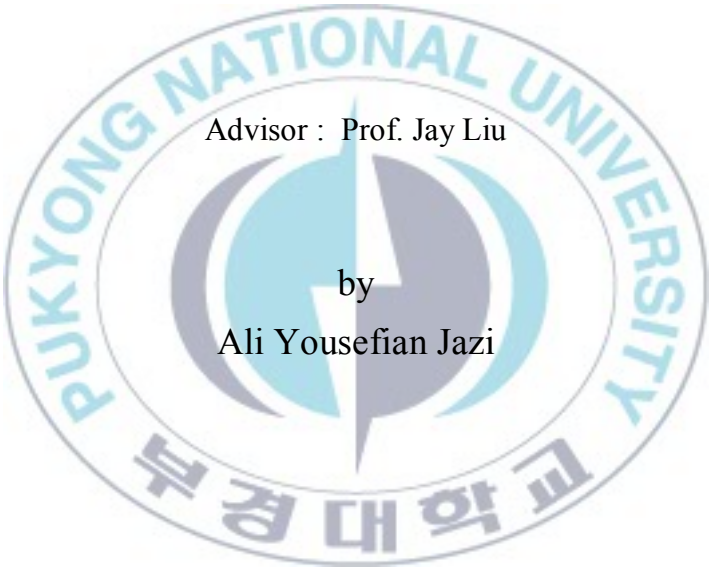
Thesis for the Degree of Master of Engineering

Decision Support in Machine Vision System for Monitoring of TFT-LCD Glass Substrates Manufacturing with Data Mining Techniques

by
Ali Yousefian Jazi
Department of Chemical Engineering
The Graduate School
Pukyong National University

August 2013

Decision Support in Machine Vision System for Monitoring of TFT-LCD Glass Substrates Manufacturing with Data Mining Techniques



Advisor : Prof. Jay Liu

by

Ali Yousefian Jazi

A thesis submitted in partial fulfillment of the requirements
for the degree of

Master of Engineering

in the Department of Chemical Engineering, The Graduate School,
Pukyong National University

August 2013

Decision Support in Machine Vision System for Monitoring of TFT-LCD Glass Substrates Manufacturing with Data Mining Techniques

A dissertation
by
Ali Yousefian Jazi

Approved by:

(Chairman) Prof. Geongbum Yi

(Member) Prof. Chang Jun Lee

(Member) Prof. Jay Liu

August 23, 2013

Contents


PUKYONG NATIONAL UNIVERSITY
부경대학교

LIST OF TABLES

LIST OF FIGURES

Decision Support in Machine Vision System for Monitoring of TFT-LCD Glass Substrates Manufacturing with Data Mining Techniques

Ali Yousefian Jazi

Department of Chemical Engineering, The Graduate School,
Pukyong National University

Abstract

The visual appearance of manufactured products is often one of the major quality attributes for certain types of products, which are used mainly for display purposes or used as the exterior part of other products. TFT-LCD (Thin Film Transistor – Liquid Crystal Display) glass substrates can serve as a representative case. Inline defect inspection plays an important role in production yields quality improvement in TFT-LCD manufacturing. The main objective of this work is presenting one decision support system for monitoring of TFT-LCD glass substrates manufacturing by using data mining techniques. This study employs optical system design to make an inline surface defect inspection system in cold process section which is carried out according to CRISP-DM standard for data mining process methodology. This study also develops an image processing methodology, wavelet co-occurrence signature, to extract the features from images and different statistical, heuristical and machine learning algorithms such as principle component analysis, simulated annealing, support vector machine (SVM), multilayer perceptron (MLP) and ensemble techniques are used as feature reduction and classification as well. Finally, the results of different feature selection methods and classifiers are compared and the best is proposed as a suitable methodology for an automatic inspection system in cold process of TFT-LCD glass substrates manufacturing.


PUKYONG NATIONAL UNIVERSITY
부경대학교

Chapter 1

1. DATA MINING PROCESS

CRISP-DM Methodology

The life cycle of a data mining project consists of six phases, shown in Fig. 1.1 [1]. The sequence of the phases is not rigid. Moving back and forth between different phases is always required. The outcome of each phase determines which phase, or particular task of a phase, has to be performed next. The arrows indicate the most important and frequent dependencies between phases. The outer circle in Fig. 1.1 symbolizes the cyclical nature of data mining itself. Data mining does not end once a solution is deployed. The lessons learned during the process and from the deployed solution can trigger new, often more-focused business questions. Subsequent data mining processes will benefit from the experiences of previous ones. In the following, we will describe each phase:

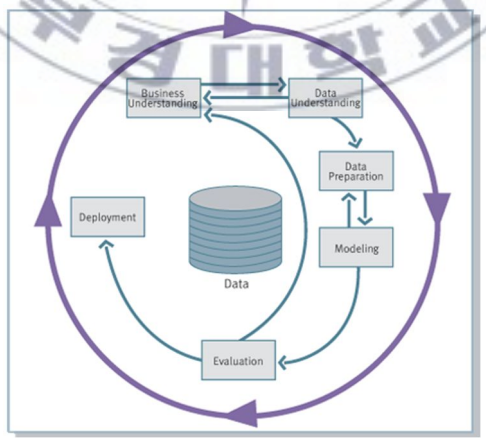


Fig. 1.1. Phases of the CRISP-DM methodology

1.1 Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data mining problem definition, and a preliminary plan designed to achieve the objectives.

1.1.1 Determine Business Objectives

The first objective of the data analyst is to understand thoroughly, from a business perspective, what the client really wants to accomplish. Often the client has many competing objectives and constraints that must be properly balanced. The analyst's goal is to uncover important factors, at the beginning, that can influence the outcome of the project. A possible consequence of neglecting this step is to expend a great deal of effort producing the right answers to the wrong questions.

1.1.2 Assess Situation

This task involves more detailed fact-finding about all of the resources, constraints, assumptions and other factors that should be considered in determining the data analysis goal and project plan. In the previous task, the objective is to get to the crux of the situation quickly. Here, the analyst wants to flesh out the details.

1.1.3 Determine Data Mining Goals

A business goal states objectives in business terms. A data mining goal states project objectives in technical terms. For example, the business goal might be "increase the quality of products in TFT-LCD manufacturing." A

data mining goal might be “detect the defects and their location on the surface of glass substrates in LCD panel.”

1.1.4 Produce Project Plan

Describe the intended plan for achieving the data mining goals and thereby achieving the business goals. The plan should specify the anticipated set of steps to be performed during the rest of the project including an initial selection of tools and techniques.

1.2 Data Understanding

The data understanding phase starts with an initial data collection. It proceeds with activities

- to get familiar with the data,
- to identify data quality problems,
- to discover first insights into the data, or
- to detect interesting subsets to form hypotheses for hidden information.

1.2.1 Collect Initial Data

Acquire within the project the data (or access to the data) listed in the project resources. This initial collection includes data loading if necessary for data understanding. For example, if applying a specific tool for data understanding, it makes perfect sense to load the data into this tool. This effort may lead to initial data preparation steps.

1.2.2 Describe Data

Examine the “gross” or “surface” properties of the acquired data and report on the results.

1.2.3 Explore Data

This task tackles the data mining questions that can be addressed using querying, visualization and reporting. These analyses may address the data mining goals directly. They may also contribute to or refine the data description and quality reports and feed into the transformation and other data preparation needed for further analysis.

1.2.4 Verify Data Quality

Examine the quality of the data, addressing questions such as: is the data complete? Is it correct? Are these missing values? If so how are they represented, where do they occur and how common are they?

1.3 Data Preparation

The data preparation phase covers all activities to construct the final dataset (data that will be fed into the modeling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection as well as transformation and cleaning of data for modeling tools.

1.3.1 Select Data

Decide on the data to be used for analysis. Criteria include relevance to the data mining goals, quality and technical constraints such as limits on data

volume or data types. Note that data selection covers selection of attributes (columns) as well as selection of records (rows) in a table.

1.3.2 Clean Data

Raise the data quality to the level required by the selected analysis techniques. Problems that can occur with “dirty data” include missing data, empty values, non-existent values, and incomplete data. Data cleaning may involve selection of clean subsets of the data, the insertion of suitable defaults or more ambitious techniques such as replacing the dirty data with derived values, or building separate models for those entities that possess dirty data. However, these approaches can introduce additional problems. Specifically, filtering the problematic data can introduce sample bias into the data and using data overlays could introduce missing values.

1.3.3 Construct Data

This task includes constructive data preparation operations such as the production of derived attributes, entire new records, or transformed values for existing attributes.

1.3.4 Integrate Data

Two methods used for integrating data are merging data and generating aggregate values. In these methods information is combined from multiple tables or other information sources to create new records or values. For example, merging tables refers to joining together two or more tables that have different information about the same objects; generating aggregate values refers to computing new values computed by summarizing information from multiple records, tables or other information sources.

1.3.5 Format Data

Formatting transformations refer to primarily syntactic modifications made to the data that do not change its meaning, but might be required by the modeling tool.

1.4 Modeling

In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, several techniques can be applied to the same data mining problem type. Some techniques require a specific form of data. Therefore, stepping back to the data preparation phase is often needed.

1.4.1 Select Modeling Technique

As the first step in modeling, select the actual modeling technique to be used. If a tool was selected in business understanding (Phase 1), this task refers to selecting the specific modeling technique, e.g., building decision trees or generating a neural network.

1.4.2 Generate Test Design

Prior to building a model, a procedure needs to be defined to test the model's quality and validity. For example, in supervised data mining tasks such as classification, it is common to use error rates as quality measures for data mining models. Therefore, if the test design specifies that the dataset should be separated into training and test sets, the model is built on the training set and its quality estimated on the test set.

1.4.3 Build Model

The purpose of building models is to use the predictions to make more informed business decisions. The most important goal when building a model is stability, which means that the model should make predictions that will hold true when it's applied to yet unseen data. Regardless of the data mining technique being used, the basic steps used for building predictive models are the same.

Models are created using data from the past in order for the model to make predictions about the future. This process is called training the model. In this step, the data mining algorithms find patterns that are of predictive value. Next, the model is refined using the test set. The model needs to be refined to prevent it from memorizing the training set. This step ensures that the model is more general (i.e. stable) and will perform well on unseen data. Then, the performance of the model is estimated using the evaluation set. The evaluation set is entirely separate and distinct from the training and test sets. The evaluation set (or hold out set) is used to assess the expected accuracy of the model when it is applied to data outside the model set. Finally, the model is applied to the score set. The score set is not pre-classified and is not part of the model set used to create the data model. The outcomes for the score set are not known in advance. The final model is applied to the score set to make predictions. The predictive sores will, presumably, be used to make more informed business decisions.

Overfitting. A problem that can occur is that model created can overfit the data. Overfitting means that the specification of a model is in large part an artifact of the idiosyncrasies of the data set used to build it (i.e., the training set). Overfitting occurs when a model essentially memorizes the data on which was built. The model should learn the patterns in order to recognize them in future unseen datasets, but the model should not memorize the

patterns. The problem with the model memorizing the training set, is that when the model scores an unknown record, it will use the results from the model set if there is a match, and if not, it will produce a random guess. In that case the model is entirely unstable, i.e. it will do no better than random for the score set.

1.4.4 Assess Model

The model should now be assessed to ensure that it meets the data mining success criteria and passes the desired test criteria. This step is a purely technical assessment based on the outcome of the modeling tasks. Two tools commonly used to assess the performance of different models are the lift chart and the confusion matrix.

A lift chart, sometimes called a cumulative gains chart, or a banana chart, is a measure of model performance. It shows how responses, (i.e., to a direct mail solicitation, or a surgical treatment for instance) are changed by applying the model. This change ratio, which is hopefully, the increase in response rate, is called the "lift". A lift chart indicates which subset of the dataset contains the greatest possible proportion of positive responses. The higher the lift curve is from the baseline, the better the performance of the model since the baseline represents the null model, which is no model at all.

A confusion matrix, sometimes called a classification matrix, is used to assess the prediction accuracy of a model. It measures whether a model is confused or not; that is, whether the model is making mistakes in its predictions. At the conclusion of the model building and assessment processes, the most appropriate model will be the model that meets the business objectives.

1.5 Evaluation

1.5.1 Evaluate Results

Previous evaluation steps dealt with factors such as the accuracy and generality of the model. This step assesses the degree to which the model meets the business objectives and seeks to determine if there is some business reason why this chosen model is deficient. Another option of evaluation is to test the model(s) on test applications in the real application if time and budget permits.

1.5.2 Review Process

At this point the resultant model appears to be satisfactory and appears to satisfy business needs. It is now appropriate to make a more thorough review of the data mining project in order to determine if there is any important factor or task that has somehow been overlooked.

1.5.3 Determine Next Steps

According to the assessment results and the process review, the analyst decides how to proceed at this stage. The analyst needs to decide whether

- to finish the project and move on to deployment (Phase 6) or
- to initiate further iterations or
- to set up new data mining projects.

1.6 Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the client can

use. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. In many cases it will be the client, not the data analyst, who will carry out the deployment steps. However, even if the analyst will not carry out the deployment effort it is important for the client to understand up front what actions will need to be carried out to make use of the models created.

1.6.1 Plan Deployment

To deploy the data mining result(s) into the business, this task takes the evaluation results and develops a strategy for deployment. If a general procedure was identified to create the relevant model(s), this procedure is documented here for later deployment.

1.6.2 Plan Monitoring and Maintenance

Monitoring and maintenance are important issues if the data mining result becomes part of the day-to-day business and its environment. A careful preparation of a maintenance strategy helps to avoid unnecessarily long periods of incorrect usage of data mining results. To monitor the deployment of the data mining result(s), the project needs a detailed plan on the monitoring process. This plan takes into account the specific type of deployment.

1.6.3 Produce Final Report

At the end of the project, the project leader and the team write up a final report. Depending on the deployment plan, this report may be only a summary of the project and its experiences (if they have not already been

documented as an ongoing activity) or it may be a final and comprehensive presentation of the data mining result(s).

1.6.4 Review Project

Assess what went right and what went wrong, what was done well and what needs to be improved.

Chapter 2

2. MANUFACTURING OVERVIEW

2.1 Sheet Glass Manufacturing

There are two different commercial processes (also called "hot" processes) for manufacturing a continuous sheet of glass for TFT-LCD glass substrates: (1) the floating process [2] and (2) the fusion process [3]. In the floating process, molten glass with a temperature around *1,100°C* is continuously delivered onto a bath of molten tin. Since the density of molten glass is lower than that of molten tin, the glass spreads on the flat layer of tin forming a continuous sheet of flat glass with uniform thickness. This thickness can be controlled by manipulating the rate at which the sheet of glass is pulled out of the tin bath. The continuous sheet of flat glass is then pulled into a long kiln (called *lehr*) for annealing and further cooling to room temperature (Fig. 2.1). In the fusion process, molten glass is delivered into a trough, called an "isopipe", where the molten glass overfills the isopipe until the glass flows evenly over both edges of the isopipe. It then fuses at the bottom of the isopipe, at which it flows downward with gravity and is further drawn down to form a continuous sheet of flat glass (Fig. 2.2).

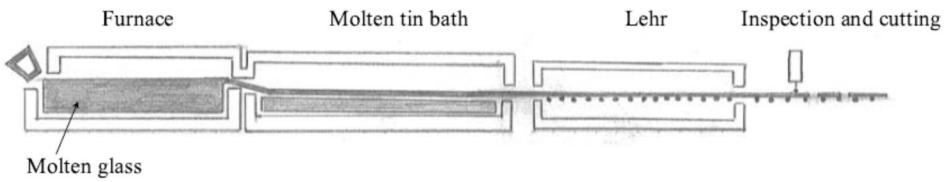


Fig. 2.1. Floating process [4]

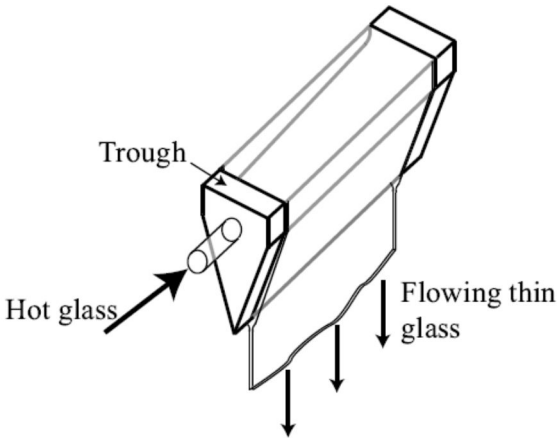


Fig. 2.2. Fusion process [4]

The continuous sheet of flat glass from a hot process is then rolled out of a hot process to be cut and inspected in a cold process. The cold process consists of cutting, chamfering, and surface polishing steps, with cleaning and inspection steps after each of the other steps. A schematic diagram of this cold process, including packaging, is shown in Fig. 2.3 The final product, a TFT-LCD glass substrate, has thickness of *0.7mm* (or *0.5mm*) and varying sizes according to a so-called TFT-LCD generation. For example, the size of a glass substrate of *generation 7* is *1870×2200* or *1950×2250* (in *mm*).

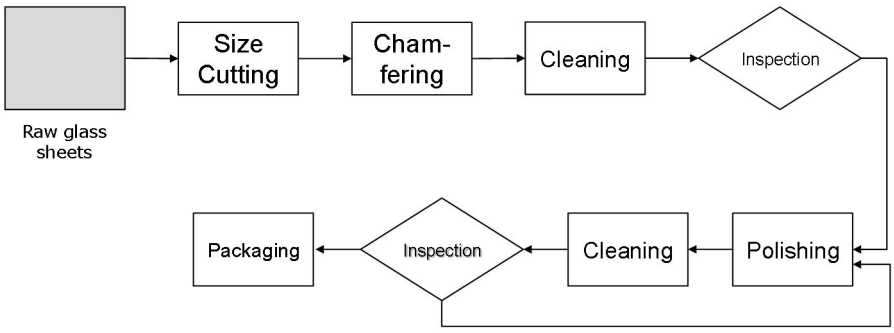


Fig. 2.3. Flow chart of cold process

2.2 Necessity of Using Automatic Optical Inspection (AOI) system

The visual appearance of manufactured products is an important quality attribute. It is an essential determining factor for product quality, especially in the case that those products are used as the exterior part of other manufactured products. One representative of those types of products includes sheet glass, or Thin Film Transistor-Liquid Crystal Display (TFT-LCD), glass substrates for Liquid Crystal Display (LCD) panels.

TFT-LCD has become the most popular Flat Panel Display (FPD) during the past decade, and the manufacturing of high quality TFT-LCD glass substrates with dimensions more than 2×2 m^2 is pushing the envelope for producing very large FPDs (bigger than of 100–*inch)*. However, one of the main weaknesses that remain with manufacturing larger glass substrates is their surface defects, such as surface warp and surface waviness. For example, the sheets with large warps, on the order of a few hundred micrometers, pose severe quality problems to the LCD panel industry.

Increased competition in this demand led the market of LCD panel manufacturers, as well as their raw material suppliers, such as sheet glass manufacturers, to make every effort to improve production yields. Nowadays, because of increasing the competition between sheet glass manufacturing, every manufacturer has been making efforts to improve product yields. Traditionally, the quality control has been often done by visual inspection of human operators. But, quality inspection by skilled human inspector is limited, open to discrepancy, time-consuming and costly for training the skillful human inspectors. On the other hand, Because of stochastic nature of the visual appearance of the products, inspection of surface waviness by human eyes is difficult and wrong judgments are easily made due to human subjectivity and eye fatigues. Recently, machine vision is being used to

automate and improve this process [5]-[7]. Since, developing an Automatic Optical Inspection (AOI) system will significantly contribute to the quality improvement of products in sheet glass manufacturing, using the AOI systems become more popular in these days. However, some manufacturing processes, such as sheet glass manufacturing, in which more than simple physical transformation of the raw materials or parts occurs, have certain types of surface defects that can hardly be detected by typical AOI.

In such manufacturing processes, defects are hard to define or measure; in other words, defects have stochastic appearances. For example, froth bubbles have very complex patterns that continuously vary in shape, size, direction, etc. There can be no discontinuous class of patterns because different patterns merge together to form more complex patterns [8]. For this reason, inspection of such stochastic defects is left for a trained human inspector. But, as mentioned before, inconsistency in human judgment has yet to be resolved in order to ensure reliable quality control; the same situations can be found in glass substrate manufacturing.

Another main difficulty in this type of the problem arises from the fact that prior information or knowledge is very limited: no accurate class labels are often available, nor is prior knowledge about important aspects of visual appearance often available. This limitedness of prior information, as well as the stochastic nature of the visual appearance of the products, makes it much more difficult to apply machine vision to the problems.

In AOI systems, first, the properties of products are specified using visual information then, they are automated by employing machine vision techniques [9]. An inline automatic optical inspection system has many advantages, such as increasing the production quality by elimination of human errors [10], and decreasing the time and costs in the process as well. In the past few years, AOI technology has been widely used in many

industries such as TFT-LCD for defect inspection and classification. There are two parts in the structure of these systems: the images are scanned and features are extracted in the first part, then defects are detected and classified by using one classification model. Fig. 2.4 shows the work-flow of the whole procedure (offline and online) for an automatic inspection system for the surface defects on TFT-LCD glass substrates which is the contribution of this work.

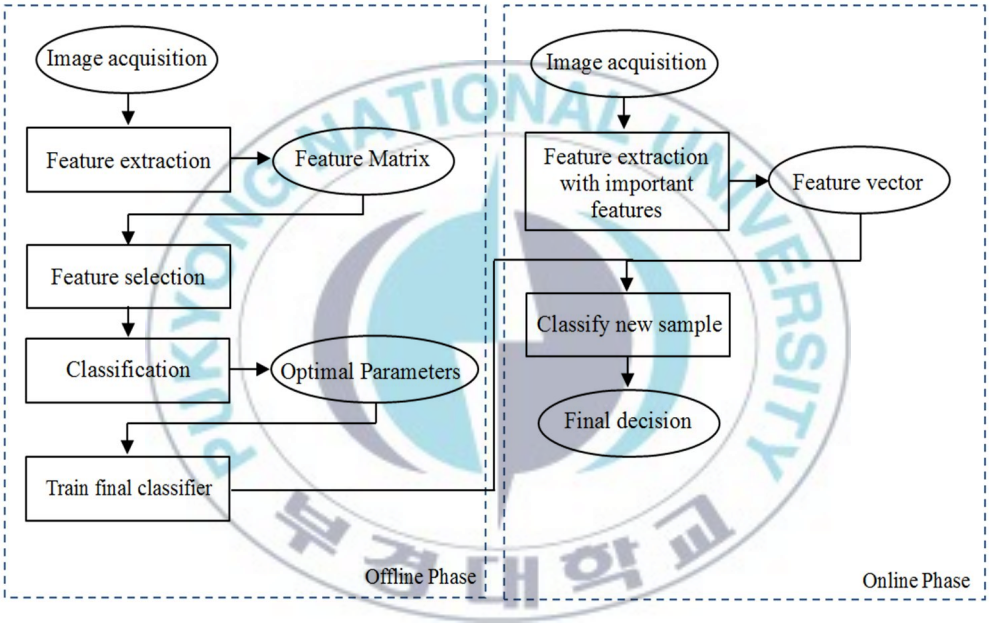


Fig. 2.4. Work-flow for automatic inspection system for surface defects on TFT-LCD glass substrates

2.3 Imaging Process

Since a glass substrate is reflective, as well as transparent, imaging glass substrates for detecting surface defects is somewhat tricky. Therefore, in the imaging process two kinds of images can be captured, reflective and transparent. Transmission images, which are taken of the light transmitting through the glasses, and reflection images, which are the light reflecting off

the glass substrates (Fig. 2.5). A TFT-LCD glass substrate has a thickness of 0.7*mm* (or 0.5*mm*) but their sizes are different based on the TFT-LCD's generation. In this work, we have worked on glass substrate of *generation 8*, which are 2200×2500*mm*. Because of the large size of mother glasses (glass substrates), and also limitations of the camera's resolution, it is not possible to take a photo of the whole surface of the mother glass and to capture defects with size of sub-millimeter by using single camera. Therefore, each mother glass was divided into 30 and 20 sections (sub-glasses) for imaging in transmission and reflection images, respectively.

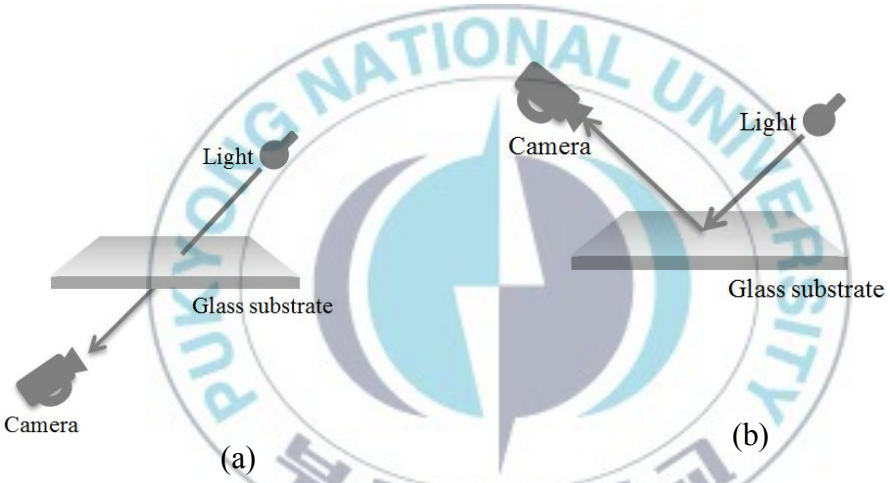
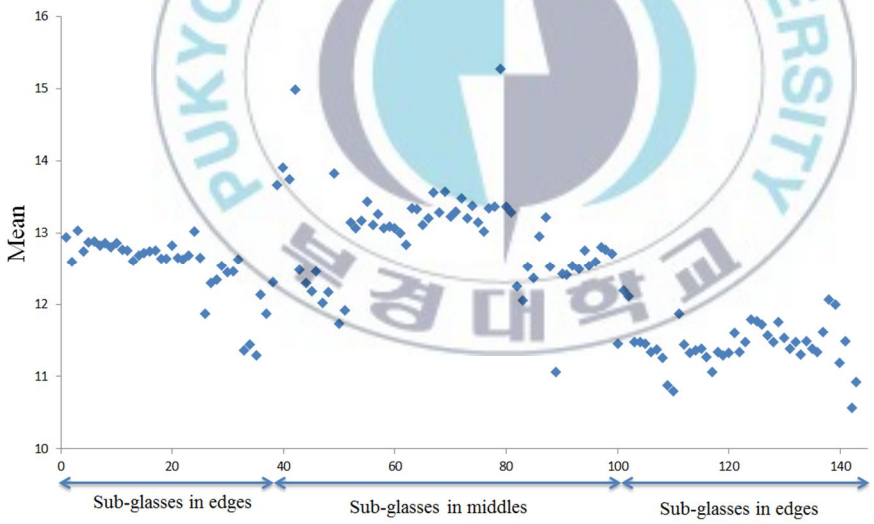


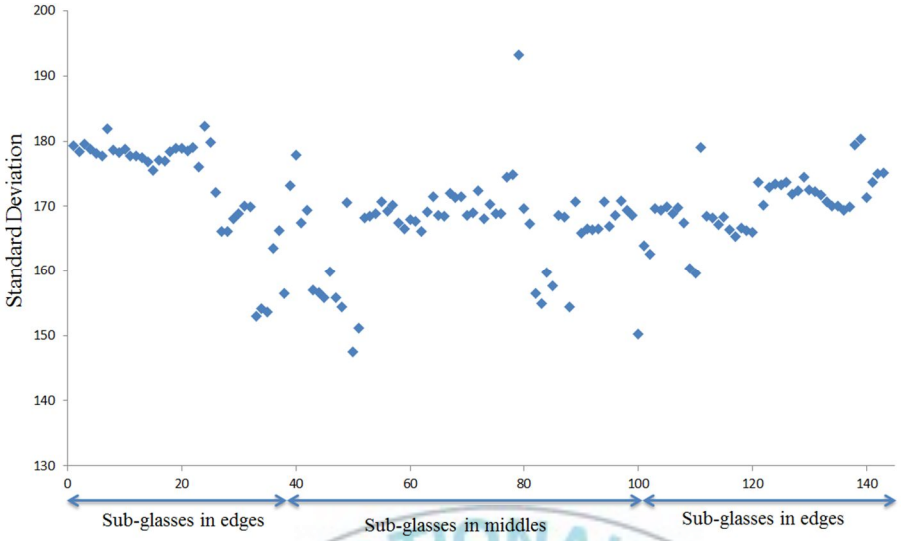
Fig. 2.5. Online (a) transmission and (b) reflection imaging system in inspection step of cold process.

On the other hand, the uniformity of pixel intensity of the sub-glass images can be affected by lighting conditions including environmental factors and the position of the sub-glasses. Fig. 2.6 shows the mean and standard deviation of pixel intensities for some sub-glasses on the edges (11^{th}, 20^{th}, 21^{th} and 30^{th}) and in the middle (14^{th}, 15^{th} and 16^{th}) of the glass substrates (mother glasses) in transmission images. As shown in this figure, the differences of mean and standard deviation of intensity between the sub-glasses on the edges and those in the center of the mother glasses can be

clearly seen. Fig. 2.7 also shows the mean and standard deviation of the pixel intensities for the 10 mother glasses and corresponding sub-glasses which are all ordered in imaging time. As shown in this figure, the differences in mean and standard deviation of intensity between the sub-glasses on the edges and those in the center of the mother glasses, as well as between the mother glasses, can be seen clearly. Also lighting conditions change over time as light sources age. This non-uniform and time-varying nature of lighting conditions is one of the most challenging problems in machine vision and this is one of the reasons why wavelet transform was chosen in this study. It is known that wavelet transform is inherently robust to lighting conditions due to its ability of multi-resolution analysis [8,11,12].

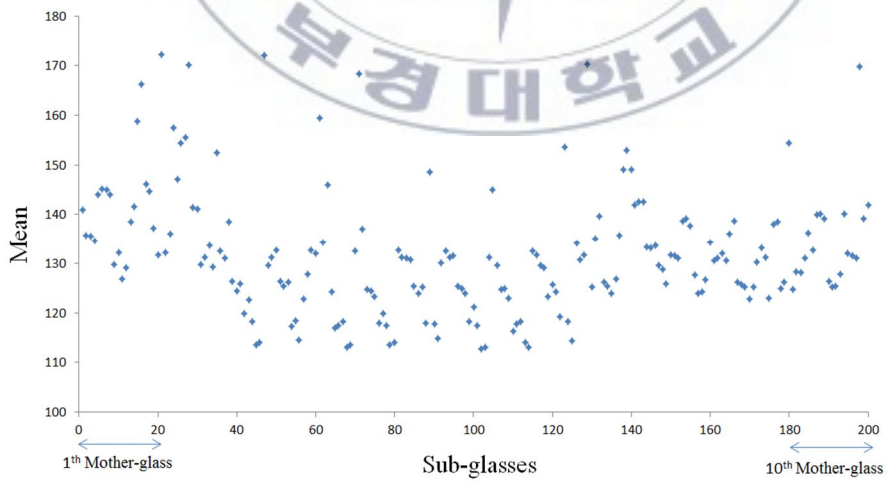


(a)

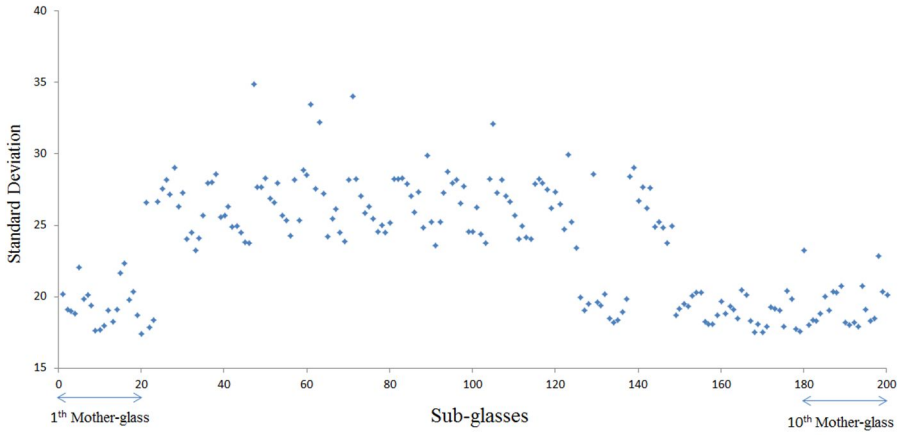


(b)

Fig. 2.6. (a) Mean and (b) Standard deviation of pixel intensities for some sub-glasses in the edges (11th, 20th, 21th and 30th) and in the middle (14th, 15th and 16th) of glass substrates.



(a)



(b)

Fig. 2.7. (a) Mean and (b) Standard deviation of pixel intensities for 10 mother glasses and all their ordered sub-glasses based on capturing time.

2.4 Surface Defects

An LCD panel consists of two glass substrates (called color filter glass and TFT glass), and a backlight unit. Liquid crystal fills the gap between the two glass substrates (see also Fig. 2.8). Since the gap is on the order of a few micrometers, surface defects over a length scale of a few micrometers can cause local brightness variation on an LCD TV screen, often called "*mura*" in the industry. Since surface defects on glass substrates greatly affect the most important quality of the final product, picture quality of an LCD TV, the LCD panel industry defines following defects that determine the surface quality of glass substrates [13]-[16]: (1) air bubbles and particles, (2) surface flaws or scratches, (3) surface roughness, (4) surface waviness, and (5) surface warp. The focus of this work is on the second and fourth defects which are characterized in transmission and reflection images, respectively, because the surface flaws or scratches and the surface waviness defects can be seen best in corresponding images. Other defects can be detected well using AOI or other equipment. For example, the first defect, air bubbles

(inside a glass substrate) and particles (inside or on a glass substrate) can be detected very well using inline AOI system, known as a “particle counter”.

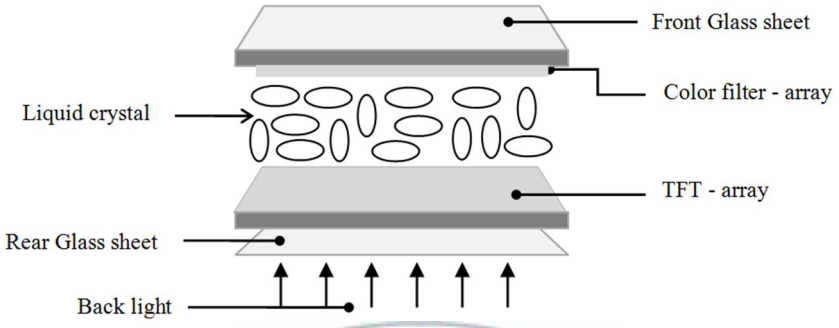
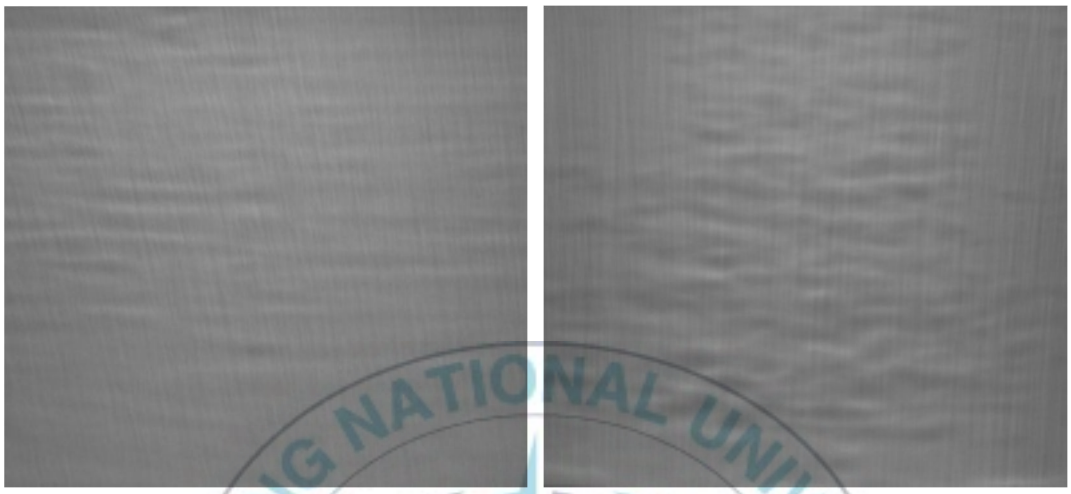


Fig. 2.8. Structure of a LCD panel.

Most surface defects can be easily seen by the bare eyes of expert human inspectors of those images. Samples of transmission and reflection images of glass substrates with and without surface flaws or scratches and waviness with some explanation of the defects are shown in Fig. 2.9 and 2.10 It should be noted that not all of these defects are easily visible to untrained eyes. These images show fractions of an entire glass substrate and a number of these “sub-glass” images depend on TFT-LCD generation, or the size of the glass substrates.

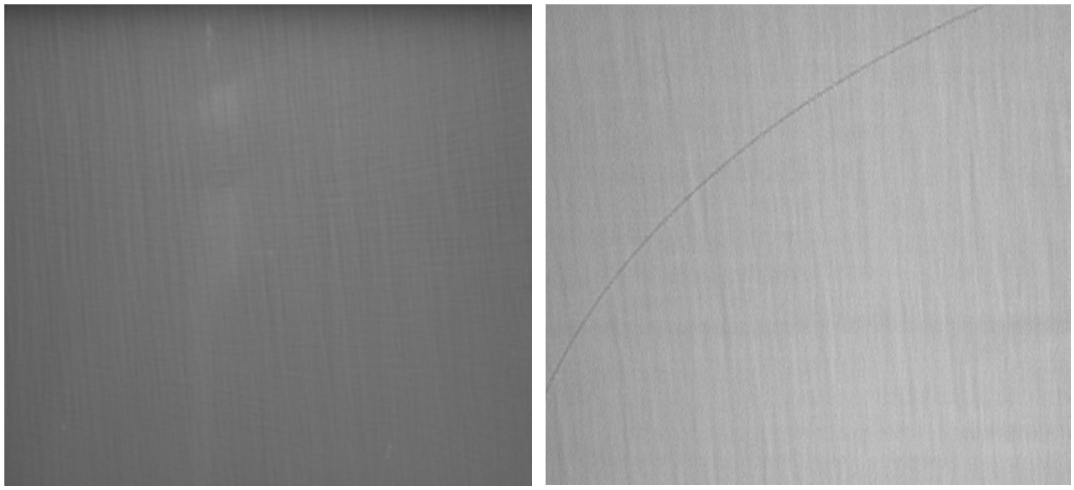
As can be seen in Fig. 2.10, the waviness surface defect can be observed to be similar to something heterogeneous, such as conveyor belt marks and water marks, which occur for a variety of reasons. Though, these heterogeneous examples can also cause local brightness variation in the images, conveyer marks have low-frequency characteristics while waviness has mid-frequency characteristics, actually, they are not defect on the glass substrates. Therefore, this study does not consider them as a defect on the glass substrates. These marks should be distinguished from our target defect,

waviness, in feature extraction part. This can be a part of our objective to use wavelet transform in this work.



Defect type A

Defect type B



Defect type C

Fig. 2.9. Examples of the three types of surface defects on transmission images. b) Type A has amorphous ripples. c) Type B has a few long arc-shaped lines. d) Type C has several faint diagonal lines.

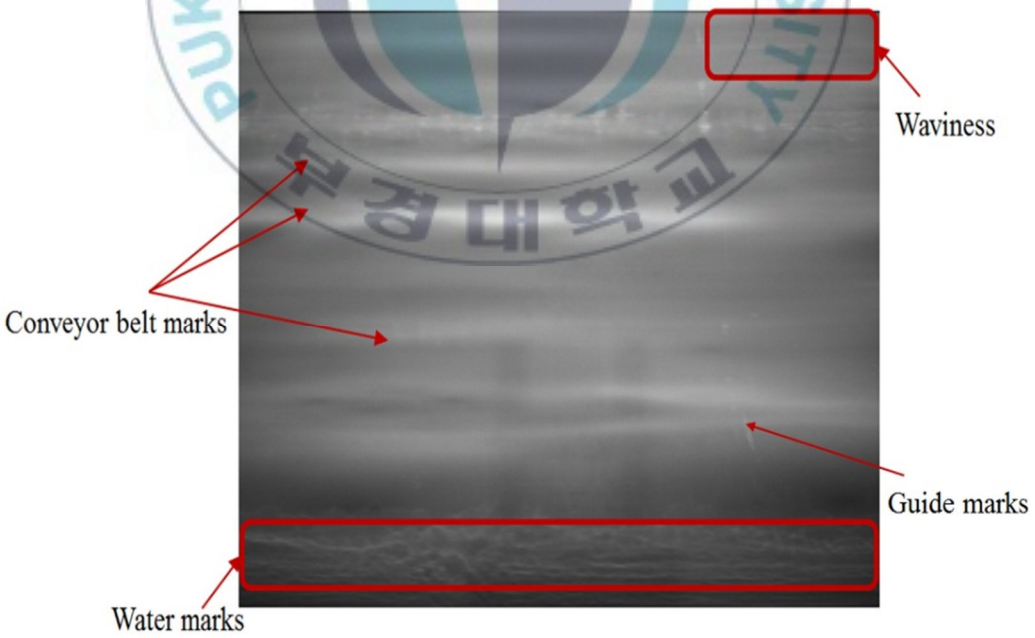


Fig. 2.10. Sample of reflection image: a sub-glass with waviness defect, conveyor belt and water marks.

Chapter 3

3. DATA PREPARATION

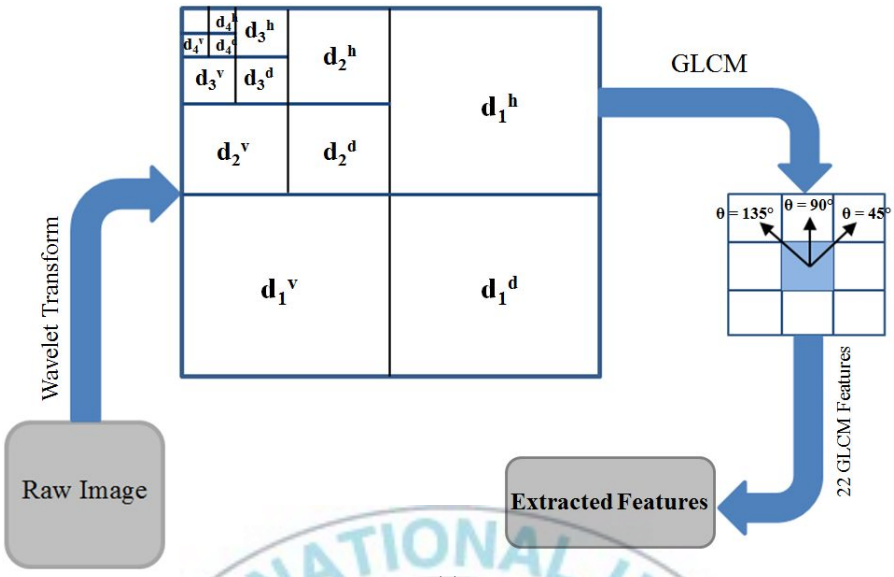
3.1 Feature Extraction

Seen from the figure of different surface defects on glass substrates (Fig. 2.9 and 2.10), detecting these defects is closely related to texture analysis since the image texture can be defined as a function of the spatial variation in pixel intensities [17]. Wavelet texture analysis is known as a very powerful, state-of-the-art method for extracting textural features from images [18]. Wavelet co-occurrence signature [19], which is a multiscale extension of the Grey Level Co-occurrence Matrix (GLCM) method [20], one of the oldest but still one of the most popular methods in texture analysis, is used in this work to extract the features from sub-images.

As shown in Fig. 3.1(a) and (b), since, the defect type *C* and waviness are not captured in the approximation wavelet sub-images, while local lighting non-uniformity and some image features such as conveyor belt marks, guide bar marks and water marks are in transmission and reflection images, the approximation wavelet detail is excluded from subsequent feature extraction steps to remove the non-uniform lighting variation and some image marks in sub-images. Fig. 3.1(c) also shows the feature extraction methodology which is used in this study.

(a)

(b)



(c)

Fig. 3.1. (a) the left is the raw image with defect type C (it also has local lighting non-uniformity between top side and the rest) and the right is the reconstructed image from level 4 approximation sub-image, (b) the left is the raw image with waviness (top-right) and also conveyor belt marks, guide bar marks, water marks and the right is the reconstructed image from level 4 approximation sub-image and (c) proposed feature extraction methodology.

3.1.1 Grey Level Co-occurrence Matrix (GLCM)

Let *I* be a given grey scale image. Let *N* be the total number of grey levels in the image. The Grey Level Co-occurrence Matrix defined by Haralick is a square matrix *G* of order *N*, where the $(i,j)^{th}$ entry of *G* represents the number of occasions a pixel with intensity *i* is adjacent to a pixel with intensity *j*. The normalized co-occurrence matrix is obtained by dividing each element of *G* by the total number of co-occurrence pairs in *G*. The adjacency can be defined to take place in each of the four directions (horizontal, vertical, left and right diagonal) as shown in Fig. 3.2. The Haralick texture features are calculated for each of these directions of adjacency [20].

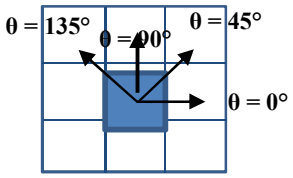


Fig. 3.2. The four directions of adjacency for calculating the Haralick texture features

The texture features are calculated by averaging over the four directional co-occurrence matrices. To extend these concepts to *n*-dimensional Euclidean space, we precisely define grey scale images in *n*-dimensional space and the above mentioned directions of adjacency in *n*-dimensional images. Following are the suggested textural features by Haralick [20].

Textural Features	
1) Angular Second Moment: $f_1 = \sum_i \sum_j \{p(i,j)\}^2$	2) Contrast: $f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ \lvert i-j \rvert = n}}^{N_g} p(i,j) \right\}$
3) Correlation: $f_3 = \dfrac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$	4) Sum of Squares: Variance $f_4 = \sum_i \sum_j (i-\mu)^2 p(i,j)$
5) Inverse Difference Moment: $f_5 = \sum_i \sum_j \dfrac{1}{1+(i-j)^2} p(i,j)$	6) Sum Average: $f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$
7) Sum Variance: $f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$	8) Sum Entropy: $f_8 = -\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$

9) Entropy: $f_9 = -\sum_i \sum_j p(i,j)\log(p(i,j))$	10) Difference Variance: f_{10} = variance of p_{x-y}
11) Difference Entropy: $f_{11} = -\sum_{i=0}^{N_{g-1}} p_{x+y}(i)\log\{p_{x-y}(i)\}$	12) Maximal Correlation Coefficient: f_{12} = (Second largest eigenvalue of $Q)^{1/2}$ where $Q(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$
13, 14) Information Measures of Correlation: $f_{13} = \frac{HXY - HXY1}{\max\{HX, HY\}}$ $f_{14} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$ $HXY = -\sum_i \sum_j p(i,j)\log(p(i,j))$	where HX and HY are entropies of p_x, and p_y, and $HXY1 = -\sum_i \sum_j p(i,j)\log\{p_x(i)p_y(j)\}$ $HXY2 = -\sum_i \sum_j p_x(i)p_y(j)\log\{p_x(i)p_y(j)\}$

3.1.2 Wavelet Co-occurrence Signature

It can be observed in Fig. 2.9 and 2.10 that the defects on a glass substrate often appear in pixel regions. The main difference between a normal pixel region and a defective one is its appearance, i.e., the spatial variation of pixel intensities. Because image texture can be defined as a function of the local spatial variation in pixel intensities [17], there is a specific type of texture in the defective pixel region, while the gray-level distribution of a normal one is almost uniform. Since texture is one of the important characteristics used for identifying objects or regions of interest in an image, detecting the different surface defects on glass substrates is closely related to texture analysis.

A wide variety of techniques have been proposed and used for analyzing texture, such as gray-level co-occurance matrix, Fractals, Gabor filters, variations of wavelet transform [17,21,22,23]. In General, methods for texture analysis can fall into one of four categories: (1) structural methods, (2) model-based methods, (3) statistical methods, and (4) transform-based methods [23]. The gray-level co-occurrence matrix (GLCM) method [20], is the oldest but still one of the most popular methods in statistical methods and texture analysis in general.

Haralick *et al*. [20] proposed the use of GLCM, which is related to the second order statistics of pixel intensities in an image, to characterize its texture using 14 textural features calculated from GLCMs. In this method, one can easily manipulate parameters such as distance (d), angle (θ), and gray level (G) in the calculation of $G \times G$ GLCMs from an image. In many applications, GLCMs with d = 1 or 2, $\theta = 0°$, 45°, 90°, and/or 135°, and G much lower than 32 were enough to provide good results with affordable computational cost. Van de Wouwer [19] applied this method to wavelet detail coefficients to get GLCMs and corresponding textural features, wavelet co-occurrence signatures. In other words, wavelet co-occurrence signatures are higher order statistics based on the co-occurrence matrix of two-dimensional (2-D) wavelet detail coefficients (also called wavelet sub-images). For more details about wavelet texture analysis, GLCM, and wavelet co-occurrence signatures, please refer to the references [17]-[20].

3.2 Feature Reduction & Selection

3.2.1 Principal Component Analysis (PCA)

The most popular statistical method for dimensionality reduction of a large data set is the Karhunen-Loeve (K-L) method, also called *Principal*

Component Analysis (PCA). Principal component analysis is a method of transforming the initial data set represented by vector samples into a new set of vector samples with derived dimensions. The goal of this transformation is to concentrate the information about the differences between samples into a small number of dimensions. More formally, the basic idea can be described as follows:

I. Firstly the data set procured from the experiment are normalized as

$$x_i'(j) = \frac{x_i(j) - \min(x(j))}{\max(x(j)) - \min(x(j))} \tag{3.1}$$

where $x_i'(j)$ is the new value of the normalized data for *j*th parameter in *i*th experiment, $x_i(j)$ is the value of *j*th parameter in *i*th experiment.

II. The new normalized multi-response array for *m* parameters and *n* experiment can be represented by matrix x' as

$$x' = \begin{bmatrix} x_1'(1) & x_1'(2) & \cdots & x_1'(m) \\ . & . & . & x_2'(m) \\ . & . & . & \vdots \\ x_n'(1) & x_n'(2) & \cdots & x_n'(m) \end{bmatrix} \tag{3.2}$$

III. The correlation coefficient array (R_{jl}) of matrix x' is written as Follows

$$R_{jl} = \frac{\mathrm{cov}(x_i'(j), x_i'(l))}{\sigma_{x_i'(j)} \times \sigma_{x_i'(l)}} \quad , \quad j = 1,2,\cdots,m \quad , \quad l = 1,2,\cdots,m \tag{3.3}$$

where $\mathrm{cov}(x_i'(j), x_i'(l))$ is the covariance of sequences $x_i'(j)$ and $x_i'(l)$; $\sigma_{x_i'(l)}$ is the standard deviation of sequence $x_i'(l)$.

IV. The eigenvalues and eigenvectors of matrix (R_{jl}) are calculated.

V. The PC are computed as follows

$$p_i(k) = \sum_{j=1}^{m} x_i'(j) \times v_k(j) \tag{3.4}$$

where $p_i(k)$ is the kth PC corresponding to ith experiment, $v_k(j)$ is jth element of kth eigenvector.

VI. The total principal component index (TPCI) corresponding to ith experiment (p_i) is computed as follows

$$p_i = \sum_{k=1}^{m} p_i(k) \times e(k) \tag{3.5}$$

$$e(k) = \frac{eig(k)}{\sum_{k=1}^{m} eig(k)} \tag{3.6}$$

where *eig(k)* is the kth eigenvalue.

VII. The TPCI for each experiment is used to realize the average factor effect at each level. The optimum parameter level that corresponds to the maximum TPCI is also predicted.

3.2.2 Parallel Genetic Algorithm (PGA)

Since increasing the dimensionality of the feature space will also result in an increased complexity of the interactions among the features and increase the degree of noise [24,25], Parallel Genetic Algorithm (PGA) was used, which is an extended version of the genetic algorithm, for reducing the dimension of the feature space in this study. Let $C = \{x_1, x_2, \ldots, x_q\}$ be the set containing all of q possible features, and Ω be the collection of all subsets of C. The goal of feature selection is to find the *best* features $\omega \in \Omega$. In a Genetic Algorithm (GA) each individual ω is represented by a binary string of length q, which is treated as the genetic code (DNA) of ω. Starting with a randomly generated population of size m ($\{\omega_1, \omega_2, \ldots, \omega_m\}$), a new generation

is produced with three genetic operations: selection, reproduction, and mutation.

Zhu & Chipman [26] first demonstrated that the GAs, although seemingly natural for the feature selection problem, are not actually easy to use or are hardly effective and they proposed a very simple modification. Their idea was to run a number of GAs in parallel without allowing each GA to fully converge, and then to unify the information from all of the individual GAs at the end. They also specified the appropriate stopping criterion as follow: given a collection of binary sequences of length q (with each sequence containing p bits), let r_j be the frequency that the j^{th} bit is equal to 1. Then the average "per bit" entropy of this collection is given by

$$entropy = -\frac{1}{q}\sum_{j=1}^{q} r_j \log_2(r_j) + (1-r_j)\log_2(1-r_j) \tag{3.7}$$

Therefore, the GA can be regarded as having converged when the entropy of the population is sufficiently close to *0*, that is, below a prespecified threshold δ (e.g., δ = 0.05). $\bar{r}_j$, which is the percentage of last-generation candidates in all of the parallel paths that contain feature j, is used as an importance measure to rank the feature j.

It was shown with a simulation study that parallel evolution, or PGA, is competitive in its ability to recover the correct model. The strength and usefulness of parallel evolution was also illustrated with both simulated and real datasets indicating its general ability to be implemented as a feature selection tool for more complex statistical models. For more details about PGA, please refer to the reference [26].

3.3 Synthetic minority over-sampling technique (SMOTE)

Imbalance data learning have increasingly attracted many researchers for more than a decade. The problem of imbalanced data is often associated with asymmetric costs of misclassifying elements of different classes (i.e., algorithm generally gives more important to correctly classify the majority class samples). One way to solve class imbalance problem is creating or modifying algorithm which includes the cost-sensitive method. The other way is using data-preprocessing techniques like sampling which is applied on data in which either new samples are added or existing samples are removed. Process of removing a sample known as under-sampling and process of adding new sample in existing is known as over-sampling.

Synthetic minority over-sampling technique (SMOTE) is well known as an over-sampling method which was proposed by Chawla *et al.* [27]. Here the minority class is over-sampled by creating "synthetic" instances in the feature space rather than by over-sampling with replacement. This approach is motivated by a technique presented by Ha and Bunke [28] in handwritten character recognition.

The minority classes are over-sampled by taking each minority class and creating synthetic instances of it along the line segments joining any/all of the k nearest neighbors' minority class, where k is predetermined. Depending upon the required over-sampling amount (b), synthetic data points are generated by randomly selecting data points. For instance, if b = 300% and k = 5, then one out of five nearest neighbors of $\mathbf{x}_0$ is randomly chosen for three times repeatedly. Each time a random k^{th} neighbor is selected to create a line connecting $\mathbf{x}_0$ to this neighbor and then a single synthetic instance is created by randomly selecting a point on the line. Therefore, synthetic instance $\mathbf{x}_{new}$ can be defined as a following:

$$\mathbf{x}_{new} = \mathbf{x}_0 + \delta * (\mathbf{x}_0^{(t)} - \mathbf{x}_0) \tag{3.8}$$

where $\mathbf{x}_0^{(t)}$ is the t^{th} nearest neighbors of $\mathbf{x}_0$ in the minority class and $\delta \in [0,1]$ is a random number. The procedure is repeated for all the minority samples. More information on the SMOTE algorithm and its pseudo code can be found in the work by Chawla et al. [27].

Chapter 4

4. MODELING

4.1 Multi-layer Perceptron (MLP)

Generally Artificial Neural Networks (ANNs) are basic input and output devices, with the neurons organized into layers. Simple perceptrons consist of a layer of input neurons, coupled with a layer of output neurons, and a single layer of weights between them. The learning process consists of finding the correct values for the weights between the input and output layer. The schematic representation given in Fig. 4.1 is often how neural nets are depicted in artificial neural networks are composed of input, hidden and output layers, and connections.

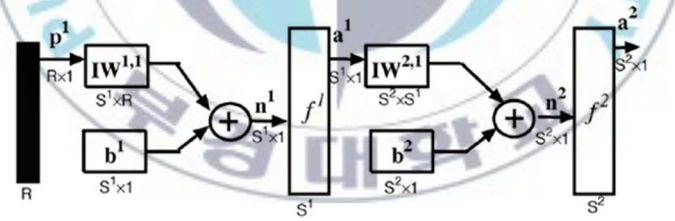


Fig. 4.1. Multilayer perceptron network

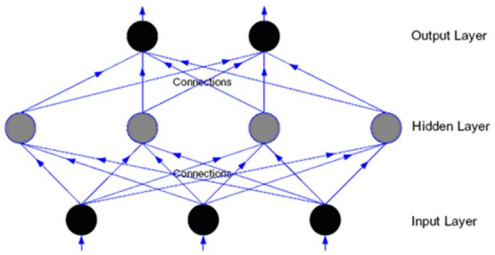


Fig. 4.2. Multilayer perceptron: 3 inputs, a hidden layer with 4 neurons and 2 outputs.

Generally data is presented at the input layer, the network then processes the input by multiplying it by the weight layer. The result of this multiplication is processed by the output layer nodes, using a function that determines whether or not the output node fires. The process of finding the correct values for the weights is called the learning rule, and the process involves initializing the weight matrix to a set of random numbers between -1 and +1. Then as the network learns, these values are changed until it has been decided that the network has solved the problem. Finding the correct values for the weights is effected using a learning paradigm called supervised learning.

The principle weakness of the simple perceptron was that it could only solve problems that were linearly separable. A Multi-Layers Perceptron (MLP) is a particular of artificial neural network [29]. An example of a multilayer network with three inputs, a hidden layer with four neurons and two outputs is shown in Fig. 4.2 MLPs can be developed to learn the relationship between inputs and outputs. They are composed of sets of embedded linear and nonlinear functions. In the MLPs, the actual output y can be defined by (Fig. 4.1)

$$a_k^2 = f^2\left(\sum_{j=1}^{s^1} w_{jk}^2 f^1\left(\sum_{i=1}^{R} w_{ij}^1 p_i + b_j^1\right) + b_k^2\right), \quad k = 1,\ldots, s^2 \tag{4.1}$$

where superscript 1 denotes hidden layer and superscript 2 denotes output layer. R, S^1 and S^2 demonstrate the numbers of the input, hidden and output units, respectively. Also, f, w and b represent transfer function, synaptic weight parameter and bias, respectively. Following is the transfer function f which is used in this study:

$$\textit{Log–sigmoid: } f(x) = \frac{1}{1+e^{-x}} \tag{4.2}$$

One of the most critical tasks in neural network design is the selection of appropriate number of hidden layers. Unlike the input and output layers, one

starts with no past information as to the number of hidden layers. If a network has too few or too many hidden nodes, it faces some problems such as following the noise in the data and leading to poor generalization for untrained data. Moreover, with increasing number of hidden layers, training becomes very time-consuming. Discovering the optimal values of neural network parameters is important to achieve a good forecast and estimation performance. In this study, the parameters of MLP model are selected by simulated annealing.

4.2 Support Vector Machine (SVM)

A training data set is defined by $D=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}$ where $\mathbf{x}_i \in \Re^N$, $y_i \in \{-1,1\}$ and n is the number of training data points. Support Vector Machines (SVMs) are learning machines, which mean that a linear function of $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x})+b$ is used to solve the classification problems in a higher dimensional version of $\mathbf{x}$, $\Phi(\mathbf{x})$. The best line is defined to be a line which minimizes the following cost function:

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \tag{4.3}$$

$$s.t. \quad \mathbf{w}^T\Phi(\mathbf{x}_i) + b \geq +1-\xi_i, \quad if \quad y_i = +1$$

$$\mathbf{w}^T\Phi(\mathbf{x}_i) + b \leq -1+\xi_i, \quad if \quad y_i = -1$$

where ξ_i is the corresponding error at the i^{th} point, C is the penalty parameter and $\Phi(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a higher-dimensional space.

The minimization of (4.3) is a standard problem in optimization theory: minimization with constraints. This can be solved by applying the Lagrangian theory. With the help of Lagrange theory, the dual formulation becomes:

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (4.4)$$

$$s.t. \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad \forall i$$

According to α_i Lagrange multipliers, the decision function is written as following:

$$y = \text{sgn}[\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b] \qquad (4.5)$$

where $K(x_i, x)$ is named the kernel function. In this study, the following Radial Basis Function (RBF) was used:

$$K(x_i, x_j) = \exp(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2) \qquad (4.6)$$

where γ is the kernel parameter [30]-[32].

4.3 Simulated Annealing (SA)

Choosing the optimal values for parameters of SVM and MLP is important to get the good results [33,34]. For this reason, simulated Annealing (SA) algorithm was used in this study. With the graphical description in Fig. 4.3 [35], a summary of SA algorithm steps is as follows:

In the first step, chose $\mathbf{x}^{(0)}$ as an initial solution and compute the value of the objective function, $F(\mathbf{x}^{(0)})$. After that, until the stopping criterion is met, do the following for *n* starting from 0:

draw a solution, **x**, at random in the neighborhood $V(\mathbf{x}^{(n)})$ of $\mathbf{x}^{(n)}$.

If $F(\mathbf{x}) \leq F(\mathbf{x}^{(n)})$ then $\mathbf{x} \rightarrow \mathbf{x}^{(n+1)}$.

If $F(\mathbf{x}) > F(\mathbf{x}^{(n)})$ then draw a number ρ at random in [0,1] and if $\rho \leq p(n, \mathbf{x}, \mathbf{x}^{(n)})$ then $\mathbf{x} \rightarrow \mathbf{x}^{(n+1)}$ else $\mathbf{x}^{(n)} \rightarrow \mathbf{x}^{(n+1)}$.

The function is often taken to be a Boltzmann function inspired from thermodynamics models:

$$p(n, \mathbf{x}, \mathbf{x}^{(n)}) = \exp\left(-\frac{1}{T_n}\Delta F_n\right), \qquad \Delta F_n = F(\mathbf{x}) - F(\mathbf{x}^{(n)}) \tag{4.7}$$

where T_n is the temperature at step n, which is a non-increasing function of the iteration counter n. In the so-called geometric cooling schedule, the temperature is kept unchanged during each successive stage, where a stage consists of a constant number L of consecutive iterations. Therefore, the SA has converged when the algorithm reaches L or $\left|F(\mathbf{x}^{(n+1)}) - F(\mathbf{x}^{(n)})\right| \le 0.05$. After each stage, the temperature is multiplied by a constant factor of $\alpha \in (0,1)$. In this work, $\alpha = 0.95$, $T_0 = 0.01$ and the initial solution $\mathbf{x}^{(0)}$ was selected randomly as well as defined classification accuracy in section 5.1.2 (Eq. 5.1) was used as an objective function.

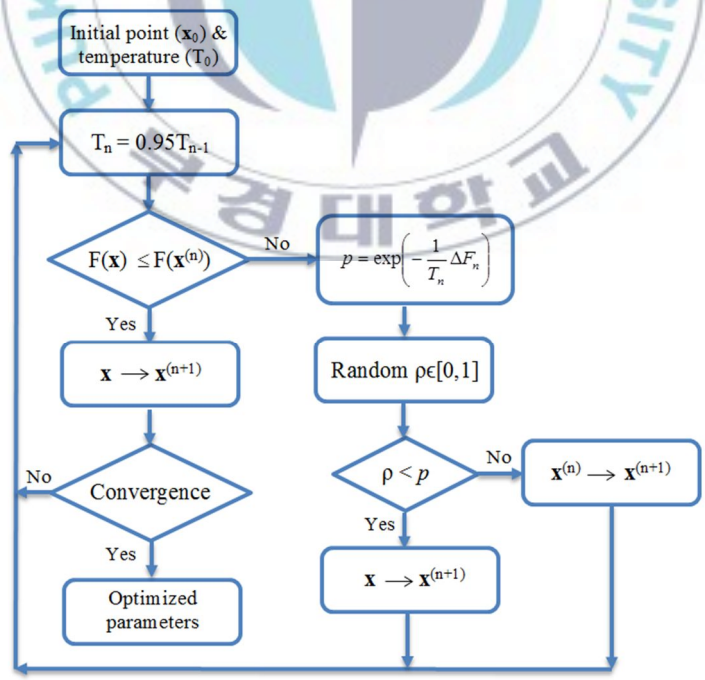


Fig. 4.3. A flow chart of simulated annealing

4.4 Classification and Regression Tree (CART)

Focusing on the data provided, this model creates a model of tree-shaped structure using inductive reasoning [36]. An important thing which we should consider in create the CART models is the construction of the "right-sized" tree [37]. In the extreme, if a tree grows so large that each terminal node contains only one entry, it may completely classify the training data, but will most probable gain significant errors in the testing data. This event is referred to as "overfitting". CART carry outs a tree-pruning procedure to avoid overfitting by using either cross-validation or a testing data set after a large tree is grown [37]. The analyses with the pruning provide an optimal tree to get purpose of prediction. The best predictor in the CART structure is chosen based on a variety of impurity or diversity measures (Gini, twoing and least-squared deviation). In this study, we used measure of Gini impurity that is used for categorical targets.

Gini Impurity Measure:

The Gini index at node *t*, *g(t)*, is defined as

$$g(t) = \sum_{k \neq l} p(k|t)\, p(l|t) \tag{4.8}$$

or

$$g(t) = 1 - \sum_{k} p^2(k|t) \tag{4.9}$$

where k and l are categories of the target variable. Suppose that m is the number of categories in this variable. When the objects in a node are equally distributed across the categories, the Gini index gets its maximum value of *1-(1/m)*. The Gini index will be 0 if all objects in the node belong to the same category. The following formula for Gini index is defined when the costs of misclassification are specified

$$g(t) = \sum_{k \neq l} C(l|k)\, p(k|t)\, p(l|t) \tag{4.10}$$

where $C(l|k)$ is the probability of misclassifying a category k case as category l. The best split, s, is the one that maximizes the following function:

$$\Phi(s,t) = g(t) - p_L g(t_L) - p_R g(t_R) \quad (4.11)$$

where p_L and p_R are the proportions of objects in t that send to the left (t_L) or right (t_R) child nodes, respectively. The value of s is reported as the improvement in the tree [37].

4.5 Cost-sensitive C5.0 Classifier

The C5.0 algorithm is a new generation of machine learning algorithms based on decision trees. It means that the decision trees are built from list of possible attributes and set of training cases, and then the trees can be used to classify subsequent sets of test cases. C5.0 was developed as an improved version of well-known and widely used C4.5 classifier and it has several important advantages over its ancestor. The generated rules are more accurate and the time used to generate them is lower. In C5.0 several new techniques were introduced:

- boosting: several decision trees are generated and combined to improve the predictions.
- variable misclassification costs: it makes it possible to avoid errors which can result in a harm.
- new attributes: dates, times, timestamps, ordered discrete attributes.
- values can be marked as missing or not applicable for particular cases.
- supports sampling and cross-validation.

Suppose there are C classes, and $Cost[i,c]$ ($i, c \in [1,...,C]$) denote the cost of misclassifying a member of the i^{th} class as being the c^{th} class, and $Cost[i]$ denote the cost of the i^{th} class. $Cost[i]$ is usually derived from $Cost[i,c]$.

There are many possible rules for the derivation, among which a popular one is $Cost[i] = \sum_{c=1}^{C} Cost[i,c]$ [38].

Threshold-moving is another benefit of using C5.0 classifier which moves the output threshold toward inexpensive classes such that examples with higher costs become harder to classify and more likely to be misclassified. Concretely, let O_i ($i \in [1,...,C]$) denote the real-value output of a C5.0 decision tree, $\sum_{i=1}^{C} O_i = 1$ and $0 \le O_i \le 1$. In standard classifiers, the class returned is, $\underset{i}{\operatorname{argmax}} O_i$; while in threshold-moving, the class returned is $\underset{i}{\operatorname{argmax}} O_i^*$. O_i^* is calculated according to (1), where η is a normalization term such that $\sum_{i=1}^{C} O_i^* = 1$ and $0 \le O^*_i \le 1$.

$$O_i^* = \eta \sum_{c=1}^{C} O_i Cost[i,c] \tag{4.12}$$

The presented threshold-moving algorithm is similar to the cost-sensitive classification method [39]. Provost in his paper said that "*the bottom line is that when studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake*" [40]. It has also been declared that trying other methods, such as sampling, without trying to simply set the threshold may be misleading [40]. Maloof showed that threshold-moving is as effective as sampling methods in solving the class imbalance problem [41].

4.6 Ensemble Technique

Ensemble learning methods train multiple component learners and then combine their predictions. Therefore, ensemble techniques improve the predictive power when compared with single learners; ensemble learning has been a hot topic during the past few years [42]. Zhou & Liu [43] proposed hard-ensemble and soft-ensemble, i.e. the combination of classifiers via hard or soft voting schemes and concluded soft-ensemble may cause negative effect on some seriously imbalanced data sets. The only difference between hard-ensemble and soft-ensemble is that using the binary votes and real-value votes, respectively. Therefore, the hard-ensemble is used to combine the classifiers' predictions.

In hard ensemble, every component learner votes for a class and then the class receiving the largest number of votes is returned. In this work, different cost-sensitive learners are trained with the threshold-moving algorithms. Therefore, it is feasible to combine these learners into an ensemble which uses binary votes of classification decisions of the component learners.

Chapter 5

5. EXPERIMENTAL RESULTS

5.1 Transmission Images (Experiment I)

5.1.1 Data Preparation and Preprocessing

For this study, 1182 sub-glass images were collected from production lines that were labeled by expert inspectors. There are four classes, including on-specification sub-glass (labeled as OK), and off-specification sub-glass that have one of three different types of defects (labeled as A, B, C). A training set of 826 sub-glass images was chosen from total of 1182 sub-glass images by random selection, and the remaining 356 images including 248, 48, 11 and 49 samples in OK, A, B, and C class, respectively, were used as a testing set. In the training set, there are 588 sub-glasses in the OK class, 110, 22 and 106 sub-glasses are in the A, B and C classes, respectively.

Among 14 GLCM features proposed by Haralick *et al*. [20], four features (angular second moment, contrast, energy, entropy) were extracted from the GLCMs of wavelet sub-images after applying a 4 level wavelet decomposition on the substrate sub-images using bior1.3 wavelet function (biorthogonal wavelets with an order one reconstruction filter and an order three decomposition filter). As for GLCMs, the values of the design parameters used were $d = 2$, $\theta = 45°$ and $135°$, and $G = 32$. The design parameters of the wavelets transform and the GLCMs were found to be appropriate by trial and error. All 12 detailed wavelet sub-images (4 levels×3 directions (horizontal, vertical, and diagonal) = 12 sub-images) were used, and two GLCMs (45° and 135°) were calculated from each wavelet sub-

image. Since four GLCM features were calculated from each GLCM, there are a total of 96 wavelet co-occurrence features extracted from one sub-image of a glass substrate; which were used as a feature vector for the sub-glass image. Therefore, the training set is an 826×96 matrix and the test set is a 356×96 matrix.

Fig. 5.1 shows the importance plots from the PGA for 75 parallel paths; each path is evolved for 24 generations. Also, mutation and crossover rates of 0.05 and 0.5, respectively, were used for each GA. In general, the number of features to be selected must be determined. This can easily be done by plotting the importance of the features from the largest to the smallest and looking for a large gap. Fig. 5.1 shows that the first seven features are clearly separated from the rest [26]. The feature ID's for the seven selected features are 23, 41, 42, 49, 85, 89, and 90. Since all GLCM features were calculated from wavelet sub-images having different spatial frequencies, only certain frequency regions capture the textural characteristics that can discriminate between the different surface defects best. Further investigation revealed that among the seven selected features, six features were calculated from the same resolution, four of them were from the angular second moment, and two of them were from the contrast, which are related to uniformity and local variation of texture, respectively.

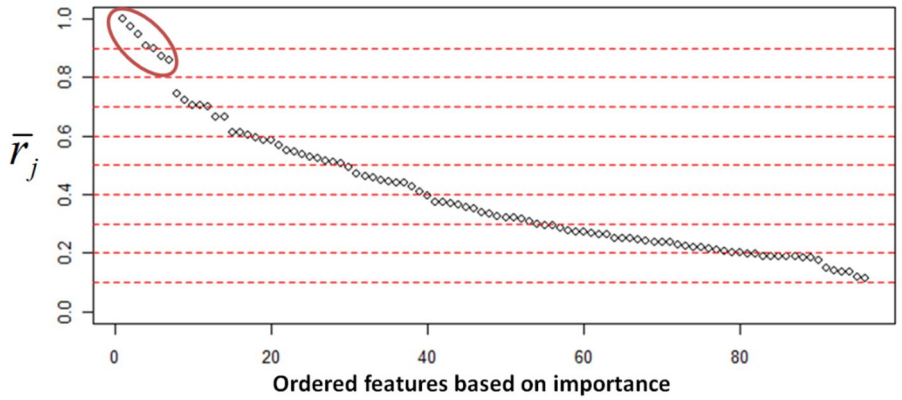


Fig. 5.1. Ordered importance plot for features by using PGA

5.1.2. Results and Discussion

Before applying classification techniques, score plots of the first three principal components of the features were plotted in Fig. 5.2 so as to see the characteristics of the classification problem. One can easily see that none of the four classes are linearly separable and the use of non-linear classification techniques may be desirable. In this investigation SVMs were used, which are considered to be the state-of-the-art classification method for pattern recognition, and the radial basis function was employed as the kernel function. The use of the radial basis function is inspired from the empirical findings that radial basis kernels tend to give good performance under general smoothness circumstances, and therefore should be considered, especially if no additional knowledge of the data is available.

As there is no structured way to choose the optimal parameters of SVMs, the values of parameters were found by a Simulated Annealing (SA) algorithm employing classification accuracy as the objective function to maximize. Table 5.1 shows the optimal values of the SVM parameters resulting from the SA analysis for classification of the four different classes of glass substrates. As shown in this table, the optimal kernel (γ) for SVM when using and when not using feature selection is 0.5 and 0.9, as well as penalty parameter (C) is 17 and 24, respectively.

Table 5.1. The optimal values of the SVM parameters resulting from the simulated annealing algorithm.

	SVM parameters	
	kernel parameter (γ)	penalty parameter (C)
Without Feature Selection	0.5	17
With Feature Selection	0.9	24

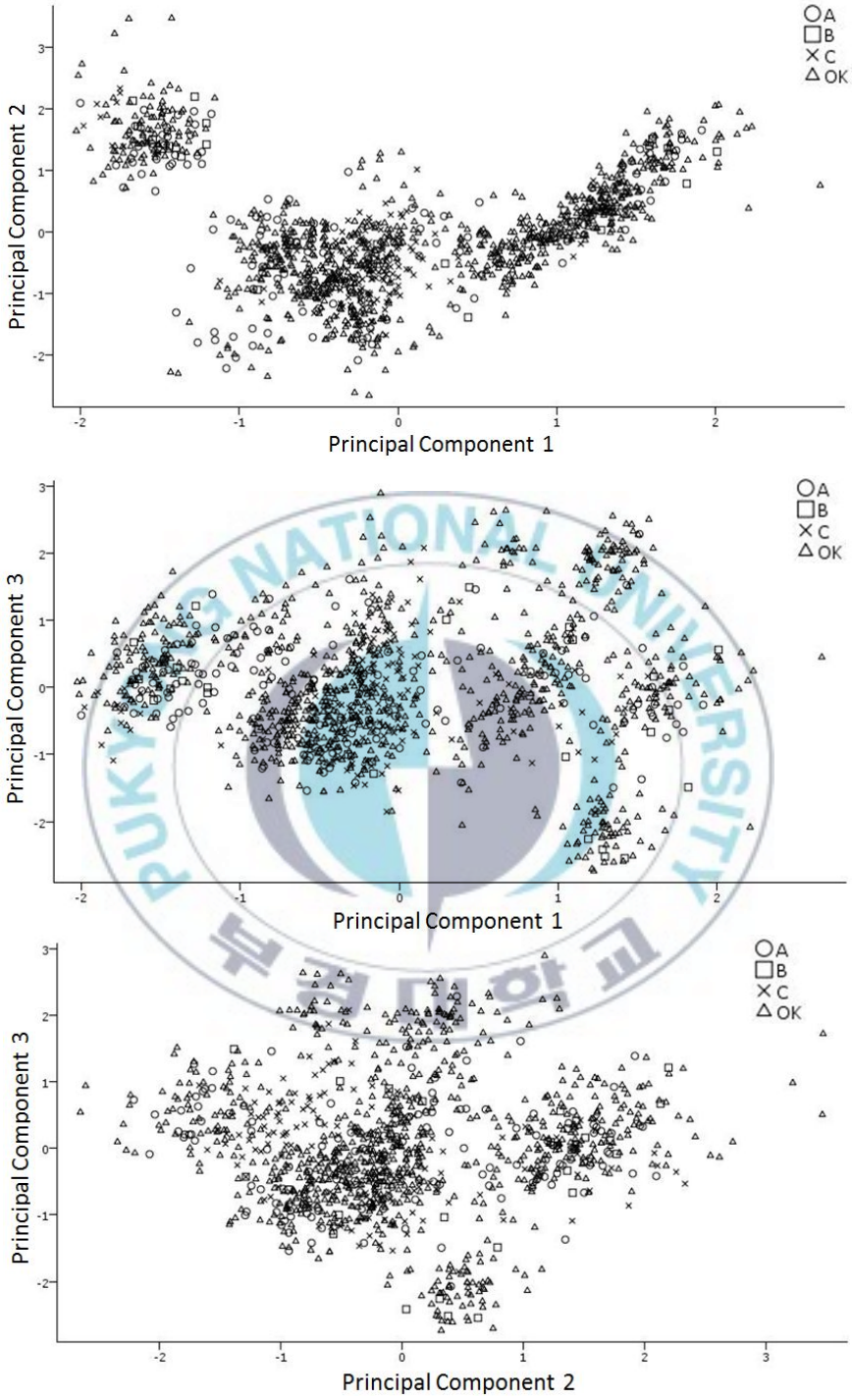


Fig. 5.2. Scatter plots of first three principal components

The training and testing results for the automatic inspection system are summarized in Table 5.2. This table shows the performance of the proposed system with and without using feature selection for the classification of TFT-LCD glass substrates. The training result with and without feature selection was nearly the same, but testing results were different. The best test accuracy value for classification using SVM with and without feature selection was 77.53% and 83.13%, respectively. In addition, the proposed method needs to be robust to the parameters of the cooling schedule of SA. Table 5.2 also shows the results of the simulation for different cooling schedules of SA under different T_0, where $L=100$, $\alpha = 0.95$. So, the optimized SVM with feature selection can classify four classes with the most satisfactory performance. The accuracy which was used to evaluate the performance of the proposed method in experiments *I* and *II* is defined below:

$$Accuracy = \frac{T_{OK} + T_A + T_B + T_C}{n} \tag{5.1}$$

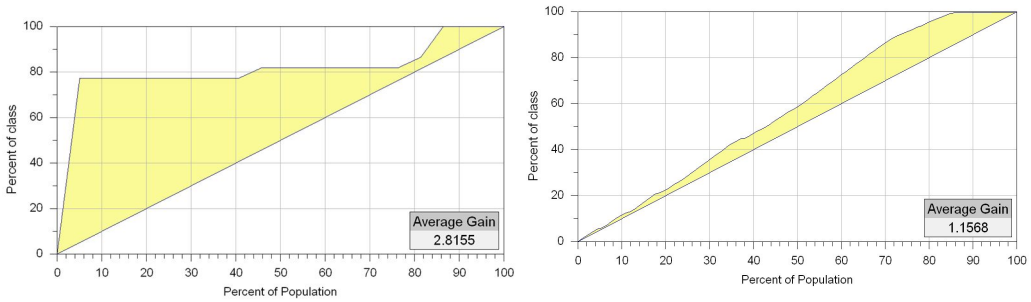
where T_{OK}, T_A, T_B and T_C are the number of samples correctly classified as the classes OK, A, B, and C, respectively, and n is the total number of sample images used in the training and testing sets.

Table 5.2. Performance of the proposed method

T_0 / Accuracy(%)	0.01		0.1		1	
	Train	Test	Train	Test	Train	Test
Without Feature Selection	100	77.53	99.99	77.51	99.99	77.52
With Feature Selection	100	83.13	100	83.13	100	83.13

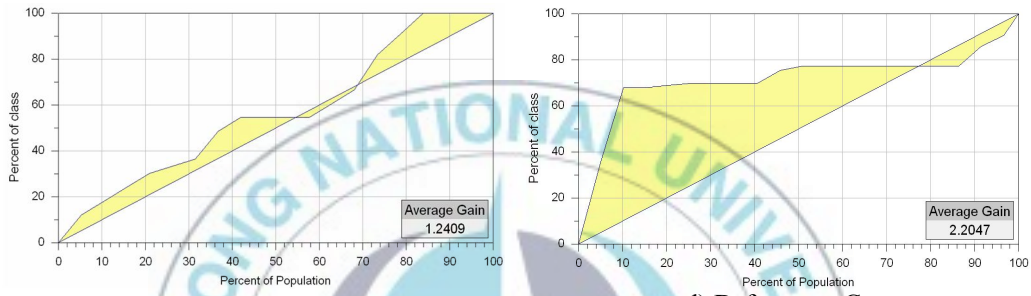
Fig. 5.3 shows the performance of the proposed method in predicting each of the four classes using the gain charts. The lift and gain are useful

tools for measuring the value of a predictive model [44]. The basic idea of lift and gain is to sort the predicted target values in decreasing order of purity based on some target category, and then compare the proportion of cases within the category in each bin with the overall proportion. The sorted row index numbers computed in the previous step are divided into n partitions, where n is the number of bins. In this work, 10 bins were used. The lift and gain values show how much improvement the model provides in picking out the best of the cases. A gain chart displays the cumulative percent of the target value on the vertical axis and the cumulative percent of population on the horizontal axis. Cumulative gain is the ratio of the expected outcome using the model to prioritize the prospects divided by the expected outcome of randomization. The straight, diagonal line shows the expected return if no model is used on the population. The shaded area between the lines shows the improvement (gain) from the model. The gain of 1.00 means that no selective targeting is done. As shown in Fig. 5.3, for the *OK* class, prediction is better by 2.8155 times when using optimized SVM as compared to the expected return of no model or randomization; this is comparable to an average gain of 1.1568, 1.2409, and 2.2047 for defect type *A*, *B*, and *C*, respectively. Fig. 5.3(a) and (d) also show about 80 and 70 percent, respectively, of the samples in the class *OK* and defect type *C* are in the first bin [45].



a) OK

b) Defect type A



c) Defect type B

d) Defect type C

Fig 5.3. Gain charts for classification of four classes of glass substrates using optimized SVM.

5.2. Transmission Images (Experiment II)

5.2.1. Data Preparation and Preprocessing

After applying 4th-level wavelet decomposition on the sub-glass images using the bior1.3 wavelet function (biorthogonal wavelets with a first order reconstruction filter and a third order decomposition filter), a total of 22 GLCM features were extracted from the GLCMs of the wavelet detail sub-images excluding the approximation sub-image. As shown in Fig. 3.1(a), since, the defect type *C* is not captured in the approximation wavelet sub-image, while local lighting non-uniformity is, the approximation wavelet detail is excluded from subsequent feature extraction steps to remove the non-uniform lighting variation in sub-images. Therefore, 12 detailed wavelet

sub-images (4 levels × 3 directions (horizontal, vertical, and diagonal) = 12 sub-images) were used, and two GLCMs (45° and 135°) were calculated from each wavelet sub-image. As for the GLCMs, the values of the design parameters used were $d = 2$, $\theta = 45°$ *and* $135°$, and $G = 32$. In addition to Haralick's original 14 GLCM features, 8 aditional features were extracted. They are Autocorrelation, Cluster prominence, Cluster shade, Dissimilarity, Energy, Homogeneity, Inverse difference and Maximum probability [46,47]. Since 22 GLCM features were calculated from each GLCM, there are a total of 528 wavelet co-occurrence signatures extracted from each sub-glass image that are then used as a feature vector for the image. Therefore, the entire dataset which is extracted from proposed feature extraction methodology (Fig. 3.1(c)) is now an 1182×528 matrix.

Since some of these extracted features are strongly correlated with each other, using a procedure to select a subset or linear combinations of the features can be good [20]. In this work, PCA was used for dimension reduction. In PCA, only the terms corresponding to the K largest eigenvalues are kept. The value of K, the number of components deemed sufficient, is determined based on Eq. 3.6. Fig. 5.4 shows the scree plot from the PCA performed on the data. The value of *threshold* was chosen to be equal to 0.9, and 34 principal components were retained as the features for the three classification methods. In other words, 90% of the variations in the original 528 features are explained by 34 principal components.

Furthermore, since our dataset has 855, 134, 37 and 156 observations in the *OK, A, B* and *C* classes, respectively, we face the issue of an imbalance in the dataset. In this case, we used SMOTE with k=5 as the number of neighbors. After using SMOTE to increase the number of observations in the minority classes, the number of observations in the resulting balanced dataset is 855, 570, 285 and 570 in each class. Therefore, the final dataset is a

2280×34 matrix, which is then randomly divided into a training (70% of the final dataset) dataset and a testing (30% of the final dataset) dataset. After modeling the training data using the CART, optimized MLP and SVM, the performance was verified by classifying the testing dataset.

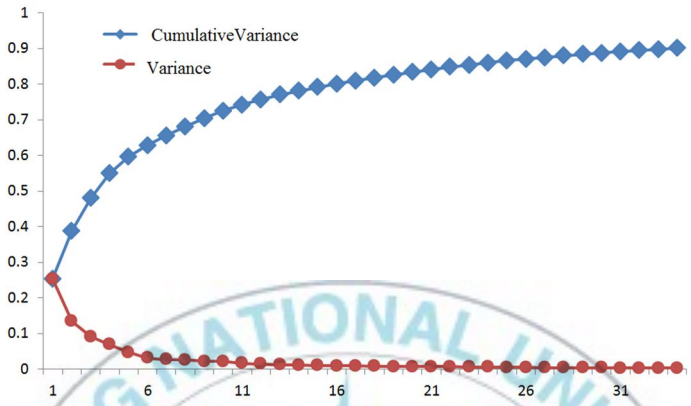


Fig. 5.4. The scree plot from the first 34 principal components of our data.

5.2.2. Results and Discussion

The radial basis function was employed as the kernel function of the SVM, which is inspired by the empirical findings that radial basis kernels tend to give good performance under general smoothness circumstances, especially if additional knowledge of the data is unavailable. Then, SA is carried out to find the optimized value of (C,γ) in the SVM and the number of nodes in each hidden layer of the MLP structure. The Kernel parameter and the penalty parameter for the SVM were found to be 0.8 and 16, respectively.

Shenouda carried out the comparison of different MLP transfer functions in classification problems and showed that the sigmoid transfer function outperforms the others [48]. Therefore, in this work, the sigmoid function was used as a transfer function and 0.001 was chosen as the learning rate for MLP. There are a number of training algorithms used to train a MLP; back

propagation is one of the most important training algorithms that is used frequently for obtaining the weights and bias [49,50]. Furthermore, by trial and error, we achieved the MLP with 2 hidden layers, 39 neurons in the first hidden layer and 26 neurons in the second hidden layer, by using SA.

The corresponding MLP and SVM parameters for the classification of four different defects on TFT-LCD glass substrates are summarized in Table 5.3 and 5.4. CART structures display a hierarchical system of decision rules for classifying the objects according to the features of the predictor variables. Table 5.5 summarizes the rules and the classified results from the tree built from the data. Table 5.5 also indicates that the third principal component (PC3) plays the most important role in the rule induction. According to Table 5.5, we can see that if an observation whose *PC3*, *PC6* and *PC16* values are greater than 0.477, -0.409 and -0.402, respectively, it falls into the terminal node whose classified class is *OK*.

The results obtained from the proposed CART, MLP and SVM models indicate that the optimized SVM model with the SA algorithm is better for classifying TFT-LCD glass substrates than the CART and MLP models. However, we have a new situation when the imbalanced dataset was used. With the imbalanced dataset, MLP outperforms the support vector machine; however these results in Table 5.6 substantiate earlier findings that SVM performs better than MLP when balanced dataset are used. The CART had the lowest accuracy of 69.4% and 67.2% for imbalanced and balanced data, respectively, followed by the proposed MLP of 78.6%, 87.9% and SVM of 74.5%, 89.5%.

Tables 5.7, 5.8 and 5.9 show the confusion matrices for the proposed methods. They show the number of cases classified correctly in each class by individual method for the balanced data. For example, Table 5.9 shows the confusion matrix for the optimized SVM model that is applied to the testing

dataset consisting of 717 instances that are roughly split evenly between the four classes. We can see from this Table that this method correctly classified 215 sub-glasses in the *OK* class, 167, 86 and 174 sub-glasses in the defect types *A, B* and *C* classes, respectively. This is the best accuracy when compared with the others.

Table 5.3. The optimal values of the MLP parameters resulting from the simulated annealing algorithm.

MLP Parameters	
No. of neurons in first hidden layer	39
No. of neurons in second hidden layer	26

Table 5.4. The optimal values of the SVM parameters resulting from the simulated annealing algorithm.

SVM Parameters	
kernel parameter (γ)	0.8
penalty parameter (C)	16.0

Table 5.5. CART analytic rules

Rule	Class
PC3 <= 0.477	
PC8 <= 0.357	
PC2 <= -0.125	
PC6 <= 0.743	
PC2 <= -1.712	C
PC2 > -1.712	B
PC6 > 0.743	
PC1 <= -1.166	A

PC1 > -1.166	OK
PC2 > -0.125	
PC4 <= 0.031	
PC8 <= -0.730	C
PC8 > -0.730	OK
PC4 > 0.031	
PC5 <= 1.165	C
PC5 > 1.165	OK
PC8 > 0.357	
PC11 <= -0.890	OK
PC11 > -0.890	
PC12 <= -0.801	OK
PC12 > -0.801	A
PC3 > 0.477	
PC6 <= -0.409	
PC6 <= -2.356	B
PC6 > -2.356	
PC33 <= 0.877	
PC4 <= 0.608	A
PC4 > 0.608	C
PC33 > 0.877	OK
PC6 > -0.409	
PC16 <= -0.402	
PC3 <= 1.073	
PC14 <= 0.303	A
PC14 > 0.303	OK
PC3 > 1.073	OK
PC16 > -0.402	OK

Table 5.6. Accuracy of CART, optimized MLP and SVM for both imbalanced and balanced data.

Accuracy (%)	Imbalanced Data		Balanced Data	
	Train	Test	Train	Test
CART	80.6	69.4	75.1	67.2
Optimized MLP	79.1	78.6	88.4	87.9
Optimized SVM	99.3	74.5	99.7	89.5

Table 5.7. Confusion matrix of CART model for the testing dataset.

		Classification Result				
		OK	**A**	**B**	**C**	**Total**
CART	**OK**	**158**	48	12	30	248
	A	57	**120**	9	6	192
	B	15	12	**53**	9	89
	C	14	19	4	**151**	188
Total						**717**

Table 5.8. Confusion matrix of proposed MLP model for the testing dataset.

		Classification Result				
		OK	**A**	**B**	**C**	**Total**
Optimized MLP	**OK**	**208**	24	11	5	248
	A	24	**167**	1	0	192
	B	4	2	**82**	1	89
	C	9	2	4	**173**	188
Total						**717**

Table 5.9. Confusion matrix of proposed SVM model for the testing dataset.

		Classification Result				
		OK	**A**	**B**	**C**	**Total**
Optimized SVM	**OK**	**215**	24	2	7	248
	A	21	**167**	0	4	192
	B	3	0	**86**	0	89
	C	10	1	3	**174**	188
Total						**717**

The CART analysis has a number of advantages over the other classification methods. One is that they are scalable for large problems and can handle smaller data sets than NN models [51]. The other advantage is that it is inherently non-parametric. In other words, no hypotheses are made regarding the underlying distribution of values of the features. Furthermore, CART can handle categorical features with either ordinal or non-ordinal structure. Thus, the use of this model became popular. On the other hand, the need for pruning in the decision tree algorithms makes CART less attractive.

Neural networks and SVMs show more potential for multi-classification problems. Brown worked on multimodal classification problems where the data sets are large with few attributes and showed that NNs do better than CART models [52]. Also, in a previous work by Burbidge *et al.* [53], a SVM classifier outperformed other standard machine learning methods. One of the most important disadvantages of these models is that there is no structured way to choose the optimal parameters of the MLP and SVM. In this study, the parameters of these algorithms are determined using simulated annealing, which produce nearly optimal results.

The comparison of the obtained results from the proposed SVM, MLP and CART models using balanced data indicated that the MLP and SVM models were, in general, better for the classification of defects than the CART model. This might be due to the stochastic nature of the defects in the images. On the other hand, the proposed MLP models in the current study were more effective in using the balanced data for defect detection of TFT-LCD glass substrates than the SVM model when the evaluation criteria are compared. However, the accuracy of the MLP model using balanced data was not considerably higher than the proposed SVM model (see Table 5.6).

5.3. Reflection Images

5.3.1. Data Preparation and Preprocessing

In this study, 4540 sub-glass images were taken by the installed cameras above the conveyor belt in the cold process and were labeled by expert inspectors as *OK* or *NG*. As mentioned in section 2.3, since each 20 sub-glasses belong to one mother glass. 227 mother glasses are considered in our study. The data set has 535 and 92 defective sub-glasses and mother glasses, respectively. Therefore, the ratios of class imbalance are approximately 0.1 and 0.4 for sub-glass and mother glass, respectively, which the ratio for the mother glass is not as much high as the ratio for the sub-glass. For testing our proposed algorithm, a total of 40 mother glasses including 17 defective glasses, were selected as a testing data set and the remaining were used to train the classifiers.

In this work, 4th-level wavelet decomposition was applied by the bior1.3 wavelet function (biorthogonal wavelets with a first order reconstruction filter and a third order decomposition filter). Then, GLCM features were extracted from the GLCMs of the wavelet detail sub-images. As shown in Fig. 3.1(b), since, the waviness is not captured in the approximation wavelet sub-

image, while image features such as conveyor belt marks, guide bar marks, (some of) water marks are. Therefore, approximation coefficients which mostly contain lighting variation across the glass were excluded in calculating GLCMs and recovering features. Furthermore, as shown in Fig. 2.10, the waviness is only in the horizontal direction in sub-images, in other word, this direction contains most of information related to waviness. Therefore, 4 horizontal wavelet coefficients were extracted and only one GLCM (90°) was calculated from each wavelet sub-image. Finally, 8 features, including: (1) Autocorrelation, (2) Cluster prominence, (3) Cluster shade, (4) Dissimilarity, (5) Energy, (6) Homogeneity, (7) Inverse difference, (8) Maximum probability, in addition to the 14 GLCM features which were defined by Haralick [45,46] were calculated from each GLCM with the parameters $d = 2$, $\theta = 90°$ and $G = 32$. Therefore, there are a total of 4×22=88 wavelet co-occurrence signatures extracted from one sub-glass image. After using the wavelet co-occurrence signature, the entire dataset is a 4540 × 88 matrix.

5.3.2. Results and Discussion

According to the imaging process for reflection images, each mother glass was divided into 20 sub-glasses. Therefore, the important points for manufactures are indicating the defective mother glasses and finding the location of these defects; both mother glass and sub-glass accuracy should be in our decision criterion. Therefore, the decision criterion in this part is defined as follows:

$$DC = \frac{MA + SA}{2} \tag{5.2}$$

where MA and SA are mother glass accuracy and sub-glass accuracy, respectively.

$$MA = \frac{T_{OK}^{M} + T_{NG}^{M}}{n^{M}} \tag{5.3}$$

$$SA = \frac{T_{OK}^{S} + T_{NG}^{S}}{n^{S}} \tag{5.4}$$

where T_{OK}^{M}, T_{NG}^{M}, T_{OK}^{S} and T_{NG}^{S} are the number of qualified (*OK*) and defective (*NG*) mother glasses and sub-glasses which are correctly classified, respectively. Moreover, n^{M} and n^{S} are the total number of mother glass and sub-glass samples. Also, for calculating *MA*, each mother glass with at least one defective sub-glass is labeled as an *NG* glass in this work.

In this study, C5.0 with a variety of cost matrices was investigated as classifiers for the ensemble technique. C5.0 is a cost-sensitive learning algorithm which considers the cost information when building and pruning the induced decision tree. Due to the high ratio of class imbalance for sub-glass, using the more cost for false positive can be helpful to get the high accuracy in sub-glass. On the other hand, since each mother glass with at least one defective sub-glass is labeled as an *NG* glass, we can put more importance in *NG* class and also have the high mother glass accuracy by using the more cost for false negative. Therefore, we evaluate more than one cost ratio; this actually increases the generality of our results. As shown in Tables 5.10, 5.11 and 5.12, for our experiments, a false positive prediction, C_{FP}, and false negative prediction for the majority of experiments, C_{FN}, is evaluated for values of 1 and 2 in the training the decision tree. The confusion matrices of the sub-glass and mother glass for all three C5.0 classifiers with different cost matrices are shown in Tables 5.13 through 5.18. The effect of each cost matrix on classifying the observations into each class based on its own defined cost can easily be concluded from these tables. In the confusion matrix with cost matrix 1 (Tables 5.13, 5.14), all of the observations in the majority class are truly classified; moreover, the mother

glass accuracy sharply decreased in the case of using cost matrix 3 as the classifier (Fig. 5.5). Therefore, we can conclude that a cost value equal to 2 can be the maximum value for the costs in this problem. Table 5.19 shows the mother glass accuracy (*MA*) and sub-glass accuracy (*SA*) for each classifier on the testing data. As one can see from Table 5.13 through 5.19, increasing the false negative cost, C_{FN}, causes the sub-glass and mother glass accuracy to decrease and increase, respectively. On the other hand, increasing the false positive cost, C_{FP}, gives us a higher *SA* and lower *MA*.

Table 5.10. Cost matrix 1.

Cost Matrix	OK	NG
OK	0	2
NG	1	0

Table 5.11. Cost matrix 2.

Cost Matrix	OK	NG
OK	0	1
NG	1	0

Table 5.12. Cost matrix 3.

Cost Matrix	OK	NG
OK	0	1
NG	2	0

Table 5.13. Sub-glass confusion matrix for cost matrix 1on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	594	0
	NG	132	74

Table 5.14. Mother glass confusion matrix for cost matrix 1 on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	23	0
	NG	8	9

Table 5.15. Sub-glass confusion matrix for cost matrix 2 on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	578	16
	NG	105	101

Table 5.16. Mother glass confusion matrix for cost matrix 2 on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	18	5
	NG	4	13

Table 5.17. Sub-glass confusion matrix for cost matrix 3 on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	550	44
	NG	72	134

Table 5.18. Mother glass confusion matrix for cost matrix 3 on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	10	13
	NG	3	14

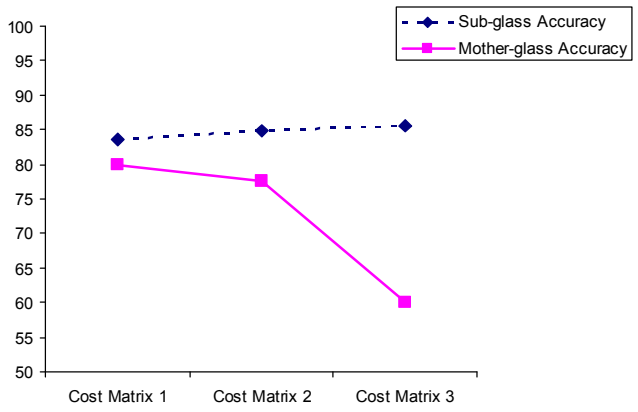


Fig. 5.5. Sub-glass and mother glass accuracy for C5.0 classifier with different cost matrices.

Table 5.19. Comparison between the accuracy of classifiers with different cost matrices.

		Cost1	Cost2	Cost3	Ensemble
ACCUURACY (%)	SA	83.5	84.88	85.5	86.37
	MA	80	77.5	60	82.5
	DF	81.5	81.19	72.75	84.44

Because we are interested in the case where both the sub-glass and mother glass accuracies are high, a combination of all three classifiers can be a good way to reach this goal. On the other hand, time is an important parameter in these systems; therefore, an ensemble technique, which is a simple voting scheme, was selected to use the three classifiers results. Although, it is expected that the performance can be improved if more learners are included, the ensembles studied in this work contain only three component learners in order to control the computational cost of system.

Tables 5.20, 5.21 show the sub-glass and mother glass confusion matrix for our proposed ensemble method on the testing data. As shown in Table 5.21, from the 23 total defective mother glasses, 21 glasses were classified correctly as the *OK* class and just 5 mother glasses in the *NG* class were classified as qualified glasses. Table 10 also shows *SA*, *MA* and *DF* using

proposed ensemble technique are 86.37%, 82.5% and 84.44%, respectively. Therefore, this discloses that threshold-moving and the ensemble methods could be a viable alternative for making an inline automatic inspection system to detect the defects on the surface of TFT-LCD glass substrates.

Table 5.20. Sub-glass confusion matrix for proposed ensemble method on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	585	9
	NG	100	106

Table 5.21. Mother glass confusion matrix for proposed ensemble method on testing data.

		PREDICTED	
		OK	NG
ACTUAL	OK	21	2
	NG	5	12

6. Conclusion

This study has successfully developed an automatic inspection system for detecting surface flaws or scratches and surface waviness on produced glass substrates in a cold process of sheet glass manufacturing. The novelties of this system are: (1) using the special imaging process and two types of imaging, transmission and reflection, to show the different defects (2) separation of the defects from the other marks on the images through the proposed feature extraction method, and (3) detection of defects with high performance and reasonable complexity by using the proposed classification technique.

This study has practical meaning because using feature selection methods enables higher classification accuracy, as well as shorter training

and testing time. In addition, the use of SA in finding the best parameters for classifiers provides optimal classification performance. In fact, this system can be operated in manufacturing to decrease the time and cost to train the human inspectors and increase the defect detection accuracy by removing the possibility of wrong judgments from the human inspectors. The experimental results using images from real production lines show that the proposed method can provide competitive performance for detecting different types of surface defects.

References

[1] P. Chapman (NCR), J. Clinton (SPSS), R. Kerber (NCR), T. Khabaza (SPSS), T. Reinartz, (DaimlerChrysler), C. Shearer (SPSS) and R. Wirth (DaimlerChrysler) "CRISP-DM 1.0 - Step-by-step data mining guide".

[2] L.A.B. Pilkington, (1969). The float glass process. Proc. of the Roy. Soc. of Lon. A., 314, 1-25.

[3] S. M. Dockerty, Sheet Forming Apparatus. United States Patent, 3,338,696, 1967.

[4] M. Akilian, "Methods of Improving the Surface Flatness of Thin Glass Sheets and Silicon Wafers," in MIT PhD Thesis, USA, 2008.

[5] S. Gunasekaran, "Computer vision technology for food quality assurance," Trends Food Sci. Tech., vol. 7, no. 8, pp. 245-256, 1996.

[6] E. Malamas, E. Petrakis, M. Zervakis, L. Petit, and J. D. Legat, "A survey on industrial vision systems applications and tools," Image Vis. Comput., vol. 21, no. 2, pp. 171-188, 2003.

[7] A. Mital, M. Govindaraju, and B. Subramani, "A comparison between manual and hybrid methods in parts inspection," Integ. Manuf. Syst., vol. 9, no. 6, pp. 344-349, 1998.

[8] J.J. Liu, J.F. MacGregor, C. Duchesne, G. Bartolacci, “Flotation froth monitoring using multiresolutional multivariate image analysis,” Miner. Eng., vol. 18, pp. 65–76, 2005.

[9] J. M. Parker, and Z. Hou, “A numerical investigation of diffuse images for effective defect detection,” in Proc. Institution of Mechanical Engineers, Part B. J. of Eng. Manuf., vol. 216, no. 7, pp. 1073-1079, 2002.

[10] A. Conci, and C. Belmiro, “A system for real-time fabric inspection and industrial decision,” Proc. 14th Int. Conf. Software engineering and knowledge engineering, pp. 707–714, 2002.

[11] M. Bharati, J. Liu, J.F. MacGregor, “Image texture analysis: methods and comparisons,” Chemometr. Intell. Lab., vol. 72, pp. 57–71, 2004.

[12] T. Randen, J.H. Husoy, “Filtering for texture classification: a comparative study,” IEEE Trans. Pattern Anal. Machine Intell. vol. 21, pp. 291–310, 1999.

[13] J. C. Lapp, (1994). Advanced glass substrates for flat panel displays. Proc. of SPIE, 2174, 129-138.

[14] K. Freischlad, (1996). Large flat panel profiler. Proc. of SPIE, 2862, 163–171.

[15] R.H. Mauch, (2000). Thin glass substrates for mobile applications. Proc. of SPIE, 4102, 162–168.

[16] H. Zhu, Q. Lin, B. Zhang, “Analysis of system error in the measurement of liquid crystal empty cell gap by means of interferometry,” Disp., vol. 21, pp. 121, 2000.

[17] M. Tuceryan, A. K. Jain, (1998). Texture analysis, in: Chen, C.H., Pau, L.F., Wang, P.S.P. (Eds), The Handbook of Pattern Recognition and Computer Vision (2nd edition), Wold Scientific publishing Co.

[18] J.J. Liu, C. Han, “Multivariate image analysis in the process industries: A review,” Chemo. and Int. Lab. Sys., 2012, in press.

[19] Van De Wouwer, G. Wavelets for multiscale texture analysis; University of Antwerp Ph.D Thesis, Belgium, 1998.

[20] R. M. Haralick, K. Shanmugam, “Dinstein, I. Textural features for image classification,” IEEE Trans. on Sys., Man, and Cyber., vol. 3, pp. 610, 1973.

[21] G. Van De Wouwer, P. Scheunders, D. Van Dyck, Statistical Texture Characterization from Discrete Wavelet Representations, IEEE Trans. Image Process. vol. 8, pp. 592-598, 1999.

[22] S.W. Myint, Fractal approaches in texture analysis and classification of remotely sensed data: comparisons with spatial autocorrelation techniques and simple descriptive statistics, Int. J. Remote Sens. vol. 24, pp. 1925-1947, 2003.

[23] C. Duchesne, J.J. Liu, J.F. MacGregor, Multivariate image analysis in the process industries: A review, Chemometr. Intell. Lab. vol. 117, pp. 116–128, 2012.

[24] John, G.H., R. Kohavi and K. Pfleger (1994). Irrelevant feature and the subset selection problem. Proc. of the Eleventh Int. Conf. on Mach. Learn., pp. 121-129.

[25] Koller, D.; Sahami, M. (1996). Toward Optimal Feature Selection. Proc. of the Thirteenth Int. Conf. on Mach. Learn., pp. 284-292.

[26] Zhu, M.; Chipman, H. A. Darwinian evolution in parallel universes: a parallel genetic algorithm for variable selection. Tech. vol. 48, no.4, pp. 491, 2006.

[27] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, J. AI Res. vol. 16, pp. 341-378, 2002.

[28] T.M. Ha, H. Bunke, Off-Line, Handwritten numeral recognition by perturbation method, IEEE Trans. Pattern Anal. Mach. Intell. vol. 19, pp. 535-539, 1997.

[29] H. Wu, J. Zelek,” A Multi-classifier Based Real-time Face Detection System”, Journal of IEEE Transaction on Robotics and Automation, 2003.

[30] Vapnik, V. Statistical Learning Theory; Wiley: New York, 1998.

[31] Cristianini, N.; Taylor, J. S. An Introduction to support vector machines and other kernel-based learning methods; Cambridge University Press: New York. 2000.

[32] Li, H.; Liang, Y.; Xu, Q. (2009). Support vector machines and its applications in chemistry. Chemo. and Int. Lab. Sys., vol. 95, pp. 188, 2009.

[33] C. Scholkopf, J. Burges, A. Smola, Advances in Kernel Methods: Support Vector Learning, MIT Press: Cambridge, 1999.

[34] X. Zhang, Y. Guo, Optimization of SVM parameters based on PSO algorithm, Proc. Fifth Int. Conf. Nat. Comput. vol. 1, pp. 536-539, 2009.

[35] R. S. Sexton, R. E. Dorsey, J. D. Johnson, Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing, Eur. J. Oper. Res. vol. 114, pp. 589-601, 1999.

[36] C.L. Chang, C.H. Chen, Applying decision tree and neural network to increase quality of dermatologic diagnosis, Expert Syst. Appl. vol. 36, pp. 4035–4041, 2009.

[37] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Inc., Monterey, California, 1984.

[38] Z. H. Zhou, and X. Y. Liu, “Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem,” IEEE Trans. Knowledge and Data Engineering, pp. 63-77, 2005.

[39] M. Kukar, and I. Kononenko, “Cost-sensitive learning with neural networks,” in Proc. 13th European Conf. Artificial Intelligence, Brighton, UK, 1998, pp.445–449.

[40] F. Provost, “Machine learning from imbalanced data sets 101,” in Working Notes of the AAAI’00 Workshop on Learning from Imbalanced Data Sets, Austin, TX, 2000, pp.1–3.

[41] M. A. Maloof, “Learning when data sets are imbalanced and when costs are unequal and unknown,” in Working Notes of the ICML’03 Workshop on Learning from Imbalanced Data Sets, Washington, DC, 2003.

[42] T. G. Dietterich, “Ensemble learning,” in The Handbook of Brain Theory and Neural Networks, 2nd edition, M.A. Arbib, Ed. Cambridge, MA: MIT Press, 2002.

[43] Z. H. Zhou, X. Y. Liu, “Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem,” IEEE Trans. Knowledge Data Eng. vol. 18, no. 1, pp. 63-77, 2006.

[44] Abdelaal, M. M.; Farouq, M. W. (2010). Using data mining for assessing diagnosis of breast cancer. Proc. of Int. Multi Conf. on Comp. Sci. and Inf. Tech., 11-17.

[45] A.Yousefian Jazi, S. Yoon, J.J. Liu, Automatic grading of TFT-LCD glass substrates using optimized support vector machines, Ind. Eng. Chem. Res. vol. 51, pp. 10887–10894, 2012.

[46] L. Soh, C. Tsatsoulis, Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices, IEEE Trans. Geosci. Remote Sens. vol. 37, pp. 780-795, 1999.

[47] D.A. Clausi, An analysis of co-occurrence texture statistics as a function of grey level quantization, Can. J. Remote Sens. vol. 28, pp. 45-62, 2002.

[48] E.A.M.A. Shenouda, A Quantitative Comparison of Different MLP Activation Functions in Classification, Lect. Notes Comput. Sc. vol. 3971, pp. 849-857, 2006.

[49] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representation by error propagation, MIT Press, Cambridge, MA, 1986, Chapter 8.

[50] I.A. Basheer, M. Hajmeer, Artificial neural networks: fundamentals, computing, design, and application, J. Microb. Methods, vol. 43, pp. 3-31, 2000.

[51] I.S. Markham, R.G. Mathieu, B.A. Wray, Kanban setting through artificial intelligence: A comparative study of artificial neural networks and decision threes, Integ. Manuf. vol. 11, pp. 239-246, 2000.

[52] D.E. Brown, V. Corruble, C.L. Pittard, A comparison of decision tree classifier with backpropagation neural networks for multimodal classification problems, Pattern Recogn., vol. 26, pp. 953– 961, 1993.

[53] R. Burbidge, M. Trotter, B. Buxton, S. Holden, Drug design by machine learning: support vector machines for pharmaceutical data analysis, Comput. Chem. vol. 26, pp. 5-14, 2001.