On-line Multi-stage Classifier for Agricultural Sorting Systems

Thesis submitted in partial fulfillment
of the requirements for the degree of
”Doctor of Philosophy”

by
Shahar Laykin

Submitted to the Senate of Ben-Gurion University
of the Negev

2006 תשס״ז

BEER - SHEVA

On-line Multi-stage Classifier for Agricultural Sorting Systems

Thesis submitted in partial fulfillment
of the requirements for the degree of
”Doctor of Philosophy”

by

Shahar Laykin

Submitted to the Senate of Ben-Gurion University
of the Negev

Approved by the advisor	Prof. Yael Edan	_____
Approved by the advisor	Dr. Victor Alchanatis	_____
Approved by the Dean of The Kreitman School of Advanced Graduate Studies		_____

2006 תשס״ז

BEER – SHEVA

This work was carried out under the supervision of

Prof. Yael Edan & Dr. Victor Alchanatis

In the Department of Industrial Engineering and Management

Faculty of Engineering Sciences

This thesis is dedicated in the memory of my late beloved sister

OSNAT LAVI

who passed away on 15/5/2007 after a long courageous and heroic fight with the breast cancer disease.

MAY SHE REST IN PEACE

To ***Ayelet*** my beloved wife

Acknowledgments

To my advisor and my friend, **Prof. Yael Edan**, thanks for the opportunity to make this research, for the guide, help and long and fruitful discussions that were so essential for my work.

To **Dr. Victor Alchantis**, my advisor and friend, thanks for all the fruitful discussions, and for sharing your huge knowledge in the agricultural aspect of this thesis.

Special thanks for **Ofir Cohen** my ***very best friend*** with whom I shared great years of discussions, thoughts, ideas, and lots of laugh.

This project was partially supported by the the Chief Scientist, Ministry of Agriculture Fund No. 463-0130,by the Paul Ivanier Center for Robotics Research & Production Management, and by the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering, Ben-Gurion University of the Negev.

To **Dr. Israel Parmet** who significantly contributed to the statistical analyses and to **Dr. Ze'ev Weismann** who drove me into the olive fields and thereby introduced me into a new domain which enabled me to obtain essential data to complete my thesis.

To **Tali Lavie**, **Peter Bak**, **Yoash Chassidim**, **Juan Wax, Avital Bechar,** my dear colleagues at the IEM department who contributed in ideas and in great times throughout the period of this research.

To **Ido Cohen** and **Yaniv Moskovich,** thank you for your assistance in running the olives panel evaluation experiments.

To **Ami Eliav** , **Nissim Abuhazira, Donny Maresse, Dror Shriki**, , **Ittai Kenet**, **Ran Hessel**, **Shahar Matza**, **Sigal Berman**, **Yossi Zahavi, Rubi Gertner, Uri Kartoun**, **Victor Livshitz**, **Gil Shayer**, **Greg Piltz** and **Yechiel Glickman** who have worked with me in the CIM lab during these wonderful years, the CIM lab has been a fun place to be because of y'all.

I'd like to thank my dear family (Laykin & Lavi), **Pinhas**, **Yardena**, **Moty** and **Osnat**. They have always provided me with love, support, encouragement, and laughter. They are the best parents, brother and sister that I could have ever hoped for.

To **Bella** and **Shlomo Saad**, for their encouragement and huge help with babysitting Shira especially in this last exhausted year.

Finally, I would like to thank my wife **Ayelet**, for her **infinite** patience and love, which gave me encouragement and motivation to pursue and finish this thesis. Last but not least, many thanks to **Shira**, my beautiful daughter, for being so amazingly sweet.

Shahar Laykin
Ben-Gurion University of the Negev
Beer Sheva, 2006

Table of Contents

List of Figures

List of Tables

Acronyms[1]

RBFL	Rule Based Fuzzy Logic
kNN	K Nearest Neighbors algorithm
FkNN	Fuzzy k Nearest Neighbors algorithm
PCA	Principal Component Analysis
ART	Adaptive Resonance Theory
PFAM	Probabilistic Fuzzy Adaptive resonance Theory
PDF	Probability Distribution function
SOM	Self-organizing map
BPANN	Back propagation neural network
FL	Fuzzy Logic
ANN	Adaptive Neural Network
NIR	Near infrared
NMR	Nuclear Magnetic Resonance
DT	Decision Tree
HIS	Hue, Intensity, Saturation
CART	Classification and Regression Tree
FCM	Fuzzy C-Mean Clustering
SVM	Support Vector Machine
MCS	Multi Classifier Systems
SFS	Stepwise Forward Selection
KS	Kalmogorov-Smirnov
FIS	Fuzzy Inference System
FIFO	First In First Out
MSPE	Mean Square Precision Error

[1] To improve the flow of the reading the abbreviations were not always used.

Abstract

This work deals with classification of agricultural produce. In agriculture, quality sorting of produce is based on a multitude of features. To define the quality of the product, a feature vector that includes all features must be derived. After extracting the feature vector of the produce, a classifier must be designed to classify the fruit into its quality grade.

In many cases, some of the produce's features are irrelevant for the classification task. These features not only cause slow classification, they also reduce performance. Furthermore, some features might appear or disappear overnight. Therefore, it is important to define the actual features employed in the classification process based on the immediate situation. In addition, the characteristics of the product change slowly with time as the fruit or vegetable ages and the season advances. If a classifier can track this change on-line, improved sorting performance can be achieved.

Extensive research has been conducted in feature selection to improve classifier performance. Due to the variability of the agricultural produce it is important to conduct feature selection on-line to enable the classifier to adapt to changes.

This work aims to provide an efficient method for solving the diversity problems in sorting agricultural produce by developing an on-line hierarchical classifier with the capability of adapting to different populations.

Methodology

Population detection is conducted by analyzing the product variability. The main idea is to check whether the current stream of produce is different from the previous one. To detect population change, an on-line unsupervised clustering algorithm was developed. When the algorithm detects a new population, it compares the *history* population overlap level with the current population. Accordingly, it decides whether to use a previous population classifier or to select a new one.

After a new population is detected, a classifier is selected for it according to the overlap level. The classifier is selected from *n* fuzzy *K*-Nearest Neighbors classifiers, each trained with a different number of features. A fuzzy logic rule-based decision system was developed to fuse the classifier results and to select the best one according to weighted feedback.

When overlapping is high, the corresponding *history* population classifier was retrieved for the current population. When overlapping is low, retraining is either human or automatic. The

human retraining procedure was defined when there is no overlap, i.e., a predefined training set for the current population is used for off-line retraining of the system. The automatic retraining procedure is applied on-line for those cases in which classifiers are retrained with the data points in the overlapping region of the *history* populations These data points are already labeled with their population classifier.

The classifier was tested with a specially designed synthetic database. To test the classifier in real world conditions and to create a well-defined database for the classification problem, a specially designated crop of olives was harvested and analyzed.

A cost analysis was developed to evaluate classifier performance in addition to classification accuracy. A cost function based on the computational cost of the features used by each classifier and its error significance was also developed. Three parameters were defined to test the performance of the classification system.

Analysis and Results

Synthetic data

The synthetic dataset was composed of six different populations. Each population contained 1000 data points with seven features. Each feature was created with a random multivariate normal distribution. Results indicate that the overall classification accuracy of the on-line classifier is better by 12% as compared to the kNN classifier that used all the features.

Time series analysis indicated system flexibility and the capability to adjust to the new populations entering the system. Sensitivity analyses indicated that the population entrance sequence has major influence on the system. Furthermore, when a new population has several overlapping populations, the problem becomes a question of which population training data to use. This pointed out the need to predefine the *population database* so that it includes populations that cover most of the feature space of the produce. In this way each population will be assigned the appropriate training set or classifier.

Agricultural data

The olives database contained 12 varieties of 10,550 olives harvested from Ramat-Negev fields in the south of Israel. A full on-line run was applied on the data resulting in an accuracy of 81%. Nevertheless, results indicate that in 13 cases (out of 21) human retrain was required. This strengthens the need for a *population database* and the improved similarity measures. The *population database* is defined a-priori using the knowledge of the overlap and similarity levels between the populations.

When using the *population database* the classifier yields higher classification accuracy performance compared to non-adaptive classifiers. The mean square precision error (MSPE) indicates this difference (0.0346 for the adaptive classifier vs. 0.2943 for the non-adaptive classifier). When compared with optimal classifiers (i.e., using all features and trained by all populations) the system yielded lower but still quite good results (85% vs. 89%; 0.0346 vs. 0.0311).

The sensitivity analysis implies that changing batch size parameters has a major influence on the system performance as well as the cost. Several parameter combinations resulted in better performance than the best that was defined off-line (0.0188 vs. 0.0286). In addition, changing the cost function characteristics – feature cost and penalty matrix cost – resulted in corresponding changes in the cost behavior.

Summary

The main contributions of the proposed classifier are: efficient detection of a new population; rapid adjustment to this population in terms of overlap and similarity measures; online feature and classifier selection via the adjustment procedure; and a cost objective function that, together with classification accuracy, tests the system performance.

The proposed on-line adaptive classifier framework selects online the most appropriate classifier and feature subsets for the incoming population. The chief benefit of our system is its ability to adapt to new populations based on previous ones using similarity measures. This ability makes it possible to decide on a classification strategy without having to train on a specific population, an approach that makes the framework more flexible to changes in the population. The capability of selecting the best feature set results in improved classification performance and lowered costs.

Keywords: Agriculture sorting systems, similarity measures, change detection, classifier selection, fuzzy rule based system, image processing, machine vision, feature selection

This thesis is in part based on the following publications:

Journal Papers

1. Laykin, S., V. Alchanatis, E. Fallik, Y. Edan. 2002. Image processing algorithms for tomato classification. Transactions of the ASAE 45(3): 851-858.

Reviewed Conference Papers

1. Laykin, S., V. Alchanatis, Y. Edan. 2000. Image processing algorithms for tomatoes classification. Proceedings of the XIV Memorial CIGR World Conference: 861-866, Tsukuba, Japan.
2. Laykin, S., V. Alchanatis, Y. Edan. 2003. On-line Hierarchical Classifier for Agricultural Sorting Systems. Paper Code 3044. ISCA 12th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE-2003), July 2003, San Francisco, USA.
3. S. Laykin, Y. Edan, and V. Alchanatis . 2004. On-line feature and classifier selection for agricultural produce. Paper Code 451-151, Proceedings of the Eighth IASTED International Conference on Artificial Intelligence and Soft Computing, September 1 – 3, 2004, Marbella, Spain.

Conference Papers

4. Laykin, S., Y. Edan, V. Alchanatis, R. Regev, F. Gross, J. Grinshpun, E. Bar-Lev, E. Fallik, S. Alkalai. 1999. Development of a quality sorting system using machine vision and impact. ASAE Paper No. 99-3144, ASAE St. Joseph, MI 49085.
5. Laykin, S., V. Alchanatis, Y. Edan. 2003. Classifier Selection for Agricultural Quality Sorting. ASAE Paper No. 03-3050, ASAE, St. Joseph, MI 49085.

1 Introduction

1.1 Description of the problem

In agriculture, quality sorting of produce is based on a multitude of features (Dull, 1986): flavor (sweetness, acidity); appearance (color, size, shape, blemishes, glossiness); and texture (firmness, mouthfeel). To measure these characteristics as efficiently and accurately as possible, appropriate sensors and algorithms must be developed focusing on a single or several quality features. To define the quality of the product, a feature vector that includes all features must be derived. Every produce has its unique feature space. After extracting the feature vector of the produce, a classifier must be designed to classify the fruit into its quality grade.

Previous research (Edan *et al*., 1994) indicated that multi-sensor quality classification improves the sorting performance. In many cases, some of the features are irrelevant for the classification task. These features not only slow classification, they also reduce performance. For example, the significance of the shape in tomato sorting is low for uniform-shaped tomatoes and high for non-uniform. In addition, some fruit features might appear overnight. For example, fruits that are damaged by hail or cold can suddenly appear and then disappear from the sorting stream depending on the weather. Therefore, it is important to define the actual features employed in the classification process based on the immediate situation.

The characteristics of the product change slowly with time as the fruit ages and the season advances. If a classifier can track this change on-line, improved performance can be achieved (Duda *et al*., 2001). This approach is similar to novelty detection, an area of research which aims to update the classifier's ability to detect whether an input is part of the data it was trained with or it is in fact unknown (Markou and Singh, 2003).

Extensive research has been conducted into feature selection to improve classifier performance (Dash and Liu, 1997, Yu *et al*., 2002, Zhang *et al*., 2004). Currently, most feature selection methods are applied off-line, before classification (Collins *et al*., 2005). However, due to the variability of the agricultural produce, which occurs on-line, feature selection should also be conducted on-line to enable the classifier to adapt to changes. In this case, the problem becomes one of how to recognize that a new population (a batch of fruits harvested

from the same field at a certain date) has arrived (Guedalia *et al*., 1999) and to adjust a new subset of features for the classification stage. For example, in case of different stages of ripeness, the system should be able to detect the changes and select a different feature space. When a new population arrives, a different classifier for it may be necessary.

Creating classifiers involves learning and adaptation procedures. These procedures enable the classifier to adjust to different produce features. The learning procedure uses training patterns (training sets) to learn or estimate the unknown parameter of the classifier (Duda *et al*., 2001). There are two types of learning procedures: supervised and unsupervised. In supervised learning, there is a specified set of classes and training sets of produce, each labeled with the appropriate class. The goal is to classify the new objects into one of the classes based upon the training objects. In unsupervised learning, no a-priori information is given on the produce labeled classes. Often, the goal in unsupervised learning is to decide which objects should be grouped together, that is to say, the system forms the classes itself (Hall, 1999). Of course, the success of classification learning heavily depends on the quality of the data provided for training— the classifier has only the input to learn from.

In this work we implement an on-line unsupervised clustering algorithm for detecting population change. It is designed for nonstationary data clustering and considers clusters which have relatively small mass. This algorithm, based on a previous work (Guedalia *et al*., 1999), has been improved in order to match the overall classifier. The improvement includes three measures for detecting population changes based on cluster size (i.e., amount of data points in a cluster) and variances of the centroid locations in the feature space.

To prevent loss of previous information, all measures related to the population that pass through the system are maintained in a *history* database. When the system detects a new population, it compares the *history* population overlap level with the current population. Accordingly, it decides whether to use a previous population classifier or to select a new one.

We propose a systematic method for classifier selection developed for on-line detection of the best-fit features. When a new population is detected, a classifier is selected for it according to the overlap level mentioned above. The classifier is selected from n fuzzy K-Nearest Neighbors classifiers, each trained with a different number of features.

Several methods for the task of combining/selecting classifiers in order to improve classification performance are well known from the literature (Kittler *et al*. 1998; Windriidge and Kittler, 2000; Kuncheva, 2004). In this thesis we used a fuzzy logic rule-based decision system to fuse the classifiers results and to select the best one according to feedback weights. The reference (training) label for this procedure is set by using a retraining in which a new batch of data points can represent the current population. This set is the new training set that must be defined for the current retraining procedure. This is done in one of two ways: by off-line human retraining or on-line automatic retraining. Human retraining is activated when there is no significance overlap between the current population and the previous ones. Automatic retraining is activated when the overlap is significant and the overlapped data points are used for the retraining procedure.

An important characteristic of the classifier/feature selection procedure is that it considers the classification error combined with the associated cost. This is in contrast to Bayesian machine learning, for example, in which the decision model minimizes the overall economic loss function (Duda *et al*., 2001). Grading of products includes two kinds of cost measures: economic losses incurred by misclassification (Miller, 1985) and computational costs. To estimate these measures, a cost function has been developed and implemented. The cost function is capable of identifying that some classification mistakes are more costly than others. In addition, there is a cost associated with the actual classification process – some features are very complicated to determine resulting in heavy computational cost.

1.2 Research objectives

The main objective of this research is to develop an on-line multi-stage classifier that includes population change detection as well as feature selection via a classifier selection methodology to enable the classifier to deal with changing and unknown populations.

1.3 Research significance

In this thesis we developed a classifier that is capable of dealing with biological products by continuously adapting itself to the current population. The developed sorting classifier changes itself according to the produce state, by selecting on-line the appropriate classifier and thereby the optimal subset of features. This is an indirect way of performing feature selection, a primary interest in this research.

By enabling feature selection we enable adaptation to time-varying features (e.g., due to weather, seasons) and to population changes (e.g., different fields, color, defects). The proposed classification system is a multi-stage, closed-loop system structure. The adaptive framework of the classification system allows it to calibrate itself according to its input produce.

System performance (selection accuracy and cost) is increased by incorporating cost analysis with error classification. This is achieved by minimizing a cost function that considers all the significant cost parameters including classifier features, computational cost and the classifier error significance.

1.4 Research contributions and innovations

The contributions and innovations of this research are twofold:

- On-line feature selection in previous research was implemented a-priori in an off-line mode. In this work we developed an on-line feature selection algorithm that consists of two levels:

1. In the first level, an on-line clustering algorithm detects new populations (i.e., changes in the feature space). It is based on a previous algorithm (Guedalia *et al*. 1999), which is extended and adjusted to fit the system's definitions. This extension includes adjustment from a two-dimensional feature space to a multi-feature dimensional space and a new methodology to define the location of population changes using three measures specifically developed in this thesis.
2. In the second level, a classifier is assigned to the present new population. According to the clustering algorithm results the classifier was selected from a sequence of *n* fuzzy K-Nearest Neighbor classifiers, each with a different number of features. The decision on the best-fit classifier is based on four possible cases of overlap that were defined between the current new population and the previous ones.

- A methodology for incorporating cost into the classifier was developed for the classifier selection stage. While most research is based only upon the classification error, this research takes into account a combined cost measure considering the algorithms' cost (computational and economical) and the risk associated with the classification error. This is done at the high level of the classification system (Figure 5). This level combines the following three criteria for the classifier selection task:

1. Classification error, based on weights assigned to the classifiers according to a fuzzy logic algorithm that fuses the classifiers results
2. Computational cost of the selected features
3. A cost function that will take into account error significance (*i.e.*, the cost of classifying a good product as bad and vise versa).

These three criteria define the objective cost function. In addition, each of the cost function components was tested for sensitivity by changing both the batch size of data defined for the population change check as well as the population entrance sequence.

2 Scientific background

2.1 Agriculture quality sorting

Quality sorting is based on a variety of features (Dull, 1986): flavor (sweetness, acidity), appearance (color, size, shape, blemishes, glossiness), and texture (firmness, mouth-feel). Previous research (Edan *et al*., 1994) indicated that quality classification base on multitude of characteristics improves overall classification and can be applied in real sorting.

Firmness, a feature indicating maturity, freshness, degree of bruising and the presence of internal spaces (i.e., internal voids or damage) has been used for years as a guide to fruit and vegetable quality (Studman and Boyd, 1994). Although there have been numerous attempts to automate firmness measurement in fruits and vegetables, there is no consensus on the recommended sensor (Pitts *et al*., 1994). Firmness sorting of apples, nectarines and kiwis was implemented in developing a commercial fruit firmness sorter (Peleg, 1999) comprises a unique conveying system which allows physical contact of the inspected items by a sensor finger. Another impact technique was to tap the fruit with a medium or small impact device. Delwiche and Sarig (1991) developed a firmness sensor of 63g to impact the fruit. Shmulevich *et al*. (2003) tested two non-destructive firmness methods on apples: low mass impact and acoustic response. Results indicated that the acoustic method might improve the sorting using impact method. Ruiz and Canavate (2005) reviewed non-destructive firmness measurements and concluded that the non-destructive firmness testing is most suitable for online fruit packing equipment. A high-speed weighting system for grading and sorting fruit implementing non-destructive factors was developed for industrial needs (Calpe *et al*., 2002).

Fruit color is an external visual property that very much affects consumer choice. Moreover, it has long been recognized as an acceptable maturity index for many fruits and vegetables such as tomatoes (Choi *et al*., 1995). Computer vision is the most important sensor for measuring color and other external features such as color homogeneity, bruises, size, shape, and stem identification (Aneshansley *et al*. 1993; Delwiche *et al*., 1994;). Feng and Qixin (2004), developed machine vision system for high-speed apples sorting. They extracted color and contour features and sort the ‘Crystal Fuji’ apples using HIS color space resulting in 90% sorting accuracy. The machine vision approach has also been used in classifying table olives

(Diaz *et al*., 2004). A three CCD camera system was applied to detect disease in citrus leafs (Pydipati *et al*., 2005).

A major issue focuses on what sort of features should be sorted. In an analysis of multiple features in Florida citrus, Miller and Druillard (2001) showed that classifying blemish features together with physical features yields better results than one just used based on blemishes. Quality classification of tomatoes was successfully applied by integrating computer vision with an impact sensor (Edan *et al*., 1997, Laykin *et al*., 2002).

The two most important parameters indicating tomato quality, firmness and color relate to ripening and shelf-life (Polderdik *et al*., 1993). Monoi and O'Brien (1980) and Polderik *et al*. (1993) investigated the relationships between tomato color, firmness, initial firmness, quality and shelf life. They found that color alone was insufficiently accurate to establish the relationship between tomato firmness and estimated shelf-life.

Several studies have examined the relationship between tomato maturity and optical features (Heron and Zacharia, 1974; O'Brien and Sarkar, 1974; Goddard *et al*., 1975; Moini and O'Brien, 1978). Moini (Moini and O'Brien, 1980; Moini *et al*., 1980) used reflectance at 670 and 960 nm wavelengths to detect mold and other surface defects. Near infrared spectroscopy (NIR) hyperspectral imaging has been used to detect bruises on apples (Lu, 2003). The spectral region between 1000-1340*nm* provided the most accurate results. He also studied the time elapsed after bruising affects results.

Nuclear Magnetic Resonance (NMR) is a nondestructive and noninvasive technique that can be used to detect the internal quality of fruits. Pathaveerat *et al*. (2001) developed an NMR sorting system for avocado maturity sorting.

Digital imaging techniques have also been used for analyzing size, shape, color, and surface defects (Sarkar and Wolfe, 1985). Maturity was classified into two stages: light red and red tomatoes were considered ripe while green indicated other ripeness stages. A color image analysis procedure was developed to classify fresh tomatoes into six maturity grades (Choi *et al*., 1995) based on hue information.

Apple orientation on conveyors was performed using shape characteristics (Throop *et al*., 2001). Hyperspectral and multispectral image analysis was used to detect defects in selected apple cultivars (Mehl *et al*., 2002). Rehkugler and Throop (1989) developed an algorithm for detecting apple-bruise with computer vision and NIR light reflectance from the apples. A real-time inspection station was developed for detecting defects on apples (Throop *et al*., 1999) while a high-speed machine vision system was developed for potato grading (Noordam *et al*., 2000). A new design for rotary trays, which presents the two sides of a vegetable for

inspection by machine vision systems, was developed for an eggplant grading system (Kondo *et al*., 2006).

Table 1: Selected examples of measuring fruits quality

Feature/Sensor	Product	Reference
Firmness: finger sensor; tapping devices	Apples, kiwi, nectarine Peaches,apples	Peleg (1999) Delwitch and Sarig (1991)
Firmness (acoustic&impact)	Apples	Shmulevich *et al*. (2003)
Non-destructive device	Dry plums	Haff *et al*. (2005)
Color Camera	Apples Olives Potatoes Citrus leaf	Feng and Qixin (2004) Diaz *et al*. (2004) Noordam *et al*.(2000) Pydipati R.(2005)
Integrated sensors	Tomatoes (Vision+impact) Florida citrus (Vision+wight)	Laykin *et al*. (2002) Miller and Druillard (2001)
NIR	Apples	Lu (2003) Rehkugler and Throop (1989)
NMR	Avocado	Pathaveerat *et al*. (2001)
Hyperspectral & multispectral	Apples (defects)	Mehl *et al*. (2002)
Firmness - Laser-based multispectral	Apples	Lu and Peng (2005)

2.2 Classification methods

The main procedure in the sorting mechanism, after extracting the produce feature vector, is to classify the fruit to one of several quality levels based on the features extracted using different sensors. The following are the most common classification methods:

2.2.1 Classical classification methods

The following is a brief description of a well-known statistical approach to the problem of classifier design (Duda *et al*., 2001).

Given *x* as the vector of features that was observed on one produce (ω_j), we need to decide how to classify the produce (ω_j) into one of ***c*** categories. We use the Bayes' rule to determine the a-posteriori probability $P(\omega_j|x)$ by using the a-priori probability $P(\omega_j)$ and the conditional probability density function $P(x|\omega_j)$:

$$P(\omega_j|x) = \frac{P(x|\omega_j) \cdot P(\omega_j)}{P(x)} \quad 1$$

$$P(x) = \sum_{j=1}^{C} P(x|\omega_j) \cdot P(\omega_j) \quad 2$$

In many pattern recognition applications, the a-priori knowledge about the probabilistic structure of the problem is unknown.

One approach to the problem is the *parameter estimation* technique that uses training data to estimate unknown probabilities and assume a form with certain probability density functions (*e.g.,* normal, linear distribution). Two common *parameter estimation* methods are maximum likelihood estimation (MLE) and Bayesian estimation (Duda *et al*., 2001). Both approaches estimate a parameter vector, $\boldsymbol{\theta}$. This vector has fixed parameter values that maximize the probability density function in the maximum likelihood case and is a random variable in the Bayesian case.

The multivariate normal distribution is very commonly used as the density function. Miller and Delwiche (1989) modeled features that were obtained from spectral analysis of peaches as normally distributed and implemented a Bayes' rule for the classification. Crow and Shimizu (1988) proposed two lognormal distributions. A Bayesian classifier was applied for sorting red apples (Shahin *et al*., 1999). When there is no assumption that the forms of the probability densities are known, the classifier can be designed by *non-parametric* approaches. The most popular and simplest approach is the *k*-nearest neighbor (*k*NN) method (Ripley, 1996). This method sorts the entire training set and calculates the distance from the new data, *X*. The labels of the k nearest neighbors are used to determine a label for the new data. Fukunaga and Flick (1985) compared the use of *1-NN* and the *2-NN* classification rules. Tomak (1979) used the k-NN method to eliminate data, of storage requirement, far from the class boundaries. Fuzzy k-NN classifiers were used for classifying airborne images (Yu *et al*., 2002). A comparison between back-propagation neural network and statistical classifiers (kNN, decision tree (DT) and Bayesian) was applied to apple sorting based on textural features (Kavdir and Guyer,2004).

Another approach to finding classification boundaries with no a-priori knowledge of the density function is based on linear discriminant functions. A two-class linear discriminant function using regression analysis was implemented to classify the spectrophotometric analysis of apples (Upchurch *et al*. 1990). Miller (1985) used the piecewise linear classifier, the term for more than two-classes, for lemon sorting. Deck *et al*. (1995) compared the Fisher discriminant function method with a neural network approach for potatoes inspection.

2.2.2 Decision tree classifiers

While the statistical classifiers reach a conclusion in one step, the decision tree classifier takes the global decision process and divides it into a number of local decisions at each level of the tree. At each branch in the tree structure, a test is applied to the input data. The answer for each level determines which branch will proceed (Duda *et al*., 2001). Final classification of the data is made at the branch end. Kanal (1979) described models of these trees and procedures for planning them. Decision trees such as C4.5 and ID3 perform feature selection as part of their construction process (Scott *et al*., 1998). A node in these trees represents the feature and the branches represent the feature's value. The classification is shown in the leaves. Decision tree, C4.5, was mentioned as the most popular tree construction algorithm by Duda *et al*. (2001). Jack and Fu (1980) report on automated classification of blood cells using quadratic classifiers at junctions in a decision tree structure. A 17% misclassification rate was achieved. A large tree classifier using heuristic search and global training (Wang and Suen, 1987) was developed for recognition of Chinese letters. In a simulation of this algorithm, a very high recognition rate of 99.9% was achieved. A decision tree based approach for discriminating apple stem and calyx was applied using CART and C4.5 methods (Unay *et al*., 2006).

A binary decision-tree-structured rule base was established for defect inspection in apple sorting (Wen and Tao, 1999). A decision tree algorithm was used to distinguish between manure and chemical fertilizer treatments in corn fields (Yang *et al*., 2001). Three different sensors were combined using a decision tree to evaluate the quality of apples (Xiaobo *et al*., 2005). Two apple varieties were classified using a decision tree classifier (Kavdir and Guyer,2004).

2.2.3 Clustering

Clustering is a well known method for unsupervised learning. Clustering algorithms divide a set of *n* observations into *g* groups so that members of the same group are more alike than members of another group (Ripley, 1996).

Figure 1 presents a typical pattern clustering activity (Jain *et al*., 1999) . The feedback path indicates that the grouping process output could affect the feature extraction and the similarity computation. Hoppner *et al*. (1999) presented the main conventional clustering techniques:

- Incomplete or heuristic techniques: geometrical methods, representation or projection techniques. Multi-dimensional data are analyzed by dimension reduction to obtain a graphical representation in two or three dimensions (using methods such as principal component analysis).
- Deterministic crisp techniques: Each data point can be assigned to exactly one cluster. The cluster partition defines an ordinary partition of the data set.
- Overlapping crisp techniques: Each data point can be assigned to at least one cluster or to several clusters simultaneously.
- Probabilistic techniques: A probability distribution is determined over the clusters such that each data point is assigned to a cluster based on its specific probability.
- Possibilistic techniques: These techniques are the fuzzy clustering algorithms. The data point is clustered according to the degrees of membership or possibility to the clusters.
- Hierarchical techniques: Divides the data into classes in several steps. The data is first separated into a few broad classes and further divided into smaller classes and so on until terminal classes are generated which can not be subdivided (Everit, 1974).
- Objective functions based techniques: An objective or evaluation function assigns each possible cluster partition with a quality value that need to be optimize. The best solution is the cluster partition that obtains the best evaluation.
- Cluster estimation techniques: These techniques use heuristic equations to build partitions and estimate cluster parameters.

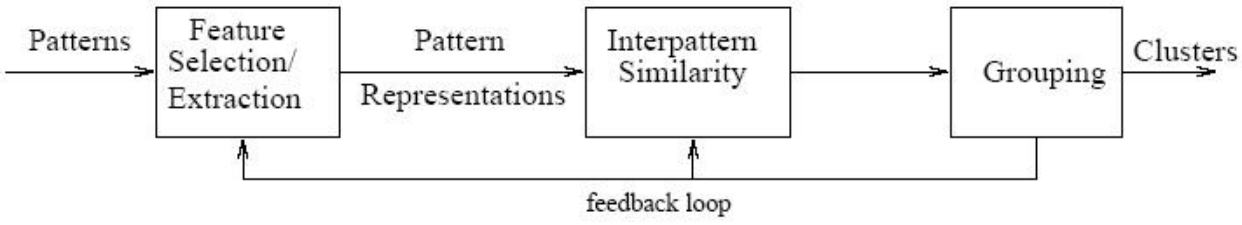

Figure 1: Stages in clustering (Jain *et al*., 1999)

K-means clustering is a crispy partitioning method in which each data point belongs to one cluster only (Jang *et al.*, 1996). The procedure starts with initialization by guessing the k-means (k-the number of clusters); continues with an iterative procedure that clusters the data according to its distance from the means; and stops when there is no change in the means location. Fuzzy C-Mean Clustering (FCM) is a clustering algorithm in which each data point belongs to a cluster to a degree specified by a membership grade (Bezdek, 1981). FCM partitions a collection of n vectors x_i, i=1,…,n into c fuzzy groups, and finds a cluster center in each group such that the cost function of a dissimilarity measure is minimized.

An algorithm is defined as on-line if no assumption is made as to the size of the dataset (Guedalia *et al*., 1995). There are three common methods for on-line clustering: a constant number of centroids; splitting centroids according to their relative importance; and inserting centroids based on external parameter. The last case introduces us to the problem of how to recognize that the data has arrived from a new cluster and how to allocate a centroid to represent it. The adaptive resonance theory (ART) is one way to deal with this problem (Carpenter and Grossberg, 1990; Georgiopoulos *et al*., 1999). Mattone (2002) presented an on-line competitive clustering algorithm and applied it to motion-based image segmentation.

Guedalia *et al*. (1999) developed an on-line agglomerative clustering method for nonstationary data. Since this algorithm fits the pattern of agricultural produce that tend to be non-stationary, it was used in this research. Apples features were classified using k-means clustering method as learning procedure in an overall classification task (Leemans and Destain, 2004). On this basis fruits were classified by quadratic discriminant an analysis into 73% classification accuracy.

2.2.4 Neural networks

Neural networks are an important tool for classification and research shows that they offer a good alternative for known classifiers (Zhang, 2000).

Lippmann (1987) describes them as connected topologies of simple processing elements. Each element computes a single-value output function based on its input vector. The network types are defined by their topology, node characteristics and learning rules.

The most common structure is the multilayer perceptron that has three layers: n inputs nodes; c output nodes where c is the number of desired classes; and a hidden layer of nodes of undetermined number. Each node input is one output of the previous layer. The neural nets are nonlinear models, which make them flexible in modeling real world problems.

To improve the performance of neural networks classifiers an on-line retrainable neural network was developed for image processing problems (Doulamis *et al*., 2000). A dynamic neural network was applied to classifying multi-sensor quality information (Guedalia and Edan, 1995). A neural network classifier was compared with the Fisher discriminant function for inspecting potatoes (Deck *et al*., 1995). Two neural networks were employed for grading carrots (Howarth and Searcy, 1991). Miller and Drouillard (2001) used three neural network configurations for classifying Florida citrus. A probabilistic neural network was applied to segmentation features in corn kernel images (Steenhoek *et al*,. 2001). An ANN classifier was applied to apple classification based on surface bruising (Shahin *et al*., 2002). Unay and Gosselin (2005) used an ANN to segment the defected region on apples by pixel-wise processing. Using a SVM classifier, they reported a 90% recognition rate. Simoes *et al.* (2002) ANN used a color classificatory for orange sorting. He concluded that for better performance the network must be trained for each new color presented to the system.

Pydipati *et al*. (2005) compared neural network and statistical classifiers to determine citrus disease resulting in 95% accuracy.

2.2.5 Fuzzy classifiers

Fuzzy pattern recognition is about any pattern classification method that involves fuzzy sets (Kuncheva, 2000). The fuzzy classifier can be described by a set of fuzzy if-then rules. Fuzzy logic applied as a decision support technique in grading apples (Kavdir and Guyer, 2003) achieved a grading accuracy of 89%.

A fuzzy model was developed to predict peanut maturity (Shahin *et al*., 2001) based on NMR signals. The hull-scrape chart that is commonly used for peanuts is boring and time-consuming. Compared to the hull-scrape chart, the fuzzy model only yielded a 73% accuracy for three maturity classes but is much faster.

Chao *et al*. (1999) applied a neuro-fuzzy based image classification system to inspecting poultry viscera.

2.2.6 Multi-classifier systems

An effective technique for achieving higher classification accuracy combines multiple classifiers. A number of fusion methods operates on the classifiers instead of their outputs, trying to improve the classification rate by optimizing classifier structures (Ruta and Gabrys, 2000). Combining classifiers means assigning a class label for x based on L classifier outputs. Soft label for x can denoted as:

$$D(x) = [\mu_1(x),....,\mu_c(x)]^T \quad 3$$

Three voting classification algorithms were compared: bagging, boosting and variants (Bauer and Kohavi, 1999). Boosting algorithms are among the most popular methods for building classifier ensembles (Kuncheva, 2003).

Figure 2 presents a schematic diagram of a multi-classifier system. The classifier outputs are combined using a suitable decision algorithm to give an overall prediction.
According to Kuncheva (2004) the use of classifier combination methods "aim at a more accurate classification decision at the expense of increased complexity"

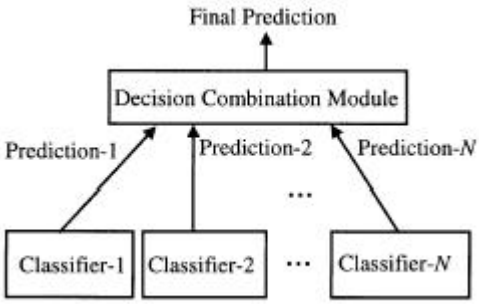

Figure 2: MCS schematic diagram (Lim and Harrison, 2003)

Kuncheva *et al*. (2001) introduced a simple rule for adapting the combination method to the application. A number of decision templates (one per class) are estimated with the same training set that is used for the classifiers. These templates are then matched to the decision profile of new, incoming objects using similarity measures.
Kuncheva (2003) compared fuzzy and non-fuzzy combination methods. The experiments show that the fuzzy combination methods performed better than the non-fuzzy methods. The fuzzy fusion is detailed in Figure 3.
The Dempster-Shafer (DS) theory was applied to multi-classifier systems to define the rejection criteria of fruit images and handwritten numbers (Theil *et al*., 2005). It is shown that this classifier fusion can boost the combined classifier accuracy to 100%.

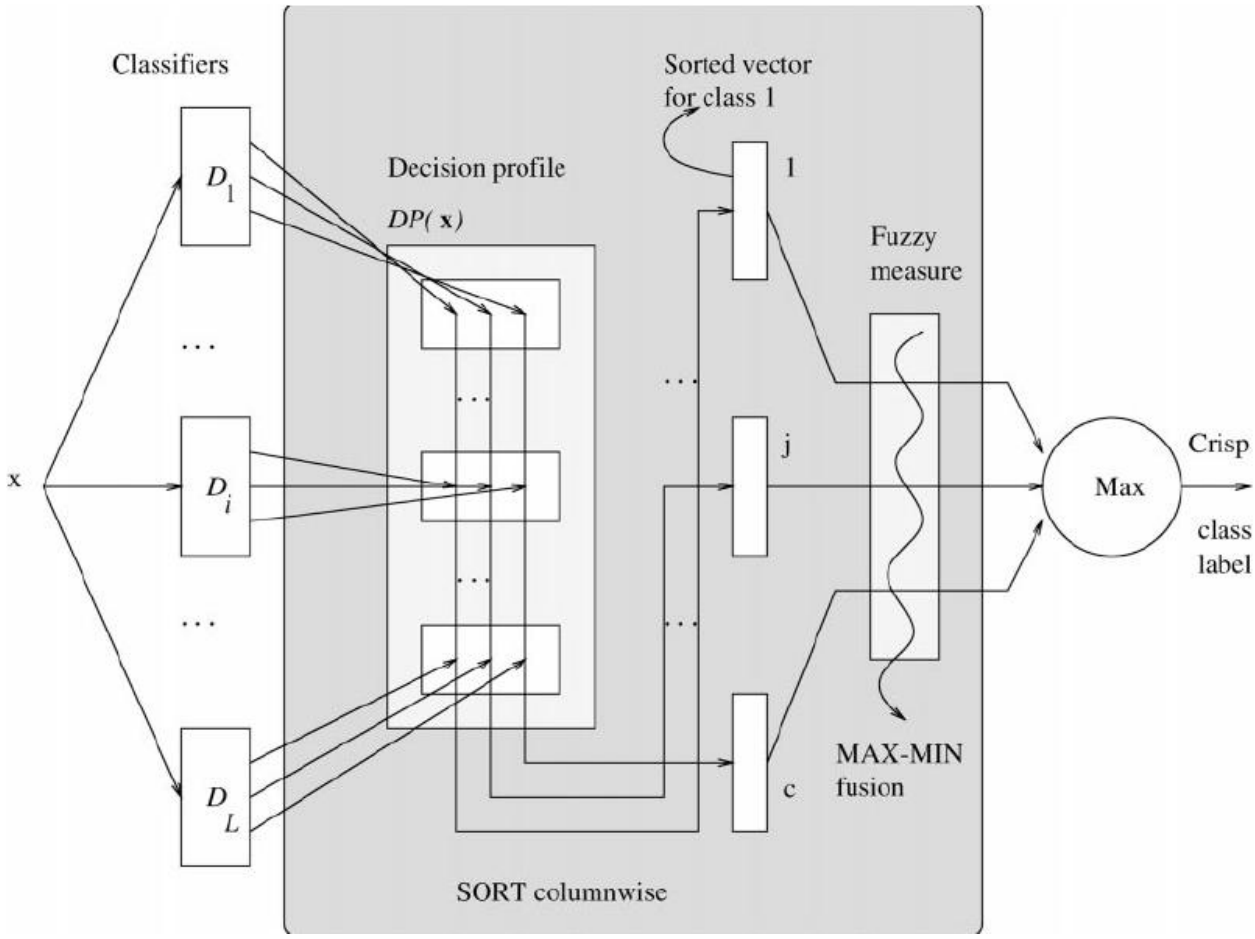

Figure 3: Fuzzy integral for classifier fusion (Kuncheva *et al*., 2001)

Windridge and Kittler (2000) combined four classifiers, k-NN, neural net, normal probability distribution function (PDF) and quadratic PDF. This was based on a method for performance-constraining a feature selection process as it relates to combine classifiers. They concluded that applying one of various tested feature selection methods could provide an alternative to other classifier combination approaches. Another approach for sequential classifiers combination uses combinations of two k-NN classifiers. The first k-NN classifier works after feature selection procedure is made. If it fails (according to threshold checking) the second is activated on all features (Last *et al*., 2002).

In addition to classifier combination there is the matter of classifier selection.

Kuncheva (2002) presents a combination of classifier fusion and selection using statistical inference to switch between the two. In this work, feature space regions, in which one classifier yields significantly better results, were classified with selection while the other regions with fusion.

Table 2: Selected examples of fruits classification methods

Method	Product	Reference
Bayes rule	Peaches Apples	Miller and Delwiche (1989) Shahin *et al*. (1999)
Knn	Apples	Kavdir and Guyer (2004)
linear discriminant functions	Apples Lemons	Upchurch *et al*. (1990) Miller (1985)
Decision tree (CART&C4.5) **Binary DT-structured rule base**	Apples (stem) Apples	Unay *et al*.(2006) Wen and Tao (1999)
k-means clustering **SOM_FCM**	Apples Edible Beans	Leemans and destain (2004) Chitioui *et al*. (2003)
Fuzzy model	Peanut maturity Apples	Shahin *et al*. (2001) Kavdir and Guyer (2003)
NN classifier **BPNN**	Potatoes Carrot Apples Apples	Deck *et al*. (1995) Howarth and Searcy (1991) Shahin *et al*. (2002) Unay and Gosselin (2005)

2.3 Adaptive models

Adaptive classification is the ability of a classifier to adjust to changes in its classification environment. A learning system should have the ability to absorb knowledge continuously and autonomously in order to deal with problems in nonstationary environments (Lim and Harrison, 2003). For example, in fruit grading, fruit features may appear overnight (Guedalia *et al*., 1999) and in some cases, apples, for instance, the defects (Lu, 2003) can appear in a matter of hours.

Carpenter and Grossberg (1987) mentioned the stability-plasticity dilemma that is very relevant to the issue of adaptive classification. The dilemma is how a learning system is able to protect useful *history* data (stability) while retaining the ability to learn new data (plasticity). In the neural network domain, the primary efforts in overcoming this problem were focused on growing and pruning networks (Lin and Lee, 1996). The issue of detecting

new data entering the system can be treated with change/novelty detection. Novelty detection is the identification of new or unknown data or signals that a machine learning system is not aware of during training (Markou and Singh, 2003).

In their research Markou and Singh differentiate between statistical and neural network approaches. The statistical approach is driven by modeling data distributions and then estimating the probability of test data belonging to such distributions. The problem is in making assumptions about the nature of training data. The main advantage, however, is its cheap computational cost. Picus and Peleg (1999), present an adaptive classification method for agriculture produce based on prototype populations. The idea was to set a classifier for each prototype population and activate it whenever a compatible population went through the system. Kuncheva, (2004) proposed a classifier ensemble for changing environments. The idea is the combination can act as an online dynamic classifier selection system that updates one classifier and combination rule for each new data point x. This idea is still in its initial stage.

2.4 Feature selection

Feature selection methods try to find a subset of features that are relevant to the target concept (e.g., the classification). An irrelevant feature neither affects the target concept nor adds anything new to it (Dash and Liu, 1997). For high dimensional data, the right selection of features has a significant effect on the cost and accuracy of an automated classifier. Reducing the number of irrelevant features will result in reducing the computational cost (running time) of a learning algorithm and yield better classification.

There are many existing methods and much work has been done in the area of feature selection. In general, feature subset selection algorithms have two components: an evaluation function that tests the fitness of feature sets and a search engine for finding these sets. Langley (1994) defines two types of feature selection frameworks derived from the evaluation function: filters and wrappers. The wrapper method uses the classifier's accuracy as the criterion (evaluation function) for the subset selection while the filter method uses various measures (distance, information, consistency etc.) and is independent of the classifier. Liu and Motoda (1998) discuss these two models in the context of machine learning and data mining approaches. Because of the huge data size used in most data mining problems, a classifier cannot directly be applied to it. Therefore, the filter method is more appropriate for data mining problems. For machine learning, on the other hand, the major concern is to improve the classifier's performance and therefore the wrapper method is more appropriate.

Jain and Zongker (1997) reviewed most of the feature selection methods (Figure 4).

A feature set that contains N features will result in 2^N optional subsets. This is a huge number and there are three main methods for solving this problem: exhaustive, heuristic and random (Dash and Liu, 1997). In the exhaustive method, the search occurs throughout the complete feature subsets to find the one with the minimum error rate. The simplest search is the stepwise selection that includes the forward selection and the backward elimination. The first starts with an empty set and adds one feature at a time, trying to maximize the evaluation function. The second starts with all available features and deletes features that reduce the performance. The two other methods based on heuristic or random search methods attempt to reduce computational complexity by compromising performance. These methods need a stopping criterion to prevent an exhaustive search of subsets.

The classic branch-and-bound method starts searching from the original feature set and removes any subset whose value is less than the bound of the evaluation function. Stepwise discriminant analysis was used for feature selection as part of a classification process of pistachio nuts (Pearson *et al*., 2001). A mathematical method called orthogonal transformation was used to find a small set of features that represent samples of wheat cultivars (Utku, 2000).

Feature selection based on the Bayes' rule for minimum cost was developed for classifying remote sensing images (Bruzzone, 2000). Another approach to feature dimension reduction is feature combination. Principal Component Analysis (PCA) is an unsupervised approach to finding the “right” features of the data (Duda *et al*., 2001). In this method d-dimensional data are projected onto a lower-dimensional subspace in a way that is optimal in a sum-square-error sense. Yeung *et al*. (2000) setup an empirical study on PCA for gene data clustering.

Two feature selection methods were implemented with a neural network input data that was used for wheat kernel color classification (Wang *et al*. 1999). The methods include PCA and divergence feature selection. Leray and Gallinari (1999) introduce a review of neural network approaches for feature selection. Feature selection by genetic algorithms was implemented for seed discrimination (Chtioui *et al*., 1998).

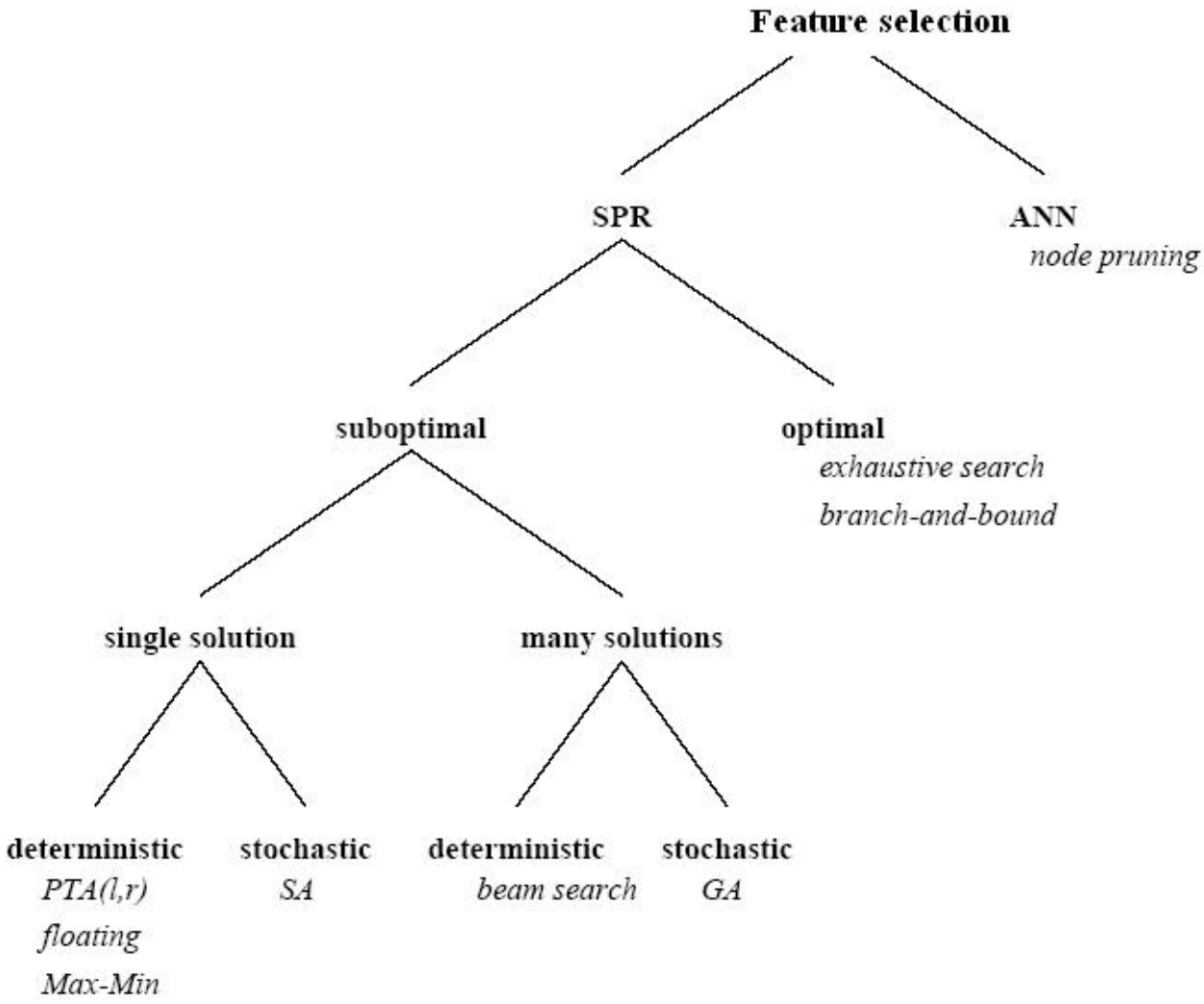

Figure 4: Taxonomy of feature selection algorithms (Jain and Zunker, 1997)

A branch-and-bound method was used to search for optimal feature subset selection (Somol *et al*., 2004). The method was applied on a number of real-data known from literature.

An on-line feature selection mechanism was applied to tracking applications (Collins *et al*., 2005). The method chooses features that minimized the potential for distraction in the next frame.

2.5 Cost analysis

Cost function analysis is important when some classification errors are more costly than the others. The simplest case of equal errors cost is most frequent case used (Duda *et al*., 2001).

The following conditional risk is used in the Bayes decision rule:

$$R(\alpha_i \mid x) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) \cdot P(\omega_j \mid x) \qquad 4$$

The loss function $\lambda(\alpha_i \mid \omega_j)$ describes the system loss in case of taking action α_i when the stat of nature (i.e.,) is ω_j. $P(\omega_j \mid x)$ is the posterior probability calculated in the Bayes formula. To minimize the overall risk calculate $R(\alpha_i \mid x)$ for $i = 1...a$ and select the action α_i that yields the minimum value.

The performance measure that is usually used in classification related works is the classification accuracy. Nevertheless, in real-world application of such work there are several types of cost that should be considered (Turney, 2000).

A cost model that takes into account additional cost attributes together with classification accuracy can yield much more realistic results.

Cost can be measured in many different units (Turney, 2000). In medical applications it may include units that related to quality of life of the patient and in image processing it might be measured in terms of computational CPU time units. Cost can also be represent with probability distribution over a range of possible costs. A Bayesian decision model for agriculture products grading was developed that includes grades and feature vector probabilities as well as penalty loss factor that was set according to the misclassification levels (Miller, 1985). Peleg (1981) developed a grading criterion based on the sorting accuracy and a sorting error index.

Turney (2000), defined most of the different cost types: misclassification error cost, test cost (e.g., blood test in medical), teacher cost (the usage of an expert), cost of intervention (e.g., intervene in a manufacturing process in order to improve it), computational cost (static and dynamic), human-computer interaction cost and the cost of instability (experiments should be repeatable).

A framework based on utility theory used a classifier combination method while considering each classifier cost (Demir and Alpydin, 2005). This framework demonstrated the ability to achieve higher utility values by changing different classifiers type considering the importance of the testing cost. The expected utility of selecting action A_i for input vector *x* is described in equation 5.

$$EU(A_i \mid x) = \sum_{k=1}^{k} P(C_k \mid x_i, A_i) \cdot u(C_k, A_i) \qquad 5$$

Where, $u(C_k, A_i) = accuracy(C_k, A_i) - \alpha \cdot \cos t(C_k, A_i)$

In this equations $u(C_k, A_i)$ is the utility (negative risk) of taking the action A_i when C_k is the state of nature.

A cost function includes three types of costs, misclassification cost, features measurements costs and the response time cost, was developed (Arnt and Zilberstein, 2004) and designed to achieve the highest quality.

The following equation (6) is the cost function that combines these three cost types. It represents the 'cost of assigning predicted label l_p to an instance F with measured attributes $meas(F)$ and actual label l_a in t time units'.

$$C(F,t) = \omega_L EC_L(cl(l_p) \mid F) + \omega_T C_T(t) + \omega_M \sum_{f_i \in meas(F)} C_M(m(f_i)) \qquad 6$$

Where $EC_L(cl(l_p) \mid F)$ is the expected misclassification cost given that classifier predicts label l_p, $C_M(m(f_i))$ is the individual attribute f_i measuring cost and $C_T(t)$ is the time cost component.

An experimental study that deals with ordinal classification problems used a cost sensitive technique that uses fixed and unequal misclassification costs between classes (Kotsiantis and Panagiotis, 2004).

3 Methodology

3.1 Overview

This chapter describes the methods used in this research. The classifier structure overview and its basic development assumptions are presented in the first section. The second section presents the classifier framework and the following sections present an overview of the methods used in this dissertation. Detailed algorithms are presented in chapter 4.

The on-line multi-stage classifier consists of two levels - a low level for population detection and a high level for classifier adjustment according to the low level input (Figure 5).

Feature selection in the on-line multi-stage classifier is achieved via classifier selection.

3.1.1 Classifier structure

At the low level a modified on-line clustering algorithm designed to cluster non-stationary data identifies the population. A centroid in the feature space represents each population. Since this type of algorithm is adaptive to the data input, each new data point becomes a new centroid while the two previous most redundant centroids are merged. Overlapping measures identify new populations. The measures themselves are determined by a known overlap volume measure (Ho and Baso, 2002) implemented on the entire feature space as well as on each individual feature.

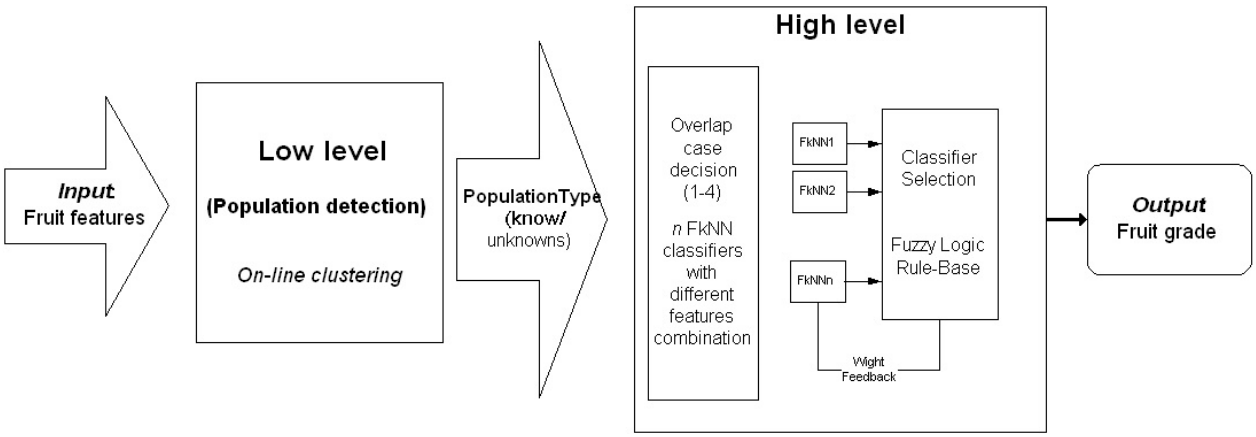

Figure 5: The on-line multi-stage classifier basic structure.

Assuming that 'H1' and 'H2' are known populations, the overlap level between these populations and a 'New Population' is assigned to one of the following four levels (Figure 6): *no overlap (1a)*, *full overlap (1b)*, *multi-overlap (1c)* (more than one overlapping population) and *partial overlap (1d)*. Each case is treated differently.

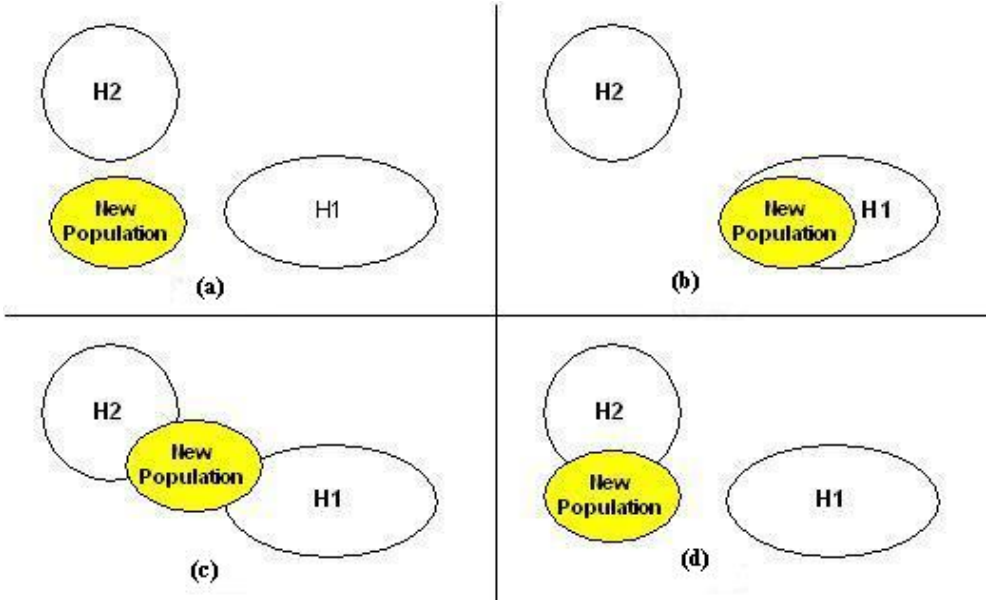

Figure 6: Population overlap cases: (a) no overlap, (b) full overlap, (c) multi-overlap, (d) partial overlap

Whenever a new population is detected at the low level, the high level stage is activated. The high level goal is to replace the previous classifier by a new one that better fits the current data.

The classifier selection procedure, which constitutes the high level stage, uses *n* fuzzy kNN classifiers, each trained with different feature combinations that function as input to a fuzzy rule-based decision system. The rule-based system is composed of three fuzzy inference systems (FIS) based on the Mamdani method (Jang *et al*., 1996).

The fuzzy kNN classifiers must be retrained whenever the classifier selection algorithm is activated for a new population. This retrain procedure is implemented automatically on-line when there is a sufficient overlap region and off-line when there is none. In the latter case, retrain is achieved using off-line human retraining.

Since there is always a gap of data points between the decisions on population change detection and overlap case determination, the data points are classified using the closest '*history*' population classifier in order to keep the system in an on-line state.

Three measures to detect population change using the amount of data points in a population and the variance of the centroids were defined.

3.1.2 Assumptions

The proposed classification system is based on three assumptions:

- The produce enters the system in batches, one batch after another.
- After the products are separated from each other, they enter the system one-by-one.
- All features exist for each product.

3.2 Overall classifier

Figure 5 describes the structure of the classifier system. Population detection is conducted by analyzing the product variability. The main idea is to check whether the current stream of produce is different from the previous one. The differences may be biological in origin, caused mainly by weather changes and/or derived from a different field source of the produce, resulting in diverse defects, major size or shape changes and various color saturation levels. Introduction of a new population may be triggered either by a feature that has significantly changed or by the appearance of a new feature (e.g., a new batch harvested after a severe weather change might contain new defects such as hail damages that were not included in previous populations) or a combination of several cases.

When a new population is announced, the overlap level of the current population is set using the overlap measures. Unless a sufficient overlap exists (case of '*full overlap'* Figure 6*b*) the *n* fuzzy kNN classifiers are retrained on the selected training set (i.e., data from the overlap region). After the classifiers are trained, the classifier selection procedure starts. At this stage we let $D = \{D_1, D_2, ..., D_n\}$ be a set of fuzzy kNN classifiers, each trained with different feature combinations, and $\Omega = \{\omega_1, ..., \omega_c\}$ is a set of class labels. Each classifier receives as its input a feature vector $x \in \Re^d$ and assigns it to a class label from Ω, i.e. $D_i : \Re^d \to \Omega$, or equivalently, $D_i(x) \in \Omega, i = 1, ..., c$. In the case of the fkNN classifiers the output is a "soft label", *c*-dimensional vector, $D_i(x) = [\mu_{i,1}(x), ..., \mu_{i,c}(x)]^T$, where $\mu_{i,j}(x)$ is the "support" that classifier D_i gives the hypothesis that *x* comes from class ω_j and is in the interval [0,1], $i = 1, ..., n$, $j = 1, ..., c$ (Kuncheva, 2004). These "soft labels" function as an input to a fuzzy rule-based decision system. The fuzzy system fuses the classifiers results and yields an overall

classification result. Based on the result, weights are assigned to each classifier. The classifiers weights are the criteria for the classifier selection. Eventually, after *l* data[2] points from the new population pass through the selection procedure, the best-fit classifier is selected. The feature subset selected to classify the current produce is defined by the features of the selected classifier.

3.3 Initial off-line feature selection

The extracted features, which form the input vectors to the system, may be statistically correlated or dependent. To improve the overall classifier efficiency, these features should be eliminated. Since the system contains two separate levels that use different features to accomplish their targets (population change detection and classifier adjustment), eliminating features at one level may affect the performance of the other level. Some features that might not change much within a certain population (e.g., size or shape descriptors) may serve as a good population representative and therefore could be good for population separation. However, since they don't influence the quality labeling they can be eliminated by the feature selection procedure. Therefore, a procedure was implemented to keep these representative features, together with the quality features that were selected with the traditional feature selection algorithm. However, for the high level that eventually yields the quality grading results, only the 'quality features' set is used. In this initial procedure the stepwise forward selection (SFS) algorithm (Dash and Liu, 1997) was used for off-line feature selection. This is conducted separately, a-priori, for each case study. The SFS algorithm starts from an empty set, and at each iteration generates new subsets by adding a feature that, together with the ones already in the set, most accurately predicts the target (i.e., the vector of results). The algorithm stops when the new feature does not significantly reduce the prediction error.

3.4 Training/Testing data sets selection

The selection of training/testing is very important for determining classifier performance. The goal is to use enough data to build the classifier (defined as a *training* set) and still leave as much unseen data as possible to test its performance (defined as a *testing* set). It is important to use a separate data set for these training/testing stages (Kuncheva, 2004). During the initial a-priori phase and at the upcoming retraining phase, a training data set for the *n* fuzzy kNN classifiers training was selected. The selection of the data for the training for both cases is

[2] The value of ***l*** was define empirically.

based on the overlap level cases (detailed in section 3.8.2). The performance of each classifier was evaluated using the cross-validation leave-one-out method (Duda *et al*., 2001). The classifiers configuration is based on the k parameter selection and is detailed in section 3.9.

3.5 Classifiers base selection

After the initial feature selection stage (described in section 3.3), a procedure for selecting the best classifiers for the high level phase is implemented. Each classifier at this level is designed with a different feature composition. This means that for *d* selected features, $\sum_{i=2}^{d-1}\binom{d}{i}$ classifiers can be generated. However, since this might be too large to use in the current phase, a classifier database, containing all possible classifiers, was determined with the best ones selected according to their classification performance on the training set. Whenever a population change is detected, the system checks all classifiers for the new population.

3.6 On-line clustering algorithm

Population identification is based on a modification of an on-line clustering algorithm designed to cluster nonstationary data (Guedalia *et al*., 1999)[3]. This algorithm takes into account clusters which have relatively small mass. For each new data point, the following steps are conducted: the closest cluster centroid is moved toward the new point; the two closest centroids are merged; and the new point becomes a centroid. The sequence of steps is important to address the criteria of adaptation to temporal changes in the data distribution which is crucial for the new population detection task. While this algorithm developers dealt with the best choice of the initial number of centroids, in this work only three were used since the goal is different. Whenever a new population is detected, the previous population details (data and centroid) is kept in a *history* database to avoid mixture of populations. A new centroid was assigned and the three-centroid process was repeated. To detect the change location, it is necessary to trace the clustering changes on-line. To do that, the clusters size together with the centroids means and variances are calculated after introducing each new data point. These calculations result in three measures that were developed (see section 3.7) to test whether the new point belongs to a new population. The full algorithm description is detailed in section 4.2.

[3] The algorithm pseudo-code, based on the original code, is presented in Appendix V.

3.7 Measures for detecting population change

The measures for detecting changes in the characteristics of the populations are activated periodically, at predetermined intervals of *m* samples (*m* is determined empirically).

Incoming data is analyzed using the following characteristics: the size of the clusters (i.e., the amount of data points) and the variance of the current centroids locations.

Let us define the function that describes the number of data points in the centroid (y) as a function of the sample number entering the system (x). The change in the number of data points that belong to each centroid is the gradient of the curve and can be calculated as follows (equation 7):

$$\frac{dy}{dx} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \qquad [7]$$

where y_i refers to the number of data points that belong to one of the centroids when sample *i* enters the system. The expression $x_i - x_{i-1}$ represents the interval of *m samples* as defined above.

When the population characteristics do not change, the cluster's centroid accumulation rate of sample points is linear. Three measures are defined to detect population changes. Two measures are based on the comparison of clusters size while the third averages the variance change between the data intervals. All measures are initialized after a new population is detected. During the process of population detection the entering data points are classified using the closest *history* population classifier in order to keep the system in an on-line state.

The performance measures for this procedure use the overlap measures between the populations structure, which are defined in section 3.11.2. Detection results were compared to the actual population change locations.

3.8 Classifier adjustment methodology

After a new population is detected, the previous classifier should be replaced by a new one that better fits the current data. The previous population's details are kept in a *history* database. These details include the population's raw data as well as the centroid and the selected classifier information. According to the overlap level between the current and previous populations, the system selects the best classifier. This procedure is detailed in a flowchart [Figure 7]. An initial database, referred as *population database*, was defined a-priori based on similarity levels between the populations. This *population database* represents most of the populations and was updated with any new population detected.

3.8.1 Overlap analysis

The new population must be checked for the overlap level with previous populations in order to decide which methodology to use to select the best-fit classifier for it. An overlap volume measure (Ho and Baso, 2002) was implemented (equation 8). This measure evaluates the overlap of two different populations. The maximum and the minimum values of each feature *fi* and class *cj* are defined to be max(*fi, cj*) and min(*fi, cj*), respectively.

$$F_{overlap} = \prod_i \frac{MIN\left(\max\left(f_i, c_1\right), \max\left(f_i, c_2\right)\right) - MAX\left(\min\left(f_i, c_1\right), \min\left(f_i, c_2\right)\right)}{MAX\left(\max\left(f_i, c_1\right), \max\left(f_i, c_2\right)\right) - MIN\left(\min\left(f_i, c_1\right), \min\left(f_i, c_2\right)\right)} \quad [8]$$

where $i = 1,....,d$ for a d-dimensional problem and *j*=1,2 for two classes (populations). Of course, the volume is zero (or negative) as long as there is at least one feature dimension in which populations range do not overlap.

Since this measure decreases rapidly along with the growing number of features, it is only used to determine whether there is an overlap or not. The measure is also only used on each individual feature. The minimum overlap is used to determine the overlap case. In addition, a simple procedure was added to provide the amount of data points that overlap from each population. To prevent noise interference, populations that indicate high degree of skew were cutoff using a 95% confidence interval on their means.

According to the overlap measure, four overlap levels between the current population data and the *history* populations were defined as mentioned above (section 3.1.1).

3.8.2 Retrain procedure

When ‘full overlap’ is determined, the corresponding history population classifier was retrieved for the current population. For the other cases, two retraining procedures were determined: automatic and human. The human retraining procedure was defined for the first case (no overlap), meaning, a predefined training set for the current population was used for off-line retraining of the system. The automatic retraining procedure applies on-line for those cases in which classifiers are retrained with the data points in the overlapping region of the *history* populations (cases 3/4). These data points are already labeled with their population classifier.

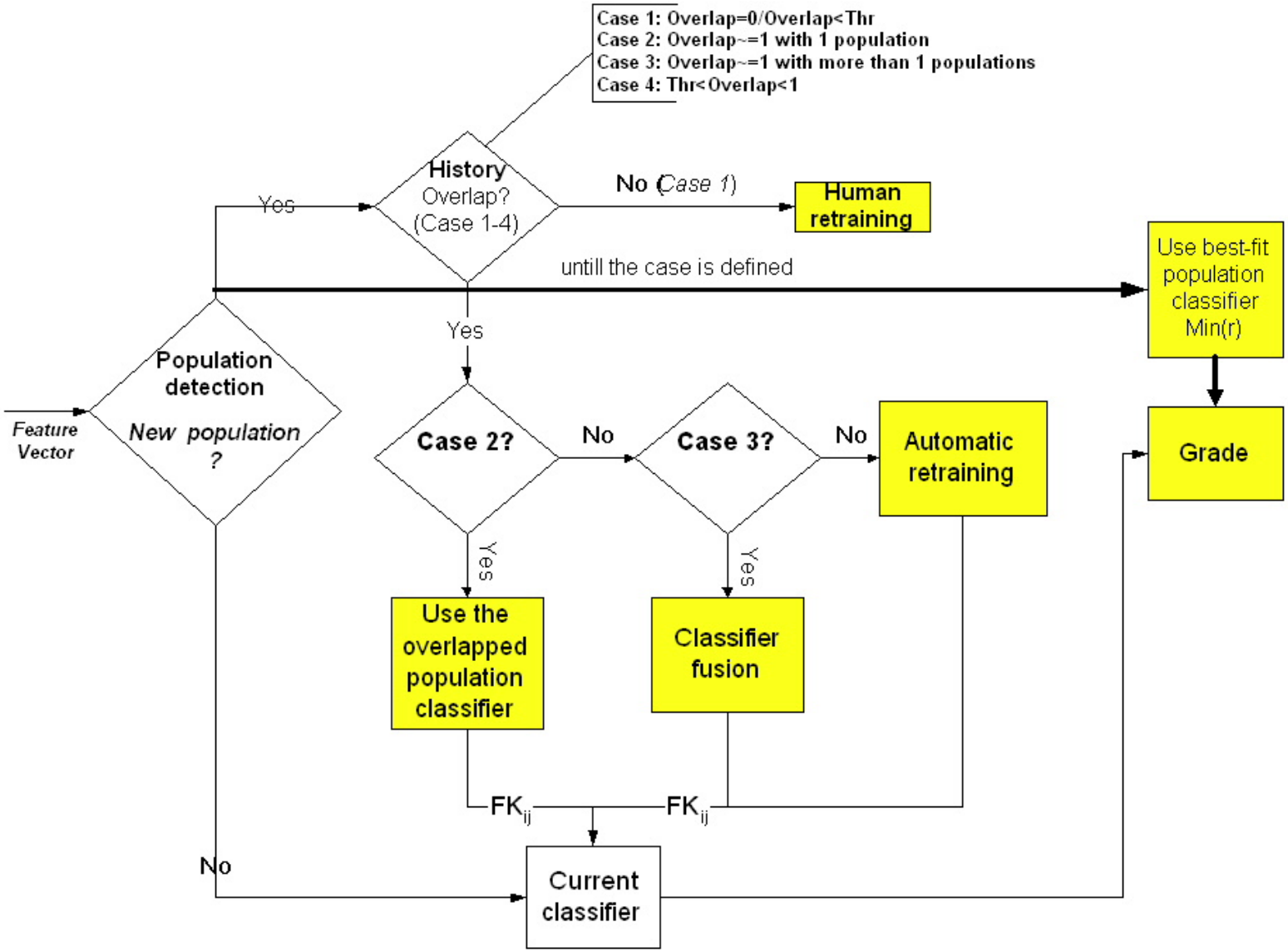

Figure 7: Detailed classifier flowchart

3.8.3 Classifier selection algorithm

The high level was implemented using *n* fuzzy kNN classifiers that function as input to a fuzzy rule-based decision system. The system was composed of three fuzzy inference systems (FIS) based on the Mamdani method (Jang *et al*., 1996). The number of quality grades is defined in advance according to the produce standards. According to the number of quality grades (*c*), each kNN classifier outputs an *c*-dimensional vector, one component for every quality grade, into the fuzzy system. The fuzzy system outputs are summed into an aggregation matrix. Each cell in the aggregated matrix sums the values that belong to it. A cell summing procedure is activated for the final decision of the quality grade.

3.9 Optimal k selection

When dealing with kNN classifiers, we must cope with the problem of selecting the best value of k, the number of neighbors considered. The leave-one-out cross-validation method was applied (Duda *et al*., 2001) to estimate the misclassification rate of the classifier for each choice of k. That is, each sample of the training set is classified by all the others, using the current set of variables in the model and the entire vector of k's. We used the set 1; 3; 5;:..15 since this has a "reasonably large" range (Buttrey and Karo, 2002).

3.10 Cost analysis

A cost analysis was developed to evaluate classifier performance in addition to classification accuracy. Since each population was assigned with its best-fit classifier containing its own features combination this analysis was applied for each population classification separately resulting in an overall system analysis.

A cost function based on the computational cost of the features used by each classifier and its error significance (the cost of classifying a good product as bad and vise versa) was developed. To deal with the computational cost, each feature was assigned a cost value that represents the execution time of each feature and its complexity.

The computational cost also depends on when, how and which features are used for grading. For example, blemish detection may be more or less difficult to determine with green fruit than with yellow fruit. Therefore, an enhanced weight factor was incorporated with the blemish feature reading associated with green fruit classification (Miller, 1985).

The cost function contains a payment function for the misclassification rate together with the classifier cost function for each population classifier and a change detection procedure cost function for the overall hierarchical classifier.

In addition, each feature has a different type of error risk. For example, if a severely bruised fruit will be detected as a good one, the cost will be much higher than a misshaped fruit that was detected as well shaped. Therefore, each feature will also be assigned an appropriate risk weight. The payment function was implemented using a penalty cost weight matrix.
A penalty value was assigned to each classifier according to the error's direction. A classifier that sorts bad fruit as a good one will be assigned a higher penalty weight than the one that sorts good fruit as bad. These weights values will be determined based on the quality standards defined for each produce (by such Israeli organizations as Agrexco – a major fruit marketing body -- Israel Institute of Standards).

For example, Table 3 presents a payment confusion matrix in which the downgrade classification penalty is equal to three times the upgrade categorization (Miller, 1985).

Table 3: Error cost downgrade=3Xupgrade

I\j	**1**	**2**	**3**
1	0	3	3
2	1	0	3
3	1	1	0

This penalty function also corresponds to the magnitude of the difference defined by the number of levels of difference. For example, a classifier that yields two level unit errors (i.e., graded 5 instead of 3 or 1 instead of 3) will receive a higher penalty than the one with only one level error (graded 4 instead of 5).
The objective function was defined as:

$$V_{\cos t} = [\sum_{i=1}^{n} C_i \cdot P_i + V_{F_i}] + \frac{\sum\sum C_i}{N} \cdot V_b \qquad [9]$$

where C_i is the classification confusion matrix of each population (n populations), P_i is the payment cost matrix of each classifier, V_{F_i} is the classifier cost, N is the batch size between two change detection operations, $\sum C_i$ is the number of products sorted by the classifier and V_b is the change detection operation cost.

The classifier cost is described by the following equation:

$$V_{F_i} = t_{F_i} \cdot V_{t_{Fi}} + \sum C_i \cdot V_{C_i} + C_{o_i} \qquad [10]$$

where t_{F_i} is the classifier active time ($t_{F_i} \alpha$ computational complexity), $V_{t_{Fi}}$ is one time unit cost, $\sum C_i$ is the number of products sorted by the classifier, V_{C_i} stands for the classifier activation cost for one sample by means of feature cost and C_{o_i} represent the general costs of the classifier (e.g., hardware operating cost).

The influence of the following parameters on the performance of the classification system was examined:

1. Detection batch size - the classifier structure includes a batch size in which it checks the data for new population entrance (i.e., for each batch of fruits the system checks if this batch belongs to current population or to a new one). Changing the batch size increase/decrease the classification accuracy.
2. '*All features*' batch size - the batch in which all features are activated. After the system assigns a new classifier for the current population it uses its features for the classification task. Nevertheless, in order to keep on detecting changes it needs to check all features occasionally. Therefore, a batch size is defined for the number of samples in which all features will be captured.
3. Current feature batch size - current classifier features are active. Following the previous batch, this is the batch size defined for the number of samples in which only the classifier features are captured.

Figure 8 presents a graphical representation of the data flow through the system and demonstrates the three aforementioned *'batch parameters'*. The change detection procedure is activated *m* times according to the 1st parameter ('*Detection batch size'* is set to *nsamples*).
After the classifier was selected, the corresponding subset of features (e.g., subset F1 may include [f1, f3, f4]; F2 may include [f7 f2 f3] etc….) was set for the current population. At this stage the 2nd and 3rd parameters are activated. This means that until a new population is detected the current subset of features is active for a ***k*** sample batch (i.e., *'Current feature batch size',* subsets are marked as *F1…FN* in Figure 8) followed by a batch of ***d*** samples in which the whole set of features is active (i.e., '*All features batch size*', marked as *All F* in Figure 8). During the procedure of classifier selection the entire feature set is in use.

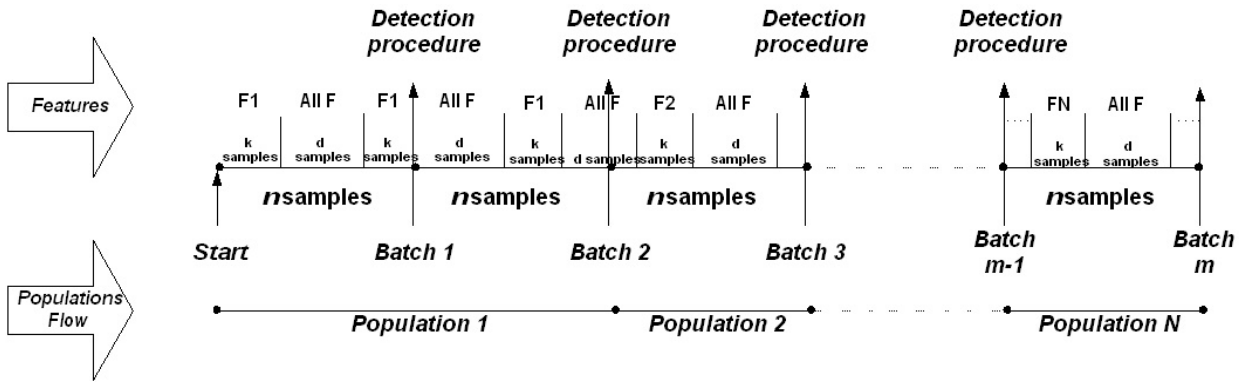

Figure 8: Classifier dataflow and the *batch parameters*

By assigning different values to these parameters different situations can be simulated.

The main objective of these simulations was to demonstrate how to get good classifier performance in less cost than the best one requires. Since determination of the exact cost values is a subjective procedure, different cost settings were compared for each of the simulations.

3.11 Evaluation

System analysis was conducted for a synthetic dataset and an agriculture dataset specially collected for this thesis. System sensitivity to changes in several parameters (e.g., batch size for population detection, data entrance order) was evaluated for the synthetic dataset. Two aspects of system performance were evaluated for the agricultural dataset: population detection and overall classification performance.

3.11.1 Datasets

The synthetic dataset included six different populations, each containing a 1000 data points with seven features. Each feature was created using a random multivariate normal distribution. The data point vectors were labeled with three quality labels.

An olive agriculture dataset was specially sampled as part of this research. A total of 10,550 olives were harvested in two seasons and at different dates. Olive images were acquired with machine vision equipment and analyzed using image processing algorithms. Thirty-one features emerged. To provide a system reference, a human panel classified the olives into four classes, according to standard classification rules for table olives (Diaz *et al.*, 2004).

3.11.2 Performance measures

The population detection performance measure is based on the population overlap. The overlap level can change from full overlap (same population) to separate groups (completely different populations). Two overlap measures were used to estimate the populations overlapping. The first is based on Fisher's discriminant ratio (Duda *et al*., 2001) estimated by the following equation:

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \qquad [11]$$

where $\mu_{1,2}, \sigma_{1,2}^2$ are the mean and variance of two populations.
The second overlap measure was mentioned earlier in this chapter (equation [8] in section 3.8.1).

These two measures are used for all possible combinations of the order in which two populations may appear. The overlap level provides a good estimation for the high level performance. Highly overlapped populations will be difficult to detect. Nevertheless, errors in detecting highly overlapped populations may not produce extreme errors since if they are so much alike and therefore the same classifier can be used. Low overlap or separate populations will be easier to detect. These statements should be tested for the high level part of the system.

The overall classification performance measure is based on two parameters calculated according to the classifier resultant confusion matrix:

- The precision parameter that indicates the ratio between the real classification grades to the results of the classifier grade.
- The mean square precision error (MSPE) parameter that is a more sensitive criterion and which gives each class grade an equal significance.

3.11.3 Sensitivity Analysis

Sensitivity analysis was tested for the following parameters:

- Entrance order – switching between batch orders. Within the synthetic database we'll change between population entrance order to check the system sensitivity for it.
- Change detection batch size – changing the frequency of change detection tests.
- ALL feature batch size – changing the batch size occasions in which all features are active.

- Current feature batch size - changing the batch size occasions in which only the current classifier features are active.
- Features cost – features activated during the simulation will be assigned with a changing cost to check their influence on the classifier overall cost.
- Payment matrix values – each misclassification is included in the payment matrix. Each misclassification degree (i.e., number of grade error units) is assigned with a cost value. Changing these values will influence the classifier overall cost.

4 Algorithms

4.1 Overview

The low level stage of the classifier contains a procedure related to on-line clustering and a procedure to determine the population change measure. The high level stage contains the algorithm for determining the overlap volume and the fuzzy logic classifier selection algorithm and the on-line automatic/human retraining procedures.

4.2 On-line clustering algorithm

The on-line clustering algorithm is based on an algorithm that clusters the produce into distinct populations (Guedalia *et al*., 1999). Each population defines a different location in the feature space. The algorithm, designed to cluster non-stationary data, takes into account clusters which have relatively small mass.

The algorithm can be summarized in three steps: (1) moving the closest centroid to the arriving data point, (2) merging the two closest centroids (creating a redundant centroid) and (3) setting the redundant centroid equals to the new data point. The well-known Euclidean-distance is used for measuring distance. A multitude of features are considered (equation 12).

$$D(X(x_1, x_{2,\ldots,} x_n), Y(y_1, y_2, \ldots, y_n) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad [12]$$

Eventually, in case of a new population, given that the initial case contains k centroids, k-1 centroids are moving towards it while one centroid (the one with the greatest weight) becomes the previous population's representative.

Figure 9 presents the algorithm's stages. One of the main changes to this algorithm that emerged in the course of developing it was in defining the number of centroids needed. The original algorithm adds centroids on-line according to the available free memory and removes the ones with negligible weight in a post-process procedure. Since the goal of this algorithm in the current research is to determine new populations, we used the basic $k=3$ centroids. Whenever a new population was detected, we designated the previous one to be a *history* population and added one new centroid to the two that were already within the area of the new population.

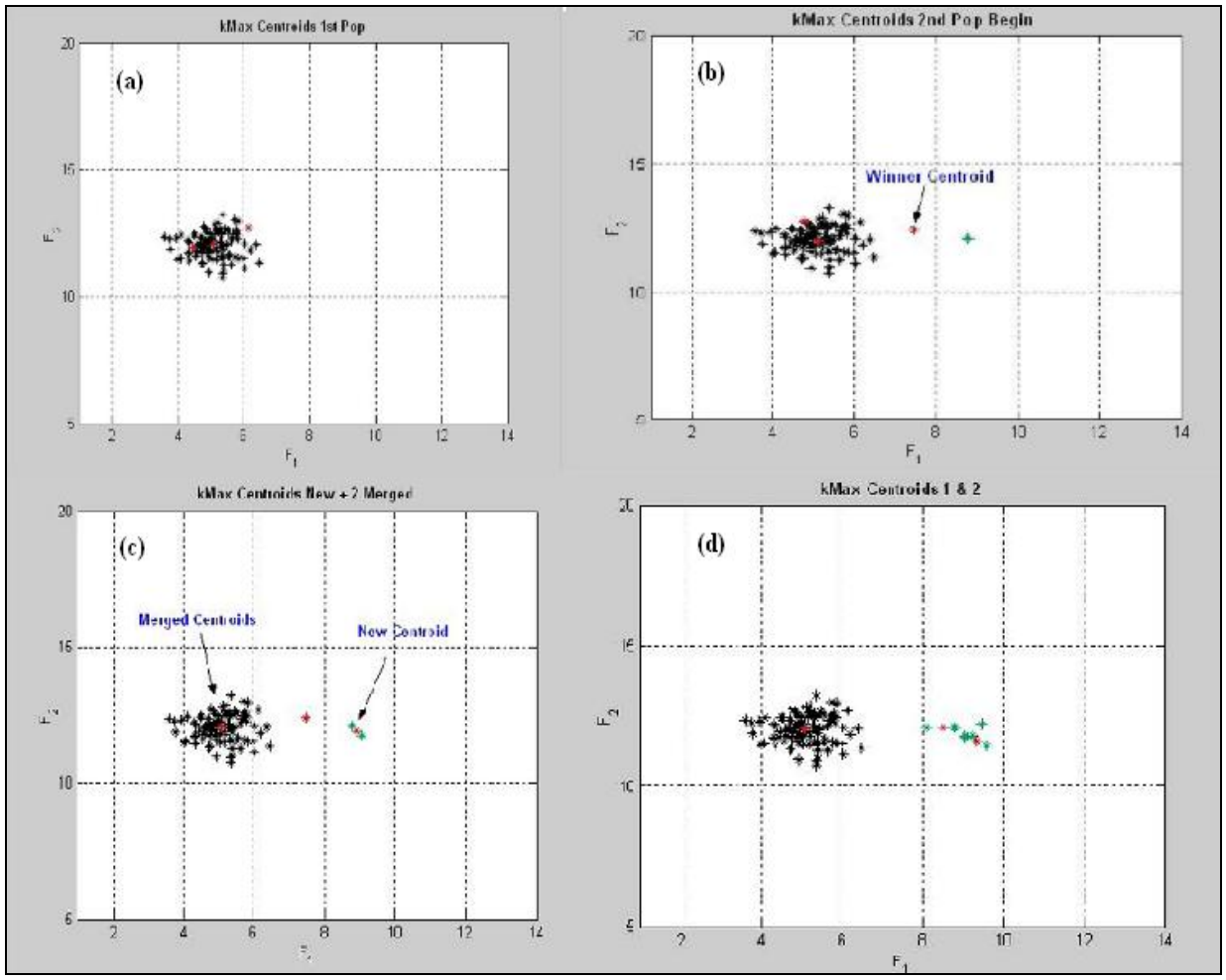

Figure 9: On-line clustering algorithm stages. (a) k (k=3) centroids (in red) after first population enters the system. (b) closest centroid (winner centroid) moves toward new data. (c) merge the two closest centroids and set a new one. (d) k-1 centroids transferred to the new population (in green) area while one centroid left within the previous populations

4.3 Population change detection measures

Population change detection is performed on-line. The system uses three measures to detect the change occurrence and to decide about whether or not to activate the high level. The three measures that were defined were based on clusters size and centroids variance. To track clusters size, their growing gradient is checked after each constant batch of points (equation 13).

$$\nabla_j = \frac{C_j(i) - C_j(i-m)}{m} \qquad [13]$$

Where *m* is the batch size, *C* is the cluster size, j=*1,2,3* and *i* is the current sample index.

This gradient is calculated for each cluster yielding a three-element 'gradient vector', which is compared from batch to batch.

Figure 10 presents changes of the size of two clusters during a sample run within three populations data flow cases. The blue graph represents the gradient of the first cluster size and the dashed green graph represents the gradient of the second cluster size. The three measures defined were:

- *Separate population measure (M1)* - designed to detect a new population completely separated from previous populations. It triggers detection of a population change when the gradient of the largest cluster (within the gradient vector) becomes approximately zero (Figure 10a).

- *Overlapped population measure (M2)* - for the case of two overlapped populations, two stages are required. In the first stage, detection is made regarding when the rate of growth of the larger cluster becomes lower than that of one of the other clusters (within the gradient vector). A population change is detected when the value is above a threshold[4] (Figure 10b), otherwise a second stage takes place after the next *n* samples. If the gradient of the smaller centroid becomes negative, this indicates highly overlapped populations and no population change is detected. As Figure 10c illustrates, this creates a "saw-tooth" graph.

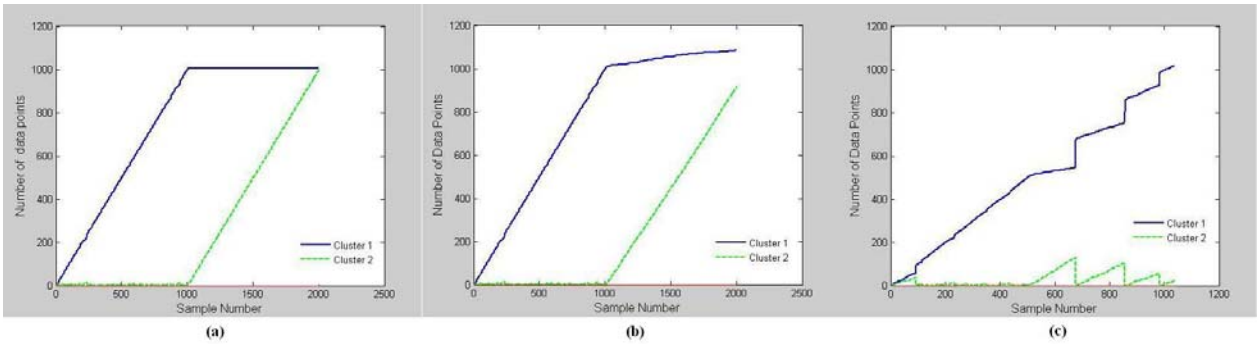

Figure 10: Cluster size in three different cases: (a) two separated populations; (b) two low overlapped populations; (c) two highly overlapped populations (saw-tooth pattern).

- *Centroid variance measure (M3)* - calculates the difference between the current and previous variances of the centroids' locations. For each data sample, the system checks the k=3 centroids for the on-line clustering algorithm. They are kept in a centroids matrix: $C_{i,j}(i = 1,2,3;\ j = 1,2,...,n)$ where n is the number of features and C is value of centroid i at feature j. After updating the centroids, the system calculates the variances of the current centroids, which are held in an aggregate matrix. This matrix contains a n-sized vectors with

[4] All thresholds were determined empirically and analyzed for their sensitivity.

the variance $\sigma_j^2 = \frac{\sum_{i=1}^{k} C_{i,j} - \mu}{N}$ where $\mu_j = \frac{\sum_{i=1}^{k} C_{i,j}}{N}$, for each $j = 1,2,...,n$ features. As each data point enters the system, the matrix grows and is checked for the variance changes in any batch of points. When one of the dimensions (features) indicates a high degree of change, it is checked again after the next n samples; if stability is detected, then a new population is announced.

Figure 11 shows one of these cases in which the centroids variance is changing significantly in one of two feature dimensions. The blue line graph is the variance change in one feature dimension and the green is for the second feature.

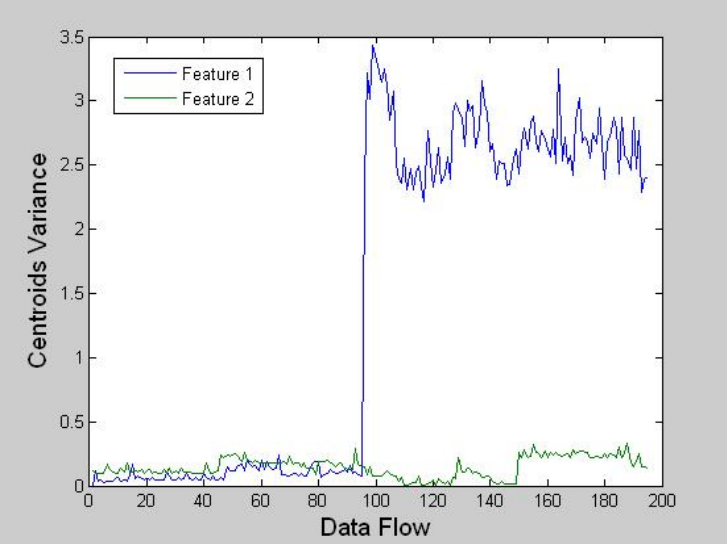

Figure 11: Variance change of the centroids of two feature space

Figure 12 presents the algorithm for detecting population changes. One of the main problems when calculating the distance between centroids with high dimensionality is that local changes may adversely affect the detection. This problem leads to the miss of some changes in the populations. To overcome this problem, the system analyzes, in each feature space, which are the two centroids that 'merged' in the second part of the on-line algorithm. Since k=3, there are three options: centroids 1&2, 1&3 or 2&3. As mentioned above, when a change occurs, k-1 (equals to two in our case) centroids move towards the new population (Figure 9(c,d)). In this situation these two centroids remain the closest for the next batch of points in the feature space that caused the change.

The selection of the same two centroids as the 'merged centroids' in more than a certain number of times in a row (the exact value is determined empirically) in one (or more) feature

space causes an alarm in the system. In this case the clustering algorithm starts running with the 'suspicious' features in which the three measures can be more easily detected.

These features 'suspicious' features are defined as *separate features* since they alarm the system for change. In order to trace these *separate features,* a feature matrix (*FM* in Figure 12*)* is defined in which each column *j* represents the *j*-th feature and each row is the index of the sample.

4.4 Overlap level decision

After the decision about a new population has been made, the system starts to adjust to it. The first stage is to define the new population location in relation to previous populations. As mentioned in section 3.8, the system deals with four optional cases defined according to the overlap measure (Figure 6). This measure (equation 2 at 3.8.1) is checked using the complete 'quality feature' space (mainly to decide on overlap or non-overlap) and then, if all features are overlapped, for each feature separately (to decide upon the overlap level).

The cases are:

No overlap case – the current batch of points (new population) completely differs from all "*history*" populations or there is a minor overlap. In this case a human retraining is activated.

Full overlap case – most of the current batch of points (new population) overlap with one of the populations in the *history* database. The classifier of the overlapped population is applied.

Multi-overlap case - high overlap with more than one population. Since more than one classifier might fit the current population classification, additional statistical measures were used to compare populations distributions across the feature space (Kalmogorov-Smirnov, see below).

Partial overlap case – only an average overlap level between current population and one or more populations. In this case we use the data points from the overlap area as training points for automatic retrain.

The overlap determination procedure is detailed in the flowcharts of Figure 13. This flowchart yields just the 1st, 2nd and 4th overlap cases for each tested population[5]. The 3rd case, as mentioned, is a competitive case in which the system needs to decide what is the best-fit population between several populations that were assigned to either the 2nd or 4th case. The output of this algorithm is a 'case table' (Table 4) where each column represents the 'tested' population case decision, its overlap measure (which is a value between zero to one) and its average Kalmogorov-Smirnov (KS) statistic.

[5] The three thresholds mentioned in the flow chart were defined empirically using several system iterations.

Table 4: The case table: each '*history*' population is graded with three measures in relation to the current population

History Populations	Overlap Case	KS level	Overlap Level
1	1/2/4	0-1	0-1
2	1/2/4	0-1	0-1
3	1/2/4	0-1	0-1
…	…	…	….
…	…	…	….
n	1/2/4	0-1	0-1

After obtaining the case table results, the system must define the suitable action to take. The algorithm, which performs this function, is presented in Figure 14. This algorithm counts the 2^{nd} and 4^{th} cases. If there is more than one 2^{nd} case, the system selects the one with the highest overlap value together with the lowest KS statistic. Case 3 is determined when the system detects more than one population labeled in the table with case 4. This means that there is more than one optional training set for the current population. In this case the problem is what population training data to use. The decision is based on comparing overlap values, KS statistics and distribution comparison.

Step 0) $ind = ind + 1$ /*ind= index of new produce feature vector enter the system*/

Step 1) Run the on-line clustering algorithm $\forall\ produce_{ind}^{f_j}$ /*update the centroids*/

Step 2) *For* $j = 1{:}N$ /* N – number of features

$MinDist^j = \min(D_{12}^j, D_{13}^j, D_{23}^j)$ /* D_{12}^j – centroids 1 & 2 distance in feature j */

$FM_{j,ind} = MinDist^j$ /* $FM_{j,ind}$ – feature matrix $f_j, sample_{ind}$

/* $MinDist = 1:3$ */

End /* End For */

Step 3) *If* $(ind \mod n)$ /* If periodic check e.g., every n samples */

Update the three measures $M1, M2, M3$

if $(M1 = 1)$ /* $M1$ – Clusters size gradient */

/* if $M1$ is 'on' $\Rightarrow M1 = 1$ */

$FlagChange = 1$ /* If $M1$ is 'on' than its a change */

End /* End if */

if $(M2 = 1)$ /* $M2$ – Saw tooth measure */

/* if $M2$ is 'on' $\Rightarrow M2 = 1$ */

$Counter_{M2} = Counter_{M2} + 1$ /*Update $Counter_{M2}$ */

if $(Counter_{M2} > 1)$

$FlagChange = 1$ /* If $M2$ repeat 2 periods than its a change */

End /* End if (counter) */

else

$Counter_{M2} = 0$ /* Set counter to zero */

End /* End if $(M2)$ */

if $(M3 = 1)$, /* $M3$ – Centroids variance measure */

/* if $M3$ is 'on' than $\Rightarrow M3 = 1$ */

$Counter_{M3} = Counter_{M3} + 1$ /*Update $Counter_{M3}$ */

if $(Counter_{M1} > 1)$

$FlagChange = 1$ /* If $M3$ repeat 2 periods than its a change */

End /* End if (Counter) */

else

$Counter_{M3} = 0$ /* Set counter to zero */

End /* End if $(M3)$ */

For $j = 1:N$ /* N – number of features

If $(\sum_{i=ind-n}^{ind} (FM_{j,i} = 3) > Thr)$ /* If merged case 3 > threshold

$FlagFeaturAlarm = 1$ /* Set the alarm flag active */

/*Update the relevant features with $FM_{ind,j}$ */

End /* End If */

End /* End For */

End /* End If */

Figure 12: New population detection procedure. The three measures and feature tracer.

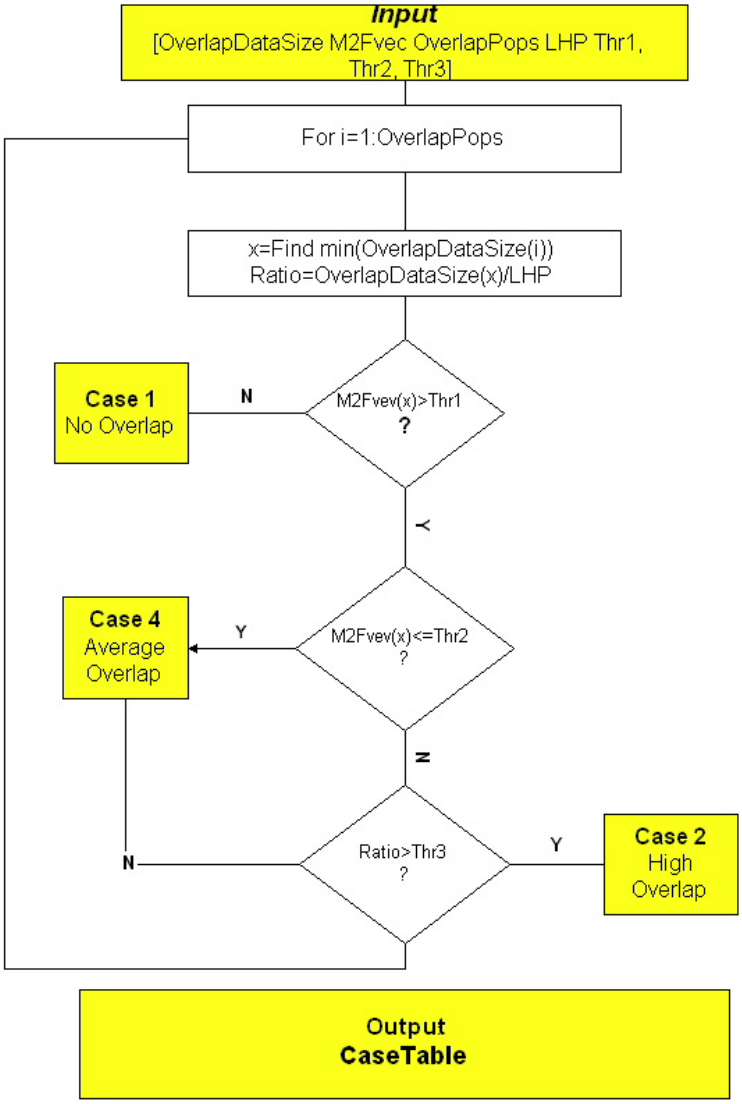

Figure 13: Overlap case determination flowchart

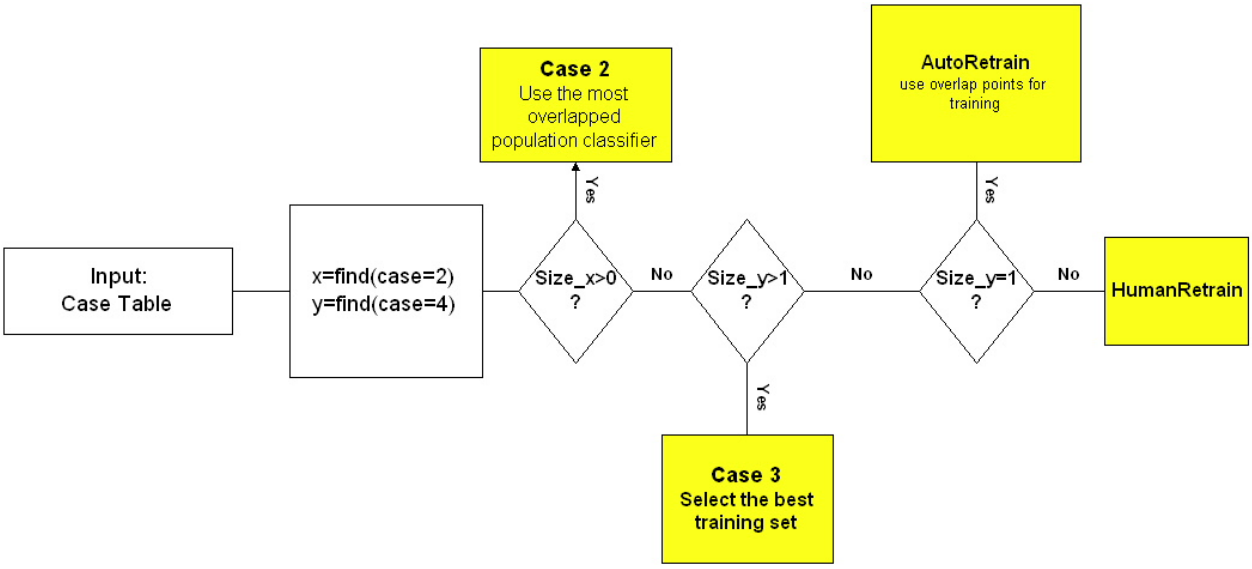

Figure 14: Procedures according to the ‘case table’

4.5 Classifier selection algorithm

In all cases, except for the *full overlap* case, the system applies a retrain procedure. Whether this procedure is automatic or manual (human) there is a need to select the best-fit classifier after all classifiers are trained with the new training set. The classifier selection algorithm was implemented using fuzzy kNN classifiers that function as an input to a fuzzy rule-based decision system. The labels of the sample neighbors are combined into the soft output label $\mu(x)$ for $x \in \Re^n$ using the Keller algorithm (Keller *et al*., 1985).

The output label, or the membership vector, $\mu(x)$, is defined as:

$$\mu_i(x) = \frac{\sum_{j=1}^{k} l_i(z^{(j)})(d_j)^{-\frac{2}{m-1}}}{\sum_{j=1}^{k}(d_j)^{-\frac{2}{m-1}}}, i = 1,...,c \qquad 14$$

where *z* belongs to the set of *k* vectors which are closest to the sample *x*; $l(z^j)$ is the soft label of the *k* nearest neighbors; *d is the* Euclidean distance; and *m* is a fuzzification parameter.

There is a different membership label related to the quality grade (*c*). Finally, a vector where each component represents the neighbor's quantity for each quality grade is obtained. For the next stage the vector value order is maintained from low to high.

The system (Figure 15) was composed of three fuzzy inference systems (FIS - Low, Average and High) using the Mamdani method (Matlab, 2002). The fuzzy system outputs are summed into an aggregation matrix. Each cell in the aggregated matrix sums the values that belong to it.

The matrix in Figure 15 fits three quality levels (A, B and C) and the "defuzzified" value of the fuzzy system input to the matrix can be seen. The columns in the matrix represent the quality level while the rows represent the FIS. For example (Figure 15), a classifier that has a low number of neighbors (*kNN_n(1,1)*) that belong to class A enters its value into the low FIS and the result, '*Low_Ca_Res*', enters the matrix in the class A column in the Low row. A cell summing procedure is activated to calculate the quality grade.

The cell summing procedure was determined using the following equation:

$$Column_i = w_1 \cdot Low_i + w_2 \cdot Ave_i + w_3 \cdot High_i \qquad 15$$

Where, $w_{1,2,3}$ - 0.25, 0.75, 1 were empirically selected.

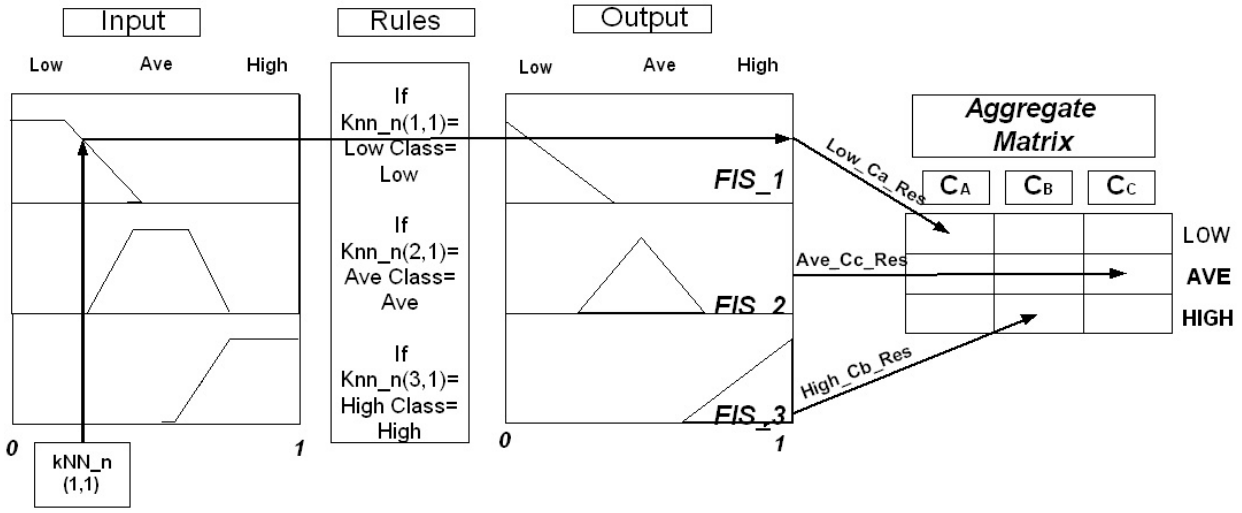

Figure 15: The fuzzy system and aggregation matrix

The overall classification result is derived from the column with the highest result (best wins). To ensure that the highest result is significant, the result variances were calculated. When the variance is low, additional comparisons are implemented on the *High* (best wins) and on the *Low* (worse wins) rows. Based on these options the classification for the current sample is defined.

Since the fruits enter the system in batches, their feature subset is employed for the next batch of fruits to save the computational cost of the redundant features.

4.6 On-line automatic retrain

Whenever the system detects cases of overlap regions (i.e., cases 3 or 4) an automatic retrain procedure starts (Figure 16). This procedure begins with the decision on the retraining points from one or more overlap populations. The next step is to retrain the fuzzy kNN classifiers. The next batch of points (together with the later detected as new population) is classified using these classifiers. The classifiers results enter the fuzzy logic rule-based system (detailed in the previous section). According to the fuzzy system results, the current population classifier is selected.

Since the population detection and the overlap decision procedures require a batch of data points that can neither be ignored nor classified by the previous classifier, the system checks for the closest population and uses its classifier to grade the points (Figure 17). In the classifier selection procedure, the data points grade is the fuzzy system output.

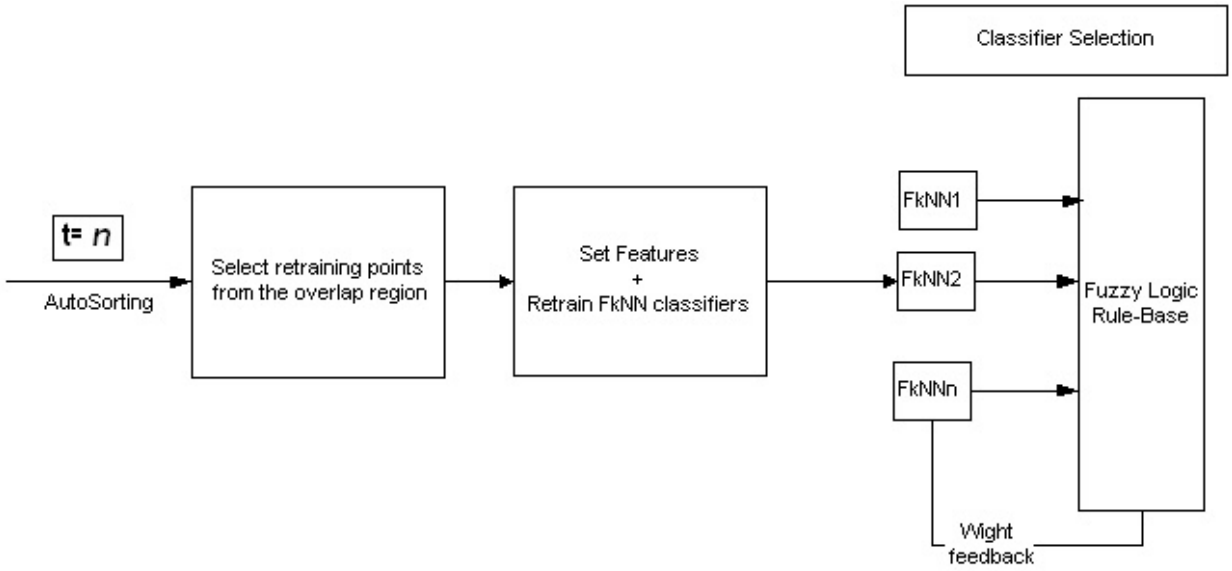

Figure 16: Automatic sorting based on the overlap regions (t=n)

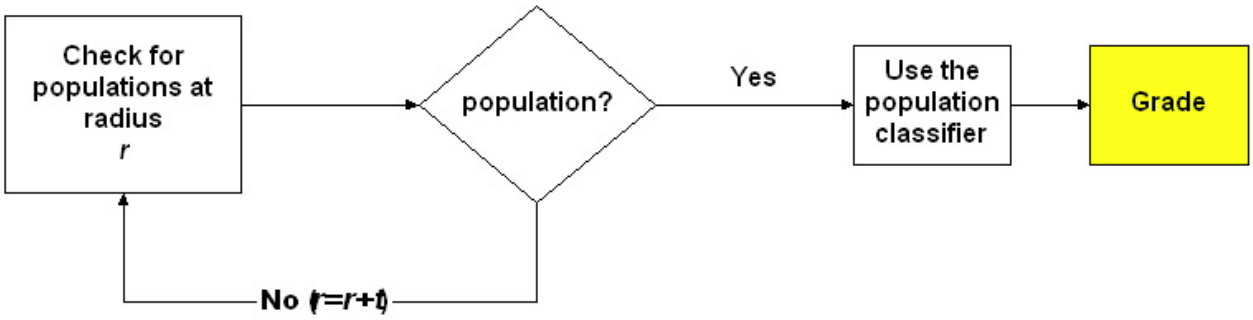

Figure 17: The default classifier selected while system adjusting

4.7 Human retrain

When the system detects a non-overlap case, it uses a pre-defined training batch of data points to initiate human retraining procedures. This means that all batch grades are known a-priori. According to this information, the system starts working in an off-line mode, performing the feature selection procedure and retraining the fuzzy kNN classifiers. As in the automatic phase, it uses the fuzzy system to select the best-fit classifier.

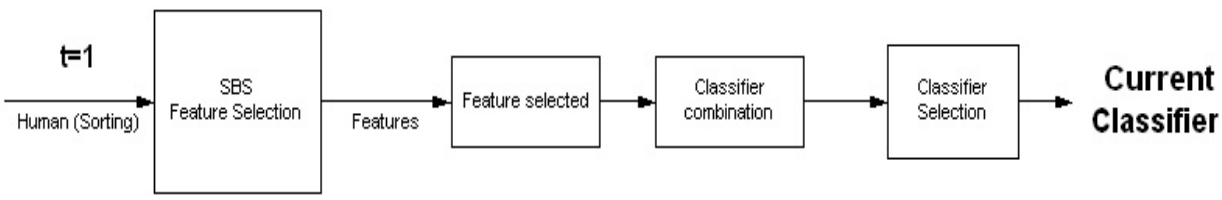

Figure 18: Human off-line sorting

5 Experiments

5.1 Overview

System performance was evaluated using a synthetic database and an agricultural database, both especially designed for this research. The synthetic database was constructed to independently evaluate each subsystem (population detection and classifier selection) and then the integrated system. These initial analyses made it possible to refine the integrated system. The agricultural database was used to evaluate the integrated system for real world conditions.

To evaluate the integrated system we compared its results to other classifiers in different configurations, i.e., different training and testing settings. Performance analysis included costs analysis. System sensitivity to the size of predefined parameters, *detection batch size, 'All features' batch size, current feature batch size*, and different costs was evaluated

5.2 Synthetic dataset

5.2.1 Structure

The synthetic dataset was composed of six different populations (Figure 19). Each population contained 1000 data points with seven features. Each feature was created with a random multivariate normal distribution.

An original population was generated based on an initial mean and variance feature vector determined empirically. Each component in this vector represents the Gaussian distribution parameters of each feature. Each feature f_i is distributed normally $N(\mu_i, \sigma_i^2)$ where $i = 1,2,...n$ - n was set to seven. Five additional populations were created by changing the third component of the original mean vector by significant value. The changing of this value simulates change of one feature dimension (e.g., color intensity feature) that influences the quality of the produce. The data point vectors were labeled with three quality labels, corresponding to three fruit grades, implementing by a constant reference function.

Each population contains different grade distributions. The first two populations are alike and highly overlap with the original population ('Pop_Base' in Figure 19) in the feature space. The third and fourth populations are alike but differ from the first three. The 5th population is completely different from all other populations.

The synthetic dataset was used to:

- Test the system capability to adjust to population changes
- Check the performance of the three measures that were developed for detecting new populations
- Check the system for the overlapping measures between the populations
- Check the system's sensitivity to the order (sequence) that populations are presented to the classification system

5.2.2 Analysis

5.2.2.1 Classifier selection

The classifier selection sub-system (high level) was analyzed in two steps. For each step it was assumed that each new population was identified correctly. In the first step, the classifiers were trained on 500 samples randomly selected from the original population. Ten different fuzzy kNN classifiers, differing in the number of features and the specific features, were empirically selected and compared for each population. This process was conducted 10 times; each time 500 different points were selected.

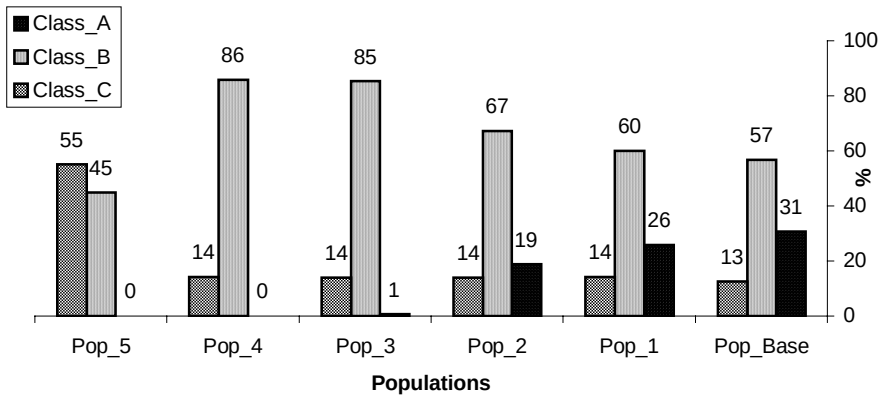

Figure 19: Quality labels distribution within each of the six synthetic populations

In the second step, the best classifier was selected based on presentation of the first 100 points of each population to the fuzzy inference system (FIS, section 4.5). The remaining 900 points were classified according to the best-fit classifier.

Classification results for the whole dataset, including all six populations, are compared to results that would have been obtained by a classifier using all features (denoted as 'All').

5.2.2.2 Population change detection

Population change detection (mentioned in section 4.3), together with the on-line clustering algorithm were evaluated by comparing the real points where there was population change to the points where the system detected the change using these measures. A time series analysis facilitated understanding of the complete on-line classification procedure.

The overall system classification results are compared to the predefined classifier that uses all features and implemented on each population separately. In both cases the classifiers were only trained with the first 500 points of the first population.

5.2.2.3 Integrated system

The main experiment was conducted on the integrated system that contains the overlapping measure and selects the actions accordingly (see the flowchart in section 4.6).

Initially, classifiers were trained with the first population but when the system found no overlapping between the current new population and the previous one, it 'announced' the need for human retrain.

A final analysis included the system sensitivity to the population's entrance sequence.

5.3 Agricultural dataset

5.3.1 Structure

To evaluate performance in real world conditions, a dataset for olives was specially constructed. The dataset contained 10,550 olives from 12 varieties, harvested from Ramat-Negev fields in the south of Israel. The olives were harvested in the course of two different seasons and at different times during the season (30/11/2004; 16/12/2004; 15/12/2005). Appendix II presents more specific details about the harvest.

The olives were classified into 4 classes (Figure 20) based on several literature references (Diez *et al*., 2004) concerning the issue of table olives classification. Image processing algorithms were developed to determine the following features: color, color homogeneity, shape and defects (Appendix I). The resulting dataset includes 10,550 feature vectors with 31 elements (features) per vector (Appendix III). A two-member panel graded the olives into four quality grades and into color and defects grades. These grades function as the real grade (label) reference to the system.

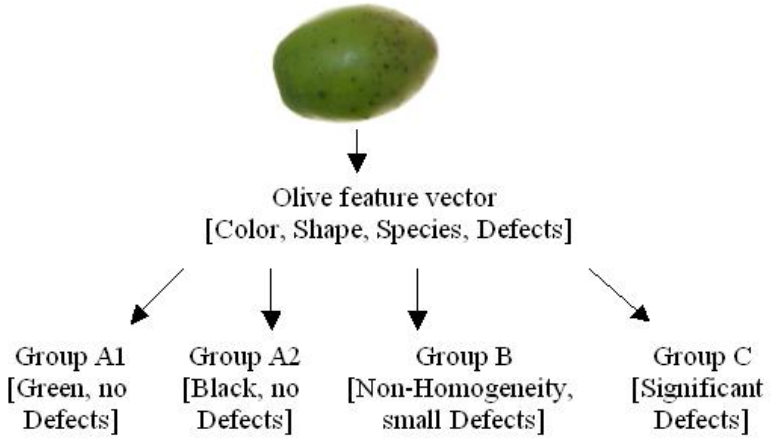

Figure 20: Olive sample and its grading options

5.3.2 Analysis

5.3.2.1 System evaluation

The on-line system was tested for two cases:

- Populations entering the system one-by-one. Each population was checked for overlap and similarity relative to the previous populations. Similar to the synthetic data experiments, the system indicates when human retraining is necessary. In this case, the system used pre-labeled training data for the current population.
- A collection of representing populations is pre-defined. A *population database* is defined a-priori using the knowledge of the overlap and similarity levels between the populations. This database, referred to as the population base, contains several populations that best represent the feature space and contain the rest of the populations in their ranges.

After the population base is set, the other populations are entered and classified on-line and their relationship (similarity) with the population base is adapted.

Each stage, population detection, feature selection and classification error, was evaluated separately.

The population base assembly was based upon an a-priori similarity test between the populations in order to find the ones that most frequently appear and the ones that best span the feature space.

The integrated system for on-line adaptive classification was compared to the following classifiers that can be found in the literature:

1. A decision tree (DT) classifier trained for each population base.
2. DT classifiers, each trained on a different population and tested on the rest of them.
3. A classifier (FkNN) trained on samples from all populations.

5.3.2.2 Cost analysis

System performance was evaluated for the following values of the batch size parameters:

1. Detection batch size - the classifier structure includes a batch size in which it checks the data for new population entrance (i.e., for each batch of fruits the system checks if this batch belongs to current population or to a new one). Changing the batch size increase/decrease the classification accuracy.
2. '*All features*' batch size - the batch in which all features are activate. After the system assigns a new classifier for the current population it uses its features to the classification task. Nevertheless, in order to keep on detecting changes it must check all features occasionally. Therefore, a batch size is defined for the number of samples in which all features will be captured.
3. Current feature batch size - current classifier features are active. This is the batch size defined for the number of samples in which only the classifier features are captured.

A total of 504 simulations were conducted (7 cases of *change detection operations*: batches of 20,30,40,50,60,70,80; 8 cases of *all features* activated: 5,10,15,20,25,30,35,40; 9 cases of *off features* activated: 1,5,10,15,20,25,30,35,40.

Sensitivity to the cost of features and the penalty of misclassification was additionally evaluated.

5.3.3 Experimental design

The final class of each olive was determined as the average value of the evaluation of two panelists. The panelists analyzed the olive images using a Graphical User Interface of Matlab (Matlab R14, 2005). See Appendix IV, Figure 58 for further details.

Classification included three quality measures: color and color homogeneity (7 grade levels), defects (5 levels) and overall quality grades (4 levels). The overall quality grades were used as the olives labels while the other grades were used to validate the image processing results.

Table 5 presents the tested populations and their best-fit base-populations. Appendix V presents the complete table for the similarity measures values between populations. This similarity was determined using the predefined overlap measure (equation 8 chapter 3.8.1) and the Kolmogorov-Smirnov test to compare the distribution of two samples. These two measures were averaged over the quality features of the populations and used for comparison.

In addition to the similarity measures, the *population database* includes the quality grades distribution of each population.

Table 5: Population data base and fit populations

Population Base	**1**	**3**	**8**	**12**	**14**	**16**	**18**	**21**
	2	6	4	7	17	5	19	15
			9	13		20		
			10					
			11					

Quality levels and their distribution varied according to the maturity of each population's produce (e.g., a premature population contains levels 1-3-4 which means green, non-homogeneous and defected olives, while other populations may contain only levels 3-4 which indicates a midway maturity population of poor quality). Each quality level was also distributed along its feature scale. The quality level distribution was different for each population.

Figure 21 presents the distribution of the quality levels in each population for a single feature (i.e., one of the tested quality features). The difference in the quality levels, in the quantity and span manner, is important for understanding the similarity domain since the overlap and distribution levels alone cannot determine population similarity. Quality grades are also necessary.

The following two figures emphasize the importance of using the overlap and KS measures in the population's similarity estimation.

Figure 22 presents a statistical comparison using the box plot method. This is a graphical way to see population range coverage. Figure 23 presents the Kolmogorov-Smirnov test of two different populations for a specific feature. Two cases are presented: two fully overlapped populations (b) and two separate populations (a). The plot-boxes include the data of each population from its first to third quarter while the borders outside of it (also known as the 'mustache') contain 1.5 interquartile ranges from these quarters. The '+' points are the outlier data points.

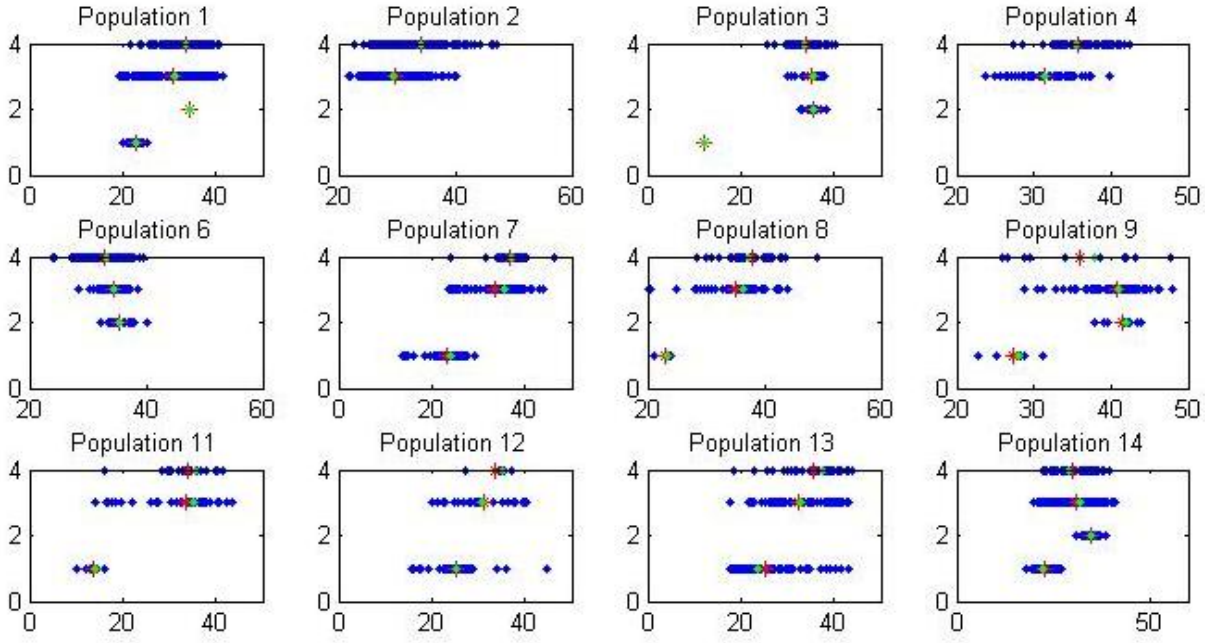

Figure 21: Population quality level distributions for a single feature. Each figure describes the feature value (horizontal axis) as a function of the grade level (1-4) (verical axis). Mean and median are marked.

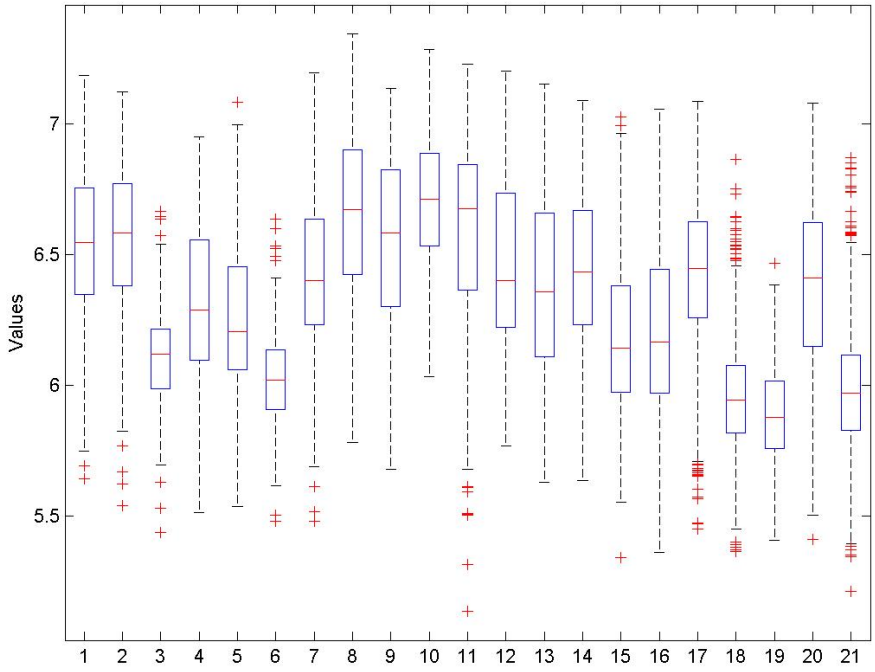

Figure 22: Box-plot of the 21 populations using the mean Hue color feature- vertical axis is the label

The KS test emphasizes the difference between the cumulative distribution functions (CDF) of the two populations checked for overlapping.

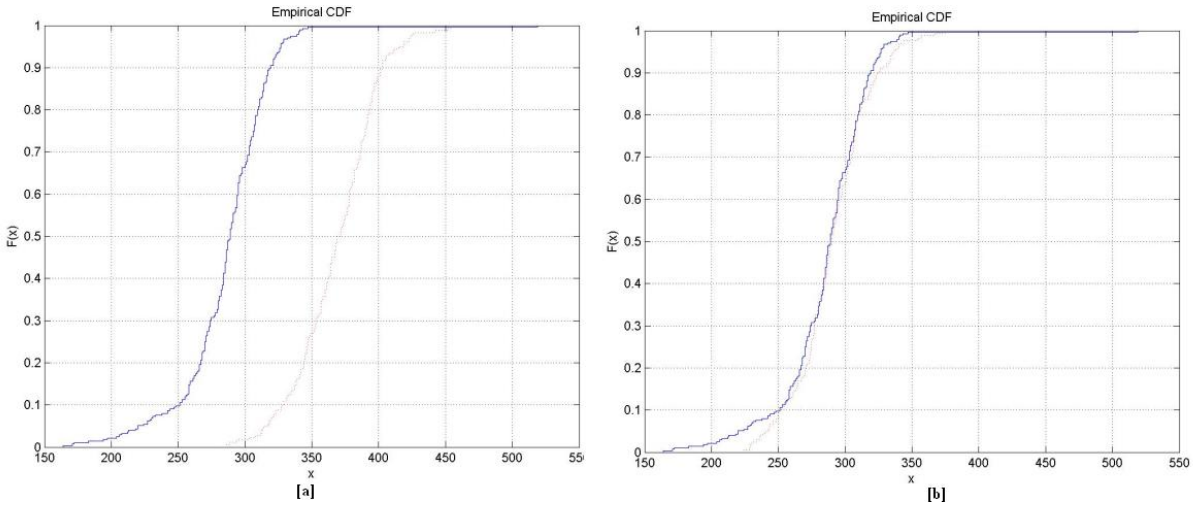

Figure 23: Two populations comparison using KS-test. (a) two separate populations and (b) two overlapped populations. The evaluation applied on the same feature.

Classification performance and its corresponding cost levels are presented using the Matlab (2005) software contours that represent 3D domain on-to 2D where one of the three batch size parameters is fixed. In addition, two main characteristics – features cost and payment matrix values - of the cost function was change to test their influence on the system cost.

5.4 Results and discussion

5.4.1 Synthetic data

Classification results for all six populations within the whole dataset are compared to results that would have been obtained by a classifier using all features (denoted as 'All').
Average results (Table 6) indicate that the system adjusts to population changes by choosing a different classifier at each step.

All classification results are poor when the new population is significantly different (e.g., Pop_5). The values marked in bold in each row indicate for each population the classification accuracy of the classifier the fuzzy system selects. It can be seen that this does not yield the best classification accuracy for each population (except for Pop_1). This is because the fuzzy system in yielding its classification during the selection procedure takes into account all classifiers results. Nevertheless, the system eventually performs well (Table 7) because it weights the classifiers for each data point so that the bad performance classifiers have a

decreasing effect on the final decision. Increasing the number of data points for selecting the classifier may improve the final results.

Table 6: Classifier selection results

Population/ Features	**All**	**1_3_5_7**	**3_6**	**1_3**	**2_5**	**4_7**	**1_5_7**	**2_3_6**	**4_5_7**	**2_4_6**	**2_3_4_6**
Original	75.0	77.5	48.9	**69.2**	49.7	55.7	76.9	49.0	56.7	52.0	51.7
Pop_1	72.5	**75.6**	55.0	72.1	52.4	56.6	75.5	53.8	57.0	53.4	56.0
Pop_2	64.9	69.5	62.4	68.1	51.4	53.6	**67.8**	60.1	53.1	51.1	55.8
Pop_3	53.9	63.6	84.6	64.9	58.9	55.5	57.8	**78.0**	51.1	56.5	67.4
Pop_4	65.5	84.2	**85.8**	90.9	59.7	53.4	53.9	85.5	50.4	55.4	82.4
Pop_5	30.2	36.4	44.7	39.9	31.8	27.4	**34.1**	44.4	26.1	27.6	39.7

Figure 24 and Table 7 summarize the overall classifier results.

The time series analysis (Figure 24) shows the values of parameters of the multi-stage classifier throughout the classification process. The time series analysis is divided into three sub-groups:

1. Fuzzy Process - denotes the instances where the high level was triggered. It also indicates whether a new population was detected.
2. Measures - denotes the instances where the three change detection measures was triggered. Instances presented as on/off states.
3. Features - this subgroup indicates which features are used at each stage.

The fuzzy process and the measures states indicate that the on-line clustering algorithm did not detect a new population during the transition to the second and third populations. These first three populations (Laykin *et al*., 2004) overlap. It can be pointed out that the second measure detects a change between P2 and P3 but at its second stage (as described earlier-0) this detection was not confirmed. The first measure detects the next three populations (P3 to P4, P4 to P5 and P5 to P6). The second and third measures support the detection of population change in two cases.

Features 2 and 4 were never selected while feature 3 is used for most of the populations. Using feature 3 is reasonable since this feature created the difference between populations. It can be seen that various features in different combinations were selected during classification.
This mainly means that the system is flexible and adjusts to the new entrance populations.
Features that are marked as 'YES' at any stage of the classification process indicate that they are included in the current classifier. For example at the transition from *P4* to *P5*, the feature subset changes from [*F1, F5, F7*] into [*F3, F6*].

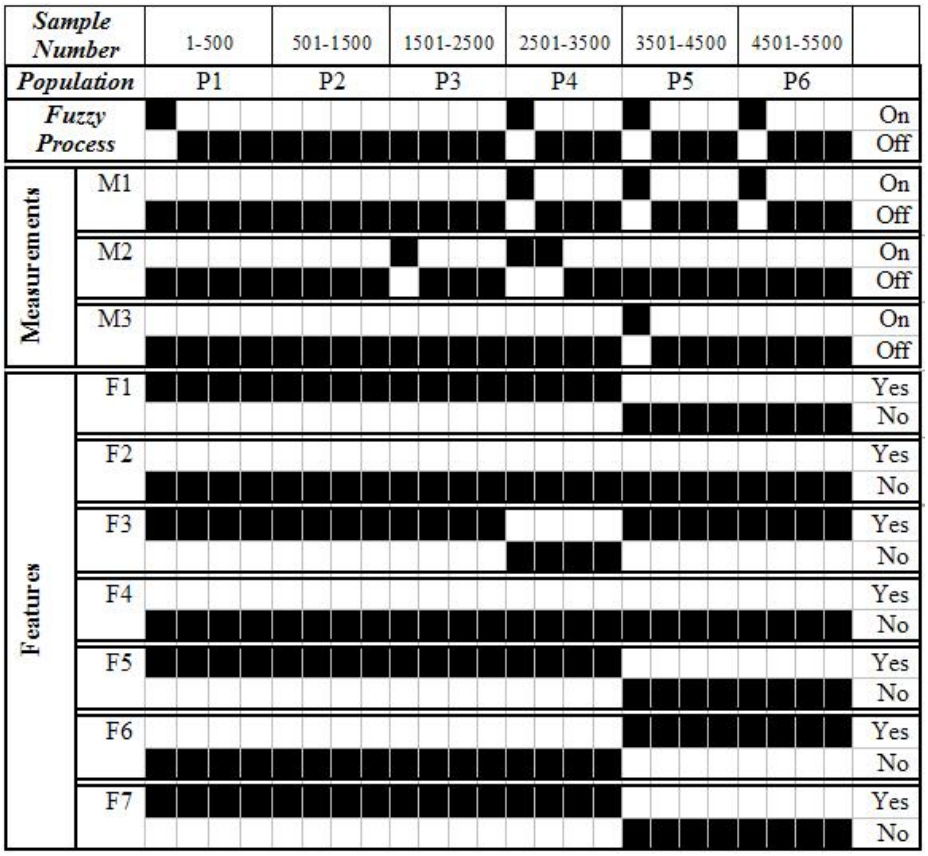

Figure 24: Parameters of the hierarchical classifier throughout the classification process

Table 7 presents classification accuracy by depicting the correct classification percentage as well as the percentage of error when classifying adjacent classes in one or two levels of proximity. Misclassified data represents the difference in quality grades.

Results indicate that the overall classification accuracy of the on-line classifier is better by 12% as compared to the kNN classifier that used all the features. Classification accuracy of individual populations improved between 5 to 12 percent with only one level of misclassification in most cases (except for Pop_5 which yielded poor classification results).

Table 7: Results comparison

Populations	**On-line Classifier**				**All_Features**		
	Classifier	Exact (%)	±1 Missed (%)	±2 Missed (%)	Exact (%)	±1 Missed (%)	±2 Missed (%)
Original	(1_3_5_7)	73.7	25.9	0.4	75.3	24.7	0
Pop_1					72.4	27.6	0
Pop_2					62.9	37.1	0
Pop_3	(1_5_7)	66.9	32.9	0.2	47.8	52.2	0
Pop_4	(3_6)	85	14.9	0.1	73.5	26.5	0
Pop_5	(3_6)	49.6	50.1	0.3	26.4	51.5	22.1
Total_Percent		**70.3**	**29.4**	**0.3**	**58.3**	**37.7**	**4**

System sensitivity to the population entrance sequence is demonstrated by two examples using the 'optimal' classifier that was trained for each population (Tables 8-9). These results imply that when the third population is trained with the overlapping data of the second population, all classifiers yield poor results (~38% misclassified - Table 8). However, if the third population is trained with the fourth population's overlapping data, a major improvement is achieved (~10% misclassified - Table 9). This implies that results will improve if the fourth population enters the system before the third population.

Table 8: Full human run versus on-line run

Pop	**offline Fknn all features (Each 500 Train, k=5)**	**Online**	**Features selected**
1	0.80080	0.809	[147]
2	0.82766	0.66	[134]
3	0.96393	0.629	[135]
4	0.94990	0.83	[345]
5	0.80321	0.81	[35]
Overall	**0.8629**	**0.75**	

Table 8 also presents the results of an off-line classifier that was trained each time with 500 data points (50%) for each population.

Table 9: On-line run with order change

Changed order	**On-Line (200 human retrain, 1st with 500)**	**Case**
Pop		
1	0.8061	human
4	0.928	human
3	**0.9118**	**Overlap(case 4)**
5	0.9014	human
2	**0.7535**	**Overlap(case 3)**
Overall	**0.8552**	

It can be concluded that for a specific population's entrance sequence, equal results can be achieved using human retrain only three times (Table 9) instead of five (Table 8, at the offline case). Furthermore, the system uses only part of the feature space for the on-line classification.

Table 10: Overlap measure values between populations
(significant values are marked (overlap>0.2))

	1:500	**501:1500**	**1501:2500**	**2501:3500**	**3501:4500**	**4501:5498**
Pop Order	1	2	3	4	5	6
1		0.7826	0.4865	0.14731	0.00786	-0.13459
2			0.3915	0.10431	0.103475	-0.143743
3				0.47117	-0.10724	0.0282779
4					-0.239710	0.2089594
5						-0.341975
6						

Table 10 illustrates another issue related to the case of more than one population overlap. Population 3 has almost the same overlap measure for both the second and fourth populations. As aforementioned, the fourth population yielded better results. This is due to the fact that populations 3 and 4 have the same grade level distribution (*b* and *c*), while the second population includes ranges of all grade levels. Accordingly, when a new population has several overlapping populations, the problem becomes a question of which population training data to use. This is solved by using distribution and symmetry measures as described in the agricultural dataset experiment design (section 5.3.3).

5.4.2 Agricultural data

Table 11 presents the results of the on-line classifier with the *population base*. The graphical stream of data and its area of change are presented in Figure 25. The overall accuracy of the on-line classifier was 85.3%.

The first two columns present a comparison of the real population change location and the system change population detection. The third and fourth columns include the quantity of data points of the new and base populations. The fifth to seventh columns include the average of the similarity measures. Columns 8-11 respectively include the overlapped data points, the overlap case (1-4), the appropriate base population (i.e., the one that was selected to classify the new population) and the final accuracy classification.

Table 11: Overlap table results for the *'populations base'* run

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Pop Index	***Real Index***	***Classifier Index***	***Size base***	***Size new***	***OM2***	***Mean OM2***	***Mean KS***	***Overlap points***	***Case***	***Base Pop***	***Class***
2	1	1	692	80	0.15558	0.86152	0.29751	575	4	1	0.9768
4	461	460	692	81	0.08053	0.81556	0.25504	364	2	1	0.83
5	626	649	145	92	0.03234	0.75869	0.23549	75	4	12	0.8337
6	1053	1086	181	95	0.00487	0.65944	0.19838	109	4	3	0.81
7	1331	1336	145	85	0.06263	0.79758	0.26319	99	4	12	0.8197
9,10	1637	1643	180	98	0.05747	0.79884	0.22608	114	2	8	0.828
11	1971	1985	180	96	0.07240	0.81346	0.19107	144	2	8	0.8985
13	2153	2181	145	80	0.12077	0.84060	0.19869	101	2	12	0.8964
15	2480	2489	974	92	0.05908	0.80238	0.15572	850	2	21	0.8656
17	3395	3403	505	98	0.08019	0.82787	0.19271	289	4	14	0.8276
19	4513	4527	704	94	0.00003	0.43760	0.31567	292	0	18	0.8476
20	4876	4887	978	94	0.00559	0.68356	0.18078	554	4	16	0.8862

All population changes were detected except of the change between populations 9 and 10. Since these two populations belong, according to Table 5, to the same population base the average accuracy remains good related to the overall accuracy.

Compared to Table 5, the results here indicate that most of the populations selected their predefined *population database*. The two exceptions are populations 4,5 and 19. In the case of population 4, the selected retrain population is 1 instead of 8 because of better overlap value (Table 20, Appendix VI). Other measures were similar.
In the case of population 5, the overlap and KS measures were similar while the skewness pointed out that population 12 is more suitable than 16 (as in Table 5).
The skewness comparison is presented in Table 12, which is partial of the full table presented in Appendix VII that contains all populations cases. The features compared here are the most useful among the base population classifiers. The comparison is between the mean and skewness values of the populations at theses common features. The base populations together with the current one, that is currently checked for similarity, are indexed in the "Pop Base" column. The selected and current populations measure values are marked in bold font. When a base population classifier is not based on one of the compared features a zero value is assigned. "Base population" that has low overlap with the current population contains a row of zero values (i.e., no need for skewness measure). The "Current rows" include all features since no sub-group was yet selected. Population 1 was more suitable to population 4 than population 8 even though the tendency to one of them is not significantly indicated in the table. Population 5, on the other hand, indicates a significance tendency to population 12 over population 16. Population 19 got low measure values for all populations and was eventually classified according to the highest values (i.e., below the threshold set for it).

Table 22 in Appendix VIII presents details of the full on-line run. Results indicate that 13 cases required human retrain. Two cases yielded very poor results (less than 70% classification accuracy) probably due to bad *history* population selection. These results together with the plurality use of human retrain is due to the system sensitivity to entrance order. The system yielded 81% classification accuracy but the extensive use of the human retrain indicates a significant need for the predefined population base.

Table 12: Skewness measure for similarity (Population 4&5)

#	PopBase	Feature 20		Feature 26		Feature 27		Feature 28	
		Mean	***Skewness***	***Mean***	***Skewness***	***Mean***	***Skewness***	***Mean***	***Skewness***
4	*Current*	**1.040473**	**0.387228**	**0.402134**	**0.510796**	**0.029511**	**0.787229**	**5.844775**	**-0.48129**
	1	0	0	0	0	**0.029479**	**0.426652**	**5.844532**	**0.057617**
	3	0	0	0	0	0	0	0	0
	8	0	0	0	0	0.023588	1.4363	6.064877	-1.11302
	12	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0
5	*Current*	**0.470174**	**1.346545**	**0.029131**	**0.38345**	**0.065649**	**-0.0697**	**4.724809**	**0.406449**
	1	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0
	12	**0.687794**	**1.317574**	0	0	**0.048626**	**0.032254**	**4.95899**	**0.483001**
	14	0	0	0	0	0	0	0	0
	16	0.575333	0.846029	0	0	0.088865	0.264578	0	0
	18	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0

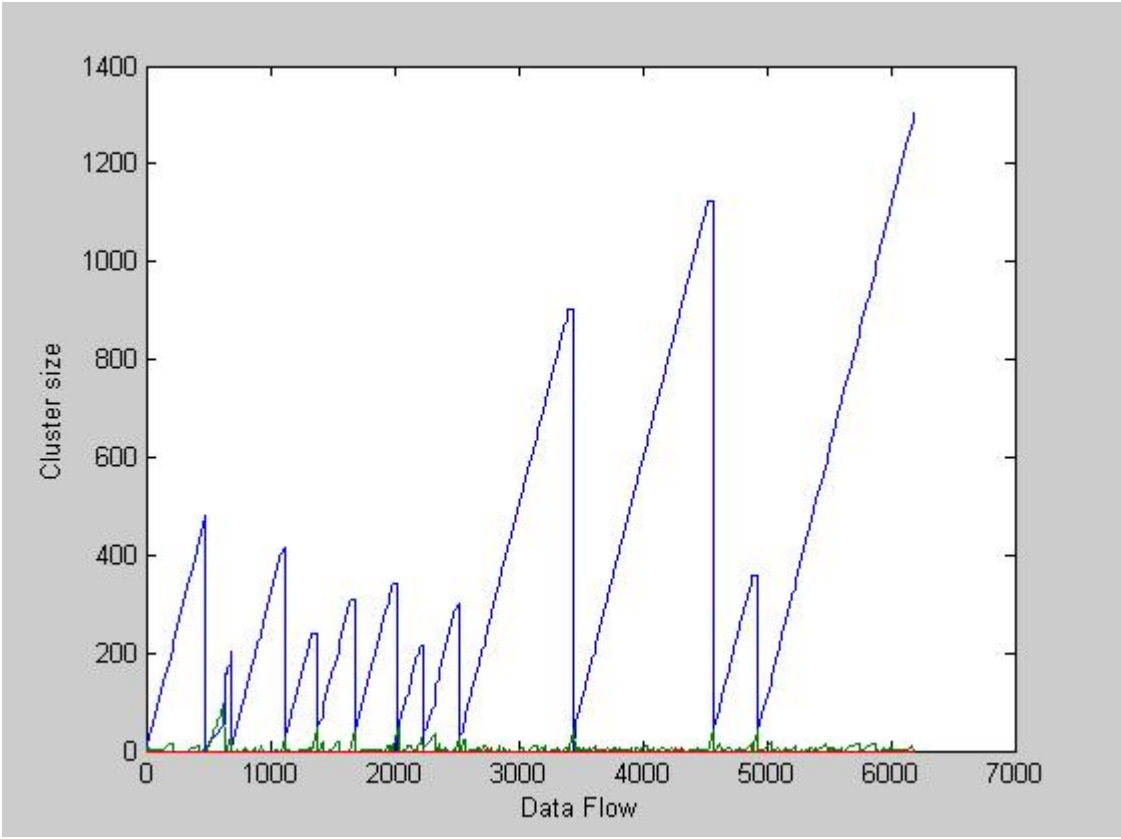

Figure 25: The population detection of the12 new populations stream

The following tables present the confusion matrix of the presented classifier in comparison to other classification options. The mean square precision error (MSPE) was evaluated for each case. Table 13 present the results of our classifier.

Table 13: Confusion matrix of the on-line classifier (using the *population database*)

	A	B	C	D
A	**0.796923**	0.04	0.14	0.023077
B	0.066558	**0.756494**	0.136364	0.040584
C	0.036353	0.051011	**0.864263**	0.048373
D	0	0.000663	0.102717	**0.89662**

$$\text{Mspe13}= \frac{1}{4}\sum_{i=1}^{4}(1-p_i)^2 = 0.25*(0.21\text{^}2+0.25\text{^}2+0.14\text{^}2+0.11\text{^}2)=0.0346$$

Table 14 includes the confusion matrix result from the decision tree c5.0. This classifier was trained, like our classifier, with the 'base-population' as the training set and the rest of the populations as testing. The decision tree eventually works as a rule-based system which is supposed to yield the best performance giving the mentioned training set. The MSPE measure it yields is almost equal to the one our classifier yields.

Table 14: Confusion matrix of decision tree C5.0 (train-'*Populations base*'; test- rest)

	A	B	C	D
A	**0.89077**	0	0.10923	0
B	0.00161	**0.74194**	0.25645	0
C	0.02260	0.01702	**0.92398**	0.03640
D	0	0.00398	0.06627	**0.92975**

$$\text{Mspe14}= \frac{1}{4}\sum_{i=1}^{4}(1-p_i)^2 = 0.25*(0.21\text{^}2+0.26\text{^}2+0.08\text{^}2+0.08\text{^}2)=0.0311$$

In comparison to our classifier, two additional classifiers are presented. These two are based again on the decision tree trained with only one population. Table 15 includes results of the classifier trained with population 16 while Table 16 presents the results when the classifier is trained on population 1. The confusion matrix together with the MSPE measure indicates severe misclassification in comparison to the adaptive classifier. The confusion matrix in Table 16 indicates that no sample was classified to grade '*B*'. This is due to the fact that the classifier was trained only with population 1 which does not contain samples of grade '*B*'.

As mentioned in section 5.3.2.2 three parameters were checked for their influence on the system cost and performance. The system performance values were based on the MSPE measure. Results in 2D and 3D (by contours) are presented.

Table 15: Confusion matrix of decision tree C5.0 trained on population 16 (test- other 20 populations)

	A	B	C	D
A	**0.56750**	0	0.43250	0
B	0	**0.63708**	0.36292	0
C	0.01847	0.00354	**0.97406**	0.00393
D	0	0.00182	0.19193	**0.80625**

Mspe15= $\frac{1}{4}\sum_{i=1}^{4}(1-p_i)^2$ =0.25*(0.44^2+0.37^2+0.03^2+0.2^2)=0.0929

Table 16: Confusion matrix of decision tree C5.0 trained on population 1 (test- other 20 populations)

	A	B	C	D
A	**0.98970**	0	0.01030	0
B	0.00109	**0**	0.99891	0
C	0.08977	0	**0.58691**	0.32332
D	0	0	0.01025	**0.98975**

Mspe16= $\frac{1}{4}\sum_{i=1}^{4}(1-p_i)^2$ =0.25*(0.02^2+1^2+0.42^2+0.02^2)=0.2943

The three parameters values, as mentioned in section 5.3.2.2, are as follow:

- 7 cases of *change detection* operations (batches of 20,30,40,50,60,70,80). The phrase 'BatchSize' is used for these values in the figures axes/legends.
- 8 cases of *all features* activate (5,10,15,20,25,30,35,40). The phrase 'On' is used for these values in the figures axes/legends.
- 9 cases of *Current feature* activate(1,5,10,15,20,25,30,35,40). The phrase 'Off' is used for these values in the figures axes/legends.

Sensitivity to the cost of features and the penalty of misclassification was additionally evaluated.

Figure 26 and Figure 27 present the overall influence of the detection batch size on the classifier performance. It always go from high performance at 20 to low performance at 80. Each line in the graph present the values of the other two parameters. For example, each point on the red line in Figure 26 is the mean of 7 values (10-40) of *Current feature* parameter (*Off*) for *All features on* value of 30.

In Figure 28, for all values of *All features on* parameter, the batch size of 20 (blue line) represents the highest performance while 70 (red line) is the worst. The cost value variations is presented in Figure 29 where the minimum value of *All feature* parameter got the minimum cost for almost all values of *batch size detection* parameter. Figure 30 and Figure 31 imply that cost values increase together with the *All feature* batch parameter and decrease when *Current features* increase. This is presented for all batch size parameter values.

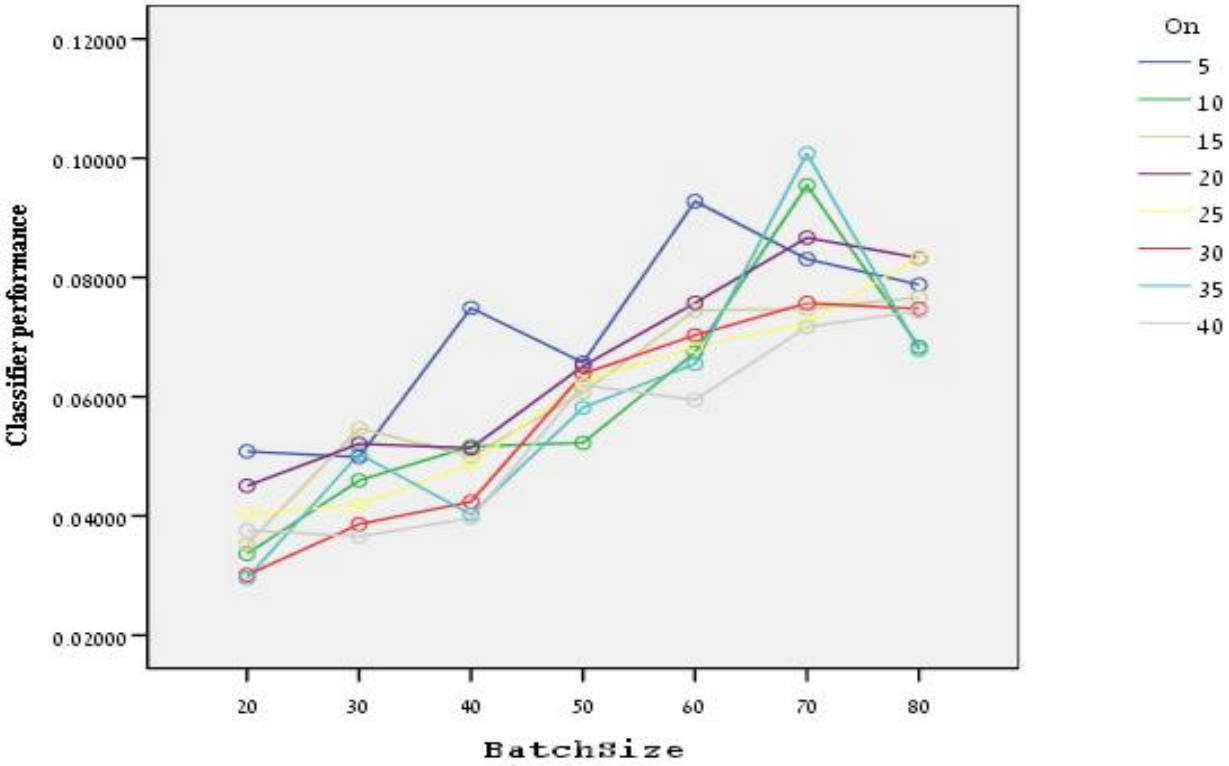

Figure 26: Batch size influence on the system performance (with *All features* (on) parameter)

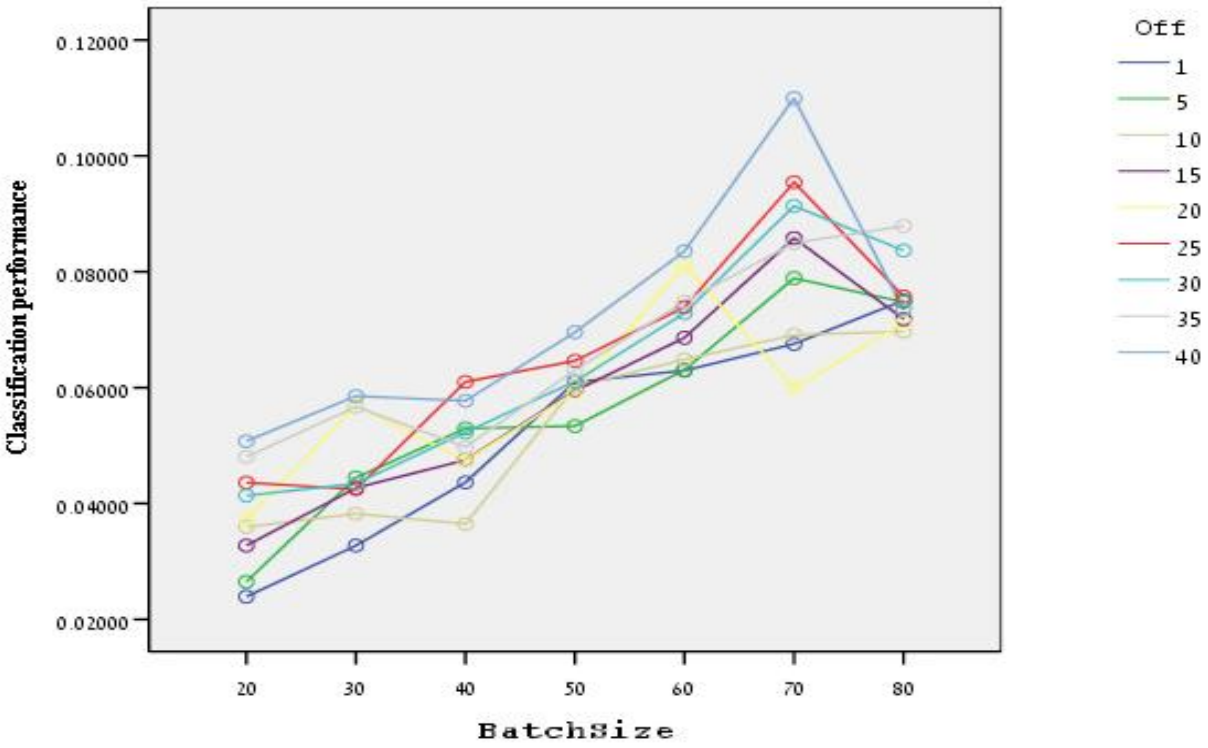

Figure 27: Batch size influence on the system performance (with *Current features* (off) parameter)

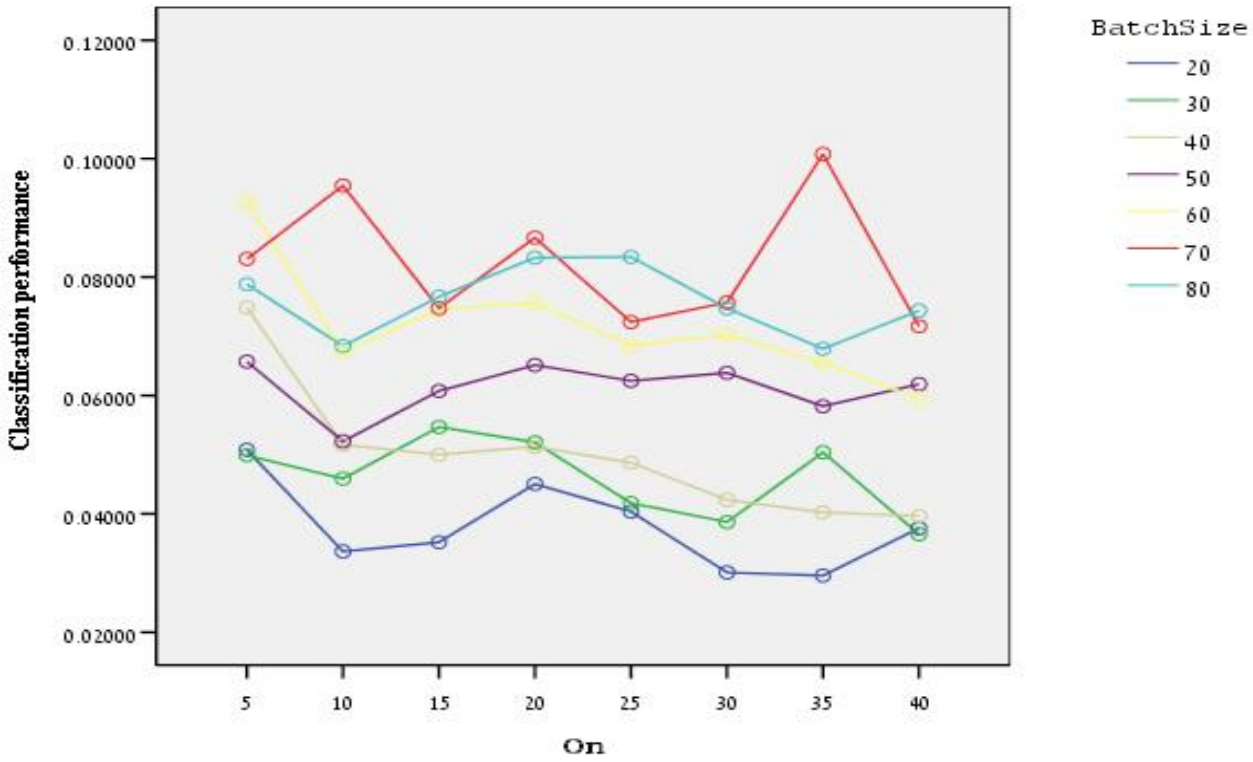

Figure 28: *All features on* parameter influence on the system performance (with *BatchSize* parameter)

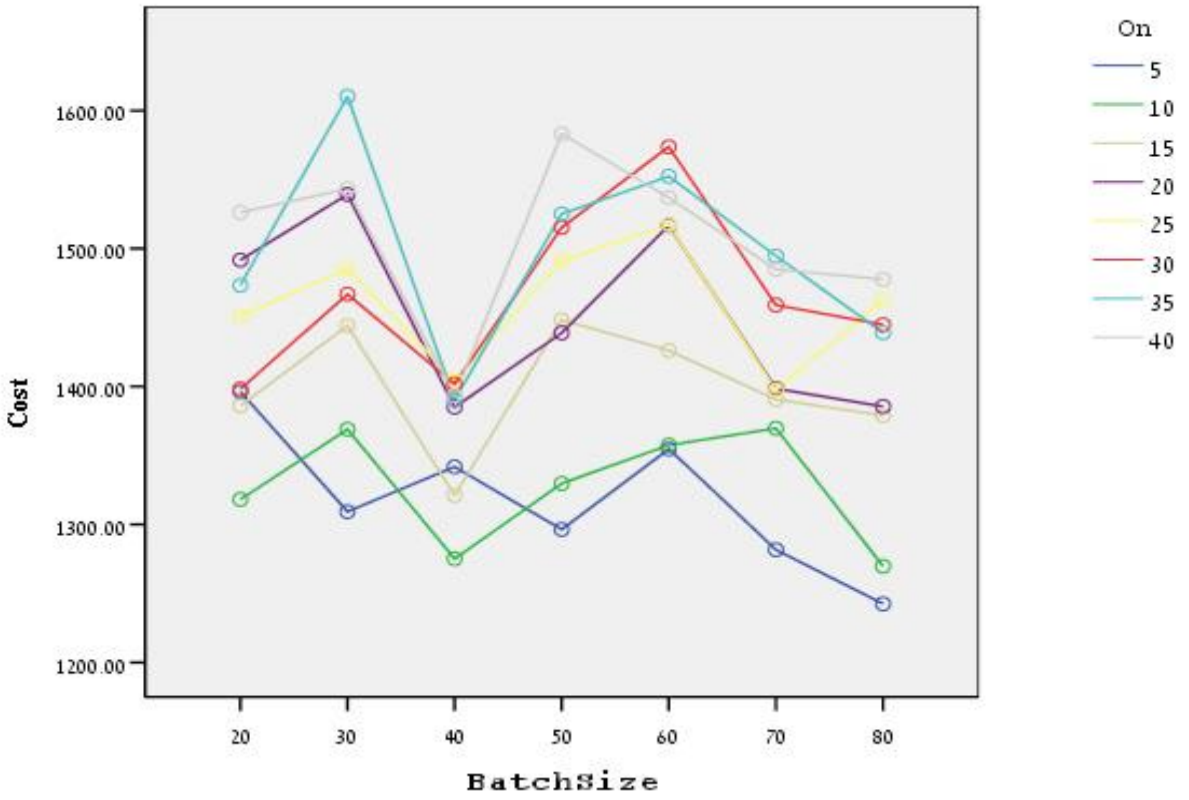

Figure 29: Batch size parameter influence on the cost levels (with *All features* (on) parameter)

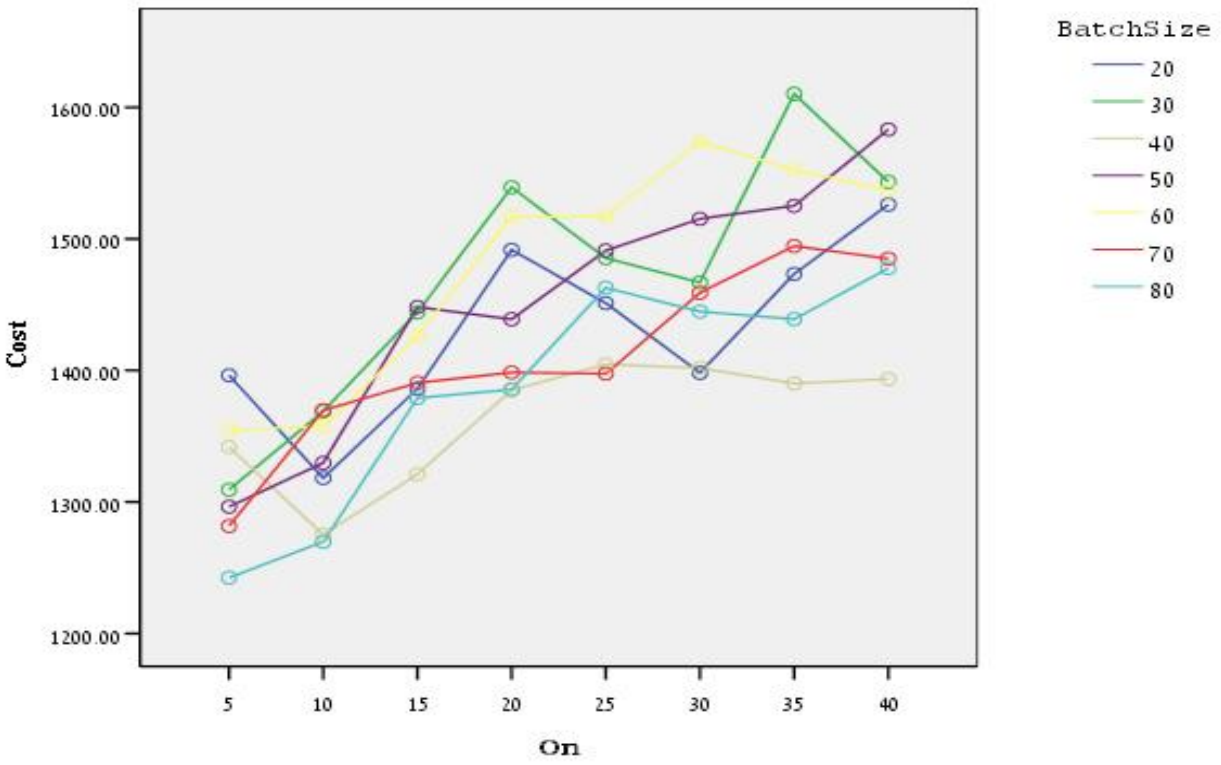

Figure 30: Cost value increasing with *All features* parameter

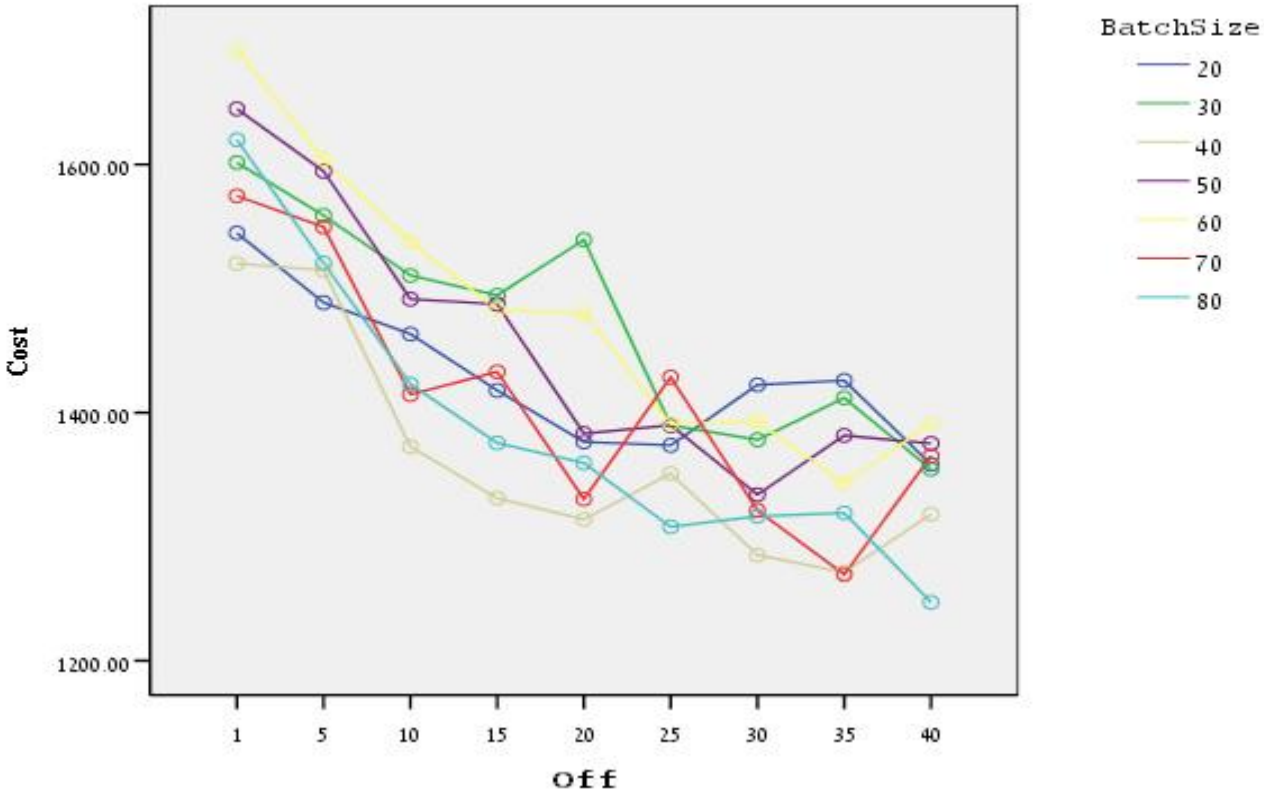

Figure 31: Cost value decrease with *Current features (off)* parameter

Contours graphs are presented to check the parameter influence in more detailed results. These results indicate can supply local trends of the parameter influence on the system.

***Change detection* batch size**

Classification performance levels as well as the corresponding cost for the case in which the batch size for the change detection procedure is set to three constant levels (20,40,80) is presented. Each contour is a boundary for certain level area. The arrows mark the gradient direction from high performance areas to the low ones. The parameters influence on performance is summarized in the following points:

1. The upper left corner of the graph indicates high performance since most of the features are used most of the time (FeatureOn>>FeatureOff) while the lower right corner indicates poor performance.
2. Several local minimum/maximum represent less expected results. For example, point [20,20] (i.e., [‘FeaturesOff’,‘FeaturesOn’]) was expected to achieve higher performance than [20,15] since less ‘FeaturesOn’ are used. The contour graph (Figure 32) shows the opposite. Same phenomenon repeats (in different locations) in other examples. These cases occur since different parameter combinations have a different influence on the change detection procedure as well as on the overall classification.
3. Comparing the classifier cost and performance contours indicates the influence of the classification accuracy on the cost function. For example, it could be expected that using point [20,5] will yield lower cost than points [20,10/15] since it uses more active features. The cost function contours presented in Figure 33 shows the opposite. This is due to the fact that the cost function incorporates classification performance. Same areas in the performance graph (Figure 32) emphasize that the first area ([20,5]) has lower performance level than the second area.
4. In several cases a high classification accuracy area corresponds to a lower cost minimum area. Two examples are points [30,10] and, even better point [20,15] in which good performance is achieved simultaneously with lower cost units.

Other cases are displayed in Appendix IX.

When comparing the different batch sizes of the change detection operations during simulation it should indicate that the classification performance levels are decreasing and the cost increasing. This can be figured out by checking the color bars of the graphs (e.g., Figure 32 min-0.02, max-0.07; Figure 36 min-0.04, max-0.13).

All feature batch size

Classification performance levels as well as the corresponding cost for the case in which the *all features on* parameter set to three constant levels (5,20,40) is presented.

The parameters influence on performance in this case is summarized in the following points:

1. The batch size of the change detection influence is best emphasized in this simulation. Each figure illustrates how the classifier performance decreases as the batch size increases. These changes tend to the right side of the figures (e.g., Figure 38) indicating that altering the change batch size together with the *features off* batch size has a high effect.
2. The highest cost area is concentrated on the left side of the figure since most of the features are activated during simulation (Figure 39).
3. Cost minimum value is at Figure 39. This means that moving from this parameter low value (5, Figure 39) to the highest (40, Figure 43) increases the overall cost since more features are active.
4. While in the lower/average parameters values (Figure 39; Figure 41) there is no specific direction to the cost gradient (i.e., arrows aim to all directions) it becomes very clear at parameter size of 40 (Figure 43) that the gradient is from left to right. This indicates that the feature cost becomes at this point more significant than the other cost constituents of the cost function.

Current features (off) batch size

Classification performance levels as well as the corresponding cost for the case in which the (current) features off parameter is set to three constant levels (5,15,40) is presented.

The parameters influence on performance in this case is summarized in the following points:

1. When moving from the value of 5 to 15 (Figure 44 to Figure 46) the low performance area increases.
2. The cost in these cases decreases. This is due to the fact that less features are involved.

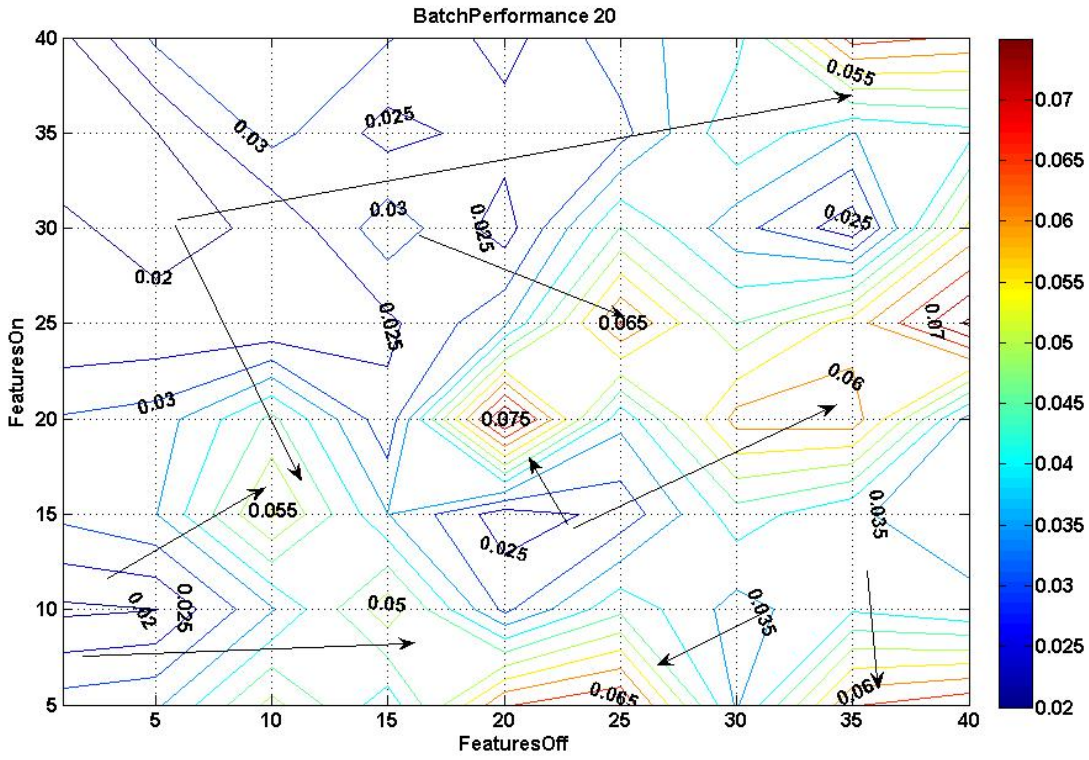

Figure 32: Classification performance measures variations where change detection is fixed on a batch of 20 and features on/off batches are changing

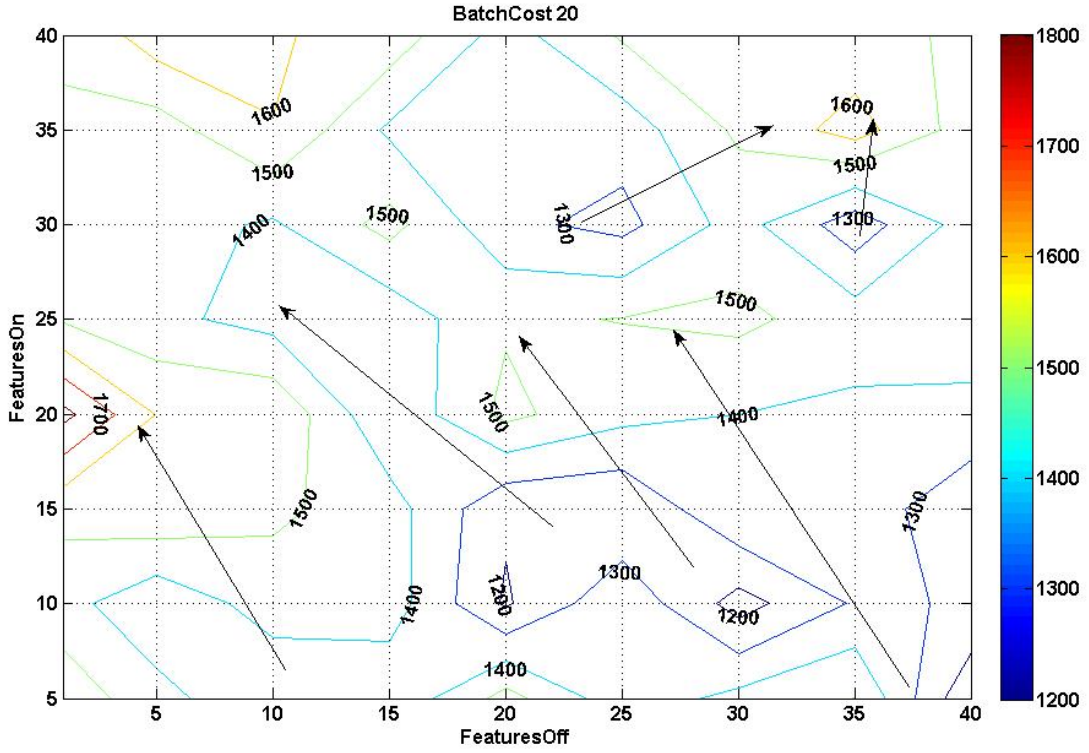

Figure 33: Cost levels variations where change detection is fixed on a batch of 20 and features on/off batches are changing

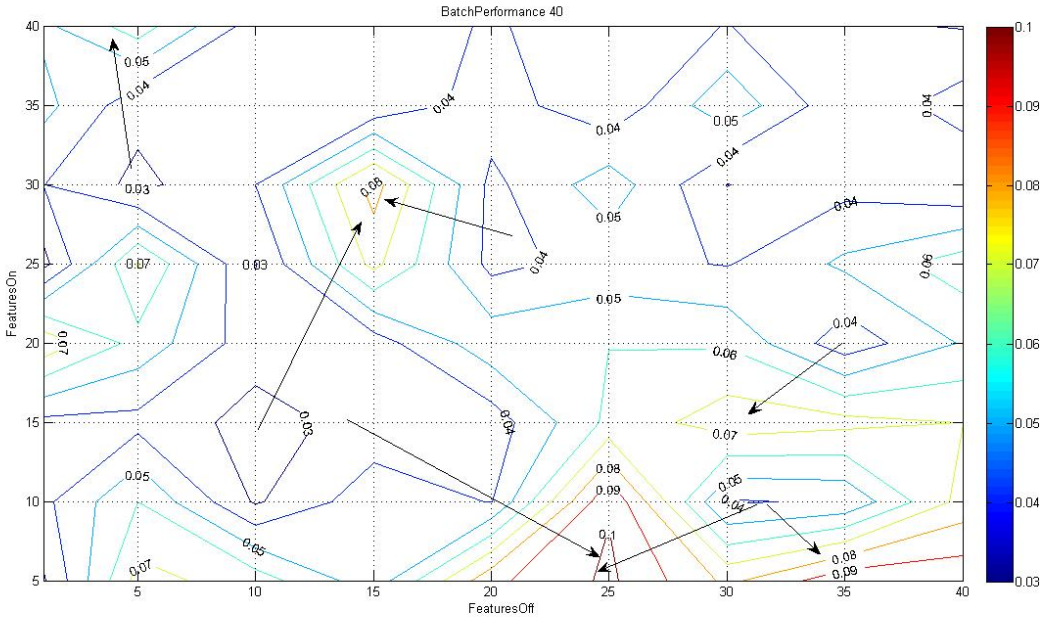

Figure 34: Classification performance measures variations where change detection is fixed on a batch of 40 and features on/off batches are changing

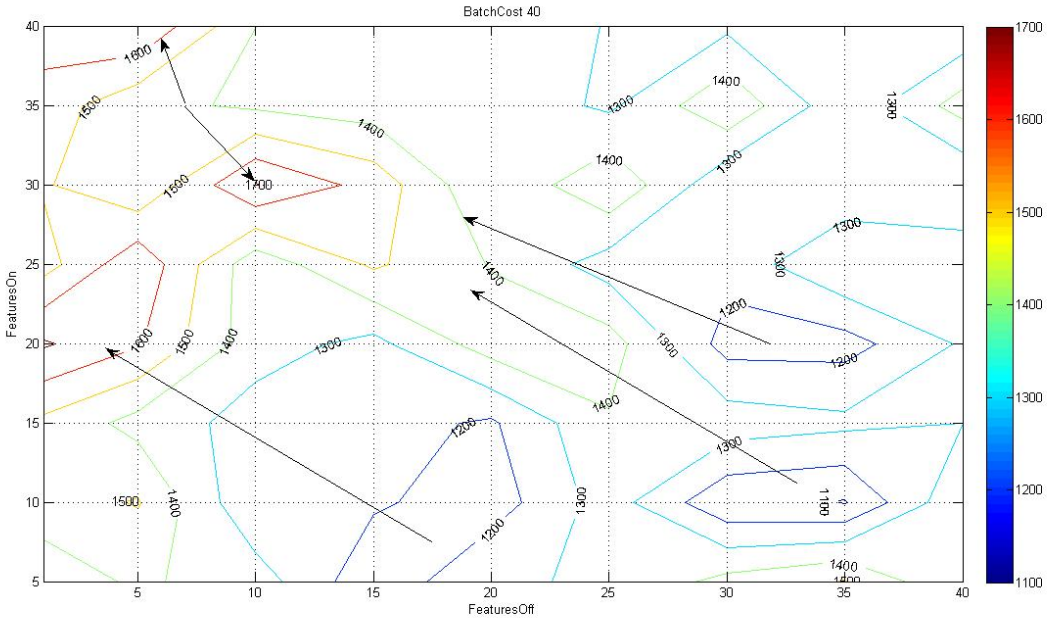

Figure 35: Cost levels variations where change detection is fixed on a batch of 40 and features on/off batches are changing

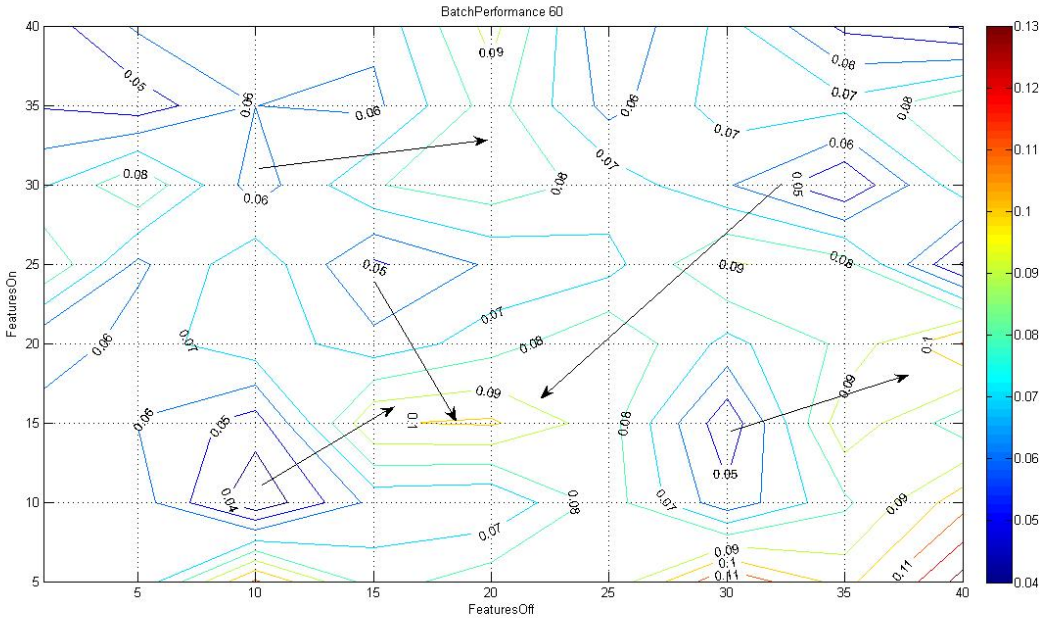

Figure 36: Classification performance measures variations where change detection is fixed on a batch of 60 and features on/off batches are changing

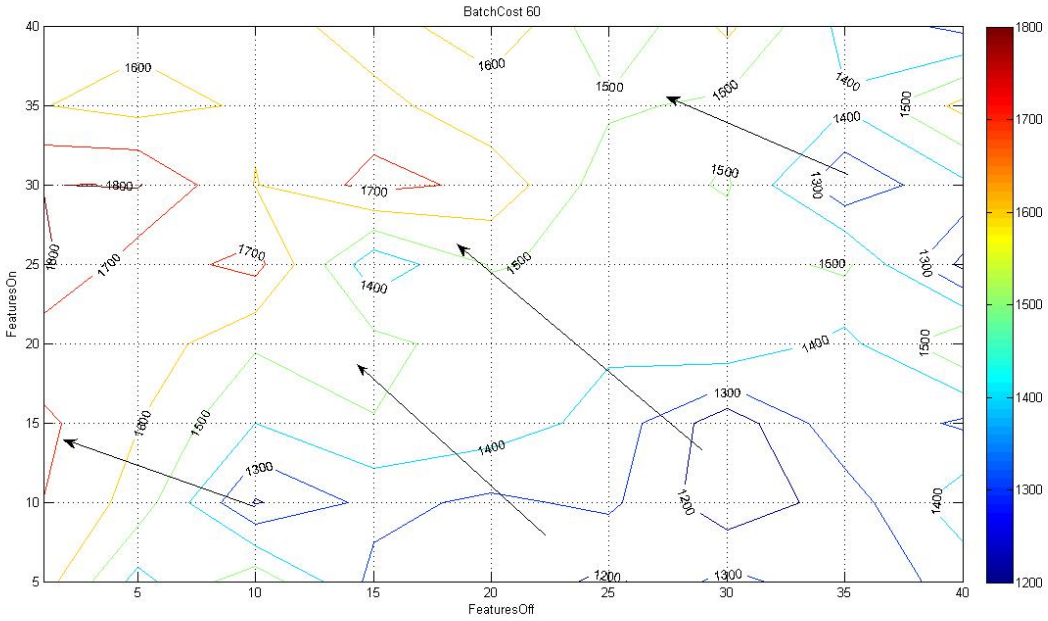

Figure 37: Cost levels variations where change detection is fixed on a batch of 60 and features on/off batches are changing

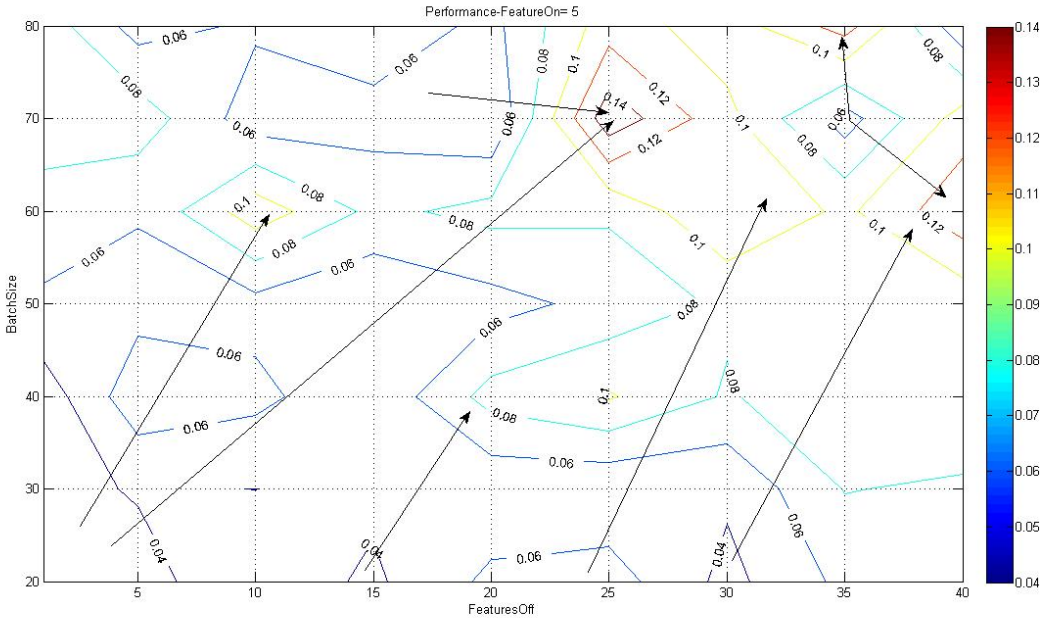

Figure 38: Classification performance measures variations where *all features on* is fixed on a batch of 5 and features off/detection batch size batches are changing

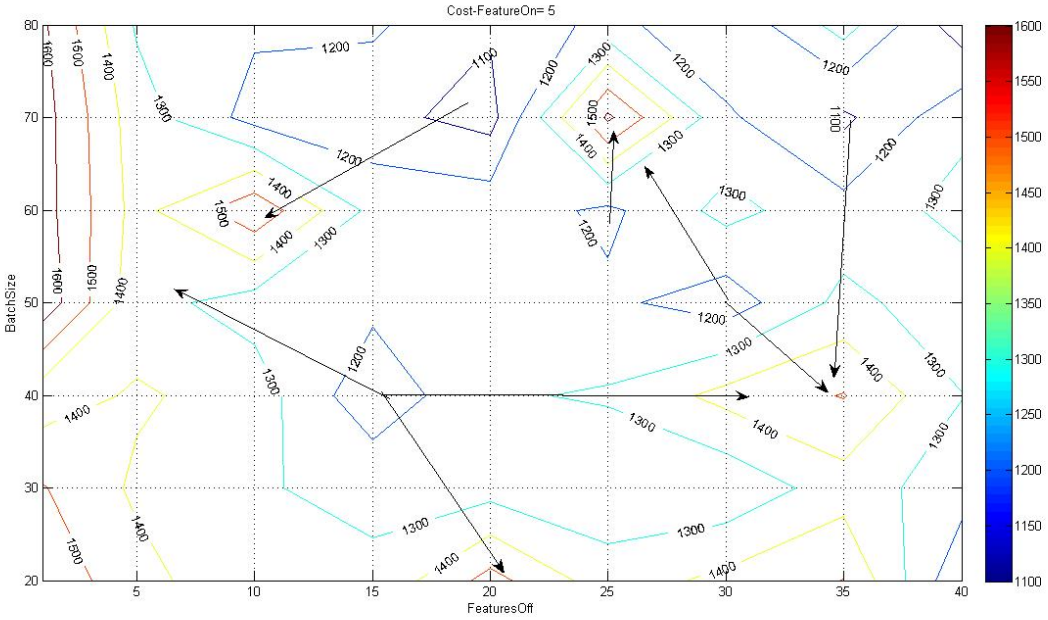

Figure 39: Cost levels variations where *all features on* is fixed on a batch of 5 and features off/detection batch size batches are changing

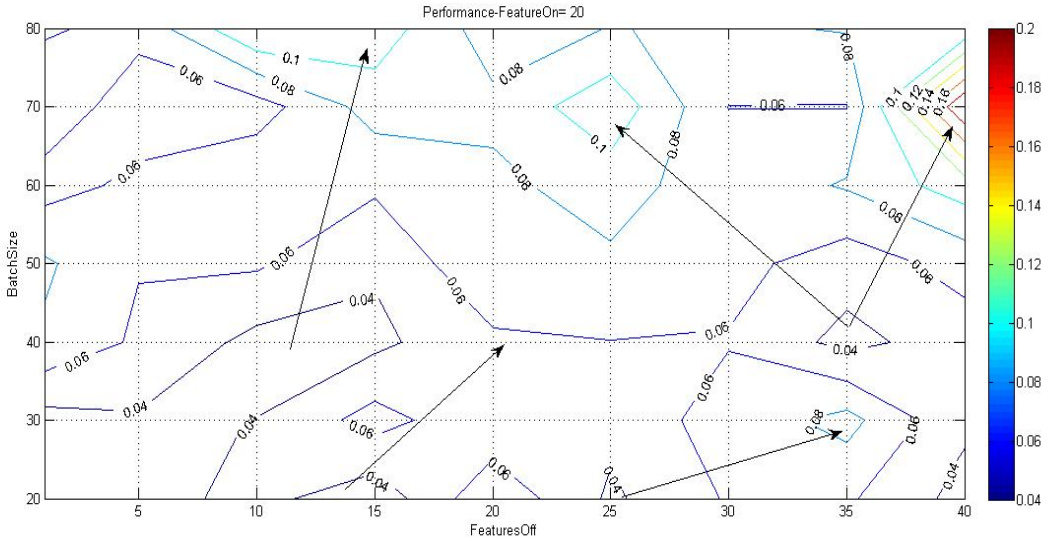

Figure 40: Classification performance measures variations where *all features on* is fixed on a batch of 20 and features off/detection batch size batches are changing

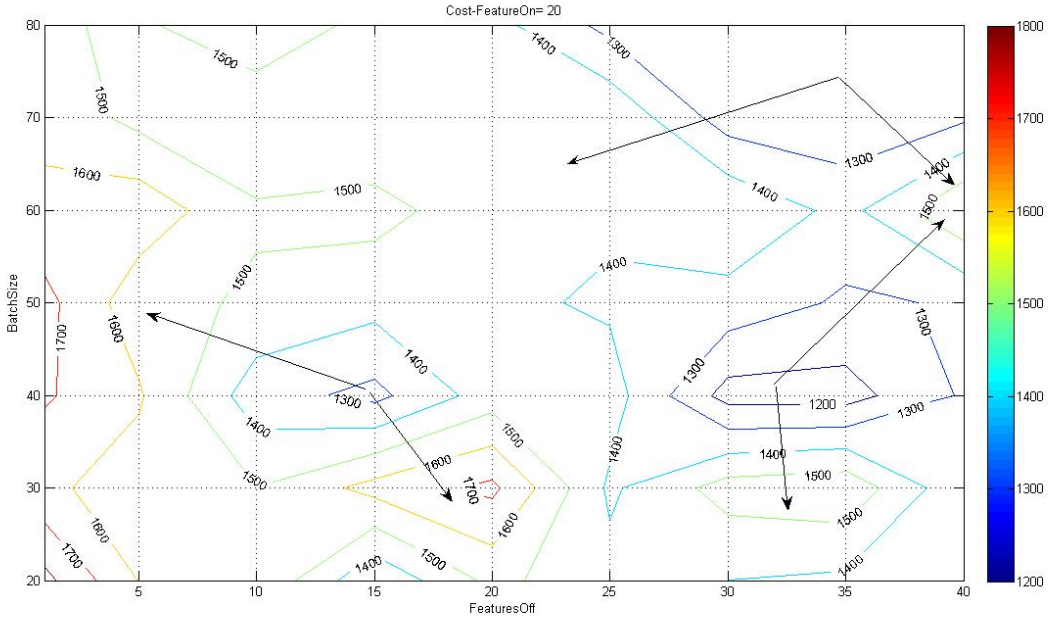

Figure 41: Cost levels variations where *all features on* is fixed on a batch of 20 and features off/detection batch size batches are changing

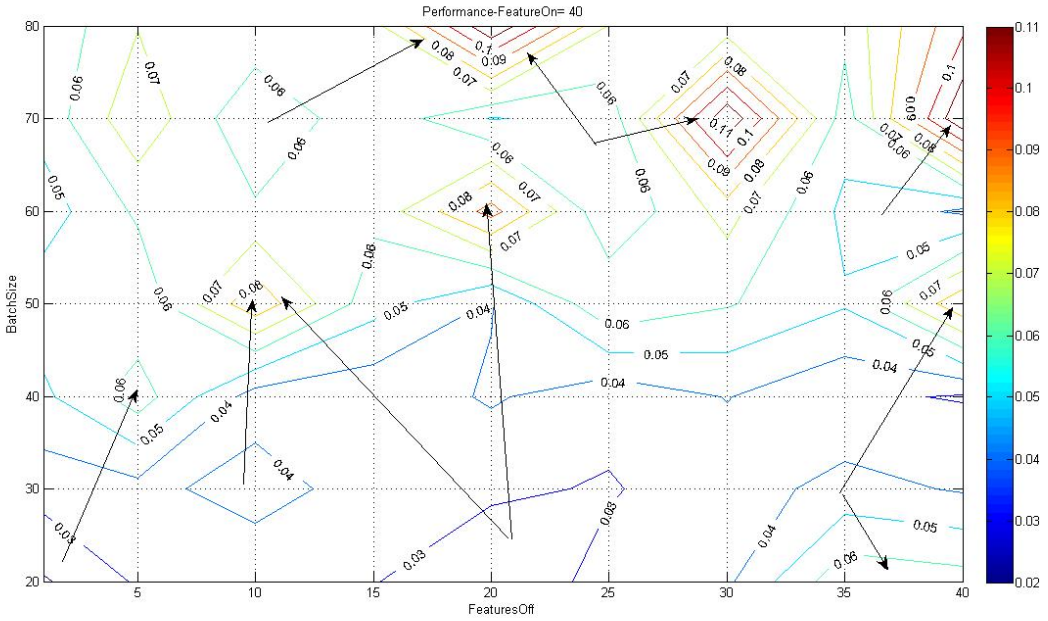

Figure 42: Classification performance measures variations where *all features on* is fixed on a batch of 40 and features off/detection batch size batches are changing

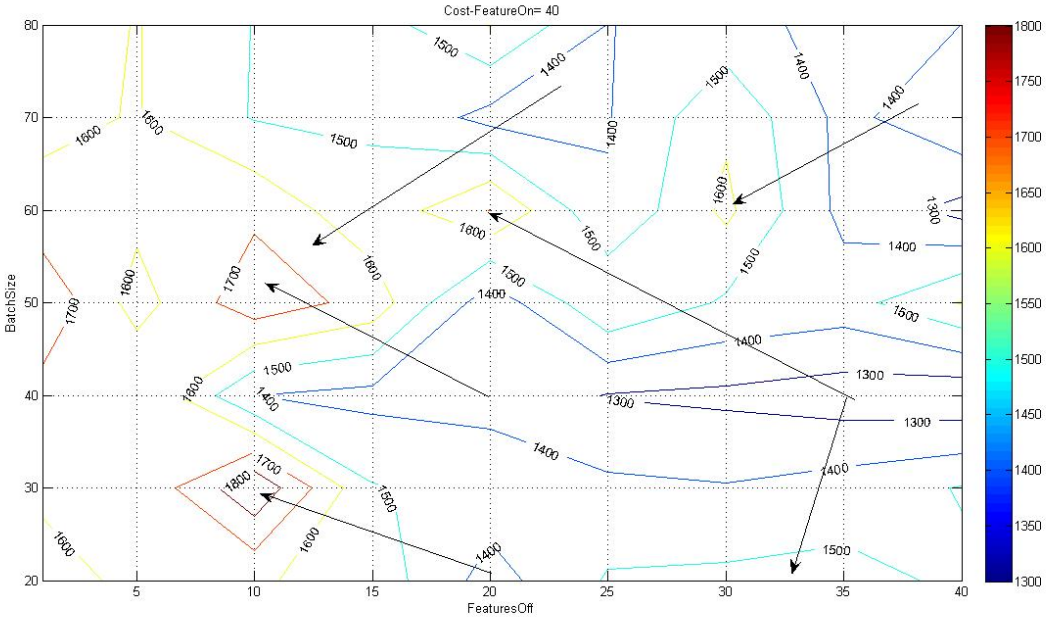

Figure 43: Cost levels variations where *all features on* is fixed on a batch of 40 and features off/detection batch size batches are changing

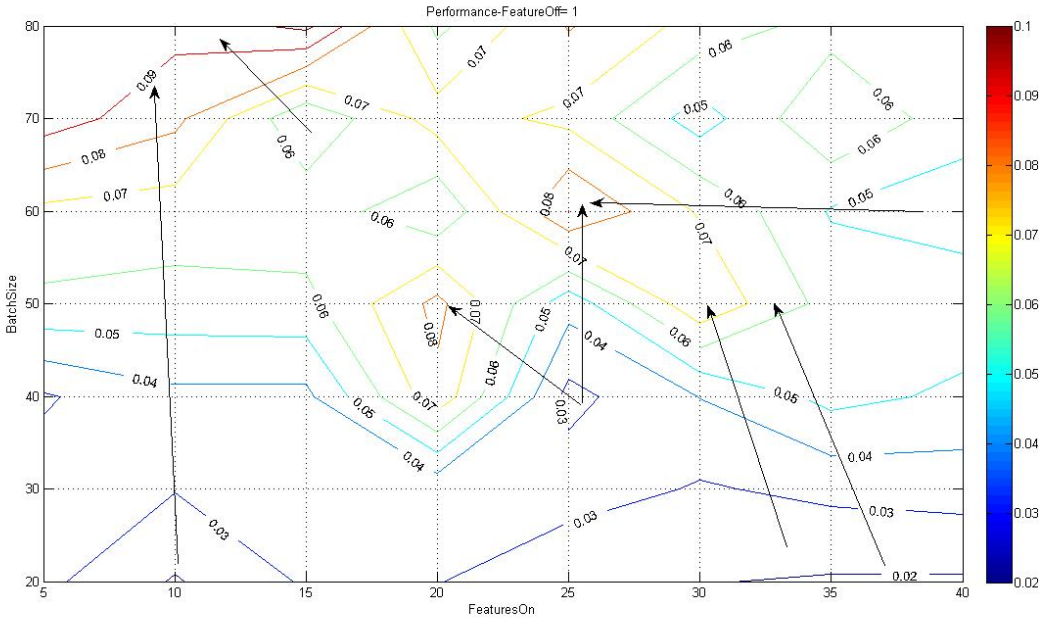

Figure 44: Classification performance measures variations where *all features off* is fixed on a batch of 1 and features on/detection batch size batches are changing

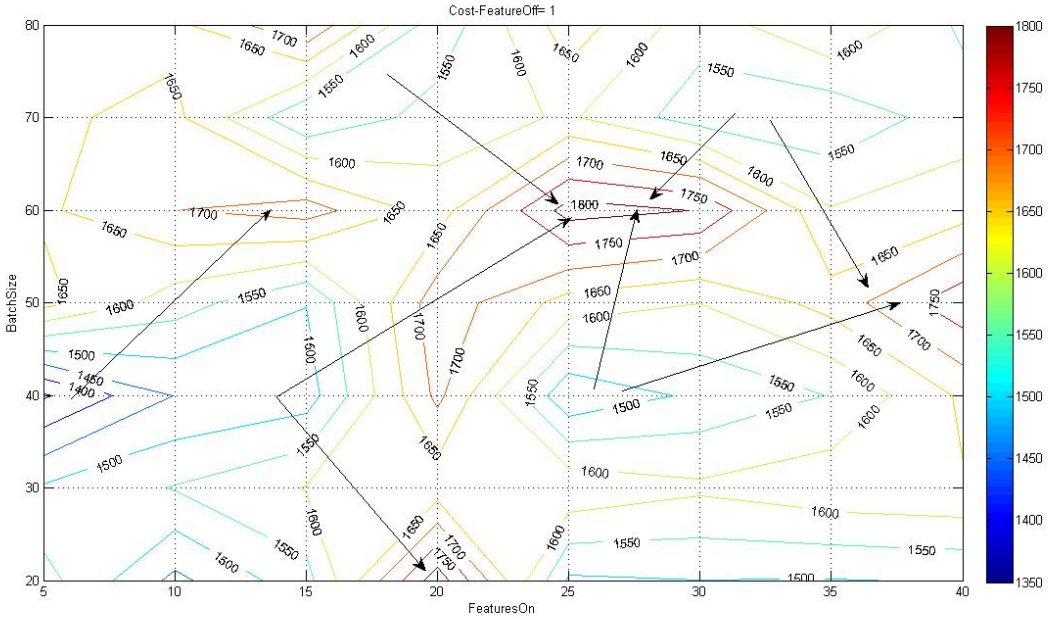

Figure 45: Cost levels variations where *all features off* is fixed on a batch of 1 and features on/detection batch size batches are changing

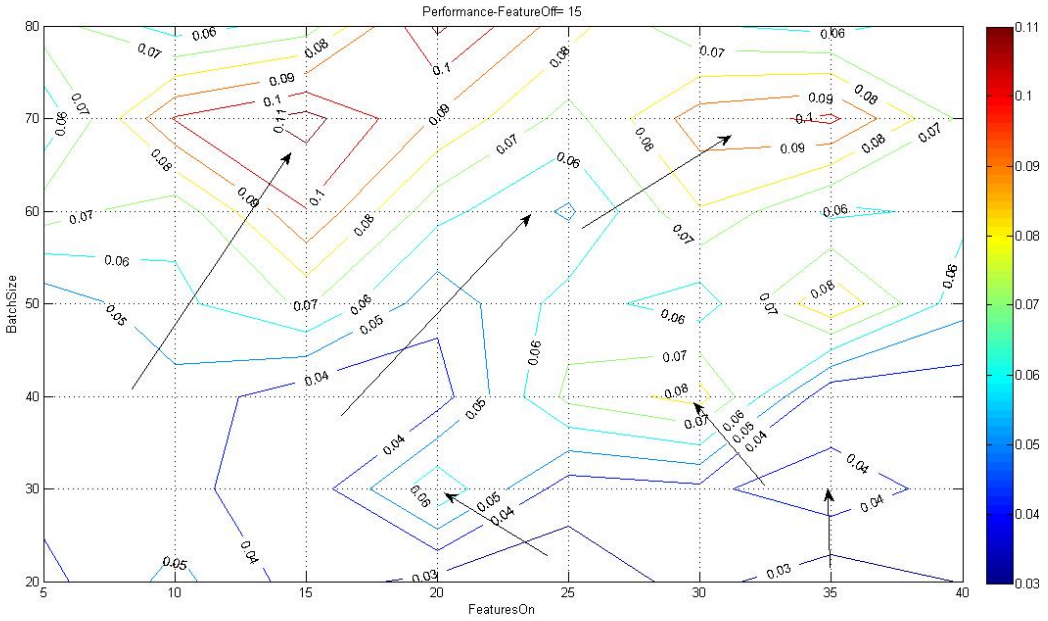

Figure 46: Classification performance measures variations where *all features off* is fixed on a batch of 15 and features on/detection batch size batches are changing

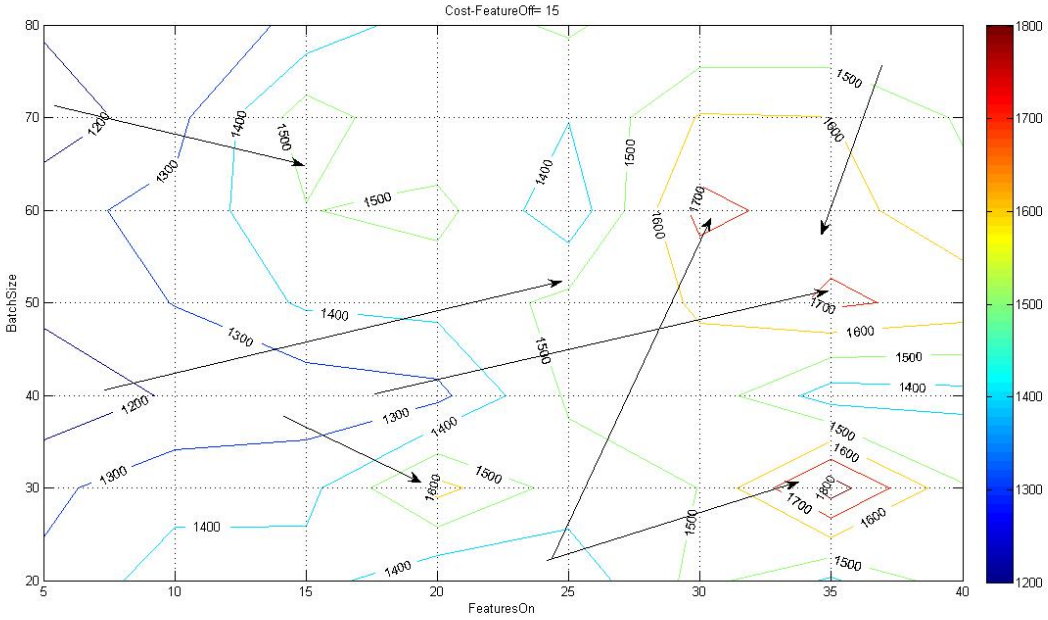

Figure 47: Cost levels variations where *all features off* is fixed on a batch of 15 and features on/detection batch size batches are changing

Sensitivity to cost changes

To check the cost function sensitivity to cost changes three values of both the feature cost and penalty cost were evaluated. Comparison of results of the first change (presented in Figure 48 and Figure 49) to the case in which all features cost was one unit and all penalties were the same for positive/negative misclassification (Figure 33) indicate:

1. Increasing the features cost by 3 units increases the significance of the feature cost. In Figure 48 we can see an obvious tend of the cost to increase from down-right corner to the up-left corner continuously. This means that by current change the feature cost is more significant than the penalty cost.
2. Multiplying the penalty cost by two makes the cost contours in Figure 49 look much like the classifier performance contours in Figure 32. Actually, cost behavior is similar to performance behavior. This result indicates the significance of the penalty cost over the feature cost.

Simulation significance

To check the changing parameters influence on the overall classifier performance a comparison is made between the best classification performance results and the best off-line classifier (Table 17). The off-line classifier was a FkNN classifier that was trained on all population- base training sets and used the best fit feature combinations of all quality features.

Table 17: Comparison between best off-line and best in the sensitivity test

	Highest performance results ParametersValue [20,40,1]	**Lowest cost results ParametersValue [20,10,30]**	**Off-line classifier Features=[20,26,29]**
Classification performance	0.01883	0.0288	0.0286
Cost	1501	1160	820 (+TrainCost)

We can see that one parameter combination (and there are more within all simulation possibilities) resulted in better performance than the best that was defined off-line. However, the cost of the simulated classifier is higher even though we must consider the training cost of the off-line classifier.

Table 17 includes also the simulation with lower accuracy performance, but still comparable to the off-line classifier, that costs much less than the best performance simulation.

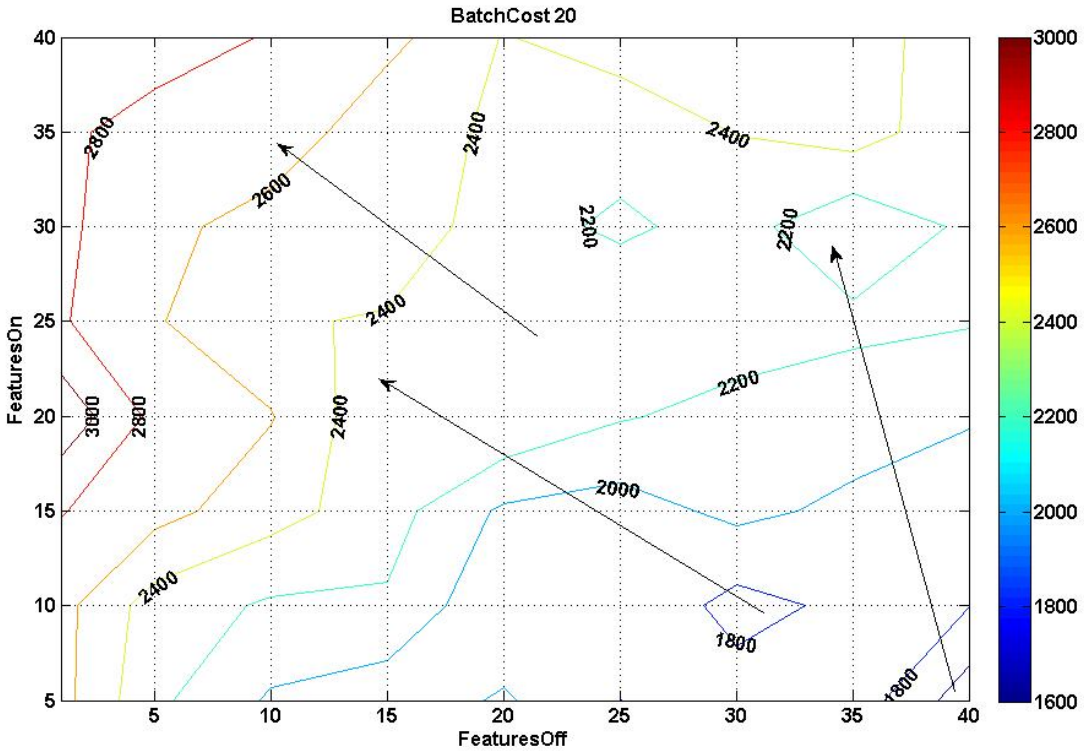

Figure 48: First batch parameter cost contours with three times feature cost

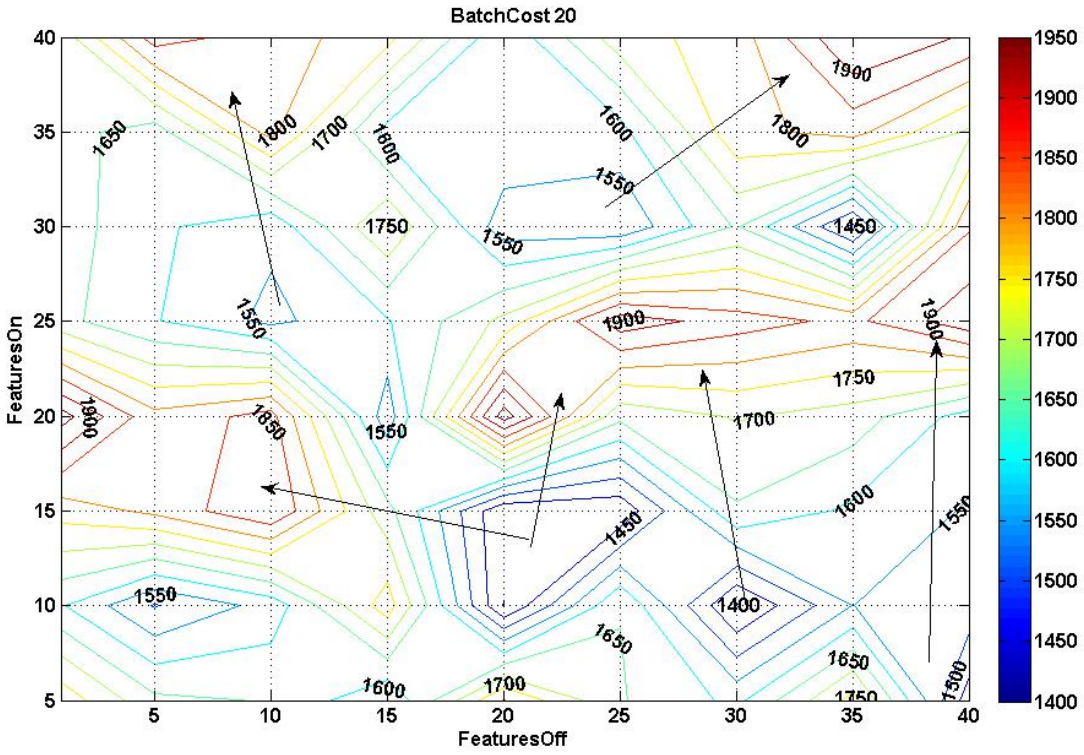

Figure 49: First batch parameter cost contours with two times the penalty cost in the payment matrix

5.5 Summary

The chapter presents system implementation analysis on synthetic and real agriculture databases. The synthetic experiment indicated the capabilities of the change detection measures in this research. The measures were precise and succeeded in detecting the significance transitions between the populations. Classification accuracy was better than the classifier that used all the features. The significance of the order of population's entrance was also presented using the synthetic data base. These experiments indicated the system sensitivity to entrance order and pointed out the need for a *population database*.

The classifier yields higher classification accuracy performance than non-adaptive classifiers for the agriculture database. The mean square precision error (MSPE) indicates this difference (0.0346 for the adaptive vrs. 0.2943 for non-adaptive). When compared with optimal classifiers (i.e., using all features and trained using all populations) the system yielded lower but still comparable results (85% vrs. 89%; 0.0346 vrs. 0.0311). Its advantage is the employment of a smaller number of features (different number for each population), which implies lower classification costs. In addition, the optimal classifier that was tested is a c5.0 tree classifier which is comparable to a rule-based system that uses all features and therefore yields the best results.

Results imply that Class B always appears with a lower accuracy than the other grades. Going through detailed results we find that these errors mainly appear for population 17. This population images of black olives and near black olives were problematic to analyze during the image-processing phase and therefore there is a higher gap between class B and C.

The sensitivity analysis implies that the defined parameters have major influence on both system performance and cost.

6 Conclusions and future work

6.1 Conclusions

With quality sorting of produce constituting a major challenge in agriculture, most research in the field today focuses on the feature extraction ability that various systems offer. The main problem is how to deal with the "*diversity of physical appearance encounters in agricultural products*" (Njoroge *et al*., 2002). This problem is difficult since agricultural features depend on growing conditions and change during the season and therefore cannot be predicted a-priori. These changes introduce a sorting problem in which the classifier must be able to classify agricultural produce as efficiently as possible using a minimum quantity of features and considering all costs in the classification process so as to maximize system profitability.

This thesis attempts to provide an efficient method for solving this problem by developing an on-line hierarchical classifier with the capability of adapting to different populations. By applying best-fit classifiers and the appropriate subset of features, we achieved improved classification accuracy.

The main contributions of the proposed classifier are: efficient detection of a new population; rapid adjustment to this population in terms of overlap and similarity measures; online feature and classifier selection via the adjustment procedure; and a cost objective function that, together with classification accuracy, tests the system performance.

A major contribution of our methodology is the ability to adapt to variations in produce using population detection and decision tools about classification strategies. Our methodology is in accordance with the observations of leading researchers in this area. For example, Lu (2003), who used NIR hyper-spectral imaging for bruise detection in apples, mentioned that "*these results indicate that bruises are affected by apple variety and bruise severity, and they change with time and at different rates, even for the same fruit. Hence, an effective detection system must have the capability to detect bruises, both new and old, for different apple cultivars*". Njoroge *et al*. (2002), in presenting the performance of his automated fruit grading system concluded: "*Incorporation of a high level intelligence into the system will enhance robustness in dealing with problems related to flexibility of the inspection product and inspection environment*".

Picus and Peleg (2000), who identified the need for adaptation in sorting agricultural produce, developed a classification system based on prototype populations and implemented it on dates. However, in their research, all prototype populations must be known in advance and used to train the classifier. Their system identifies a new population from an existing one. In our research, not all populations must be known in advance. When a new population is detected, the system compares it to previous populations using an overlap measure and two similarity statistic measures (KS for distribution and skewness for the class similarity). Since the classifier deals with many different relationships between the populations it is more flexible and suitable for real world problems.

The classifier that we developed has several advantages:

1. Employs a multitude of features

   The developed methodology deals with a multitude of features, detecting the features that cause the change and define the best-fit feature set for the new population decision.

2. On-line change detection

   An on-line clustering algorithm was expanded to enable on-line change detection. This includes adjustment of the clustering algorithm to provide information to three measures developed for global and local changing detection. Previous work dealing with on-line change detection was mainly implemented for signal processing and abrupt changes [Markou and Singh, 2003].

3. Change detection in the feature space depends on the batch size that is defined for the change detection operation timing.

   In a population classifier based on a FIFO stack methodology (Picus, 2000), in which the classifier is trained with a known, a-priori population training set and where the reference is the population index, the problem is that the stack must contain samples from the same population in order to detect it. In case of two populations in the stack, the classifier detects either a mix with no reference or a third population. The previous population classifier misclassified the samples that, meanwhile, went through the stack.

   Although in our classifier we encounter the same problem of misclassified samples, a situation in which a mixed population is detected and classified as unknown cannot arise. Our system achieves this by separating the detection (of the new population) from the action (set new classifier).

4. Our population base approach is based on all relevant features and not only on the best ones. As a result of this approach, the classifier detects changes in more than one feature domain and uses the population's similarity measures to improve performance.

5. While most similarity measures are based on a rough comparison between two clusters/classes (see Hu and Basu, 2002; Rond and Wang, 2004) our system diagnoses not only the overlapping level between the groups but also determines whether the classification levels within the populations are similarly distributed. Whenever a new population enters the system, it might be found highly overlapped with several populations but only similar to part of them. This is a very important point since even though the overlap level is high, the two populations might contain different grade distributions (e.g., Figure 4). Comparing the skewness measure of both populations for each selected feature provides the system with a better population choice. The issue of the classification level span inside the populations requires additional research.
6. The classifier selection procedure also defines the best-fit feature subset. This is achieved by the procedure developed for using data from populations that have the best overlap level with the new population as a retrain data set for the classifier selection procedure. This procedure uses a fuzzy logic rule-based system to select the best-fit classifier from a batch of *n* fuzzy kNN classifiers. These classifiers use different feature combinations.

Sensitivity analysis indicates, as expected, that the population entrance order is important and adversely influences the classification accuracy. To overcome this problem a population base was selected using an averaged overlap and KS statistic between the populations. This population base was the reference for new populations that use its known classifiers in case of full adjustment or just the retrain data according to the overlap levels. The advantage of our classifier is its ability to use several pre-defined populations in order to build a new retrain database for the adaptation stage.

Lim and Harrison (2003), faced with the problems of on-line pattern recognition using probabilistic fuzzy adaptive resonance theory (PFAM), mentioned the need to "*absorb knowledge continuously and autonomously without corrupting or forgetting previously acquired knowledge*". This is called the stability-plasticity dilemma (Carpenter and Grossberg, 1987). The need for this population base can also be seen from the research of Lim and Harrison (2003) who concluded: "*One practical strategy is to employ a dual-mode learning approach where each PFAM classifier is first trained, offline, with a set of input samples with*

different orderings. This approach helps establish a knowledge base in the classifier before online, incremental learning is engage"'.

Picus (2000) defined prototype populations, a concept which is similar to our 'population base' using the dendogram clustering method. The best separating feature dendogram was selected for this task. The system detects populations that fit the prototype population and sets its classifiers for it. Populations that were not fully similar to one prototype population could not be classified. Our classifier yields higher classification accuracy performance when compared to non-adaptive classifiers. The mean square precision error (MSPE) indicates this difference (0.0346 for the adaptive Ver. 0.2943 for non-adaptive). When compared with optimal classifiers (i.e., classifier using all features and all populations and trained on them) the system yielded lower but still quite good results (85% versus 89%). Furthermore, employing a lower number of features (a different amount for each population), results in lower classification costs. Sensitivity analysis dealing with the batch size also indicates the cost benefit that is achieved using a small subset of features for each population.

To test the classifier in real world conditions, we harvested and analyzed a specially designated crop of olives to create a well-defined database for the classification problem. Commonly employed databases used for performance analyses of classifiers (e.g., UCI machine learning repository, 1998, ELENA dataset, etc.) could not be used since they do not contain multiple populations. The application of the classifier to the problem of olive quality sorting resulted in contributions for olive quality sorting: image processing algorithms, classification of table olives and olive oil quality prediction.

To summarize, the proposed on-line adaptive classifier framework selects online the most appropriate classifier and feature subsets for the incoming population. The chief benefit of our system is its ability to adapt to new populations based on previous ones using similarity measures. This ability makes it possible to decide on a classification strategy without having to train on a specific population, an approach that makes the framework more flexible to changes in the population. The capability of selecting the best feature set results in improved classification performance and lowered costs.

6.2 Future work

Future work may include the following directions:

6.2.1 Algorithms

Optimal classifier design

The system presented in this work is based on several methods that were combined into one hierarchical system. Using several sensitivity analyses, we tried to improve system performance. To improve the system, optimization procedures should be conducted for some of its elements. The classifier selection procedure can be based on systems other than the fuzzy method (e.g., voting method, Dempster-Shafer method). The fuzzy procedure itself can be tested for various membership functions with different rules. A classifier combination method should be applied to define the training data points selected for a new population. Several population classifiers that have high similarity with the new population must be combined to obtain the best classification grades. Additionally, statistical tools, designed for process optimization (Montgomery, 2001), might be considered in order to get the best performance based on all system parameters (e.g., several batch sizes, cost parameters.).

Virtual population structures

After building the population base there should be a way of combining these populations into virtual populations to maximize the coverage of the feature space. This option should provide the system with the ability to deal with additional types of populations. This virtual process should include growing populations from the initial *population database* in order to keep the nature of quality labels.

6.2.2 Parameters

Cost analysis

In the current research we presented an off-line cost function that was applied after implementation. A basic cost analysis of the classifier selection procedure was conducted on-line in which classifiers with equal accuracy were compared over their feature quantity (i.e., the classifier with minimum features was selected). Additional parameters should be incorporated into this on-line cost analysis. These parameters can include most of the predefined parameters in our off-line cost function – classifier operation cost, misclassification penalty and change detection routine cost. In this way the system can be made more sensitive to the cost influence during a system run. This sensitivity to cost will improve the suitability of system performance in other domains as well, including those mentioned above.

As the cost analysis demonstrated in the experiment chapter there is a way to set classifier parameters to achieve high performance classification with much lower costs. Therefore further optimization analysis research should be conducted on the simulations results.

Changing batch size

In this work we tested the influence of the batch size in which change detection is checked. Future research might check the possibility of a variable batch size or a new parameter that can be more successful in the performance means. The system can increase or decrease the batch size according to its changing gap (e.g., after several short batch sizes, checking the use of a long one for the detection and then repeating the loop).

In addition, the overlap measure and the classifier selection procedures were applied to constant batches that have been empirically predefined. The system sensitivity to those batches size should be tested.

Population similarity measures

In this research we implemented three measures (overlapping, KS statistic and skewness) for defining population similarity and selecting the classification action accordingly. The overlapping measure is specifically fitted to this work and other related measures should be tested. The overlapping domain, which related to a changing online environment, should be further examined.

The skewness measure that determines the classification grade level patterns (e.g., Figure 20 in section 5.3.2) inside the population feature space used in this research is very general. A combination of this measure with other statistical ones or its extension may improve its performance.

6.2.3 Experiments

Extended experimentation

Further research must be carried out on additional agricultural produce. Additionally, the suitability of the system for additional domains should be evaluated. Among the features to be included in these domains are: multiple features, vague border definition and time change dependency. We propose that GIS domains and medical systems as offering suitable features for further research. The former might include satellite vision implementation to classify surface patterns (e.g., different agricultural field variations or any other ground cover application) while the latter can include blood tests or diseases with changing variables between patients. Additional domains could include economic problems (e.g., classify personal loans/accounts as ‘good’ or ‘bad’ during some time periods based on the application characteristics) and data mining (e.g., handwriting detection of people in changing states).

7 References

[1] Aneshnsley D. J., Affeldt H. A., Brusewitz G.H., Chen P., Delwiche M. J., Peleg K., Searcy S., Singh N., Throop J. A., Upchurch B. L. and Zion B. 1993. Detecting Surface Defects (Wounds, Bruises and Decay). In Nondestructive Technologies for quality evaluation of fruits and vegetables, Proceedings of the international workshop funded by BARD, Spokane,Washington 15-19 June 1993. Pages: 72-79. ASAE, St. Joseph MI-49085.

[2] Arnt A. and Zilberstein S. 2004. Attribute measurement policies for time and cost sensitive classification. Proceedings of Forth IEEE International Conference on Data Mining (ICDM). P:323-326.

[3] Bauer E. and Kohavi R. 2004. An empirical comparison of voting classification algorithms: bagging, boosting and variants. Machine Learning. 36(1-2):105-139.

[4] Bezdek J. C. 1981. Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York.

[5] Boudaoud N. and Masson M. 2000. Diagnosis of transient states: A pattern recognition approach. Journal Européen des Systèmes Automatisés. 34(5): 689-708.

[6] Bruzzone L. 2000. An approach to feature selection and classification of remote sensing images based on the bayes rule for minimum cost, IEEE Transactions on Geoscience and Remote Sensing. 38 (1): 429-438.

[7] Buttrey S. E. and Karo C. 2002. Using k-nearest-neighbor classification in the leaves of a tree. Computational Statistics and Data Analysis. 40(2002): 27-37.

[8] Calpe J., Soria M., Martinez M., Frances J.V., Rosado A., Gomez-Chova L., Vila J. 2002. high-speed weighting system based on DSP. Proceedings of the IEEE 2002 28th Annual Conference of the Industrial Electronics Society (IECON 02). Vol 2: 1579-1583.

[9] Carpenter G. A. and Grossberg S. 1990. Adaptive resonance theory: Neural network architectures for self- organizing pattern recognition. In R. Eckmiller, G. Hartmann, and G. Hauske (Eds.), Parallel processing in neural systems and computers (pp. 383– 389). Amsterdam: North- Holland.

[10] Carpenter G. A. and Grossberg S. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision Graphics and Image Processing, Vol 37: 54-115.

[11] Chao K., Chen Y. R., Early H. and Park B. 1999. Color image classification systems for poultry viscera inspection. Applied Engineering in Agriculture. 15(4): 363-369.

[12] Chamberlain D. W. 1976. Electro-optic field tomato color sorter. American Society of Agricultural Engineering Paper No. 761533. St. Joseph Mich.: ASAE.

[13] Choi K., Lee G., Han Y.J. and Bun, J.M. 1995. Tomato maturity evaluation using color image analysis. Transactions of the American Society of Agricultural Engineering 38(1): 171-176.

[14] Chtioui Y., Bertrand D. and Barba D. 1998. Feature selection by a genetic algorithm. Application to seed discrimination by artificial vision. Journal of the Science of Food and Agriculture 76(1998): 77-86.

[15] Chitioui Y., Panigrahi S. and Backer L.F. 2003. Self-organizing map combined with a fuzzy clustering for color image segmentation of edible beans. Transactions of the American Society of Agricultural Engineering, 46(3):831-838.

[16] Collins R.T., Liu Y. and Leordeanu M. 2005. Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27 (10): 1631-1643.

[17] Crow E. and Shimizu K. 1988. Lognormal distributions, theory and applications. Marcel Dekker, New York.

[18] Dash M. and Liu H. 1997. Feature selection for classification. Intelligent Data Analysis. 1(1997) : 131-156. Elsevier-Science.

[19] Davis J. M. and Gardner G. 1994. Harvest maturity effects fruit yield, size and grade of fresh-market tomato cultivars. HortScience 29(6): 225-229.

[20] Deck S. H., Morrow C. T., Heinemann P. H. and Sommer H. J. III. 1995. Comparison of a neural network and traditional classifier for machine vision inspection of potatoes, Applied Engineering in Agriculture. 11(2):319-326.

[21] Delwiche M. J., Affeldt H. A., Birth G., Brown G. K., Guyer D. E., Hetzroni A., Peleg K. and Thai C. N. 1994. Surface Color Measurement of Fruits and Vegetables. In Proceedings of the International Workshop on Nondestructive Technologies for quality evaluation of fruits and vegetables. ASAE, St. Joseph MI-49085.

[22] Delwiche, M. J. and Sarig, Y. 1991. A probe impact sensor for fruit texture measurement. Transactions of the American Society of Agricultural Engineering 34(1): 187 -192.

[23] Demir C. and Alpaydin E. 2005. Cost-conscious classifier ensembles. Pattern Recognition Letters. 26(2005): 2206-2214.

[24] Diaz R. ,Gil L., Serrano C., Blasco M., Molto E., Blasco J. 2004. Comparison of three algorithms in the classification of table olives by means of computer vision, Journal of Food Engineering 61 (2004): 101–107.

[25] Doulamis A. D., Doulamis N. D., Kollias S. D. 2000. On-line retrainable neural networks: Improving the performance of neural networks in image analysis problems. IEEE Transactions on Neural Networks.11(1): 137-155.

[26] Duda R. O., Hart P. E., Stotk D. G. 2001. Pattern Classification. 2nd ed. New York, N.Y.: John Wiley and Sons.

[27] Dull G. G. 1986. Nondestructive evaluation of quality of stored fruits and vegetable. Food Technology. 40(5): 106-110.

[28] Edan Y., Shmulevich I., Rachmani D., Fallik E., and Grinberg S. 1994. Neural networks for quality grading of tomatoes based on mechanical properties. Proceedings of the Food Automation III Conference. Pp:346-355.

[29] Edan Y., Pasternak H., Shmulevich I., Rachmani D., Guedalia D., Grinberg S. and Fallik E.. 1997. Color and firmness classification of tomatoes. Journal of Food Science 62(4): 793-796.

[30] Everitt B. 1974. Cluster Analysis. 1st Ed. Heinemann Educational Books Ltd, London.

[31] Fukunaga K. and Flick T. 1985. The 2-NN rule or more accurate NN risk estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(1):107-112.

[32] Feng G. and Qixin C. 2004. Study on color image processing based intelligent fruit sorting system. Proceedings of the 5th world congress on intelligent control and automation, June 15-19,2004, Hangzhou, P.R. China.

[33] Georgiopoulos M., Dagher I., Heileman G. L. and Bebis G. 1999. Properties of learning of a fuzzy ART variant. Neural Networks 12(1999):837-850.

[34] Goddard W.B., O'Brien M., Lorenzen C. and Williams D.W. 1975. Development of criteria for mechanization of grading processing tomatoes. Transactions of the ASAE 18(1): 190-193.

[35] Guedalia I. D., London M. and Werman M. 1999. An on-line agglomerative clustering method for nonstationary data. Neural Computation 11: 521-540.

[36] Guedalia I. D., Edan Y. and Werman M. 1995. A new method for on-line clustering of sparse data. ASAE Paper No. 95-3606. ASAE, St. Joseph MI-49085.

[37] Haff R.P., Jackson E.S. and Pearson T.C. 2005. Non-destructive detection of pits in dried plums. Applied Engineering in Agriculture. 21(6): 1021-1026.

[38] Hall M.A. 1999. Correlation-based feature selection for machine learning, PhD Thesis, Department of Computer Science, The University of Waikato, Hamilton, NewZealand.

[39] Heron J. R. and Zacharia G.L. 1974. Automatic sorting of processing tomatoes. Transactions of the American Society of Agricultural Engineering 17(5): 987-992.

[40] Ho T. K. and Basu M. 2002. Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(3): 289-300.

[41] Hoppner F., Kruse R., Klawonn F. and Runkler T. 1999. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. 1st ed. John Wiley & Sons, Inc. New York.

[42] Howarth M. S. and Searcy S. W. 1991. Comparison of Bayesian and neural network classifiers for grading carrots. ASAE Paper No. 91-7012. ASAE, St. Joseph MI-49085.

[43] Jain A.K., Murty M.N. and Flynn P.J. 1999. Data clustering: A review. ACM Computing Surveys. 31(3): 264-323.

[44] Jain A. and Zongker D. 1997. Feature selection: Evaluation, application and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence. 19(2): 153-158.

[45] Jack K. and Fu K. 1980. Automated classification of nucleated blood cells using a binary tree. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2(5):429-443.

[46] Jang J. S. R, Sun C. T., Mizutani E. 1996. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. 1st Ed. Prentice Hall.

[47] Kanal L. 1979. Problem solving models and search strategies for pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1(2):193-201.

[48] Kavdir D. and Guyer E. 2004, Comparison of Artificial Neural Networks and Statistical Classifiers in Apple Sorting using Textural Features, Biosystems Engineering. 89(3): 331–344

[49] Kavdir D. and Guyer E.. 2003. Apple grading using fuzzy logic, Turkish Journal of Agriculture and Forestry, 27(2003): 375-382.

[50] Keller J. M., Gray M. R. and Givens J. A. 1985. Fuzzy K-nearest neighbor algorithm. IEEE Transactions on System Man and Cybernetics. 15(4) :580-585.

[51] Kittler J., Hatef M., Duin R. P.W. and Matas J. 1998. On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3): 226-239.

[52] Kondo N., Kamata J., Ninomiya K., Monta M. and Ting K.C.. 2006. Rotary Tray for Machine Vision Inspection of Whole Eggplant Fruit, American Society of Agricultural Engineering Annual Meeting, Paper No. 066063.

[53] Kotsiantis S.B. and Pintelas P.E., 2004. A cost sensitive technique for ordinal classification problems. Proceedings of the 3rd Hellenic Conference on Artificial Intelligence, SETN 2004, LNAI 3025, pp. 220-229, 2004. Springer-Verlag berlin Heidelberg 2004.

[54] Kuncheva L.I. 2000. Fuzzy classifier design (studies in fuzziness and soft computing, Vol. 49). 1st ed. Phisica-Verlag Heidelberg, A Springer-Verlag Company. NY.

[55] Kuncheva L. I., Bezdek J. C., Duin R. P. W. 2001. Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition. 34(2001): 299-314.

[56] Kuncheva L. I. 2002. Switching between selection and fusion in combining classifiers: an experiment. IEEE Transactions on System Man and Cybernetics – Part B: Cybernetics. 32(2): 146-156.

[57] Kuncheva L. I. 2003. `Fuzzy' vs. `Non-fuzzy' in combining classifiers designed by boosting, IEEE Transactions on Fuzzy Systems, 11(6): 729-741.

[58] Kuncheva L. I. 2004. Classifier ensembles for changing environment. Proceedings of the 5th Intetnational Workshop on Multiple Classifier Systems, Cagliari, Italy. Springer-Verlag, LNCS, 3077, pp:1-15.

[59] Kuncheva L.I. 2004. Combining Pattern Classifiers. 1st ed. New York, N.Y.: Wiley-interscience, John Wiley and Sons.

[60] Langley P. 1994. Selection of relevant features in machine learning. In Proceedings of AAAI Fall Symposium on Relevance. AAAI, September 1994.

[61] Last M., Bunke H., and Kandel A. 2002. Fuzzy Modeling of the Complexity vs. Accuracy Trade-off in a Sequential Two-Stage Multi-Classifier System. Proceedings of Fuzzy Sets and Fuzzy Systems 2002 Conference (FSFS 2002), pp. 12-17, Interlaken, Switzerland, February 2002.

[62] Laykin S., Edan Y. and Alchanatis V. 1999. Development of a quality sorting machine using vision and impact. ASAE Paper No. 99-3144. ASAE, St. Joseph MI-49085.

[63] Laykin S., Alchanatis V. and Edan Y. 2002. Image-processing algorithms for tomato classification. Transactions of the American Society of Agricultural Engineering. 45(3): 851-858.

[64] Leeman V. and Destain M.F., 2004. A real-time grading method of apples based on features extracted from defects. Journal of Food Engineering. 61(2004): 83–89.

[65] Leray P. and Gallinari P. 1999. Feature selection with neural networks. Behaviormetrika. 26(1): 145-166.

[66] Lim C. P. and Harrison R. F. 2003. Online pattern classification with multiple neural network systems: An experimental study. IEEE Transactions on Systems, Man and Cybernetics – Part C: Application and Reviews. 33(2):235-247.

[67] Lin C. T. and Lee C. S. G. 1996. Neural fuzzy systems. Englewood Cliffs, NJ: Prentice-Hall.

[68] Lippmann R. 1987. An introduction to computing with neural nets. IEEE Acoustics, Speech and Signal Processing Magazine. 4(2): 4-22.

[69] Liu H. and Motoda H. 1998. Feature selection for knowledge discovery and data mining. 1st Ed. Kluwer Academic Publisher.

[70] Lu R. 2003. Detection of bruises on apples using near-infrared hyperspectral imaging. Transactions of the American Society of Agricultural Engineering. 46(2): 523-530

[71] Lu R. and Peng Y., 2005. A laser-based multispectral imaging system for real-time detection of apple fruit firmness. Proceedings of SPIE -- Volume 5996 Optical Sensors and Sensing Systems for Natural Resources and Food Safety and Quality 59960F (Nov. 8, 2005).

[72] Matlab. 2002. The Mathwork. Inc, South Narick, MA 01760.

[73] Markou M. and Singh S. 2003. Novelty detection: a review – part 1: statistical approaches. Signal Processing. 83 (2003): 2481-2497.

[74] Markou M. and Singh S. 2003. Novelty detection: a review – part 2: neural network based approaches. Signal Processing. 83 (2003): 2499-2521.

[75] Mattone R. 2002. The growing neural map: An on-line competitive clustering algorithm. Proceedings of IEEE International Conference on Robotic and Automation (ICRA 02). Vol.4. p:3888-3893.

[76] Mehl P. M., Chao K., Kim M. and Chen Y. R. 2002. Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. Applied Engineering in Agriculture. 18(2): 219-226.

[77] Miller W. M. 1985. Decision model for computer-based grade separation of fresh produce, Transactions of the American Society of Agricultural Engineering. 28(4):1341-1345.

[78] Miller W. M., Drouillard G. P. 2001. Multiple feature analysis for machine vision grading of Florida citrus. Applied Engineering in Agriculture. 17(5):627-633.

[79] Miller B. K. and Delwich M. J. 1989. Color vision system for peach grading. Transactions of the American Society of Agricultural Engineering. 7(4):1484-1490.

[80] Moini S., O'Brien M. and Chen P. 1980. Spectral properties of mold and defects of processing tomatoes. Transactions of the American Society of Agricultural Engineering. 23(4): 1062-1065.

[81] Moini S. and O'Brien M. 1978. Tomato color measurement versus maturity. Transactions of the American Society of Agricultural Engineering 21(4): 797-800.

[82] Moini S. and O'Brien M. 1980. Reflectance as tomato grade category standards. Transactions of the American Society of Agricultural Engineering 23(4): 1066-1067.

[83] Njoroge J.B., Ninomia K., Kondo N. and Toita H. 2002. Automated fruit grading system using image processing. Proceedings of the 41st Society of Instruments and Control Engineering (SICE) annual conference, Vol. 2, p.1346-1351

[84] Noordam J. C. , Otten G. W., Timmermans A. J. M. and Van Zwol B. H. 2000. High speed potato grading and quality inspection based on a color vision System, Department Production & Control Systems, ATO. P. O. Box 17, 6700 AA, Wageningen, the Netherlands.

[85] O'Brien M. and Sarkar S.C. 1974. System for optical transmission characteristics for computerized grading tomatoes. Transactions of the American Society of Agricultural Engineering. 17(2): 193-194.

[86] Pathaveerat S., Chen P., McCarthy M.J., 2001. On-line NMR evaluation of avocado fruit quality. ASAE Paper No. 01-3003. ASAE, St. Joseph MI-49085.

[87] Pearson T. C., Doster M. A., Michailides T. J. 2001. Automated detection of pistachio defects by machine vision, Applied Engineering in agriculture. 17(5): 729-732.

[88] Peleg K. 1981. Quality criteria of sorting operations. Transactions of the American Society of Agricultural Engineering. 24(6): 1459-1465.

[89] Peleg K., Ben-Hanan U. 1992. Adaptive sorting by prototype population. Pattern recognition letters 15(1994):111-123.

[90] Peleg, K., 1999. Development of a commercial fruit firmness sorter. Journal of Agriculture Engineering Research. 72(3): 231-238.

[91] Picus M., Peleg K. 1999. Optimal adaptive classification of Agricultural Produce, Computer and Electronics in Agriculture. 22(1999), 11-27.

[92] Pitts M. J., Abott, J.A., Amstrong, P.R., Brown, G.K., Brusewitz, G.H., David, D.C., Delwich, M.J., Galili, N., Gan-Mor, S. Haugh, C.G., Massey, D. Mizrach, A., Nahir, D., Peleg, K., Rohrbach, R.P., Sarig, Y., Schaare, P.N., Schmilovitch, Z., Shmulevich, I., Stone, M.L., Stroshine, R.L., and Younce, F.L. 1993. Sensing fruit and vegetable firmness. In Proceedings of the Intl. Workshop on Nondestructive Technologies for quality evaluation of fruits and vegetables. ASAE, St. Joseph MI-49085.

[93] Polderdik J. J., Tijskens L. M. M., Robberts J. E., and Van der Valk H.C.P. 1993. Predictive model of keeping quality of tomatoes. Postharvest Biology and Technology 2: 179-185.

[94] Pydipati R., Burks T.F., Lee W.S. 2005. Statistical and neural network classifiers for citrus disease detection using machine vision. Transactions of the American Society of Agricultural Engineering. 48(5): 2007-2014.

[95] Ripley B. D. 1996. Pattern Recognition and Neural Networks. Cambridge University Press 1996.

[96] Ruiz M. and Chen, P. 1982. Use of the first derivative of spectral reflectance to detect mold of tomatoes. Transactions of the American Society of Agricultural Engineering. 25(3): 759-762.

[97] Ruiz-Altisent M. and Ortiz-Canavate J., 2005. Instrumentation and procedures for commercial non-destructive determination of firmness of various fruits. ASAE Paper No. 05-6176. ASAE, St. Joseph MI-49085.

[98] Ruta D. and Gabrys B. 2000 . An overview of classifier fusion methods, Computing and Information Systems, 7 (2000):1-10.

[99] Rong J. G. and Wang X. Z.2004. A study on similarity measures for the feature selection method offs. Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai. Pp: 1869-1873.

[100] Sarkar N. and Wolfe R. R. 1985. Computer vision based system for quality separation of fresh market tomatoes. Transactions of the American Society of Agricultural Engineering. 28(5): 1714-1718.

[101] Scott M. J. J., Niranjan M. and Prager R.W. 1998. Parcel: feature subset selection in variable cost domains. Thesis work. Cambridge University Engineering Department Trumpington Street Cambridge CB2 1PZ, England .

[102] Shahin M. A., Tollner E. W., Evans M. D. and Arabina H. R. 1999. Watercore features for sorting red delicious apples: a statistical approach. Transactions of the American Society of Agricultural Engineering. 42(6):1889-1896.

[103] Shahin M. A., Verma B. P. and Tollner E. W. 2001. Fuzzy logic model for predicting peanut maturity. Transactions of the American Society of Agricultural Engineering. 43(2):483-490.

[104] Shahin M. A., Tollner E. W., McClendon R. W. and Arabina H. R. 2002. Apple classification based on surface bruises using image processing and neural network. Transactions of the American Society of Agricultural Engineering. 45(5):1619-1627.

[105] Shewfelt R. L., Prussia S. E., Resurreccion V. A., Hurst W. C. and Campbell D. T. 1987. Quality changes of vine-ripened tomatoes within the postharvest handling system. Journal of Food Science. 52(3): 661-664.

[106] Shmulevich I., Galili N. and Howarth M.S. 2003. Nondestructive dynamic testing of apples for firmness evaluation. Postharvest Biology and Technology. 29 (2003):287-299.

[107] Somol P., Pudil P. and Kittler J. 2004. Fast branch & bound algorithm for optimal feature selection. IEEE Transactions on Pattern Analysis and Machine Vision. 26(7): 900-912.

[108] Steenhoek L. W., Misra M. K., Batchelor W.D. and Davidson J. L. 2001. Probabilistic neural network for segmentation of features in corn kernel images. Applied Engineering in Agriculture. 17(2):225-234.

[109] Stephenson K. Q. 1976. Color sorting of tomatoes. In: Quality Detection of Foods. Ed. Gaffny J., ASAE Monograph, St. Joseph, Mich 199-201.

[110] Studman C. and Boyd L. 1994. Measurement of firmness in fruits and vegetables. AgEng. Conference, Milan Report No. #94-G-066.

[111] Thiel C., Schwenker F. and Palm G. 2005. Using Dempster-Shafer theory in MCF systems to reject samples. Proceedings of the 6th International Workshop on Multiple Classifier Systems. MSC 2005. p: 118-127.

[112] Throop J. A., Rehkugler G. E. 1989. Image processing algorithm for apple defect detection. Transactions of the American Society of Agricultural Engineering. 32(1): 267-272.

[113] Throop J. A., Aneshansly D. J., Upchurch B. L. and Anger B. 2001. Apple orientation on two conveyors: performance and predictability based on fruit shape characteristics. Transactions of the American Society of Agricultural Engineering 44(1): 99-109.

[114] Throop J. A., Aneshansley D. J. and Anger B. 1999. Inspection Station Detects Defects on Apples in Real Time, ASAE Paper No. 993205 ASAE, St. Joseph MI-49085.

[115] Tomak I. 1979. Two modifications of CNN, IEEE Transactions on Systems, Man and Cybernetics, SMC-17(1):769-772 .

[116] Turney P. 2000. Types of cost in inductive concept learning. Proceedings Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning. P:15-21, Stanford University, California.

[117] Unay D., Gosselin B. and Debeir O. 2006. Apple stem and calyx recognition by decision trees. Proceeding of IASTED (541) Visualization, Imaging, and Image Processing-2006 .

[118] Upchurch B., Affeldt H., Hruschaka W., Norris K. and Throop J. 1990. Spectrophotometric study of bruises on whole 'red delicious' apples, Transactions of the American Society of Agricultural Engineering. 33(2):585-589.

[119] Utku H. 2000. Application of the feature selection method to discriminant digitized wheat varieties, Journal of Food Engineering. 46 (2000): 211-216, 2000.

[120] Wang Q. R. and Suen C. Y. 1987. Large tree classifier with heuristic search and global training. IEEE Transactions on Pattern Analysis and Machine Intelligence. 9(1):91-102.

[121] Wang D., Dowell F. E. and Lacey R. E. 1999. Single wheat kernel color classification using neural networks. Transactions of the American Society of Agricultural Engineering. 42(1): 233-240.

[122] Wen Y. and Tao Y. 1999. Building a rule-based machine vision system for defect inspection on apple sorting and packing lines. Expert Systems with Applications. 16(1999):307-313.

[123] Windridge D. and Kittler J. 2000. Combined classifier optimization via feature selection. Proceedings of the Joint IAPR International Workshop on SSPR& SPR 2000, pp. 687-695.

[124] Xiaobo Z. and Jiewen Z. 2005. Apple quality assessment by fusion three sensors. Proceedings of the Fourth IEEE Conference on Sensors 2005. P:389-392.

[125] Yang C.C. , Prasher S.O., Whalen J. and Goel P.K. 2001. Application of data mining technology for hyperspectral imagery classification in agricultural fields, ASAE Paper No. 01-3116 ASAE, St. Joseph MI-49085.

[126] Yeung K.Y. and Ruzzo W. A. 2000. An empirical study on Principal Component Analysis for gene expression data, Technical report UW-CSE-2000-11-3, November, 2000.

[127] Yu S., De Backer S. and Scheunders P. 2002. Genetic Feature Selection Combined with Composite Fuzzy NN Classifiers for Hyperspectral Satellite Imagery. Pattern Recognition Letters. 23(2002): 183-190.

[128] Zhang G. P. 2000. Neural networks for classification: A survey. IEEE Transactions on Systems Man and Cybernetics-Part C: Applications and Reviews. 30(4): 451-462.

[129] Zhang L., Gang Sun, Jun Guo. 2004. Feature Selection for Pattern Classification Problems. Proceedings of the The Fourth International Conference on Computer and Information Technology (CIT'04). pp:233 – 237.

8 Appendices

Appendix I Vision system & Image processing algorithms

The vision station is presented in the following figures.

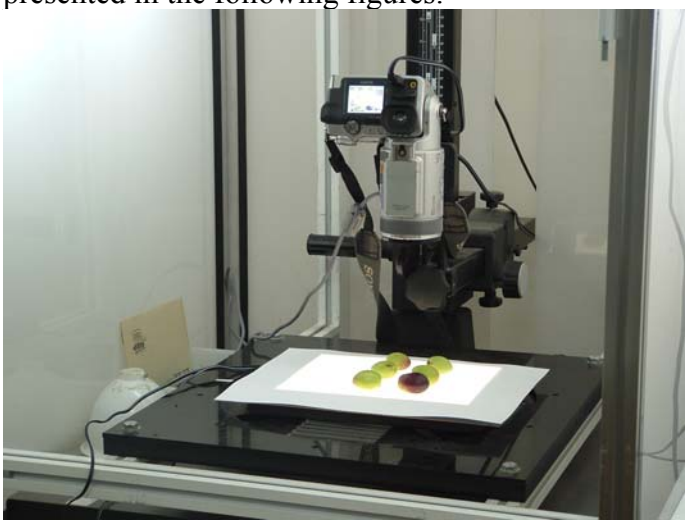

Figure 50: Grabbing system(1) – The camera and light table.

The system composed of :

- Color CCD camera (Sony Cyber-shot DSC-F717).
- Four 17 W halogen spotlights were placed, two on each wall with a constant distance between them to ensure uniform illumination distribution in the sampling area. The walls are made of diffuse structure that enables the light uniformity.

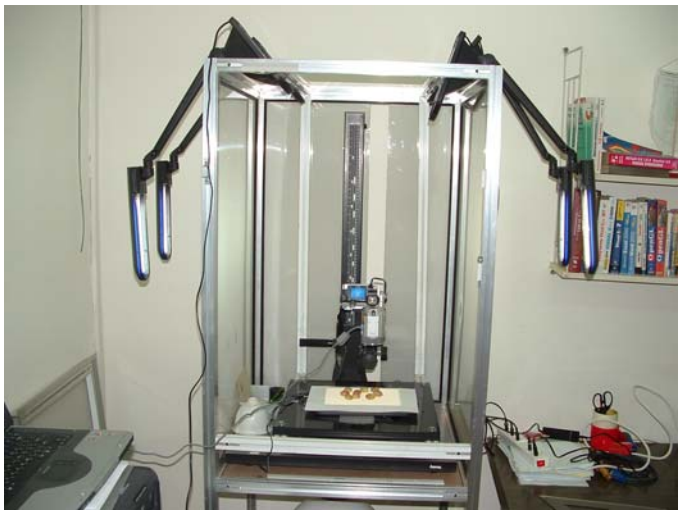

Figure 51: Grabbing system (2) – Overall system

- The acquired RGB image was transformed to an HSI (hue, saturation, intensity) image using a standard software procedure (Matlab, 7.1).

Images of Olives Harvested Varieties:

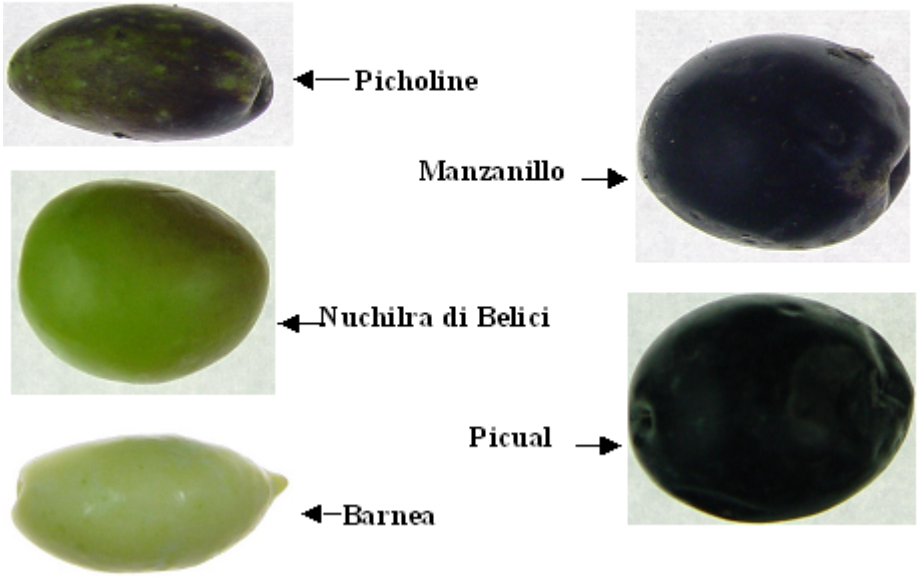

Figure 52: Olives varieties

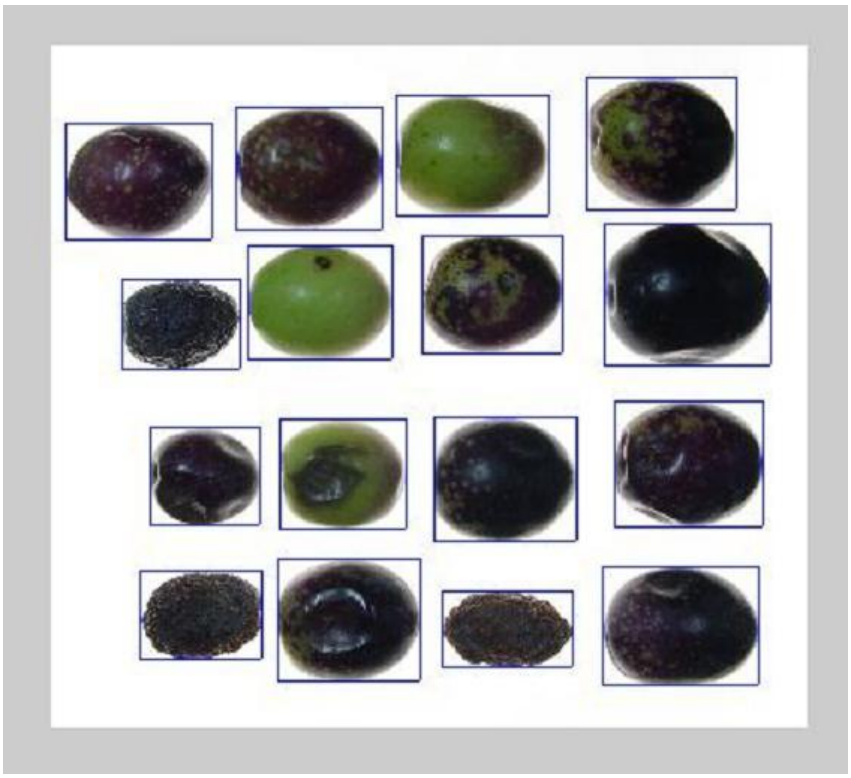

Figure 53: Extracted olives

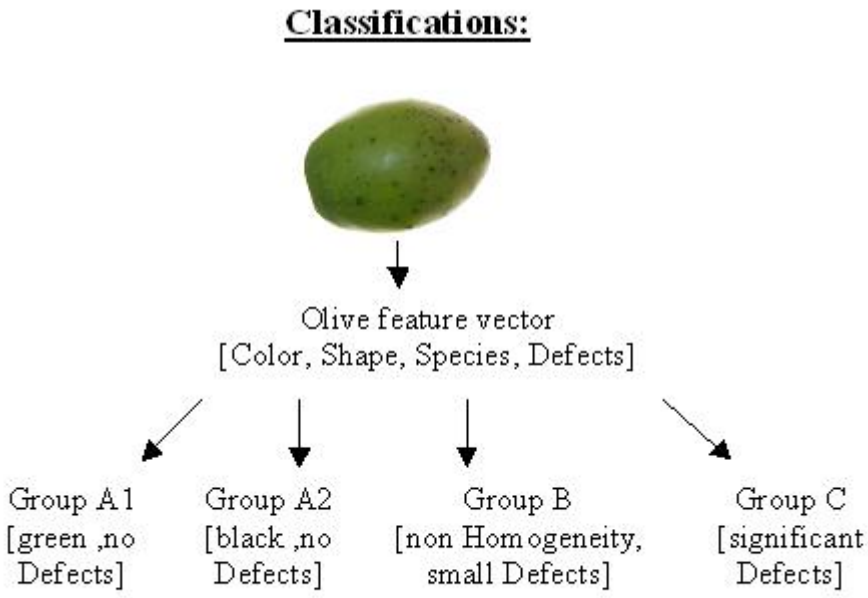

Figure 54: Olives grades

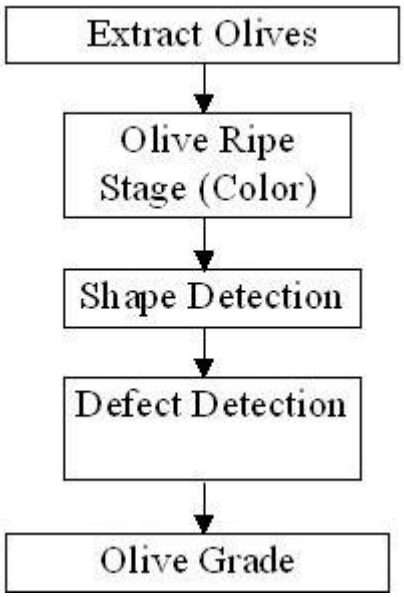

Figure 55: Image processing flow chart

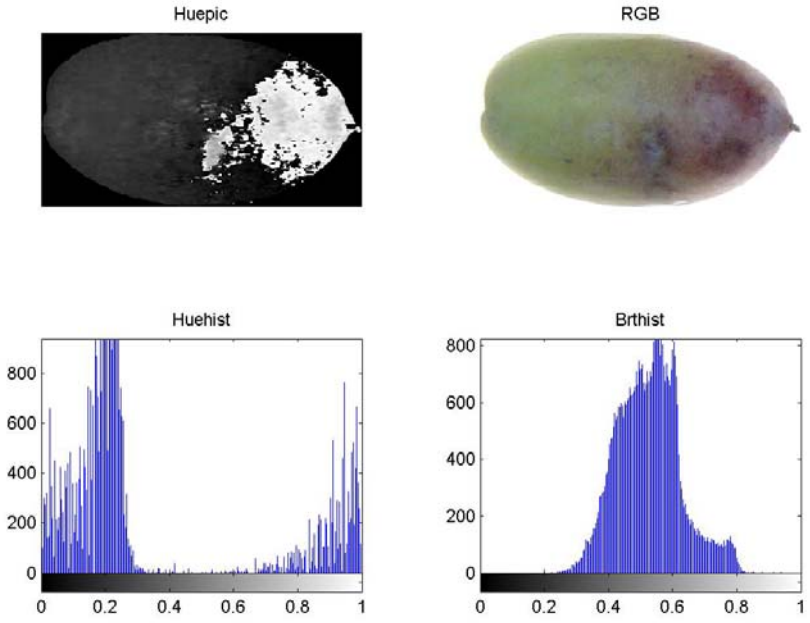

Figure 56: Color histograms (midway maturity olive)

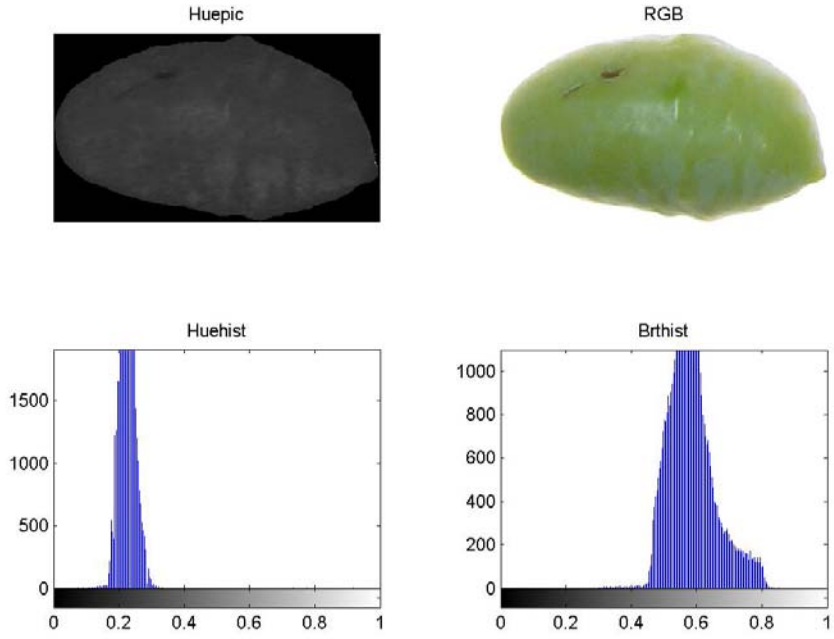

Figure 57: Color histograms (green olive)

Appendix II Olives data description

21 populations based on different spices of olives and from different location and dates:

Table 18: Olives varieties

	Population Name	# Samples	Location (RamatNegev)	Special Features	Harvest Day[6]
1	Barnea	693	Tree 1	Shape	15/12/2005
2	Barnea 2	460	Tree 2	Shape	15/12/2005
3	Manzanilo	182			15/12/2005
4	Pishulin	165		Bad Quality	15/12/2005
5	Nuchilera Belizi	427		Big size	15/12/2005
6	Pikual	279		BIG Black	15/12/2005
7	Barnea	184/305	Tree 1	Shape	H30/11/2004 G 2/12/2005
8	Barnea (Tree 2)	104/180	Tree 2	Shape	H30/11/2004 G 2/12/2005
9	Ochiblanka	137/203		Color Homogeneity	H30/11/2004 G 2/12/2005
10	Barnea (defective Tree)	96/132	Tree 3	Bad Quality	H30/11/2004 G 2/12/2005
11	Blanketa	114/182		Small size	H30/11/2004 G 2/12/2005
12	Nuchilera Belizi	73/145		Big size	H30/11/2004 G 2/12/2005
13	Prolin	222/327		Small size	H30/11/2004 G 2/12/2005
14	Nuchilera Belizi	505		Big size	H 16/12/2004 G 17/12/2004
15	Unknown (~Barnea)	915	Taken from packing	Ripe stage	H 16/12/2004 G 17/12/2004
16	Prolin	978		Small	H 16/12/2004 G 24/12/2004
17	Arbikina	1118	First Tree	SMALL	H 16/12/2004 G 24/12/2004
18	Ochiblanka	704		Color Homogeneity	H 16/12/2004 G 25/12/2004
19	Pikual	364	Taken from packing	BIG Black	H 16/12/2004 G 25/12/2004
20	Kurineiky	1313		SMALLEST	H 16/12/2004 G 26/12/2004
21	Barnea	974	Taken from packing (Bad Quality)	Shape+ Ripe stage	H 16/12/2004 G 27/12/2004

[6] H – harvest day; G – grabbing day – in-between kept in refrigerator
At 16/12/2004 – Rainy day.

Appendix III Olives features description

Table 19: Features description

Feature name	***implementation***	***description***
Area	BW matrix	Number of relevant pixels
Length	BW matrix	Length of Olive_Bounding_Box
Width	BW matrix	Width of Olive_Bounding_Box
Eccentricity		L1/L2: Ellipse (same second moment as the region) centers distance (L1) by Long Axis (L2) (~1 – Circle; ~0 - Area Line).
Orientation		
Equiv. Diameter		Diameter of same area circle
Compactness (Shape)		(Perimeter/Area^2)
Max_Distance		maximum distance from center to edge
Length_vec		The distance vector length
FD1 (1st FFT coefficient)		Average Radius
FD2 (2nd FFT coefficient)		Bendingness
FD3		Elongation
FD4		
FD5		
Ratio_Elong		(FD(1)-2*(FD(3)))/(FD(1)+2*(FD(3)))
Ave_G		Color
Var_G		Color Variance
DefectPer		
Texture (Mean)		Ave intensity (GrayImage).
Texture (Std)		Standard deviation – measure of average contrast
Texture (Smoothness)		$R = 1 - 1/(1 + \sigma^2)$
Texture (3rd moment)		Measure of the skewness of the histogram
Texture (Uniformity)		$U = \sum_{i=0}^{L-1} p^2(z_i)$
Entropy		Measure of randomness

Appendix IV Expert Panel GUI

The Matlab (7^{TM}) GUI that was used for the two panelists work is at Figure 58.
The GUI contains three operation areas. The first is the olive label and color (**b** and **c** in the figure). These are the general grades – *labels* contains 4 grades and *color* contains 5 grades.
The '*defect type and severity*' (**d, e** and **f**) section is the more detailed one. First the user mark the defect type (**e**) and then the defect severity level (**f**). The five defect types mentioned in the GUI are the most common defects: *fly defect* (automatically get the worst grade), *wrinkles*, *pressure*, *mechanical* and *rot* defects.
The last section (**g**) is the brightness bar which give the user the option of improving the image visibility.

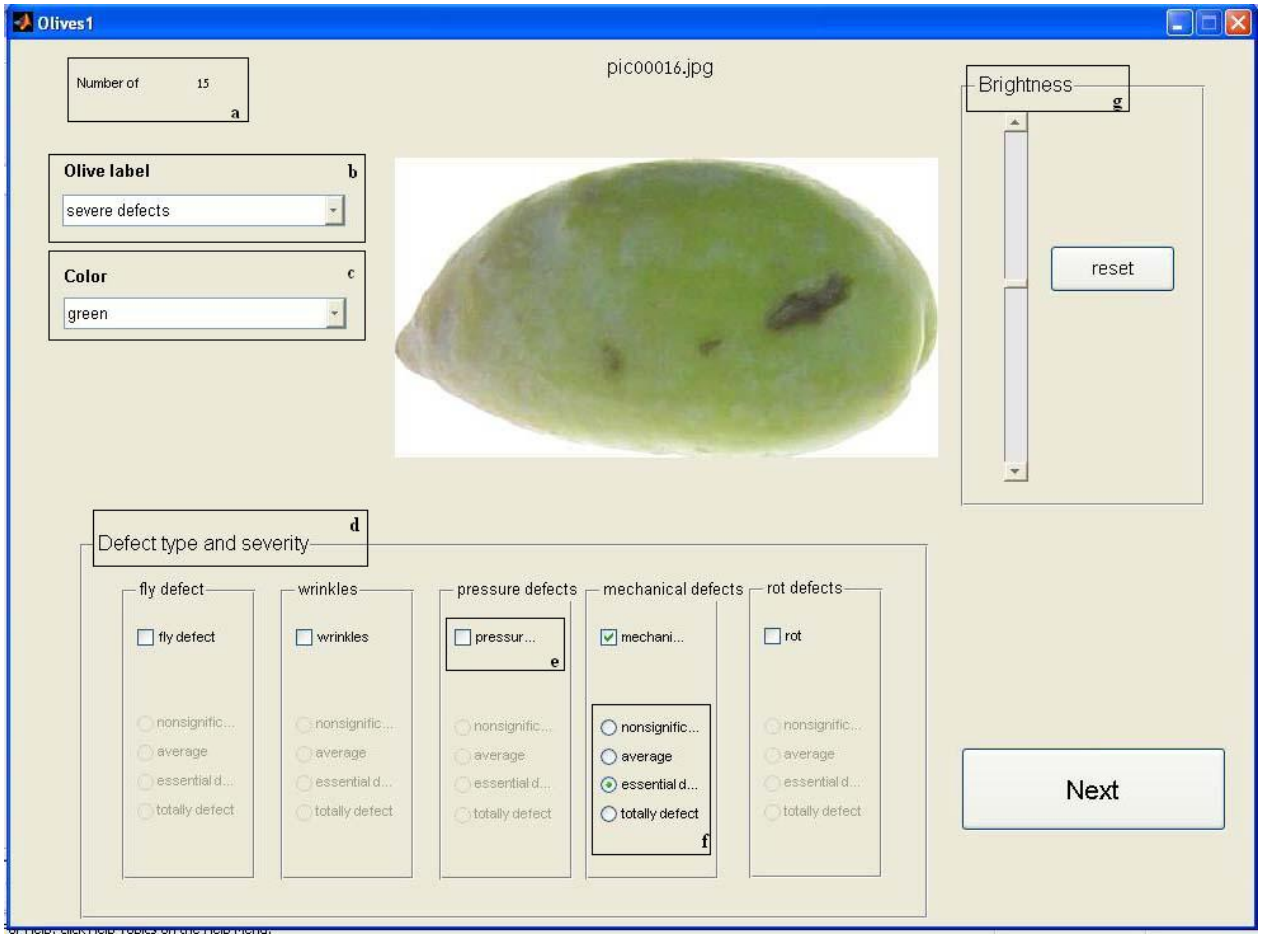

Figure 58: Olives panel GUI

Appendix V The on-line clustering algorithm

The original on line clustering algorithm (Guedalia *et al.*, 1999) is detailed in the following pseudo code. For each centroid α, let y_α be the location and c_α the counter of the centroid:

Step 0) Set $k=3$ /* set the first 3 data points to be the initial k centroids. */

Step 1) Get data point x

Step 2) $winner = \alpha \ \underline{s.t.}\ \|y_\alpha - x\|$ is minimal /*$winner$ - The centroid closest to the data point.*/

Step 3) $y_{winner} \leftarrow y_{winner} + \frac{x - y_{winner}}{c_{winner} + 1}$ /*Update the location and counter of the $winner$ centroid */

$c_{winner} \leftarrow c_{winner} + 1$

Step 4) $\{\gamma, \delta\} = \arg\min_{\gamma \neq \delta} \|y_\gamma - y_\delta\|$ /*Find the two most similar centroids (closest to each other)*/

Step 5) $y_\gamma \leftarrow \frac{y_\gamma c_\gamma + y_\delta c_\delta}{c_\gamma + c_\delta}$ /*Merge the two redundant centroids.*/

$c_\gamma \leftarrow c_\gamma + c_\delta$

Step 6) $y_\delta = x; c_\delta = 0$ /*Initialize the new centroid with the last data point. It may indicate the arrival of a new cluster of data (population)*/

Step 7) Go to step 1.

Figure 59: On-line clustering algorithm

Appendix VI Similarity measures

Table 20: Offline Similarity Measure

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
1	**2**	**16**	**0.2599717**	**0.0726101**	**0.8554033**	**1**
1	3	16	0.4843019	0.0000011	0.5060942	0
1	**4**	**16**	**0.2543785**	**0.0537339**	**0.8383415**	**1**
1	5	16	0.4289169	0.1295740	0.8934223	0
1	6	16	0.4956528	0.0000002	0.4626533	0
1	**7**	**16**	**0.2873446**	**0.0396341**	**0.8241855**	**1**
1	**8**	**16**	**0.2563744**	**0.0423501**	**0.8336830**	**1**
1	9	16	0.3411934	0.1248844	0.8864424	1
1	10	16	0.3272320	0.0258910	0.8066193	1
1	**11**	**16**	**0.2141131**	**0.0489578**	**0.8389789**	**1**
1	12	16	0.4125023	0.0452449	0.8328870	0
1	13	16	0.4268367	0.0555402	0.8431684	0
1	14	16	0.3073054	0.0166107	0.8022292	1
1	15	16	0.3934981	0.0103934	0.7733074	1
1	16	16	0.4568689	0.0071551	0.7623228	0
1	**17**	**16**	**0.2430299**	**0.0253214**	**0.8152759**	**1**
1	18	16	0.6002140	0.0027664	0.7092093	0
1	19	16	0.6212220	0.0000000	0.4348616	0
1	20	16	0.5023976	0.0016429	0.7121305	0
1	21	16	0.4807657	0.0007170	0.6857637	0
2	1	16	0.2599717	0.0726101	0.8554033	1
2	3	16	0.6446618	0.0000004	0.4739157	0
2	4	16	0.3240864	0.0268389	0.8031989	1
2	5	16	0.4900826	0.0199640	0.8071495	0
2	6	16	0.6496801	0.0000001	0.4328719	0
2	7	16	0.4412021	0.0146560	0.7829015	0
2	8	16	0.2715743	0.0436923	0.8300227	1
2	9	16	0.3629259	0.0471526	0.8354614	1
2	10	16	0.3472392	0.0255501	0.8010878	1
2	11	16	0.3096585	0.0388623	0.8300572	1
2	12	16	0.4427491	0.0209877	0.7972867	0
2	13	16	0.4861413	0.0112119	0.7715701	0
2	14	16	0.3655478	0.0067897	0.7768335	0
2	15	16	0.4503086	0.0050205	0.7408830	0
2	16	16	0.5043974	0.0013832	0.6980878	0
2	17	16	0.3506489	0.0040980	0.7396472	0
2	18	16	0.7118687	0.0007643	0.6608950	0
2	19	16	0.7432353	0.0000000	0.3975812	0
2	20	16	0.5122602	0.0003117	0.6540368	0
2	21	16	0.5536444	0.0002512	0.6364184	0
3	1	16	0.4843019	0.0000011	0.5060942	0
3	2	16	0.6446618	0.0000004	0.4739157	0
3	4	16	0.3794952	0.0000109	0.5731300	0
3	5	16	0.5930157	0.0000004	0.4700016	0
3	6	16	0.1699748	0.0211458	0.8037918	1
3	7	16	0.5498698	0.0000001	0.4247965	0
3	8	16	0.5348589	0.0000001	0.4126882	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
3	10	16	0.5599689	0.0000001	0.4199258	0
3	11	16	0.5072556	0.0000001	0.4304739	0
3	12	16	0.6339349	0.0000000	0.4122828	0
3	13	16	0.6373792	0.0000001	0.4415902	0
3	14	16	0.5693598	0.0000003	0.4591029	0
3	15	16	0.4443617	0.0000061	0.5398621	0
3	16	16	0.6232087	-0.0000002	0.4794044	0
3	17	16	0.5036724	0.0000007	0.4817347	0
3	18	16	0.5006201	0.0000083	0.5649087	0
3	19	16	0.5493722	0.0000391	0.6425998	0
3	20	16	0.7463929	0.0000008	0.4503804	0
3	21	16	0.4560956	0.0000242	0.6059211	0
4	1	16	0.2543785	0.0537339	0.8383415	1
4	2	16	0.3240864	0.0268389	0.8031989	1
4	3	16	0.3794952	0.0000109	0.5731300	0
4	5	16	0.4684317	0.0199336	0.8018438	0
4	6	16	0.4036667	0.0000022	0.5313479	0
4	7	16	0.3983584	0.0057436	0.7412084	0
4	8	16	0.2482983	0.0060858	0.7450437	0
4	9	16	0.3008026	0.0417567	0.8289540	1
4	10	16	0.3065487	0.0040040	0.7255240	0
4	11	16	0.2969666	0.0047781	0.7375812	0
4	12	16	0.4933828	0.0064406	0.7465026	0
4	13	16	0.4981634	0.0146564	0.7857308	0
4	14	16	0.3725760	0.0042502	0.7379810	0
4	15	16	0.2870388	0.0231298	0.8074718	1
4	16	16	0.5043155	0.0058107	0.7566843	0
4	17	16	0.2829799	0.0068146	0.7514136	0
4	18	16	0.5459757	0.0233037	0.8073543	0
4	19	16	0.5995259	0.0000007	0.4860504	0
4	20	16	0.5807913	0.0007885	0.6762674	0
4	21	16	0.4092446	0.0057986	0.7610421	0
5	1	16	0.4289169	0.1295740	0.8934223	0
5	2	16	0.4900826	0.0199640	0.8071495	0
5	3	16	0.5930157	0.0000004	0.4700016	0
5	4	16	0.4684317	0.0199336	0.8018438	0
5	6	16	0.6004060	0.0000001	0.4340520	0
5	7	16	0.2842674	0.0483424	0.8331084	1
5	8	16	0.4418710	0.0115464	0.7801165	0
5	9	16	0.5004796	0.0379223	0.8311367	0
5	10	16	0.5654017	0.0084242	0.7568640	0
5	11	16	0.3944239	0.0152383	0.7911874	1
5	12	16	0.2502602	0.0316528	0.8155679	1
5	13	16	0.2360910	0.0997739	0.8706733	1
5	14	16	0.2812121	0.0484871	0.8341936	1
5	15	16	0.5041840	0.0143578	0.7841573	0
5	16	16	0.2260171	0.0329079	0.8201560	1
5	17	16	0.3801370	0.0762269	0.8566035	1
5	18	16	0.6018378	0.0085787	0.7610942	0
5	19	16	0.6167108	0.0000000	0.4660355	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
5	20	16	0.2985383	0.0088844	0.7704347	0
5	21	16	0.5685319	0.0008588	0.6988238	0
6	1	16	0.4956528	0.0000002	0.4626533	0
6	2	16	0.6496801	0.0000001	0.4328719	0
6	3	16	0.1699748	0.0211458	0.8037918	1
6	4	16	0.4036667	0.0000022	0.5313479	0
6	5	16	0.6004060	0.0000001	0.4340520	0
6	7	16	0.5453830	0.0000000	0.3879166	0
6	8	16	0.5633319	0.0000000	0.3723522	0
6	9	16	0.6276653	0.0000001	0.4301515	0
6	10	16	0.6051584	0.0000000	0.3720624	0
6	11	16	0.5116109	0.0000000	0.3901721	0
6	12	16	0.6363950	0.0000000	0.3704352	0
6	13	16	0.6389169	0.0000000	0.4023266	0
6	14	16	0.5702769	0.0000001	0.4297225	0
6	15	16	0.4357158	0.0000015	0.5018398	0
6	16	16	0.6203610	0.0000001	0.4535806	0
6	17	16	0.5127878	0.0000002	0.4529719	0
6	18	16	0.4822143	0.0000024	0.5230795	0
6	19	16	0.5108687	0.0000661	0.6144997	0
6	20	16	0.7466479	0.0000001	0.4138759	0
6	21	16	0.4305641	0.0000142	0.5805596	0
7	1	16	0.2873446	0.0396341	0.8241855	1
7	2	16	0.4412021	0.0146560	0.7829015	0
7	3	16	0.5498698	0.0000001	0.4247965	0
7	4	16	0.3983584	0.0057436	0.7412084	0
7	5	16	0.2842674	0.0483424	0.8331084	1
7	6	16	0.5453830	0.0000000	0.3879166	0
7	8	16	0.3794785	0.0402995	0.8294378	1
7	9	16	0.4242165	0.0335473	0.8146384	0
7	10	16	0.4758961	0.0149149	0.7799698	0
7	11	16	0.2855943	0.0951499	0.8734668	1
7	12	16	0.2222876	0.1130222	0.8745762	1
7	13	16	0.2404971	0.1501164	0.8902521	1
7	14	16	0.2699866	0.0087097	0.7596694	0
7	15	16	0.4803093	0.0031481	0.7213648	0
7	16	16	0.3644653	0.0030962	0.7147496	0
7	17	16	0.2499214	0.0124644	0.7681994	1
7	18	16	0.5689227	0.0010443	0.6746958	0
7	19	16	0.5834282	0.0000000	0.4161854	0
7	20	16	0.3793237	0.0012455	0.6869199	0
7	21	16	0.5392759	0.0001051	0.6268722	0
8	1	16	0.2563744	0.0423501	0.8336830	1
8	2	16	0.2715743	0.0436923	0.8300227	1
8	3	16	0.5348589	0.0000001	0.4126882	0
8	4	16	0.2482983	0.0060858	0.7450437	0
8	5	16	0.4418710	0.0115464	0.7801165	0
8	6	16	0.5633319	0.0000000	0.3723522	0
8	7	16	0.3794785	0.0402995	0.8294378	1
8	9	16	0.2595184	0.0530756	0.8426866	1

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
8	10	16	0.1621573	0.0899169	0.8639525	1
8	11	16	0.2012264	0.0527045	0.8420213	1
8	12	16	0.4194305	0.0780225	0.8608434	0
8	13	16	0.4458699	0.0318996	0.8205812	0
8	14	16	0.3673118	0.0015512	0.7000203	0
8	15	16	0.4386579	0.0006801	0.6587464	0
8	16	16	0.5222519	0.0004578	0.6476486	0
8	17	16	0.3402941	0.0021006	0.7024167	0
8	18	16	0.5908964	0.0002167	0.6187591	0
8	19	16	0.6981040	0.0000000	0.3618007	0
8	20	16	0.5505010	0.0002061	0.6220498	0
8	21	16	0.5393642	0.0000409	0.5855221	0
9	1	16	0.3411934	0.1248844	0.8864424	1
9	2	16	0.3629259	0.0471526	0.8354614	1
9	3	16	0.5976295	0.0000006	0.4757074	0
9	4	16	0.3008026	0.0417567	0.8289540	1
9	5	16	0.5004796	0.0379223	0.8311367	0
9	6	16	0.6276653	0.0000001	0.4301515	0
9	7	16	0.4242165	0.0335473	0.8146384	0
9	8	16	0.2595184	0.0530756	0.8426866	1
9	10	16	0.3204040	0.0312353	0.8174103	1
9	11	16	0.3485005	0.0215217	0.7993269	1
9	12	16	0.4684119	0.0870180	0.8653651	0
9	13	16	0.4855170	0.0753373	0.8580219	0
9	14	16	0.4302437	0.0048765	0.7421277	0
9	15	16	0.4987217	0.0055824	0.7400815	0
9	16	16	0.5651108	0.0029410	0.7210952	0
9	17	16	0.3971101	0.0074928	0.7570209	0
9	18	16	0.6435480	0.0022723	0.6988242	0
9	19	16	0.7213284	0.0000001	0.4213995	0
9	20	16	0.5862683	0.0008319	0.6791752	0
9	21	16	0.6041799	0.0005722	0.6631930	0
10	1	16	0.3272320	0.0258910	0.8066193	1
10	2	16	0.3472392	0.0255501	0.8010878	1
10	3	16	0.5599689	0.0000001	0.4199258	0
10	4	16	0.3065487	0.0040040	0.7255240	0
10	5	16	0.5654017	0.0084242	0.7568640	0
10	6	16	0.6051584	0.0000000	0.3720624	0
10	7	16	0.4758961	0.0149149	0.7799698	0
10	8	16	0.1621573	0.0899169	0.8639525	1
10	9	16	0.3204040	0.0312353	0.8174103	1
10	11	16	0.2727703	0.0131100	0.7738044	1
10	12	16	0.5554475	0.0330082	0.8139934	0
10	13	16	0.5796734	0.0097596	0.7580359	0
10	14	16	0.4745256	0.0012393	0.6821496	0
10	15	16	0.5010291	0.0008395	0.6632452	0
10	16	16	0.6300221	0.0004295	0.6444504	0
10	17	16	0.4304929	0.0015047	0.6881391	0
10	18	16	0.6412143	0.0001866	0.6091905	0
10	19	16	0.7550361	0.0000000	0.3642806	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
10	20	16	0.6624778	0.0001671	0.6105340	0
10	21	16	0.5767249	0.0000218	0.5711791	0
11	1	16	0.2141131	0.0489578	0.8389789	1
11	2	16	0.3096585	0.0388623	0.8300572	1
11	3	16	0.5072556	0.0000001	0.4304739	0
11	4	16	0.2969666	0.0047781	0.7375812	0
11	5	16	0.3944239	0.0152383	0.7911874	1
11	6	16	0.5116109	0.0000000	0.3901721	0
11	7	16	0.2855943	0.0951499	0.8734668	1
11	8	16	0.2012264	0.0527045	0.8420213	1
11	9	16	0.3485005	0.0215217	0.7993269	1
11	10	16	0.2727703	0.0131100	0.7738044	1
11	12	16	0.3560850	0.0491170	0.8374280	1
11	13	16	0.3900060	0.0302732	0.8157136	1
11	14	16	0.3389114	0.0024132	0.7254925	0
11	15	16	0.4339089	0.0011619	0.6936432	0
11	16	16	0.4411637	0.0008349	0.6765897	0
11	17	16	0.3033797	0.0034681	0.7253663	0
11	18	16	0.5579865	0.0001833	0.6177679	0
11	19	16	0.6138113	0.0000000	0.3755258	0
11	20	16	0.4853955	0.0002869	0.6390845	0
11	21	16	0.5196518	0.0000347	0.6083255	0
12	1	16	0.4125023	0.0452449	0.8328870	0
12	2	16	0.4427491	0.0209877	0.7972867	0
12	3	16	0.6339349	0.0000000	0.4122828	0
12	4	16	0.4933828	0.0064406	0.7465026	0
12	5	16	0.2502602	0.0316528	0.8155679	1
12	6	16	0.6363950	0.0000000	0.3704352	0
12	7	16	0.2222876	0.1130222	0.8745762	1
12	8	16	0.4194305	0.0780225	0.8608434	0
12	9	16	0.4684119	0.0870180	0.8653651	0
12	10	16	0.5554475	0.0330082	0.8139934	0
12	11	16	0.3560850	0.0491170	0.8374280	1
12	13	16	0.1377009	0.1584922	0.8955915	1
12	14	16	0.2533267	0.0048308	0.7353890	0
12	15	16	0.5297810	0.0018394	0.7055053	0
12	16	16	0.3173308	0.0011817	0.6734300	0
12	17	16	0.3241443	0.0061600	0.7388949	0
12	18	16	0.6207247	0.0005680	0.6563426	0
12	19	16	0.6318503	0.0000000	0.3897274	0
12	20	16	0.3099720	0.0005413	0.6486717	0
12	21	16	0.5879698	0.0000333	0.6090850	0
13	1	16	0.4268367	0.0555402	0.8431684	0
13	2	16	0.4861413	0.0112119	0.7715701	0
13	3	16	0.6373792	0.0000001	0.4415902	0
13	4	16	0.4981634	0.0146564	0.7857308	0
13	5	16	0.2360910	0.0997739	0.8706733	1
13	6	16	0.6389169	0.0000000	0.4023266	0
13	7	16	0.2404971	0.1501164	0.8902521	1
13	8	16	0.4458699	0.0318996	0.8205812	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
13	9	16	0.4855170	0.0753373	0.8580219	0
13	10	16	0.5796734	0.0097596	0.7580359	0
13	11	16	0.3900060	0.0302732	0.8157136	1
13	12	16	0.1377009	0.1584922	0.8955915	1
13	14	16	0.2458246	0.0094172	0.7595280	0
13	15	16	0.5256249	0.0058803	0.7509991	0
13	16	16	0.2782480	0.0070464	0.7470290	0
13	17	16	0.3363009	0.0248997	0.7996618	1
13	18	16	0.6070641	0.0037698	0.7316612	0
13	19	16	0.6221249	0.0000000	0.4378312	0
13	20	16	0.2613589	0.0032848	0.7235473	0
13	21	16	0.5893321	0.0002334	0.6657616	0
14	1	16	0.3073054	0.0166107	0.8022292	1
14	2	16	0.3655478	0.0067897	0.7768335	0
14	3	16	0.5693598	0.0000003	0.4591029	0
14	4	16	0.3725760	0.0042502	0.7379810	0
14	5	16	0.2812121	0.0484871	0.8341936	1
14	6	16	0.5702769	0.0000001	0.4297225	0
14	7	16	0.2699866	0.0087097	0.7596694	0
14	8	16	0.3673118	0.0015512	0.7000203	0
14	9	16	0.4302437	0.0048765	0.7421277	0
14	10	16	0.4745256	0.0012393	0.6821496	0
14	11	16	0.3389114	0.0024132	0.7254925	0
14	12	16	0.2533267	0.0048308	0.7353890	0
14	13	16	0.2458246	0.0094172	0.7595280	0
14	15	16	0.4029486	0.0167880	0.7827042	0
14	16	16	0.2580863	0.0864705	0.8682509	1
14	17	16	0.1692952	0.1043595	0.8737741	1
14	18	16	0.5805032	0.0030627	0.7058208	0
14	19	16	0.6013830	0.0000005	0.4559773	0
14	20	16	0.2877834	0.0102519	0.7873463	1
14	21	16	0.4894162	0.0004072	0.6376048	0
15	1	16	0.3934981	0.0103934	0.7733074	1
15	2	16	0.4503086	0.0050205	0.7408830	0
15	3	16	0.4443617	0.0000061	0.5398621	0
15	4	16	0.2870388	0.0231298	0.8074718	1
15	5	16	0.5041840	0.0143578	0.7841573	0
15	6	16	0.4357158	0.0000015	0.5018398	0
15	7	16	0.4803093	0.0031481	0.7213648	0
15	8	16	0.4386579	0.0006801	0.6587464	0
15	9	16	0.4987217	0.0055824	0.7400815	0
15	10	16	0.5010291	0.0008395	0.6632452	0
15	11	16	0.4339089	0.0011619	0.6936432	0
15	12	16	0.5297810	0.0018394	0.7055053	0
15	13	16	0.5256249	0.0058803	0.7509991	0
15	14	16	0.4029486	0.0167880	0.7827042	0
15	16	16	0.4834830	0.0290267	0.8080174	0
15	17	16	0.3333540	0.0309065	0.8175760	1
15	18	16	0.4492822	0.0315536	0.8144107	0
15	19	16	0.5264469	0.0000113	0.5343581	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
15	20	16	0.5921017	0.0030458	0.7149050	0
15	21	16	0.1801447	0.0158162	0.7853906	1
16	1	16	0.4568689	0.0071551	0.7623228	0
16	2	16	0.5043974	0.0013832	0.6980878	0
16	3	16	0.6232087	-0.0000002	0.4794044	0
16	4	16	0.5043155	0.0058107	0.7566843	0
16	5	16	0.2260171	0.0329079	0.8201560	1
16	6	16	0.6203610	0.0000001	0.4535806	0
16	7	16	0.3644653	0.0030962	0.7147496	0
16	8	16	0.5222519	0.0004578	0.6476486	0
16	9	16	0.5651108	0.0029410	0.7210952	0
16	10	16	0.6300221	0.0004295	0.6444504	0
16	11	16	0.4411637	0.0008349	0.6765897	0
16	12	16	0.3173308	0.0011817	0.6734300	0
16	13	16	0.2782480	0.0070464	0.7470290	0
16	14	16	0.2580863	0.0864705	0.8682509	1
16	15	16	0.4834830	0.0290267	0.8080174	0
16	17	16	0.3462631	0.1253339	0.8810044	1
16	18	16	0.5892214	0.0128277	0.7704514	0
16	19	16	0.5834617	0.0000027	0.5162810	0
16	20	16	0.2062121	0.0511505	0.8447287	1
16	21	16	0.5354194	0.0019732	0.7071628	0
17	1	16	0.2430299	0.0253214	0.8152759	1
17	2	16	0.3506489	0.0040980	0.7396472	0
17	3	16	0.5036724	0.0000007	0.4817347	0
17	4	16	0.2829799	0.0068146	0.7514136	0
17	5	16	0.3801370	0.0762269	0.8566035	1
17	6	16	0.5127878	0.0000002	0.4529719	0
17	7	16	0.2499214	0.0124644	0.7681994	1
17	8	16	0.3402941	0.0021006	0.7024167	0
17	9	16	0.3971101	0.0074928	0.7570209	0
17	10	16	0.4304929	0.0015047	0.6881391	0
17	11	16	0.3033797	0.0034681	0.7253663	0
17	12	16	0.3241443	0.0061600	0.7388949	0
17	13	16	0.3363009	0.0248997	0.7996618	1
17	14	16	0.1692952	0.1043595	0.8737741	1
17	15	16	0.3333540	0.0309065	0.8175760	1
17	16	16	0.3462631	0.1253339	0.8810044	1
17	18	16	0.5367667	0.0088206	0.7549276	0
17	19	16	0.5676004	0.0000004	0.4985060	0
17	20	16	0.3746492	0.0315854	0.8306306	1
17	21	16	0.4210161	0.0008515	0.6828896	0
18	1	16	0.6002140	0.0027664	0.7092093	0
18	2	16	0.7118687	0.0007643	0.6608950	0
18	3	16	0.5006201	0.0000083	0.5649087	0
18	4	16	0.5459757	0.0233037	0.8073543	0
18	5	16	0.6018378	0.0085787	0.7610942	0
18	6	16	0.4822143	0.0000024	0.5230795	0
18	7	16	0.5689227	0.0010443	0.6746958	0
18	8	16	0.5908964	0.0002167	0.6187591	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
18	9	16	0.6435480	0.0022723	0.6988242	0
18	10	16	0.6412143	0.0001866	0.6091905	0
18	11	16	0.5579865	0.0001833	0.6177679	0
18	12	16	0.6207247	0.0005680	0.6563426	0
18	13	16	0.6070641	0.0037698	0.7316612	0
18	14	16	0.5805032	0.0030627	0.7058208	0
18	15	16	0.4492822	0.0315536	0.8144107	0
18	16	16	0.5892214	0.0128277	0.7704514	0
18	17	16	0.5367667	0.0088206	0.7549276	0
18	19	16	0.3075506	0.0000480	0.5961390	0
18	20	16	0.6922194	0.0018573	0.6902983	0
18	21	16	0.3754440	0.0372762	0.8288024	1
19	1	16	0.6212220	0.0000000	0.4348616	0
19	2	16	0.7432353	0.0000000	0.3975812	0
19	3	16	0.5493722	0.0000391	0.6425998	0
19	4	16	0.5995259	0.0000007	0.4860504	0
19	5	16	0.6167108	0.0000000	0.4660355	0
19	6	16	0.5108687	0.0000661	0.6144997	0
19	7	16	0.5834282	0.0000000	0.4161854	0
19	8	16	0.6981040	0.0000000	0.3618007	0
19	9	16	0.7213284	0.0000001	0.4213995	0
19	10	16	0.7550361	0.0000000	0.3642806	0
19	11	16	0.6138113	0.0000000	0.3755258	0
19	12	16	0.6318503	0.0000000	0.3897274	0
19	13	16	0.6221249	0.0000000	0.4378312	0
19	14	16	0.6013830	0.0000005	0.4559773	0
19	15	16	0.5264469	0.0000113	0.5343581	0
19	16	16	0.5834617	0.0000027	0.5162810	0
19	17	16	0.5676004	0.0000004	0.4985060	0
19	18	16	0.3075506	0.0000480	0.5961390	0
19	20	16	0.6976587	0.0000003	0.4936878	0
19	21	16	0.4752364	0.0001344	0.6069113	0
20	1	16	0.5023976	0.0016429	0.7121305	0
20	2	16	0.5122602	0.0003117	0.6540368	0
20	3	16	0.7463929	0.0000008	0.4503804	0
20	4	16	0.5807913	0.0007885	0.6762674	0
20	5	16	0.2985383	0.0088844	0.7704347	0
20	6	16	0.7466479	0.0000001	0.4138759	0
20	7	16	0.3793237	0.0012455	0.6869199	0
20	8	16	0.5505010	0.0002061	0.6220498	0
20	9	16	0.5862683	0.0008319	0.6791752	0
20	10	16	0.6624778	0.0001671	0.6105340	0
20	11	16	0.4853955	0.0002869	0.6390845	0
20	12	16	0.3099720	0.0005413	0.6486717	0
20	13	16	0.2613589	0.0032848	0.7235473	0
20	14	16	0.2877834	0.0102519	0.7873463	1
20	15	16	0.5921017	0.0030458	0.7149050	0
20	16	16	0.2062121	0.0511505	0.8447287	1
20	17	16	0.3746492	0.0315854	0.8306306	1
20	18	16	0.6922194	0.0018573	0.6902983	0

p1	p2	#features	MeanK	OM2	MeanOM2	0/1
20	19	16	0.6976587	0.0000003	0.4936878	0
20	21	16	0.6453449	0.0001203	0.5989998	0
21	1	16	0.4807657	0.0007170	0.6857637	0
21	2	16	0.5536444	0.0002512	0.6364184	0
21	3	16	0.4560956	0.0000242	0.6059211	0
21	4	16	0.4092446	0.0057986	0.7610421	0
21	5	16	0.5685319	0.0008588	0.6988238	0
21	6	16	0.4305641	0.0000142	0.5805596	0
21	7	16	0.5392759	0.0001051	0.6268722	0
21	8	16	0.5393642	0.0000409	0.5855221	0
21	9	16	0.6041799	0.0005722	0.6631930	0
21	10	16	0.5767249	0.0000218	0.5711791	0
21	11	16	0.5196518	0.0000347	0.6083255	0
21	12	16	0.5879698	0.0000333	0.6090850	0
21	13	16	0.5893321	0.0002334	0.6657616	0
21	14	16	0.4894162	0.0004072	0.6376048	0
21	15	16	0.1801447	0.0158162	0.7853906	1
21	16	16	0.5354194	0.0019732	0.7071628	0
21	17	16	0.4210161	0.0008515	0.6828896	0
21	18	16	0.3754440	0.0372762	0.8288024	1
21	19	16	0.4752364	0.0001344	0.6069113	0
21	20	16	0.6453449	0.0001203	0.5989998	0

Appendix VII Full Skewness table

Skewness overall results for the overall run with populations database.

Table 21: Pop-Base SkewTable

Sample	Population	SK1	SK2	SK3	SK4
1.000	0.000	0.511	0.320	3.037	-1.594
1.000	1.000	0.000	0.000	0.427	0.058
1.000	2.000	0.000	0.000	0.000	0.000
1.000	3.000	0.000	0.000	1.436	-1.113
1.000	4.000	0.000	0.000	0.000	0.000
1.000	5.000	0.000	0.000	0.000	0.000
1.000	6.000	0.000	0.000	0.000	0.000
1.000	7.000	0.000	0.000	0.000	0.000
1.000	8.000	0.000	0.000	0.000	0.000
460.000	0.000	0.387	0.511	0.787	-0.481
460.000	1.000	0.000	0.000	0.427	0.058
460.000	2.000	0.000	0.000	0.000	0.000
460.000	3.000	0.000	0.000	1.436	-1.113
460.000	4.000	0.000	0.000	0.000	0.000
460.000	5.000	0.000	0.000	0.000	0.000
460.000	6.000	0.000	0.000	0.000	0.000
460.000	7.000	0.000	0.000	0.000	0.000
460.000	8.000	0.000	0.000	0.000	0.000
460.000	9.000	0.706	0.308	2.112	0.000
649.000	0.000	1.347	0.383	-0.070	0.406
649.000	1.000	0.000	0.000	0.000	0.000
649.000	2.000	0.000	0.000	0.000	0.000
649.000	3.000	0.000	0.000	0.000	0.000
649.000	4.000	1.318	0.000	0.032	0.483
649.000	5.000	0.000	0.000	0.000	0.000
649.000	6.000	0.846	0.000	0.265	0.000
649.000	7.000	0.000	0.000	0.000	0.000
649.000	8.000	0.000	0.000	0.000	0.000
649.000	9.000	0.000	0.000	0.000	0.000
649.000	10.000	0.000	0.000	0.000	0.000
1086.000	0.000	-0.247	-0.475	-0.189	1.031
1086.000	1.000	0.000	0.000	0.000	0.000
1086.000	2.000	-0.012	-2.000	0.068	0.000
1086.000	3.000	0.000	0.000	0.000	0.000
1086.000	4.000	0.000	0.000	0.000	0.000
1086.000	5.000	0.000	0.000	0.000	0.000
1086.000	6.000	0.000	0.000	0.000	0.000
1086.000	7.000	0.000	0.000	0.000	0.000
1086.000	8.000	0.000	0.000	0.000	0.000
1086.000	9.000	0.000	0.000	0.000	0.000
1086.000	10.000	0.000	0.000	0.000	0.000
1086.000	11.000	0.000	0.000	0.000	0.000
1336.000	0.000	0.501	-1.553	1.437	0.171
1336.000	1.000	0.000	0.000	0.000	0.000
1336.000	2.000	0.000	0.000	0.000	0.000

Sample	Population	SK1	SK2	SK3	SK4
1336.000	6.000	0.000	0.000	0.000	0.000
1336.000	7.000	0.000	0.000	0.000	0.000
1336.000	8.000	0.000	0.000	0.000	0.000
1336.000	9.000	0.000	0.000	0.000	0.000
1336.000	10.000	0.000	0.000	0.000	0.000
1336.000	11.000	0.993	0.000	0.183	0.239
1336.000	12.000	0.000	0.000	0.000	0.000
1643.000	0.000	-0.324	0.886	1.696	-0.770
1643.000	1.000	0.000	0.000	0.000	0.000
1643.000	2.000	0.000	0.000	0.000	0.000
1643.000	3.000	0.000	0.000	1.436	-1.113
1643.000	4.000	0.000	0.000	0.000	0.000
1643.000	5.000	0.000	0.000	0.000	0.000
1643.000	6.000	0.000	0.000	0.000	0.000
1643.000	7.000	0.000	0.000	0.000	0.000
1643.000	8.000	0.000	0.000	0.000	0.000
1643.000	9.000	0.000	0.000	0.000	0.000
1643.000	10.000	0.000	0.000	1.981	-1.288
1643.000	11.000	0.000	0.000	0.000	0.000
1643.000	12.000	0.000	0.000	0.000	0.000
1643.000	13.000	0.000	0.000	0.000	0.000
1985.000	0.000	0.083	0.918	0.893	-0.436
1985.000	1.000	0.000	0.000	0.427	0.058
1985.000	2.000	0.000	0.000	0.000	0.000
1985.000	3.000	0.000	0.000	1.436	-1.113
1985.000	4.000	0.000	0.000	0.000	0.000
1985.000	5.000	0.000	0.000	0.000	0.000
1985.000	6.000	0.000	0.000	0.000	0.000
1985.000	7.000	0.000	0.000	0.000	0.000
1985.000	8.000	0.000	0.000	0.000	0.000
1985.000	9.000	0.000	0.000	0.000	0.000
1985.000	10.000	0.000	0.000	1.981	-1.288
1985.000	11.000	0.000	0.000	0.000	0.000
1985.000	12.000	0.000	0.000	0.000	0.000
1985.000	13.000	0.425	0.000	0.863	0.180
1985.000	14.000	0.000	0.000	1.391	-0.807
2181.000	0.000	1.501	0.290	0.023	0.839
2181.000	1.000	0.000	0.000	0.000	0.000
2181.000	2.000	0.000	0.000	0.000	0.000
2181.000	3.000	0.000	0.000	0.000	0.000
2181.000	4.000	1.318	0.000	0.032	0.483
2181.000	5.000	0.000	0.000	0.000	0.000
2181.000	6.000	0.846	0.000	0.265	0.000
2181.000	7.000	0.000	0.000	0.000	0.000
2181.000	8.000	0.000	0.000	0.000	0.000
2181.000	9.000	0.000	0.000	0.000	0.000
2181.000	10.000	0.000	0.000	0.000	0.000
2181.000	11.000	0.993	0.000	0.183	0.239
2181.000	12.000	0.000	0.000	0.000	0.000
2181.000	13.000	0.000	0.000	0.000	0.000

Sample	Population	SK1	SK2	SK3	SK4
2181.000	14.000	0.000	0.000	0.000	0.000
2181.000	15.000	0.000	0.000	0.000	0.000
2489.000	0.000	-0.440	1.421	0.699	-0.100
2489.000	1.000	0.000	0.000	0.000	0.000
2489.000	2.000	0.000	0.000	0.000	0.000
2489.000	3.000	0.000	0.000	0.000	0.000
2489.000	4.000	0.000	0.000	0.000	0.000
2489.000	5.000	0.000	0.000	0.000	0.000
2489.000	6.000	0.000	0.000	0.000	0.000
2489.000	7.000	0.000	0.000	0.000	0.000
2489.000	8.000	0.358	0.000	0.000	0.000
2489.000	9.000	0.000	0.000	0.000	0.000
2489.000	10.000	0.000	0.000	1.981	-1.288
2489.000	11.000	0.000	0.000	0.000	0.000
2489.000	12.000	0.000	0.000	0.000	0.000
2489.000	13.000	0.000	0.000	0.000	0.000
2489.000	14.000	0.000	0.000	0.000	0.000
2489.000	15.000	0.000	0.000	0.000	0.000
2489.000	16.000	0.000	0.000	0.000	0.000
3403.000	0.000	-0.150	1.528	0.859	-0.602
3403.000	1.000	0.000	0.000	0.427	0.058
3403.000	2.000	0.000	0.000	0.000	0.000
3403.000	3.000	0.000	0.000	0.000	0.000
3403.000	4.000	0.000	0.000	0.000	0.000
3403.000	5.000	0.547	0.000	1.090	0.000
3403.000	6.000	0.000	0.000	0.000	0.000
3403.000	7.000	0.000	0.000	0.000	0.000
3403.000	8.000	0.000	0.000	0.000	0.000
3403.000	9.000	0.000	0.000	0.000	0.000
3403.000	10.000	0.000	0.000	1.981	-1.288
3403.000	11.000	0.000	0.000	0.000	0.000
3403.000	12.000	0.000	0.000	0.000	0.000
3403.000	13.000	0.425	0.000	0.863	0.180
3403.000	14.000	0.000	0.000	0.000	0.000
3403.000	15.000	0.000	0.000	0.000	0.000
3403.000	16.000	0.000	0.000	0.000	0.000
3403.000	17.000	0.000	0.000	0.000	0.000
4527.000	0.000	-0.201	-5.821	0.570	0.345
4527.000	1.000	0.000	0.000	0.000	0.000
4527.000	2.000	0.000	0.000	0.000	0.000
4527.000	3.000	0.000	0.000	0.000	0.000
4527.000	4.000	0.000	0.000	0.000	0.000
4527.000	5.000	0.000	0.000	0.000	0.000
4527.000	6.000	0.000	0.000	0.000	0.000
4527.000	7.000	0.000	0.000	0.000	0.000
4527.000	8.000	0.000	0.000	0.000	0.000
4527.000	9.000	0.000	0.000	0.000	0.000
4527.000	10.000	0.000	0.000	0.000	0.000
4527.000	11.000	0.000	0.000	0.000	0.000
4527.000	12.000	0.000	0.000	0.000	0.000

Sample	Population	SK1	SK2	SK3	SK4
4527.000	13.000	0.000	0.000	0.000	0.000
4527.000	14.000	0.000	0.000	0.000	0.000
4527.000	15.000	0.000	0.000	0.000	0.000
4887.000	0.000	1.431	3.273	0.402	0.542
4887.000	1.000	0.000	0.000	0.000	0.000
4887.000	2.000	0.000	0.000	0.000	0.000
4887.000	3.000	0.000	0.000	0.000	0.000
4887.000	4.000	0.000	0.000	0.000	0.000
4887.000	5.000	0.846	0.000	0.265	0.000
4887.000	6.000	0.000	0.000	0.000	0.000
4887.000	7.000	0.000	0.000	0.000	0.000
4887.000	8.000	0.000	0.000	0.000	0.000
4887.000	9.000	0.000	0.000	0.000	0.000
4887.000	10.000	0.000	0.000	0.000	0.000
4887.000	11.000	0.000	0.000	0.000	0.000
4887.000	12.000	0.000	0.000	0.000	0.000
4887.000	13.000	0.000	0.000	0.000	0.000
4887.000	14.000	0.000	0.000	0.000	0.000
4887.000	15.000	0.000	0.000	0.000	0.000
4887.000	16.000	0.000	0.000	0.000	0.000
4887.000	17.000	0.000	0.000	0.000	0.000
4887.000	18.000	0.000	0.000	0.000	0.000

Appendix VIII The overlap table for the full online algorithm

Table 22: Full overlap run

Real Index	Classifier Index	Base	Size base	Size new	OM2	Mean OM2	Mean KS	Overlap points	Case	Overlap Pop	Class
1	620	1	618	82	0.0377	0.84612	0.10148	483	2	1	0.9746
461	737	1	116	85	0.0067	0.7725	0.1847	67	4	2	0.969
1153	1156	4	-	86	0	0.50	0.416	1	1	Human	0.8
1334	1340	2	116	82	0.0008	**0.7096**	**0.2307**	40	0		**0.738**
1449	1568	2	116	94	0.0002	0.6632	0.3931	62	0	**0.8065**	**0.8065**
1637	1643	3	180	98	0.05747	0.79884	0.22608	114	2	8	0.82
1926	1965	2	116	97	0.0000	0.4152	0.4455	3	1	**0.7056**	**0.7056**
2205	2212	2	116	90	0.0001	0.6248	0.3243	33	0	**0.66**	**0.66**
2510	2518	2	116	84	0.0044	**0.7577**	**0.2286**	**57**	**4**	**2510**	**0.6**
2690	2753	2	116	89	0.0000	0.6397	0.4148	3	1	**0.77**	**0.77**
3024	3049	2	116	93	0.0000	0.4844	0.3600	0	1	**0.6471**	**0.6471**
3206	3218	2	116	84	0.0005	0.7025	0.3332	31	0	**0.5**	**0.5**
3351	3362	2	116	80	0.0000	0.5280	0.3549	2	1	**0.72**	**0.72**
3678	3691	2	116	91	0.0000	0.5502	0.3421	16	1	**0.65**	**0.65**
4183	4214	2	116	88	0.0000	0.5568	0.3266	10	1	**0.89**	**0.89**
5098	5114	2	116	88	0.0000	0.5230	0.3803	5	1	**0.8**	**0.8**
6075	6083	2	116	99	0.0000	0.0000	0.0000	0	0	**0.89**	**0.89**
7194	7235	2	116	87	0.0000	0.5120	0.4575	3	1	**0.81**	**0.81**
7898	7913	2	116	89	0.0000	0.2959	0.4430	0	1	**0.86**	**0.86**
8261	8280	2	116	82	0.0000	0.0000	0.0000	0	0	**0.92**	**0.92**
9612	9612	2	116	90	0.0000	0.5571	0.3831	4	1	**0.87**	**0.87**

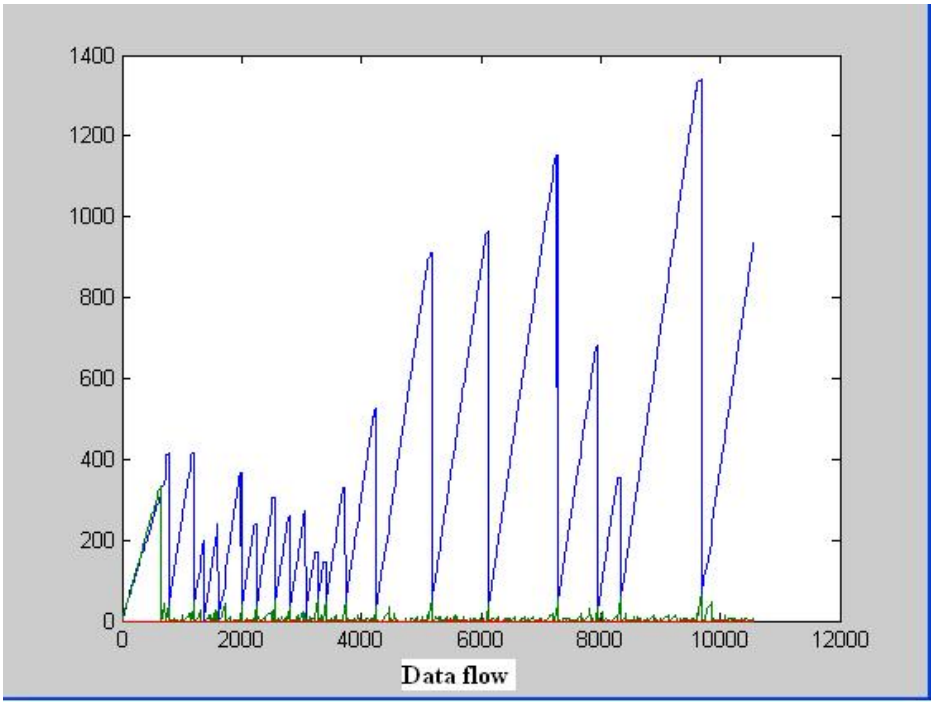

Figure 60 : Full run change detection

Appendix IX Graphical simulations

Additional experiment is dealing with the performance and cost and is detailed in the graph.

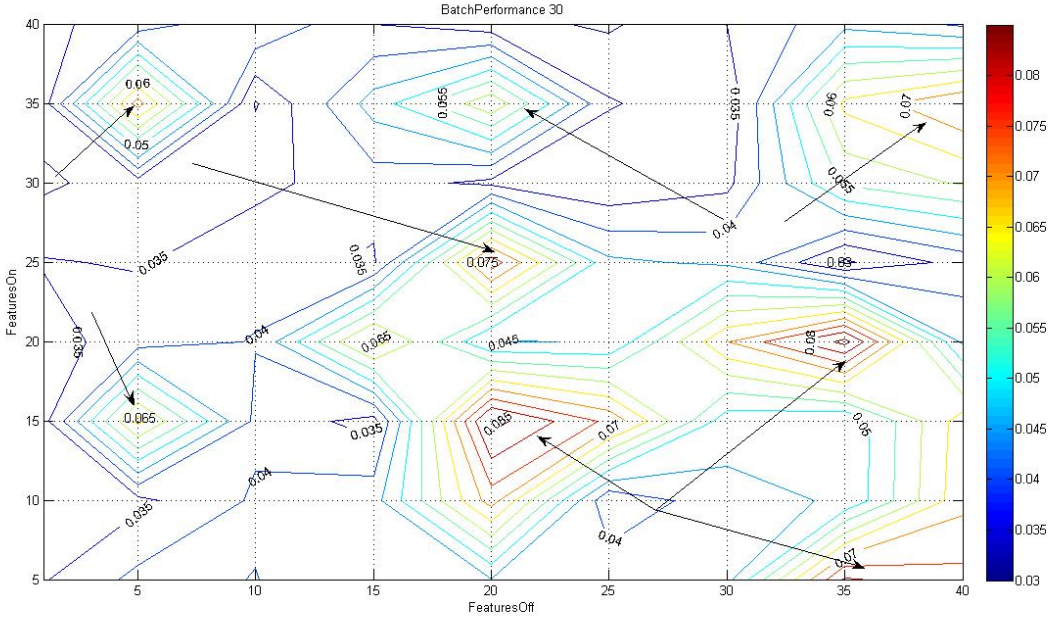

Figure 61: Classification performance measures variations where *batch size* is fixed on a batch of 30 and features on/detection batch size batches are changing

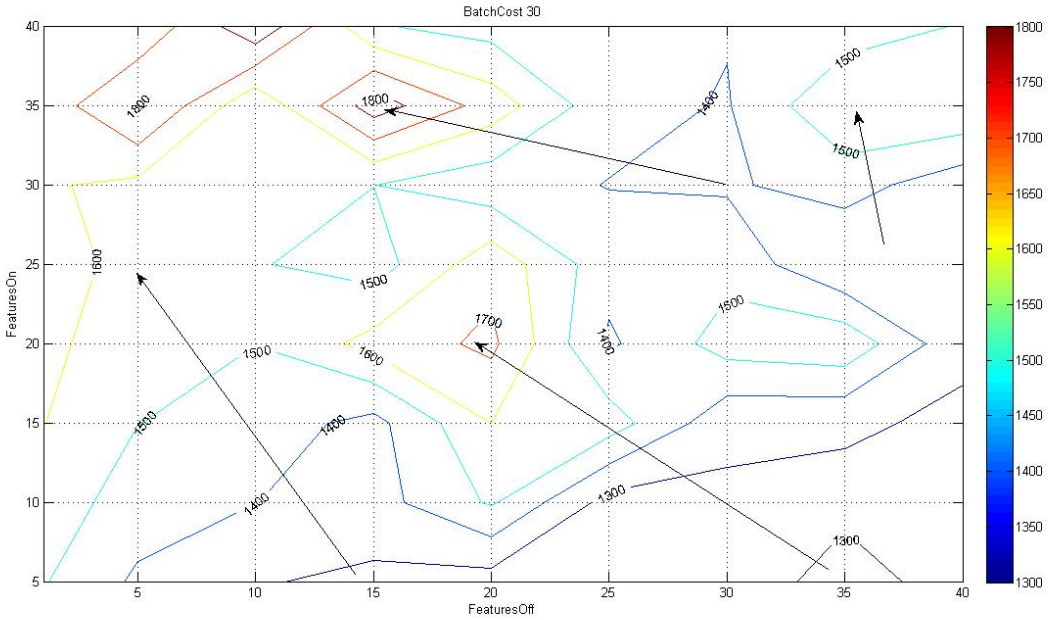

Figure 62: Cost levels measures variations where *batch size* is fixed on a batch of 30 and features on/detection batch size batches are changing

Appendix X Population description

Populations composition is provide in the following tables.

Table 23: All Populations

	Class1	Class2	Class3	Class4		
1	22	0	287	383		692
2	0	0	268	192	2 class	460
3	0	17	36	128		181
4	0	0	71	94	2 class	165
5	151	0	149	127		427
6	0	33	89	157		279
7	85	14	143	63		305
8	24	0	69	87		180
9	10	22	149	21		202
10	0	0	53	79	2 class	132
11	31	0	102	49		182
12	54	0	63	28		145
13	145	0	114	68		327
14	87	24	330	64		505
15	0	145	372	398		915
16	155	33	621	169		978
17	58	247	716	97		1118
18	0	162	458	84		704
19	0	136	207	20		363
20	170	19	978	144		1311
21	0	60	446	468		974

Table 24 provides another perspective on the similarity between the population, KS measure is the presented values.

Table 24: Populations confusion similarity matrix

	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000
1	0.0000	**0.2600**	0.4843	0.2544	0.4289	0.4957	0.2873
2	0.2600	0.0000	0.6447	0.3241	0.4901	0.6497	0.4412
3	0.4843	0.6447	0.0000	0.3795	0.5930	**0.1700**	0.5499
4	0.2544	0.3241	0.3795	0.0000	0.4684	0.4037	0.3984
5	0.4289	0.4901	0.5930	0.4684	0.0000	0.6004	0.2843
6	0.4957	0.6497	**0.1700**	0.4037	0.6004	0.0000	0.5454
7	0.2873	0.4412	0.5499	0.3984	0.2843	0.5454	0.0000
8	0.2564	0.2716	0.5349	**0.2483**	0.4419	0.5633	0.3795
9	0.3412	0.3629	0.5976	0.3008	0.5005	0.6277	0.4242
10	0.3272	0.3472	0.5600	0.3065	0.5654	0.6052	0.4759
11	**0.2141**	0.3097	0.5073	0.2970	0.3944	0.5116	0.2856
12	0.4125	0.4427	0.6339	0.4934	0.2503	0.6364	**0.2223**
13	0.4268	0.4861	0.6374	0.4982	**0.2361**	0.6389	0.2405
14	0.3073	0.3655	0.5694	0.3726	0.2812	0.5703	0.2700
15	0.3935	0.4503	0.4444	0.2870	0.5042	0.4357	0.4803
16	0.4569	0.5044	0.6232	0.5043	**0.2260**	0.6204	0.3645
17	0.2430	0.3506	0.5037	0.2830	0.3801	0.5128	0.2499
18	0.6002	0.7119	0.5006	0.5460	0.6018	0.4822	0.5689
19	0.6212	0.7432	0.5494	0.5995	0.6167	0.5109	0.5834
20	0.5024	0.5123	0.7464	0.5808	0.2985	0.7466	0.3793
21	0.4808	0.5536	0.4561	0.4092	0.5685	0.4306	0.5393

	8.0000	9.0000	10.0000	11.0000	12.0000	13.0000	14.0000
1	0.2564	0.3412	0.3272	0.2141	0.4125	0.4268	0.3073
2	0.2716	0.3629	0.3472	0.3097	0.4427	0.4861	0.3655
3	0.5349	0.5976	0.5600	0.5073	0.6339	0.6374	0.5694
4	0.2483	0.3008	0.3065	0.2970	0.4934	0.4982	0.3726
5	0.4419	0.5005	0.5654	0.3944	0.2503	0.2361	0.2812
6	0.5633	0.6277	0.6052	0.5116	0.6364	0.6389	0.5703
7	0.3795	0.4242	0.4759	0.2856	**0.2223**	0.2405	0.2700
8	0.0000	**0.2595**	**0.1622**	**0.2012**	0.4194	0.4459	0.3673
9	0.2595	0.0000	0.3204	0.3485	0.4684	0.4855	0.4302
10	**0.1622**	0.3204	0.0000	0.2728	0.5554	0.5797	0.4745
11	0.2012	0.3485	0.2728	0.0000	0.3561	0.3900	0.3389
12	0.4194	0.4684	0.5554	0.3561	0.0000	**0.1377**	**0.2533**
13	0.4459	0.4855	0.5797	0.3900	0.1377	0.0000	0.2458
14	0.3673	0.4302	0.4745	0.3389	0.2533	0.2458	0.0000
15	0.4387	0.4987	0.5010	0.4339	0.5298	0.5256	0.4029
16	0.5223	0.5651	0.6300	0.4412	0.3173	0.2782	**0.2581**
17	0.3403	0.3971	0.4305	0.3034	0.3241	0.3363	0.1693
18	0.5909	0.6435	0.6412	0.5580	0.6207	0.6071	0.5805
19	0.6981	0.7213	0.7550	0.6138	0.6319	0.6221	0.6014
20	0.5505	0.5863	0.6625	0.4854	0.3100	0.2614	0.2878
21	0.5394	0.6042	0.5767	0.5197	0.5880	0.5893	0.4894

	15.0000	16.0000	17.0000	18.0000	19.0000	20.0000	21.0000
1	0.3935	0.4569	0.2430	0.6002	0.6212	0.5024	0.4808
2	0.4503	0.5044	0.3506	0.7119	0.7432	0.5123	0.5536
3	0.4444	0.6232	0.5037	0.5006	0.5494	0.7464	0.4561
4	0.2870	0.5043	0.2830	0.5460	0.5995	0.5808	0.4092
5	0.5042	**0.2260**	0.3801	0.6018	0.6167	0.2985	0.5685
6	0.4357	0.6204	0.5128	0.4822	0.5109	0.7466	0.4306
7	0.4803	0.3645	0.2499	0.5689	0.5834	0.3793	0.5393
8	0.4387	0.5223	0.3403	0.5909	0.6981	0.5505	0.5394
9	0.4987	0.5651	0.3971	0.6435	0.7213	0.5863	0.6042
10	0.5010	0.6300	0.4305	0.6412	0.7550	0.6625	0.5767
11	0.4339	0.4412	0.3034	0.5580	0.6138	0.4854	0.5197
12	0.5298	0.3173	0.3241	0.6207	0.6319	0.3100	0.5880
13	0.5256	0.2782	0.3363	0.6071	0.6221	0.2614	0.5893
14	0.4029	0.2581	**0.1693**	0.5805	0.6014	0.2878	0.4894
15	0.0000	0.4835	0.3334	0.4493	0.5264	0.5921	**0.1801**
16	0.4835	0.0000	0.3463	0.5892	0.5835	**0.2062**	0.5354
17	0.3334	0.3463	0.0000	0.5368	0.5676	0.3746	0.4210
18	0.4493	0.5892	0.5368	0.0000	**0.3076**	0.6922	0.3754
19	0.5264	0.5835	0.5676	0.3076	0.0000	0.6977	0.4752
20	0.5921	0.2062	0.3746	0.6922	0.6977	0.0000	0.6453
21	**0.1801**	0.5354	0.4210	**0.3754**	0.4752	0.6453	0.0000

Appendix XI Simulations results

Table 25 provides the 504 simulations results. The first three columns contains the three simulation parameters values that was mentioned in section 3.10. Each simulation results in three aspects: classification **accuracy**, classification performance (**MSPE**) and the **cost**.

Table 25: The simulation results

BatchSize	On	Off	Accuracy	PM	Cost
20	5	1	0.83168	0.03227	1569.75
20	5	5	0.82818	0.03404	1436.51
20	5	10	0.82737	0.05172	1316.83
20	5	15	0.82657	0.03669	1318.31
20	5	20	0.78570	0.06464	1536.21
20	5	25	0.80882	0.06974	1345.84
20	5	30	0.81420	0.03421	1427.78
20	5	35	0.78489	0.06480	1502.47
20	5	40	0.83033	0.06920	1113.24
20	10	1	0.86905	0.01909	1436.48
20	10	5	0.86717	0.01988	1322.61
20	10	10	0.84888	0.02963	1323.53
20	10	15	0.81931	0.05318	1452.69
20	10	20	0.85776	0.02889	1186.77
20	10	25	0.81823	0.04387	1376.43
20	10	30	0.85265	0.03286	1160.28
20	10	35	0.82576	0.03952	1310.33
20	10	40	0.82603	0.03596	1294.34
20	15	1	0.85050	0.03118	1530.90
20	15	5	0.82737	0.03479	1579.82
20	15	10	0.82388	0.05580	1521.01
20	15	15	0.83087	0.03513	1444.45
20	15	20	0.86260	0.02223	1216.79
20	15	25	0.85830	0.02655	1206.88
20	15	30	0.82011	0.04281	1388.91
20	15	35	0.82468	0.03544	1348.39
20	15	40	0.85292	0.03300	1237.34
20	20	1	0.81877	0.03045	1830.98
20	20	5	0.83194	0.03222	1597.35
20	20	10	0.81689	0.04614	1588.16
20	20	15	0.85964	0.02636	1313.15
20	20	20	0.81070	0.07584	1525.83
20	20	25	0.82684	0.03650	1431.09
20	20	30	0.83302	0.06202	1399.85
20	20	35	0.83436	0.06291	1379.39
20	20	40	0.82979	0.03297	1359.31
20	25	1	0.86690	0.02038	1491.06
20	25	5	0.86582	0.02072	1426.10
20	25	10	0.86609	0.02022	1361.63
20	25	15	0.86287	0.02387	1335.58
20	25	20	0.82818	0.03404	1486.82
20	25	25	0.81689	0.06567	1503.20

BatchSize	On	Off	Accuracy	PM	Cost
20	25	30	0.82011	0.04494	1524.15
20	25	35	0.82092	0.05749	1449.34
20	25	40	0.80586	0.07609	1482.74
20	30	1	0.86690	0.02039	1498.88
20	30	5	0.86824	0.01917	1440.06
20	30	10	0.86663	0.02043	1387.43
20	30	15	0.83141	0.03318	1533.19
20	30	20	0.86152	0.02259	1325.49
20	30	25	0.87066	0.04514	1268.71
20	30	30	0.83114	0.03174	1439.69
20	30	35	0.86206	0.02194	1242.61
20	30	40	0.82092	0.05631	1447.78
20	35	1	0.86798	0.01883	1498.30
20	35	5	0.86717	0.01988	1451.67
20	35	10	0.83221	0.03182	1598.26
20	35	15	0.86152	0.02305	1384.21
20	35	20	0.85453	0.02717	1375.22
20	35	25	0.85265	0.02821	1345.98
20	35	30	0.81850	0.04412	1516.35
20	35	35	0.80909	0.03489	1641.27
20	35	40	0.82388	0.03825	1448.87
20	40	1	0.86798	0.01883	1501.87
20	40	5	0.83168	0.03107	1655.17
20	40	10	0.83194	0.03186	1610.06
20	40	15	0.83490	0.03039	1561.03
20	40	20	0.86340	0.02297	1357.62
20	40	25	0.83141	0.03318	1510.91
20	40	30	0.82253	0.03820	1522.20
20	40	35	0.81258	0.06737	1533.30
20	40	40	0.82361	0.06428	1482.18
30	5	1	0.8196	0.0346	1506.80
30	5	5	0.8123	0.0414	1381.48
30	5	10	0.8129	0.0399	1307.16
30	5	15	0.8069	0.0438	1278.51
30	5	20	0.8080	0.0448	1259.30
30	5	25	0.8094	0.0435	1231.71
30	5	30	0.8094	0.0435	1220.55
30	5	35	0.7806	0.0808	1354.70
30	5	40	0.8163	0.0766	1243.72
30	10	1	0.8258	0.0304	1553.30
30	10	5	0.8198	0.0337	1453.87
30	10	10	0.8120	0.0407	1401.51
30	10	15	0.8091	0.0431	1360.85
30	10	20	0.7771	0.0723	1512.48
30	10	25	0.8166	0.0357	1261.13
30	10	30	0.8094	0.0435	1278.29
30	10	35	0.8067	0.0455	1267.97
30	10	40	0.8207	0.0685	1231.65
30	15	1	0.8198	0.0330	1602.86

BatchSize	On	Off	Accuracy	PM	Cost
30	15	5	0.8241	0.0666	1501.74
30	15	10	0.8142	0.0387	1450.67
30	15	15	0.8188	0.0329	1368.34
30	15	20	0.7663	0.0870	1600.17
30	15	25	0.7741	0.0737	1550.17
30	15	30	0.8064	0.0470	1327.54
30	15	35	0.8104	0.0454	1315.21
30	15	40	0.8174	0.0678	1280.95
30	20	1	0.8188	0.0327	1619.90
30	20	5	0.8142	0.0378	1552.77
30	20	10	0.8107	0.0402	1501.75
30	20	15	0.7806	0.0679	1634.92
30	20	20	0.7747	0.0445	1723.74
30	20	25	0.8080	0.0454	1383.15
30	20	30	0.7787	0.0698	1542.74
30	20	35	0.7674	0.0866	1573.30
30	20	40	0.8104	0.0439	1321.23
30	25	1	0.8177	0.0354	1638.99
30	25	5	0.8177	0.0346	1571.06
30	25	10	0.8145	0.0382	1507.71
30	25	15	0.8185	0.0342	1452.62
30	25	20	0.7725	0.0780	1666.35
30	25	25	0.8048	0.0461	1439.54
30	25	30	0.7991	0.0438	1468.33
30	25	35	0.8266	0.0292	1300.01
30	25	40	0.8161	0.0370	1323.35
30	30	1	0.8255	0.0290	1609.88
30	30	5	0.8188	0.0327	1577.02
30	30	10	0.8198	0.0337	1516.54
30	30	15	0.8153	0.0376	1501.31
30	30	20	0.8188	0.0335	1437.08
30	30	25	0.8225	0.0305	1396.87
30	30	30	0.8193	0.0337	1387.23
30	30	35	0.8241	0.0562	1443.03
30	30	40	0.8266	0.0606	1333.28
30	35	1	0.8204	0.0326	1633.98
30	35	5	0.7749	0.0723	1821.92
30	35	10	0.8252	0.0295	1514.65
30	35	15	0.7698	0.0472	1852.52
30	35	20	0.7841	0.0640	1655.37
30	35	25	0.8177	0.0355	1433.40
30	35	30	0.8241	0.0311	1393.01
30	35	35	0.7809	0.0663	1594.05
30	35	40	0.7768	0.0751	1594.13
30	40	1	0.8185	0.0342	1646.70
30	40	5	0.8150	0.0367	1610.91
30	40	10	0.7749	0.0448	1884.78
30	40	15	0.8190	0.0349	1507.88
30	40	20	0.8217	0.0315	1459.80

BatchSize	On	Off	Accuracy	PM	Cost
30	40	25	0.8266	0.0292	1421.00
30	40	30	0.8204	0.0351	1406.43
30	40	35	0.8083	0.0435	1446.37
30	40	40	0.8099	0.0390	1506.04
40	5	1	0.8492	0.0288	1343.16
40	5	5	0.8069	0.0736	1414.77
40	5	10	0.8086	0.0653	1351.10
40	5	15	0.8346	0.0441	1126.61
40	5	20	0.7986	0.0876	1289.79
40	5	25	0.8120	0.1020	1309.63
40	5	30	0.7717	0.0777	1433.12
40	5	35	0.7776	0.0973	1505.35
40	5	40	0.8104	0.0974	1302.56
40	10	1	0.8389	0.0376	1450.99
40	10	5	0.8102	0.0598	1507.04
40	10	10	0.8486	0.0288	1209.28
40	10	15	0.8314	0.0461	1213.12
40	10	20	0.8411	0.0396	1150.10
40	10	25	0.7921	0.0991	1346.41
40	10	30	0.8387	0.0389	1119.40
40	10	35	0.8395	0.0415	1093.67
40	10	40	0.7978	0.0737	1386.74
40	15	1	0.8397	0.0374	1474.93
40	15	5	0.8422	0.0369	1366.35
40	15	10	0.8524	0.0264	1257.31
40	15	15	0.8424	0.0335	1237.35
40	15	20	0.8438	0.0345	1186.08
40	15	25	0.8059	0.0624	1390.99
40	15	30	0.8024	0.0759	1354.03
40	15	35	0.8072	0.0731	1323.40
40	15	40	0.8086	0.0699	1299.90
40	20	1	0.7983	0.0770	1712.76
40	20	5	0.8096	0.0564	1612.37
40	20	10	0.8411	0.0343	1341.29
40	20	15	0.8435	0.0351	1271.98
40	20	20	0.8096	0.0564	1450.39
40	20	25	0.8059	0.0598	1442.52
40	20	30	0.8548	0.0587	1161.13
40	20	35	0.8427	0.0337	1160.06
40	20	40	0.8276	0.0509	1312.43
40	25	1	0.8505	0.0269	1458.18
40	25	5	0.7981	0.0708	1679.39
40	25	10	0.8486	0.0297	1333.54
40	25	15	0.8042	0.0729	1515.49
40	25	20	0.8346	0.0370	1393.73
40	25	25	0.8349	0.0437	1255.32
40	25	30	0.8371	0.0393	1234.05
40	25	35	0.8115	0.0519	1400.00
40	25	40	0.8061	0.0654	1372.53

BatchSize	On	Off	Accuracy	PM	Cost
40	30	1	0.8384	0.0402	1510.73
40	30	5	0.8510	0.0273	1409.07
40	30	10	0.8005	0.0399	1704.06
40	30	15	0.7967	0.0842	1561.94
40	30	20	0.8397	0.0374	1308.43
40	30	25	0.8086	0.0557	1484.60
40	30	30	0.8481	0.0297	1219.05
40	30	35	0.8397	0.0368	1216.14
40	30	40	0.8457	0.0301	1202.32
40	35	1	0.8287	0.0531	1552.05
40	35	5	0.8465	0.0335	1428.37
40	35	10	0.8454	0.0324	1383.34
40	35	15	0.8454	0.0311	1349.42
40	35	20	0.8311	0.0453	1366.58
40	35	25	0.8449	0.0322	1282.21
40	35	30	0.8091	0.0575	1481.25
40	35	35	0.8473	0.0321	1224.18
40	35	40	0.8239	0.0450	1444.08
40	40	1	0.8155	0.0480	1658.65
40	40	5	0.8021	0.0651	1700.45
40	40	10	0.8435	0.0351	1400.59
40	40	15	0.8416	0.0329	1371.54
40	40	20	0.8360	0.0413	1365.60
40	40	25	0.8438	0.0331	1294.85
40	40	30	0.8395	0.0404	1278.85
40	40	35	0.8473	0.0316	1247.18
40	40	40	0.8489	0.0293	1223.35
50	10	20	0.8443	0.0346	1115.18
50	30	10	0.8392	0.0372	1386.38
50	40	20	0.8389	0.0393	1333.77
50	35	20	0.8381	0.0402	1323.18
50	35	30	0.8373	0.0425	1273.16
50	5	15	0.8207	0.0426	1227.09
50	20	15	0.8142	0.0429	1433.62
50	25	1	0.8153	0.0436	1630.25
50	10	5	0.8147	0.0446	1479.76
50	25	5	0.8131	0.0456	1582.01
50	35	40	0.7951	0.0471	1561.25
50	25	25	0.8166	0.0478	1377.98
50	10	30	0.8153	0.0479	1247.55
50	20	35	0.8172	0.0494	1284.43
50	10	10	0.8118	0.0495	1405.83
50	15	35	0.8086	0.0496	1406.98
50	15	25	0.8174	0.0496	1281.21
50	40	35	0.8064	0.0511	1455.62
50	40	5	0.8198	0.0522	1557.85
50	5	5	0.8163	0.0526	1336.04
50	5	20	0.8107	0.0527	1246.30
50	30	5	0.8077	0.0528	1707.50

BatchSize	On	Off	Accuracy	PM	Cost
50	5	10	0.8161	0.0529	1257.48
50	40	15	0.8032	0.0537	1660.65
50	30	15	0.8080	0.0543	1610.71
50	35	5	0.8045	0.0554	1646.18
50	40	1	0.8061	0.0559	1785.34
50	15	10	0.8147	0.0562	1415.63
50	35	1	0.8094	0.0563	1669.03
50	10	1	0.8112	0.0563	1573.30
50	30	40	0.8059	0.0564	1383.72
50	15	1	0.8405	0.0570	1501.29
50	10	15	0.8129	0.0572	1303.46
50	15	30	0.8010	0.0578	1334.34
50	5	1	0.8040	0.0579	1667.04
50	35	25	0.8142	0.0597	1406.05
50	10	35	0.8018	0.0598	1251.23
50	10	40	0.7991	0.0599	1248.68
50	10	25	0.8026	0.0605	1342.73
50	40	30	0.8056	0.0609	1486.30
50	20	5	0.8123	0.0612	1537.19
50	15	5	0.7771	0.0626	1910.00
50	20	10	0.8075	0.0627	1483.88
50	30	30	0.8094	0.0641	1383.50
50	35	35	0.8002	0.0645	1434.94
50	25	15	0.7973	0.0645	1528.46
50	15	20	0.8048	0.0646	1464.72
50	30	35	0.7986	0.0650	1538.63
50	25	30	0.7983	0.0652	1438.69
50	5	25	0.8021	0.0664	1221.56
50	20	30	0.8069	0.0665	1362.14
50	25	10	0.8048	0.0666	1523.32
50	20	40	0.8051	0.0673	1309.92
50	40	25	0.7994	0.0686	1594.64
50	25	35	0.8069	0.0691	1350.62
50	35	10	0.7994	0.0691	1691.62
50	15	15	0.8061	0.0718	1414.52
50	25	40	0.7986	0.0732	1346.82
50	20	25	0.7965	0.0763	1385.49
50	20	20	0.8051	0.0769	1421.61
50	15	40	0.8069	0.0777	1305.08
50	30	1	0.8223	0.0778	1600.99
50	30	20	0.8002	0.0789	1517.37
50	20	1	0.8120	0.0830	1731.57
50	5	30	0.8067	0.0837	1144.75
50	25	20	0.7862	0.0866	1642.83
50	5	40	0.7943	0.0868	1237.11
50	40	10	0.7903	0.0873	1767.53
50	30	25	0.7895	0.0880	1508.56
50	40	40	0.7919	0.0881	1607.49
50	35	15	0.7892	0.0888	1720.68

BatchSize	On	Off	Accuracy	PM	Cost
50	5	35	0.7806	0.0959	1330.12
60	5	1	0.8034	0.0676	1642.27
60	5	5	0.8048	0.0617	1364.12
60	5	10	0.7736	0.1117	1575.01
60	5	15	0.7959	0.0751	1271.49
60	5	20	0.7895	0.0863	1262.38
60	5	25	0.8056	0.0830	1176.90
60	5	30	0.7755	0.1192	1332.98
60	5	35	0.7892	0.0960	1232.55
60	5	40	0.7698	0.1344	1335.01
60	10	1	0.8010	0.0651	1698.92
60	10	5	0.8013	0.0651	1559.23
60	10	10	0.8459	0.0312	1191.25
60	10	15	0.8037	0.0633	1329.66
60	10	20	0.8053	0.0604	1279.30
60	10	25	0.7873	0.0850	1321.96
60	10	30	0.8258	0.0527	1127.95
60	10	35	0.7938	0.0780	1245.20
60	10	40	0.7800	0.1060	1464.35
60	15	1	0.8016	0.0663	1723.30
60	15	5	0.8059	0.0596	1603.44
60	15	10	0.8314	0.0450	1401.25
60	15	15	0.7833	0.0995	1495.68
60	15	20	0.7835	0.1014	1455.18
60	15	25	0.7876	0.0841	1362.88
60	15	30	0.8333	0.0429	1136.03
60	15	35	0.7868	0.0973	1373.89
60	15	40	0.7965	0.0741	1284.12
60	20	1	0.8217	0.0514	1626.53
60	20	5	0.8013	0.0653	1664.48
60	20	10	0.7938	0.0770	1513.72
60	20	15	0.8131	0.0634	1533.26
60	20	20	0.7921	0.0753	1442.27
60	20	25	0.7967	0.0893	1416.14
60	20	30	0.7970	0.0669	1490.92
60	20	35	0.7956	0.0822	1367.87
60	20	40	0.7712	0.1109	1596.19
60	25	1	0.7884	0.0901	1821.93
60	25	5	0.8077	0.0578	1644.51
60	25	10	0.7857	0.0775	1734.00
60	25	15	0.8303	0.0483	1329.31
60	25	20	0.8056	0.0616	1507.14
60	25	25	0.8104	0.0662	1477.26
60	25	30	0.7881	0.0914	1445.90
60	25	35	0.7919	0.0840	1523.02
60	25	40	0.8395	0.0389	1172.27
60	30	1	0.7946	0.0689	1798.06
60	30	5	0.7862	0.0890	1806.87
60	30	10	0.8091	0.0548	1595.59

BatchSize	On	Off	Accuracy	PM	Cost
60	30	15	0.7825	0.0793	1733.95
60	30	20	0.7868	0.0862	1675.28
60	30	25	0.7954	0.0763	1443.55
60	30	30	0.8021	0.0611	1509.69
60	30	35	0.8319	0.0407	1221.39
60	30	40	0.7986	0.0765	1379.35
60	35	1	0.8309	0.0491	1603.15
60	35	5	0.8279	0.0445	1561.66
60	35	10	0.8064	0.0601	1615.21
60	35	15	0.7983	0.0575	1642.93
60	35	20	0.7876	0.0845	1515.24
60	35	25	0.8126	0.0563	1516.82
60	35	30	0.7916	0.0751	1477.40
60	35	35	0.7973	0.0730	1409.71
60	35	40	0.7782	0.0895	1629.19
60	40	1	0.8271	0.0451	1627.86
60	40	5	0.8207	0.0616	1629.14
60	40	10	0.8013	0.0616	1676.49
60	40	15	0.8008	0.0626	1527.12
60	40	20	0.7852	0.0938	1703.01
60	40	25	0.8150	0.0511	1409.82
60	40	30	0.7911	0.0737	1622.33
60	40	35	0.8274	0.0475	1369.50
60	40	40	0.8416	0.0383	1267.53
70	5	1	0.7809	0.0955	1642.78
70	5	5	0.8172	0.0919	1346.15
70	5	10	0.8295	0.0486	1164.19
70	5	15	0.8220	0.0515	1130.48
70	5	20	0.8344	0.0406	1061.91
70	5	25	0.7066	0.1536	1619.64
70	5	30	0.7752	0.1060	1217.11
70	5	35	0.8215	0.0506	1078.92
70	5	40	0.7763	0.1093	1274.40
70	10	1	0.7930	0.0826	1662.26
70	10	5	0.7835	0.1043	1577.24
70	10	10	0.7599	0.1379	1583.09
70	10	15	0.8147	0.1008	1266.14
70	10	20	0.8341	0.0411	1145.19
70	10	25	0.7795	0.1122	1392.00
70	10	30	0.8338	0.0875	1165.13
70	10	35	0.8061	0.0714	1196.20
70	10	40	0.7978	0.1213	1340.04
70	15	1	0.8236	0.0516	1504.09
70	15	5	0.7887	0.0937	1599.80
70	15	10	0.8131	0.0640	1355.78
70	15	15	0.7731	0.1139	1553.14
70	15	20	0.8139	0.0595	1293.81
70	15	25	0.8233	0.0884	1259.73
70	15	30	0.8051	0.0695	1283.44

BatchSize	On	Off	Accuracy	PM	Cost
70	15	35	0.8287	0.0646	1143.87
70	15	40	0.7709	0.0673	1521.93
70	20	1	0.8209	0.0745	1571.07
70	20	5	0.8220	0.0476	1468.58
70	20	10	0.8231	0.0509	1401.15
70	20	15	0.7973	0.0888	1412.03
70	20	20	0.7905	0.0852	1488.98
70	20	25	0.7787	0.1135	1483.52
70	20	30	0.8193	0.0598	1250.85
70	20	35	0.8193	0.0593	1226.87
70	20	40	0.8094	0.2005	1283.58
70	25	1	0.8056	0.0675	1606.66
70	25	5	0.8325	0.0371	1437.30
70	25	10	0.8392	0.0683	1374.71
70	25	15	0.8099	0.0662	1404.07
70	25	20	0.7822	0.0952	1545.25
70	25	25	0.8118	0.0626	1342.50
70	25	30	0.8322	0.0837	1255.39
70	25	35	0.8126	0.0651	1272.43
70	25	40	0.8018	0.1059	1339.24
70	30	1	0.8271	0.0451	1522.31
70	30	5	0.7865	0.0905	1697.00
70	30	10	0.8077	0.0628	1497.55
70	30	15	0.7825	0.0957	1607.46
70	30	20	0.8233	0.0570	1338.78
70	30	25	0.7932	0.1233	1544.26
70	30	30	0.8279	0.0870	1340.95
70	30	35	0.8099	0.0609	1330.07
70	30	40	0.8193	0.0589	1253.03
70	35	1	0.8379	0.0699	1510.43
70	35	5	0.7908	0.0885	1666.48
70	35	10	0.8330	0.0692	1446.99
70	35	15	0.7825	0.1021	1601.72
70	35	20	0.8166	0.0520	1400.15
70	35	25	0.8172	0.0561	1392.26
70	35	30	0.7978	0.1189	1475.27
70	35	35	0.7817	0.2529	1537.39
70	35	40	0.7895	0.0972	1420.59
70	40	1	0.8204	0.0537	1578.16
70	40	5	0.8029	0.0775	1605.02
70	40	10	0.8228	0.0510	1493.27
70	40	15	0.8094	0.0676	1487.94
70	40	20	0.8239	0.0493	1367.95
70	40	25	0.8163	0.0537	1393.82
70	40	30	0.7739	0.1184	1580.78
70	40	35	0.8180	0.0547	1369.83
70	40	40	0.7973	0.1193	1487.83
80	5	1	0.7889	0.0916	1619.32
80	5	5	0.822	0.0515	1287.73

BatchSize	On	Off	Accuracy	PM	Cost
80	5	10	0.8169	0.0632	1215.50
80	5	15	0.8061	0.0751	1215.73
80	5	20	0.825	0.0541	1110.93
80	5	25	0.7787	0.1107	1234.32
80	5	30	0.8051	0.089	1114.87
80	5	35	0.762	0.1291	1342.32
80	5	40	0.8274	0.0448	1041.02
80	10	1	0.7905	0.0934	1637.05
80	10	5	0.8212	0.0733	1410.41
80	10	10	0.8373	0.039	1219.69
80	10	15	0.8263	0.0547	1210.03
80	10	20	0.8137	0.0657	1219.11
80	10	25	0.8255	0.0536	1147.92
80	10	30	0.8155	0.061	1148.20
80	10	35	0.7897	0.0981	1308.03
80	10	40	0.8188	0.0764	1127.25
80	15	1	0.7809	0.1027	1748.08
80	15	5	0.8139	0.0648	1439.97
80	15	10	0.8142	0.0595	1393.40
80	15	15	0.8163	0.0648	1332.05
80	15	20	0.8301	0.0465	1231.72
80	15	25	0.8145	0.063	1272.68
80	15	30	0.7776	0.1293	1372.07
80	15	35	0.8137	0.0703	1225.65
80	15	40	0.7776	0.0892	1395.52
80	20	1	0.8247	0.0575	1516.05
80	20	5	0.8163	0.0662	1479.53
80	20	10	0.7779	0.1202	1601.51
80	20	15	0.7986	0.1122	1454.15
80	20	20	0.7981	0.069	1433.44
80	20	25	0.8341	0.0807	1267.35
80	20	30	0.8048	0.0768	1296.70
80	20	35	0.8314	0.0816	1207.10
80	20	40	0.8282	0.0852	1213.90
80	25	1	0.7943	0.0808	1677.53
80	25	5	0.7814	0.1114	1671.97
80	25	10	0.8088	0.0613	1490.94
80	25	15	0.7911	0.0841	1516.27
80	25	20	0.7989	0.0768	1445.01
80	25	25	0.8158	0.0633	1331.92
80	25	30	0.7688	0.1222	1504.72
80	25	35	0.8336	0.0819	1264.58
80	25	40	0.8107	0.0688	1262.70
80	30	1	0.8163	0.0662	1570.15
80	30	5	0.7806	0.1028	1724.67
80	30	10	0.8123	0.0884	1505.36
80	30	15	0.8172	0.0609	1408.41
80	30	20	0.8077	0.0726	1404.16
80	30	25	0.8244	0.0513	1309.41

BatchSize	On	Off	Accuracy	PM	Cost
80	30	30	0.8204	0.0605	1307.93
80	30	35	0.7749	0.1089	1528.59
80	30	40	0.8155	0.061	1244.61
80	35	1	0.8045	0.0559	1651.42
80	35	5	0.8177	0.0587	1545.81
80	35	10	0.8196	0.0585	1461.15
80	35	15	0.822	0.0566	1412.82
80	35	20	0.8129	0.0654	1424.85
80	35	25	0.7981	0.1137	1502.02
80	35	30	0.8163	0.0672	1348.40
80	35	35	0.8126	0.0702	1312.91
80	35	40	0.8139	0.0648	1291.06
80	40	1	0.825	0.0522	1539.50
80	40	5	0.8051	0.0695	1604.43
80	40	10	0.8158	0.0675	1496.03
80	40	15	0.8163	0.0662	1454.56
80	40	20	0.7954	0.1193	1605.50
80	40	25	0.8139	0.0696	1397.30
80	40	30	0.8013	0.0635	1438.20
80	40	35	0.8077	0.0633	1363.03
80	40	40	0.7975	0.0979	1400.20

תקציר

עבודה זו עוסקת בסיווג מוצרים חקלאיים. סיווג לאיכות של מוצר חקלאי מבוסס על ריבוי מאפיינים. ווקטור המאפיינים של המוצר הוא הכרחי להגדרת האיכות שלו. לאחר שמתקבל המידע לגבי מאפייני הפרי יש צורך בתכנון מסווג שיוכל לשייך אותו לקבוצת האיכות המתאימה.

במקרים רבים חלק מהמאפיינים לא רלוונטיים למשימת הסיווג. מאפיינים אלו עשויים לגרום להאטה במהירות הסיווג ואף לירידה ברמת הביצועים של המסווג.

ישנם מאפיינים, כגון פגמים מסוגים שונים, שעשויים להופיע בתוך זמן קצר. עקב כך ישנה חשיבות מרובה לבחירת המאפיינים בהתאם למצב נתון מיידי. בנוסף, מאפייני המוצר משתנים עם הזמן ובהתאם לעונות השנה. מסווג שיוכל לזהות שינויים אלו באופן מקוון ישפר את רמת הביצוע של הסיווג.

מחקר רב בוצע בתחום בחירת המאפיינים לשיפור רמת הסיווג. השינויים הרבים שמאפיינים את המוצרים החקלאיים מחייבים ביצוע בחירה זו באופן מקוון על מנת לאפשר למסווג להסתגל לשינוי.

מטרת העבודה העיקרית היא לפתח שיטת סיווג יעילה כדי להתגבר על בעיית השינוי במוצר. פותח מסווג הירארכי מקוון עם יכולות הסתגלות לשינויים באוכלוסיית המוצר.

<u>השיטה</u>

זיהוי הגעתה של אוכלוסייה חדשה למערכת מתאפשר ע״י ניתוח השונות במוצר. הרעיון העיקרי הוא לבחון אם זרם המוצרים הנוכחי שונה מקודמו.

לשם ביצוע בדיקה זו יישמנו ושיפרנו אלגוריתם אשכול מקוון. כשהאלגוריתם זיהה שינוי באוכלוסיית הפרי הוא בודק את רמת החפיפה שלה עם אוכלוסיות שכבר סווגו במערכת. בהתאם לרמת החפיפה מתקבלת החלטה האם להשתמש במסווג של אוכלוסייה קודמת או שיש צורך לאמן מסווג חדש.

באם רמת החפיפה גבוהה ניתן להשתמש במסווג קודם. במידה ולא נמצא מסווג כזה המערכת בוחרת מסווג חדש מבין n מסווגים מסוג K השכנים הקרובים. כל מסווג כזה מורכב ממאפיינים שונים ומערכת שילוב מסווגים מבוססת חוקים בלוגיקה עמומה. מערכת זו בוחרת במסווג עם המשקל הגבוה ביותר בסוף תהליך הבחירה. כאשר אין חפיפה בכלל יש צורך לאמן את המערכת בעזרת מומחה אנושי. עבור חפיפה חלקית משמשות נקודות מאזור החפיפה לאימון מחדש של המערכת. שתי אפשרויות אלו מוגדרות במערכת – אימון אנושי ואימון אוטומטי.

המסווג נבחן בעזרת שני בסיסי מידע: בסיס מידע מלאכותי שנבנה באופן שיתאים לבעיות הקיימות במוצרים חקלאיים ובנוסף נבחן המסווג על מוצר חקלאי אמיתי – בסיס מידע של זיתים שולחניים שנקטפו במהלך העונה.

ניתוח עלות בוצע על מנת להעריך את ביצועי המסווג במושגי עלות בנוסף לדיוק הסיווג. במהלך העבודה פותחה פונקצית עלות המבוססת על העלות החישובית של מאפיינים, אשר היו בשימוש במהלך הסיווג, ועל רמת הטעות של המסווג. שלושה מאפיינים הוגדרו כדי לשערך את יכולות המערכת.

<u>ניתוח ותוצאות</u>

<u>נתונים מלאכותיים</u>

הנתונים המלאכותיים הורכבו משש אוכלוסיות שונות. כל אחת מהן מכילה 1000 נקודות עם שבעה מאפיינים. כל מאפיין נוצר מתוך התפלגות נורמלית רב-מימדית. תוצאות הסיווג של המערכת הושוו למסווג מסוג K השכנים הקרובים שהשתמש בכל המאפיינים עבור כל הדגימות. המסווג המקוון שפותח במסגרת עבודה זו הראה שיפור של 12% יחסית למסווג זה.

ניתוח זמן הריצה של המסווג הראה את גמישותו לשינויים ויכולתו להסתגל לאוכלוסיות החדשות שנכנסו למערכת. ניתוח רגישות הראה כי לסדר כניסת האוכלוסיות למערכת יש השפעה רבה על יכולות הסיווג שלה. בנוסף, במקרים של חפיפה עם יותר מאוכלוסייה אחת הפכה הבעיה להחלטה באיזה קבוצת אימון לבחור. בעיה זו הראתה את הצורך בהגדרת אמדי דמיון בין אוכלוסיות ועל הצורך להגדיר אוכלוסיות בסיס המכסות את רוב מרחב המאפיינים של המוצר.

<u>נתונים חקלאיים</u>

10550 זיתים מ-12 זנים שונים נמסקו ממטעי עצי זית ברמת נגב בדרום הארץ. סה"כ הוצגו למערכת 21 אוכלוסיות. תוצאות ראשוניות של המערכת שסיווגה באופן מקוון את כל האוכלוסיות הראו על אחוזי דיוק של 81% אבל ב-13 מקרים נזדקקה המערכת לאימון מחדש (אנושי).

8 אוכלוסיות בסיס נבחרו ושני מדדי דמיון נוספו למדד החפיפה. במצב זה ביצועי המסווג השתפרו בהשוואה למסווגים לא מסתגלים. בהשוואה למסווג מיטבי שביצע אימון על כל אוכלוסיות הבסיס וסיווג בשימוש כל המאפיינים (מסווג מסוג עץ החלטה) ביצועי המערכת היו נמוכים יותר אבל עדיין ברמה מספקת (85% דיוק למערכת המוצעת לעומת 89% למסווג המיטבי).

ניתוח הרגישות העלה כי שינוי בפרמטרים שהוגדרו משפיע מאוד על ביצועי המערכת ועל מחירה. מספר סימולציות בשימוש שילוב הפרמטרים הביאו את המערכת לרמות ביצוע גבוהות ממסווג שהוגדר באופן לא מקוון. שינוי במאפייני פונקצית העלות הביאו לשינויים מתאימים.

סיכום

עיקר התרומה של המערכת: זיהוי יעיל של אוכלוסייה חדשה, הסתגלות לאוכלוסייה במובנים של חפיפה ואמדי דמוית, בחירת מאפיינים\מסווגים באופן מקוון, פיתוח פונקציית עלות. המסווג בוחר בצורה מקוונת את המסווג המתאים ביותר ואת המאפיינים שעמם. יתרון המערכת הוא גם ביכולתה להסתגל לאוכלוסייה חדשה בהתבסס על מדדי דמיון.

מילות מפתח: מערכות מיון חקלאיות, מדדי דמיון, זיהוי שינוי, בחירת\איחוד מסווגים, מערכת מבוססת חוקי לוגיקה עמומה, עיבוד תמונה, מכונות ראייה, בחירת מאפיינים

עבודה זו מוקדשת לזכרה של אחותי האהובה

אסנת לביא

שנפטרה בטרם עת ב-15/5/2007 לאחר מאבק אמיץ

וממושך במחלת סרטן השד

מי ייתן שתנוח על משכבה בשלום

העבודה נעשתה בהדרכתם של פרופ' יעל אידן וד"ר ויקטור אלחנתי

המחלקה להנדסת תעשיה וניהול

הפקולטה למדעי ההנדסה

מסווג רב-שלבי מקוון למערכות מיון חקלאיות

מחקר לשם מילוי חלקי של הדרישות לקבלת

"דוקטורט לפילוסופיה"

מאת

שחר לייקין

הוגש לסינאט אוניברסיטת בן-גוריון בנגב

אישור מנחה פרופ' יעל אידן _____

אישור מנחה דר' ויקטור אלחנתי _____

אישור דיקן בית הספר ללימודי מחקר מתקדמים _____

תשס"ז 2006

באר-שבע

מסווג רב-שלבי מקוון למערכות מיון חקלאיות

מחקר לשם מילוי חלקי של הדרישות לקבלת

"דוקטורט לפילוסופיה"

מאת

שחר לייקין

הוגש לסינאט אוניברסיטת בן-גוריון בנגב

תשס"ז 2006

באר-שבע