividing each digital image into an array of regions. The method can further include generating a data structure from the arrays of the digital images. The data structure can include gradient features based on stroke shapes on small distances, structural features based on stroke trajectories on extended distances, and concavity features based on stroke relationships.

16 Claims, 7 Drawing Sheets

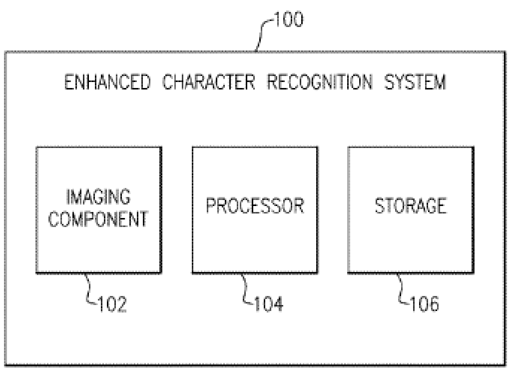


Figure A.5: Systems and Method for Recognition of Unreadable Characters on Printed Circuit Boards.

6. IMPI Technical State Report Patent 1 —see Figure A.6.

IMPI
INSTITUTO MEXICANO
DE LA PROPIEDAD
INDUSTRIAL

IT/S/2018/000034

DIRECCIÓN DIVISIONAL DE PROMOCIÓN Y SERVICIOS DE INFORMACIÓN TECNOLÓGICA
SUBDIRECCIÓN DIVISIONAL DE SERVICIOS DE INFORMACIÓN TECNOLÓGICA
COORDINACIÓN DEPARTAMENTAL DEL CENTRO DE INFORMACIÓN TECNOLÓGICA

Folio de Salida: IT/S/2018/000034
Folio de Solicitud: IT/B/2017/001462
Folio del Expediente: IT/E/2017/001466

Ciudad de México, 17 de enero de 2018

REPORTE DE BÚSQUEDA DEL ESTADO DE LA TÉCNICA

SKYWORKS SOLUTIONS DE MEXICO S DE RL DE C. V.
Presente

En relación a su solicitud de búsqueda de información tecnológica con fecha de presentación 20 de diciembre de 2017, correspondiente a:

METODO AUTOMATIZADO PARA EL RECONOCIMIENTO OPTICO DE CARACTERES CON DAÑO SUPERFICIAL EN CIRCUITOS IMPRESOS (PBC's) A TRAVES DE LA APLICACION DE INTERFACES INTELIGENTES DENTRO DEL PROCESO DE MANUFACTURA DE MICROCIRCUITOS

A continuación se presentan los siguientes resultados:

BASE DE DATOS	ESTRATEGIA DE BÚSQUEDA	DOCUMENTOS ENCONTRADOS
INTERNET	inspeccion optica circuitos electricos and bayesianos	7
DERWENT	CTB=((detec* AND (damage or defect or failure)) or inspection) AND CTB=(PRINTED ADJ CIRCUITS) AND CTB=(optic*) AND CTB=(text or character) AND IC=(h05k);	110
DERWENT	CTB=(Automated Optical Inspection) AND CTB=(text or character);	52
IMPI (SIGA)	CLAS:h05k* AND (inspeccion or detec*) and ((optic* or vision*) or (error or falla)) (circuito or impreso) AND (inspeccion or detec*) and ((optic* or vision*) and ((error or falla)))	2
	TOTAL DE DOCUMENTOS ENCONTRADOS	171

Figure A.6: Metodo Automatizado para el Reconocimiento Optico de Caracteres con Daño Superficial en Circuitos Impresos PCB a traves de la Aplicacion de Interfaces Inteligentes dentro del Proceso de Manufactura de Microcircuitos.

7. IMPI Technical State Report Patent 2 —see Figure A.7.

IMPI
INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL

IT/S/2018/000025

DIRECCIÓN DIVISIONAL DE PROMOCIÓN Y SERVICIOS DE INFORMACIÓN TECNOLÓGICA
SUBDIRECCIÓN DIVISIONAL DE SERVICIOS DE INFORMACIÓN TECNOLÓGICA
COORDINACIÓN DEPARTAMENTAL DEL CENTRO DE INFORMACIÓN TECNOLÓGICA

Folio de Salida: IT/S/2018/000025
Folio de Solicitud: IT/B/2017/001463
Folio del Expediente: IT/E/2017/001467

Ciudad de México, 12 de enero de 2018

REPORTE DE BÚSQUEDA DEL ESTADO DE LA TÉCNICA

SKYWORKS SOLUTIONS DE MEXICO S DE RL DE C. V.
Presente

En relación a su solicitud de búsqueda de información tecnológica con fecha de presentación 20 de diciembre de 2017, correspondiente a:

SISTEMA AUTOMATIZADO PARA EL RECONOCIMIENTO OPTICO DE CARACTERES CON DAÑO SUPERFICIAL EN CIRCUITOS IMPRESOS (PCB¿s) PARA DISTINTOS PROCESOS DE MANUFACTURA DE MICROCIRCUITOS CON LA APLICACIÓN DE INTERFACES INTELIGENTES.

A continuación se presentan los siguientes resultados:

BASE DE DATOS	ESTRATEGIA DE BÚSQUEDA	DOCUMENTOS ENCONTRADOS
IMPI (SIGA)	G06K09/03	9
DERWENT	character AND (error OR diffus*) AND G06K09/03	704
	TOTAL DE DOCUMENTOS ENCONTRADOS	713

Figure A.7: Sistema Automatizado para el Reconocimiento Optico de Caracteres con DañO Superficial en Circuitos Impresis OCB para Distintos Procesis de Manufactura de Microcircuitos con la Aplicacion de Interfaces Inteligentes.

8. The AIA Vision Show Conference 2016, 2018 —see Figure A.8.

- **Neural Vision - Machine Vision's Move To Artificial Intelligence.**
- **A Process For Transforming Fully Manual Inspection To Fully Automated Inspection.**
- **Giving Machine Vision Data Its Due As Part Of A Manufacturing 4.0 Strategy.**
- **Grooming The Next Generation Automation Capable Workforce For Industry 4.0.**
- **Machine Learning For Vision Based Inspection And Verification.**
- **Deep Learning, Classification And Their Applications In Machine Vision.**
- **Deep Learning For Industrial Vision Applications On Heterogeneous Embedded Vision Systems.**
- **Demystifying Deep Learning: Semantic Segmentation and Embedded Deployment.**

Figure A.8: The AIA Vision Show 2016, 2018.

9. 2019 Innovation Disclosure Records for Patents —see Figure A.9.

Innovation Disclosures\ By Docket No

< Previous Next >>

Docket No.	Status	Fiscal Qtr.	Submit Date	Title of Disclosures as Submitted
2011-IDR-0131	ASSIGNED	1Q12	12/09/2011	X-Out Defects Detection Storing Algorithm.
2011-IDR-0133	ASSIGNED	1Q12	12/12/2011	X-Out Virtual Server Protection System for Production Data Manipulation.
2012-IDR-0091	FILED - UTILITY	4Q12	07/06/2012	Automatic X-Out Mark Project: Laser Mark - 2 Parallel Cameras Vision.
2012-IDR-0094	FILED - UTILITY	4Q12	07/11/2012	Automatic X-Out Mark Project: X-Out Inker Less Mapping-Laser Module.
2019-IDR-0127	NO RANK	3Q19	06/13/2019	Automatic Vision Method with a Mixture of Global and Local Threshold Processing in OCR for Damaged Characters in PCBs
2019-IDR-0128	NO RANK	3Q19	06/13/2019	Automatic Vision Method for OCR of Unreadable Damaged Characters on PCBs Using PCA and Bayesian Discriminant Functions

< Previous Next >>

Figure A.9: 2019 Innovation Disclosure Records for Patents.

10. ICAI2016 Presentation —see Figure A.10.

July 27

03:00 - 03:40pm: POSTER SESSION C
(INCLUDES POSTER + SHORT & POSITION PAPERS)
July 27, 2016 (Wednesday)
(LOCATION: Hallways of Ballrooms 1-5)

o. Generic Object Recognition Using Bag-of-Features and Image Concept-Base
Hirokazu Watabe, Misako Imono, Eriko Yoshimura, Seiji Tsuchiya
Department of Intelligent Information Engineering and Science, Doshisha University, Kyoto, Japan
o. Judging Emotion from EEGs Using SVM and Principal Component Analysis
Seiji Tsuchiya, Mayo Morimoto, Misako Imono, Hirokazu Watabe
Department of Intelligent Information Engineering and Science, Doshisha University, Kyoto, Japan
o. Deep Recurrent Neural Network and Psychoacoustic Modeling for Speech Enhancement
Yu Yong Jeon, Gyu Seok Park, Jang-Woo Kwon, Sang Min Lee
Department of EE, Inha University, Incheon, South Korea; Division of Computer Engneering and Information, Inha University, Incheon, South Korea; Institute for Information and Electronics Research, Inha University, South Korea
o. Dual Sub-swarm Interaction QPSO Algorithm Based on Different Correlation Coefficients
Tao Wu, Xi Chen, Xi Wu
Department of CS, Chengdu University of Information Technology, Chengdu, P. R. China; School of Computer Science and Technology, Southwest University for Nationalities, P. R. China
o. Legalized Path Planning of Mobile Real Time Location System in Emergency Department
Shachi Singh, Andrew R. Winton
ECE, University of Manitoba, Winnipeg, MB, Canada; Department of Computer Science, University of Manitoba, Canada
o. ...

SESSION 10-ICAI: INTELLIGENT SYSTEMS, NOVEL AI CONCEPTS, PATTERN RECOGNITION AND APPLICATIONS
Chairs: Prof. Lin-Ching Chang
Department of EECS, The Catholic University of America, USA; and
Prof. Eman El-Sheikh,
Director, Center for Cybersecurity & Professor of CS, University of West Florida, USA
July 27, 2016 (Wednesday); 03:40pm - 04:40pm
(LOCATION: Ballroom 5)

03:40 - 04:00pm: An Intelligent Web-Based System for Measuring Students' Attention Levels
Omer Useche, Eman El-Sheikh
Department of CS, University of West Florida, Pensacola, Florida, USA

04:00 - 04:20pm: AI Inferences Utilizing Occam Abduction
James A. Crowder
Raytheon Intelligence, Information, and Services, Aurora, Colorado, USA

04:20 - 04:40pm: OCR for Unreadable Damaged Characters on PCBs Using GSC Algorithm and K-NN Classifier
Carlos F. Nava-Duenas, Felix F. Gonzalez-Navarro
Skyworks Solutions, Inc.; Engineering Institute, UABC, Mexicali, Mexico

96

Figure A.10: ICAI2016 Presentation.

11. CSCI2015 Presentation —see Figure A.11.

ISBN # 1-60132-425-1; American Council on Science & Education

SYMPOSIUM ON SIGNAL & IMAGE PROCESSING, COMPUTER VISION & PATTERN RECOGNITION (CSCI-ISPC)

CSCI5145 - ISPC

OCR for Unreadable Damaged Characters on PCBs Using Principal Component Analysis and Bayesian Discriminant Functions

Carlos Fabian Nava-Dueñas, Felix Fernando Gonzalez-Navarro
Skyworks Solutions, Inc., Mexico; UABC Engineering Institute, Mexico

Abstract: In the last few decades, new computer vision technologies and image processing techniques have been very important in the improvement and automation of manual processes in many technical areas, e.g., in the semiconductor industry. In this paper, we propose to change the actual pattern matching methods implemented to have optical character recognition by the use of principal component analysis method to extract the principal characteristics and features of damaged or unreadable numerical digit characters from images on printed board circuits (PCBs) and compute linear and quadratic Bayesian discriminant functions to classify and find the correct numerical character that corresponds to those features. In the first step of this work, grayscale color images are acquired from a charge-coupled device (CCD) camera, then image segmentation is manually computed to create a dataset of 500 matrix images for the character digits from 0 to 9. Then, a feature extraction method is applied to get the principal components that will be used in the character recognition state. Finally, our results show that applying Bayesian linear and quadratic discriminants to the principal component features can improve optical character recognition (OCR) detectability of damaged characters from actual 95–97% to 99.88% in early tests. This suggests to us that the problem probably follows a linear model where linear hyperplanes separate decision regions with satisfactory (almost no) errors.

CSCI5142 - ISPC

Field Phenomics: A web based image analysis platform using open source tools

Henry Alonso Ruiz, Felix Fernando Navarro, Michael Selvaraj, Milton Valencia, Alfredo Delgado
Baja California University (UABC), Engineering Institute, Mexico;
International Center for Tropical Agriculture (CIAT), Colombia;
Texas A&M University, Texas, USA

Abstract: The main goal of any field phenomics platform is to enable plant breeders to detect and evaluate dynamic traits which so far are difficult or inefficient to measure while also requiring destructive sampling. Plant phenotyping is frequently slower and more expensive than genotyping due to the time interval data capture for traits under different environments. Advanced remote sensing technologies such as aerial phenotyping may help expedite the development and selection of climate smart plant varieties. This is an ongoing research project. The main objective of this research is to build an open source web based image analysis platform using OpenCV (open source computer vision) to analyze field data in the form of images. Computation of vegetative indices to predict plant performance under real field conditions would be a provided outcome.

CSCI5132-ISPC

Image Analysis for Automatic Characterization of Nanomaterials

Jesús Caro-Gutiérrez, Félix F. González-Navarro, Benjamín Valdez-Salas, Mario A. Curiel-Álvarez
Engineering Institute, Autonomous University of Baja California, Mexicali, Mexico

Abstract: Nowadays, the development of new technology depends strongly on nanomaterials study, which is carried out usually by microscopy techniques allowing the images acquisition of materials for their posterior characterization. When this feature extraction is performed through a human observer it can become slow, laborious

~ 33 ~

Figure A.11: CSCI2015 Presentation.

References

Ballerini, L. (2012). Non-melanoma skin lesion classification using colour image data in hierarchical knnclassifier. In *Proceedings of 2012 IEEE International Symposium on Biomedical Imaging*. 107

Boden, M.A. (2006). *Mind As Machine: A History of Cognitive Science*. Oxford - Clarendon Press, Oxford, England. 14

Casey, R. (1996). A survey of methods and stategies in character segmentation. In *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 18, 690–706. 96, 106

Cover, T. (1967). Nearest neighbor pattern classification. In *IEEE Transactions on Information Theory*, vol. 13, 21–27. 76

Dawson, B. (2014). Smart cameras yesterday, today and tomorrow. 16

Desrochers, D. (2001). Ocr readability study and algorithms for testing partially damaged characters. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001*. 96, 106

Duda, R. (2001). *Pattern Classification*. John Wiley and Sons, New York City, New York, USA. 99, 106

Favata, J. (1996). A multiple feature/resolution approach to handprinted digit and character recognition. In *Proceedings of the International Journal of Imaging Systems and Technology*, vol. 7, 304–311. 106

Gonzalez, R. (2008). *Digital Image Processing*. Pearson - Prentice Hall, New York City, New York, USA. 24, 25, 30, 32

Gour, R. (2019). Neural network algorithms, learn how to train ann. 82

Hu, J. (2007). Predicting protein subcelluar localizations using weighted euclidian distance. In *Proceedings of 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, 1370–1373. 107

Ibraheem, I. (2011). Linear and quadratic classifier to detection of skin lesions epicutaneus. In *2011 5th International Conference on Bioinformatics and Biomedical Engineering*. 99

Jadhav, S. (2016). Comparative study of k-nn, naive bayes and decision tree classification techniques. In *International Journal of Science and Research (IJSR)*, vol. 5, 1842–1845. 76, 78

Jolliffe, I. (2002). *Principal Component Analysis, Springer Series in Statistics*. Springer Publishing, New York City, New York, USA. 96

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence*, 1137–1143. 104

Lecun, Y. (1989). Backpropagation applied to handwritten zip code recognition. In *Neural Computation*, 541–551. 18

Li, F.F. (2009). Imagenet database. 18

Li, F.F. (2019). Introduction to computer vision. 18

Misiak, M. (2008). Character recognition using radon transformation and principal component analysis in postal applications. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, 495–500. 96, 97

Mori, S. (1992). Historical review of ocr research and development. In *Proceedings of the IEEE*, vol. 80, 1029–1058. 96, 106

NEDEVSCHI, S. (2012). Pca type algorithm applied in face recognition. In *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*. 96

OGNJANOVSKI, G. (2014). Everything you need to know about neural networks and backpropagation, machine learning easy and fun. 80

PAPERT, S. (1966). The summer vision project. 15

PING, D. (2013). A review on image feature extraction and representation techniques. In *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, 385–396. 43

PRINCE, S. (2012). *Computer Vision: models, learning and inference*. Cambridge University Press, Cambridge, United Kingdom. 34

QING-SONG, X. (2001). Monte carlo cross validation. In *Chemometrics and Intelligent Laboratory Systems*, vol. 56, 1–11. 116

ROBERTS, L.G. (1963). *Machine Perception of Three-Dimensional Solids*. Ph.D. thesis, Massachusetts Institute of Technology. 15

RODRIGUEZ, C. (2002). An incremental and hierarchical knnclassifier for handwritten characters. In *Proceedings of 2002 IEEE International Conference on Pattern Recognition*, vol. 3, 98–101. 107

SHI, J. (2000). Normalized cuts and image segmentation. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 888–995. 18

SYSTEMS, E. (2019). Quick history of machine vision. 17

SZELISKI, R. (2010). *Computer Vision: Algorithms and Applications*. Springer, Salmon Tower Building, New York City. 11, 16, 17, 18

UMBAUGH, S. (2010). *Digital Image Processing and Analisis*. CRC Press, Boca Raton, Florida, USA. 26, 27, 30, 35, 41

YANG, Y. (2005). Automatic target classification and experiments on the mstar sar images. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Network*, 2–7. 99