SIGNAL AND IMAGE PROCESSING ALGORITHMS FOR AGRICULTURAL APPLICATIONS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Berkan Dülek

November, 2005

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ahmet Enis Çetin(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Uğur Güdükbay

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute Engineering and Science

ABSTRACT

SIGNAL AND IMAGE PROCESSING ALGORITHMS FOR AGRICULTURAL APPLICATIONS

Berkan Dülek
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ahmet Enis Çetin
November, 2005

Medical studies indicate that acrylamide causes cancer in animals and certain doses of acrylamide are toxic to the nervous system of both animals and humans. Acrylamide is produced in carbohydrate foods prepared at high temperatures such as fried potatoes. For this reason, it is crucial for human health to quantitatively measure the amount of acrylamide formed as a result of prolonged cooking at high temperatures. In this thesis, a correlation is demonstrated between measured acrylamide concentrations and NABY (Normalized Area of Brownish Yellow regions) values estimated from surface color properties of fried potato images using a modified form of the k-means algorithm. Same method is used to estimate acrylamide levels of roasted coffee beans. The proposed method seems to be a promising approach for the estimation of acrylamide levels and can find applications in industrial systems.

The quality and price of hazelnuts are mainly determined by the ratio of shell weight to kernel weight. Due to a number of physiological and physical disorders, hazelnuts may grow without fully developed kernels. We previously proposed a prototype system which detects empty hazelnuts by dropping them onto a steel plate and processing the acoustic signal generated when kernels hit the plate. In that study, feature vectors describing time and frequency nature of the impact sound were extracted from the acoustic signal and classified using Support Vector Machines. In the second part of this thesis, a feature domain post-processing method based on vector median/mean filtering is shown to further increase these classification results.

Keywords: Acrylamide, fried potatoes, coffee, k-means, image analysis, color, segmentation, median/mean filtering, hazelnuts, acoustics, classification, aflatoxin.

ÖZET

TARIMSAL UYGULAMALAR İÇİN SİNYAL VE İMGE İŞLEME ALGORİTMALARI

Berkan Dülek
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Ahmet Enis Çetin
Kasım, 2005

Tıbbi araştırmalar akrilamidin hayvanlarda kansere neden olduğunu ve belirli dozlarının hayvan ve insan sinir sistemleri üzerinde toksik etkisinin bulunduğunu göstermiştir. Patates kızartması gibi yüksek sıcaklıklarda hazırlanan karbonhidratlı besinlerde akrilamide rastlanılmaktadır. Dolayısıyla besinlerin yüksek sıcaklıklarda uzun süreli pişirilmesi sonucunda oluşan akrilamid miktarının nicel olarak ölçülebilmesi insan sağlığı açısından büyük önem taşımaktadır. Bu tezde, k-ortalama algoritmasına dayanan bir yöntem geliştirilerek deneysel olarak ölçülmüş akrilamid konsantrasyonları ile patates kızartması ve kahve imgelerinin yüzeysel renk analizinden tahmin edilen NABY (Kahverengimsi Sarı Bölgelerin Standartlaştırılmış Alanı) değerleri arasında bir ilintinin varlığı gösterilmektedir. Önerilen yöntem akrilamid seviyelerinin sağlıklı tahmini için umut verici bir yaklaşım olarak gözükmekte ve endüstriyel alanda uygulanabilirliği bulunmaktadır. Paketleme hatlarına kameralar yerleştirilerek patates kızartmalarına ait imgeler gerçek-zamanda analiz edilebilir ve yüksek NABY değerlerine sahip olanlar ayrıştırılabilir.

Fındıkların kalite ve fiyatlarının belirlenmesinde temel unsur çekirdek ağırlığının çekirdek içi ağırlığa oranıdır. Susuzluk, besleyici öğelerin azlığı ve kurtlanma gibi fizyolojik ve fiziksel sebeplerle fındıkların içi tam olarak gelişemeyebilir. Dolayısıyla boş ve dolu fındıkların güvenilir bir şekilde otomatik olarak ayrıştırılabilmesi büyük önem taşımaktadır. Önceki bir çalışmamızda fındıkları çelik bir plakanın üzerine atıp çarpma esnasında ortaya çıkan akustik sinyali işleyerek boş fındıkları bulan bir sistem önermiştik. Çarpma sesinin zaman ve frekans bölgesine ait özelliklerini açıklayan öznitelik vektörleri çıkarılmış ve Destek Vektör Makineleri kullanılarak sınıflandırma yapılmıştı. Bu tezin ikinci kısmında, vektör ortanca/ortalama temelli süzmeye dayanan bir öznitelik bölgesi art-işleme yöntemi tasarlanarak fındıklara ait ayrıştırma sonuçlarının arttığı gösterilmektedir.

Anahtar sözcükler: Akrilamid, patates kızartması, kahve, k-ortalama, imge analizi, renk, bölütleme, ortanca/ortalama süzgeci, fındık, akustik, sınıflandırma, aflatoksin.

Acknowledgement

I would like to express my deep gratitude to my supervisor Prof. Dr. Ahmet Enis Çetin for his instructive comments and constant support throughout this study.

I would like to express my special thanks to Prof. Dr. Özgür Ulusoy and Asst. Prof. Dr. Uğur Güdükbay for showing keen interest to the subject matter and accepting to read and review the thesis.

I would also like to thank Assoc. Prof. Dr. Vural Gökmen and Asst. Prof. Dr. Selim Aksoy for many helpful suggestions and discussions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the development of fast and reliable computer technologies, digital signal and image processing algorithms have found vast application areas such as automation, defense, agriculture, health and robotics. In this thesis, we focus on two specific agricultural applications and propose algorithms based on signal and image processing techniques. The first application is the estimation of acrylamide levels in fried potato chips and roasted coffee beans using digital color images. The second is the detection of empty hazelnuts from fully developed nuts using impact acoustics. Following sections explain the motivations behind these intriguing applications, summarize the previous work and conclude with an outline of the organization of this thesis.

1.1 Image Analysis for Acrylamide Formation

1.1.1 Motivation

Acrylamide is a chemical that is used to make polyacrilamide materials. Polyacrylamide is used in the treatment of drinking-water and waste water to remove particles and other impurities. It is also used to make glues, paper and cosmetics. Polyacrylamide materials contain very small amounts of acrylamide. Acrylamide is also used in the construction of dam foundations and tunnels, and appears to be produced in some foods prepared at high temperatures such as fried potatoes. The levels of acrylamide found in some foods are much higher than the levels recommended for drinking-water, or levels expected to occur as a result of contact between food and food packing (from paper) or use of cosmetics. The highest levels found so far are in starchy foods (potato and cereal products).[1]

Acrylamide formation was found to occur during the browning process by Maillard reaction of reducing sugars with asparagine at temperatures above 120 °C [1, 2, 3, 4]. Colored products are also formed in foods during heating as a result of Maillard reaction [5, 6, 7]. These brown polymers have significant effect on the quality of food, because color is an important food attribute and a key factor in consumer acceptance. Mechanism of the formation of brown color is not fully understood yet [8].

The problem with acrylamide is that it is known to cause cancer in animals. Also, certain doses of acrylamide are toxic to the nervous system of both animals and humans. For this reason, it is crucial for human health to quantitatively measure the amount of acrylamide formed as a result of prolonged cooking at high temperatures. If a correlation is demonstrated between acrylamide concentration and surface color properties of thermally processed food images, a machine vision based system can be designed to remove those products having high levels of acrylamide from a packaging line by means of a surface image analysis.

1.1.2 Related Work

Since color can easily be measured, it may be used as an indicator of Maillard reaction products like acrylamide. Color of foods is usually measured in $L^*a^*b^*$ units which is an international standard for color measurements, adopted by the Commission Internationale d'Eclairage (CIE) in 1976. L^* is the luminance or lightness component (black to white), and parameters a^* (from green to red) and

[1] More information about acrylamide is available on http://www.who.int/foodsafety/publications/chem/acrylamide_faqs/en/index.html

b^* (from blue to yellow) are the two chromatic components [9]. Amrein et al. reported a significant correlation between the L^* values and the acrylamide content during baking at 180 °C [10]. Surdyk et al. also reported a highly significant correlation between color and acrylamide content in bread crust during baking [11]. Pedrechi et al. reported that L^* and b^* values did not show considerable changes as those shown by a^* during frying of potato chips [6]. A linear correlation was found between the acrylamide concentration and the color of potato chips represented by the redness component a* at temperatures of 120, 150 and 180 °C for up to 5 minutes of frying. However, the effect of prolonged frying on acrylamide concentration and color was not mentioned by these researchers. Taubert et al. investigated the relation between the level of surface browning and acrylamide concentration of French fries with linear regression. They reported that there could be a close correlation for small-surface material being fried [12]. A somewhat less close correlation was observed for intermediate-surface material, while no correlation was observed for large-surface material.

Although these findings suggest that surface color may be correlated with acrylamide concentration in thermally processed foods, the measurement of surface image and its color properties need to be investigated in more detail to establish a useful correlation. As illustrated in Figure 1.1, the amount of measured acrylamide increases rapidly at the onset of frying, reaching an apparent maximum concentration of 10963 ng/g. However, the acrylamide concentration in potato chips decreases exponentially as the time passes. These results suggest that acrylamide forms as an intermediate product during Maillard reaction and its concentration begins to decrease as the rate of degradation exceeds the rate of formation during heating. Although the rapid increase is conveniently modeled by CIE a^* parameter, the exponential fall is not captured. The study also shows that CIE L^* and b^* values decrease exponentially during frying at 170 °C and they keep decreasing as the frying proceeds.

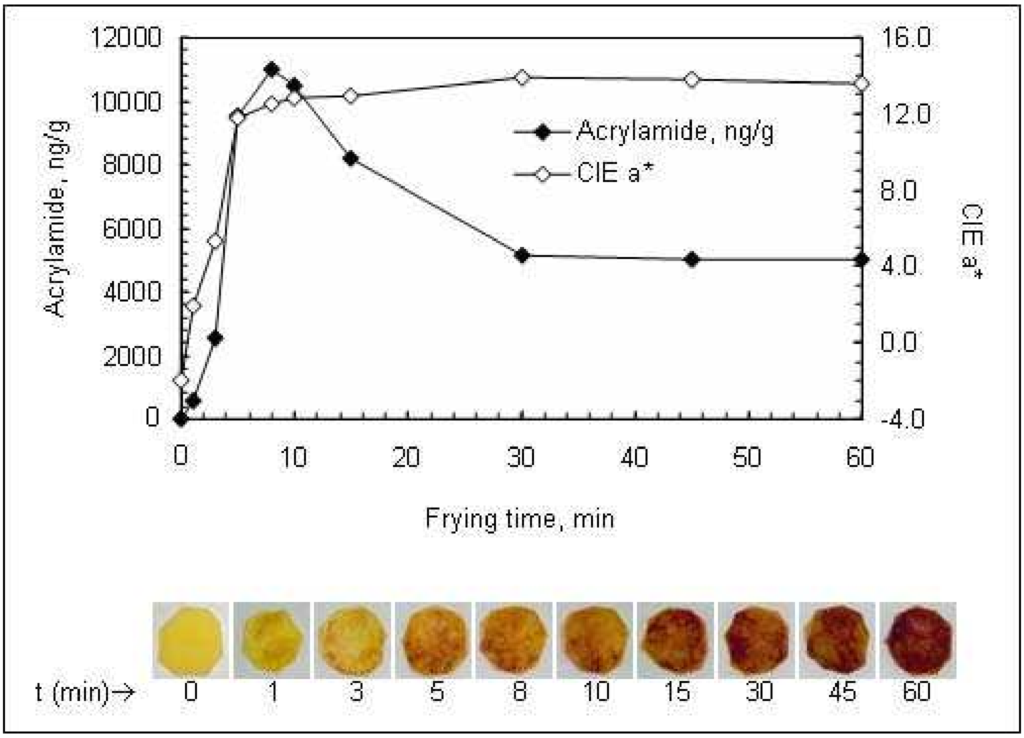


Figure 1.1: Change of acrylamide concentration and CIE redness parameter a^* in potato chips during frying at 170 °C

1.2 Detection of Empty Hazelnuts using Impact Acoustics

1.2.1 Motivation

The quality and price of hazelnuts is mainly determined by the ratio of shell weight to kernel weight. Due to physiological disorders such as plant stress, dehydration and lack of nutrients, hazelnuts may grow without fully developed kernels. Physical disorders such as insect infestation also prevent hazelnuts to develop into a healthy form by intervening the maturation process. It is usually the case that empty hazelnuts and nuts with undeveloped kernels contain a cancer causing material, called aflatoxin. Currently, pneumatic devices are employed to segregate between empty and full hazelnuts. However, these devices suffer from

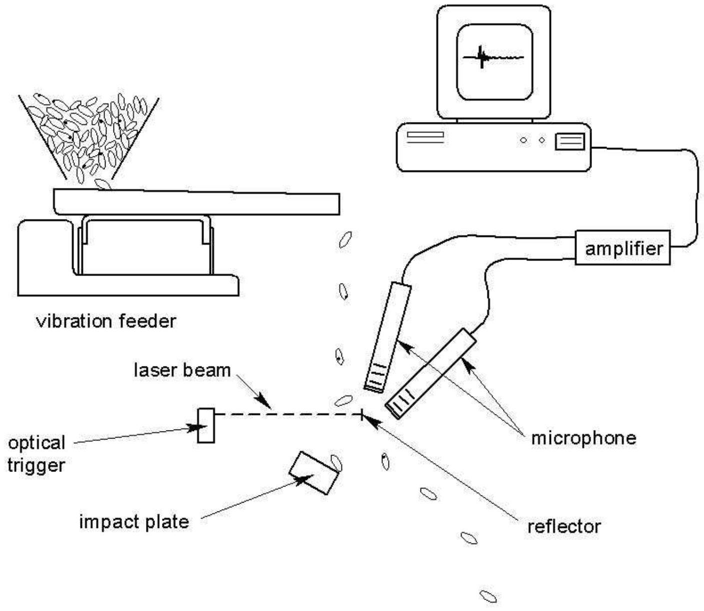


Figure 1.2: Schematic of experimental apparatus for collecting acoustic emissions from hazelnuts

high classification error rates. Therefore, it becomes a necessity both industrially and in terms of food safety to provide a reliable separation between these two types of product in an autonomous manner.

1.2.2 Related Work

Previously, a high-throughput (20-40 nuts/second), low-cost acoustical prototype system was developed to separate pistachio nuts with closed shells from those with cracked shells in real time [13, 14, 15]. A similar system was proposed to detect empty hazelnuts by dropping them onto a steel plate and processing the acoustic signal generated when the kernel hits the plate [16]. An air valve can be used to separate detected empty hazelnuts from the process stream. The schematic diagram of the system is shown in Figure 1.2. The proposed system works reliably in a food processing environment with little maintenance or skill required to operate. In addition, signal processing part can be carried out in an ordinary PC with 44kHz sound sampling capability.

1.3 Organization of the Thesis

Chapter 2 develops a modified version of the well-known k-means clustering algorithm [17, 18] so that it can be used effectively in any supervised classification framework. A performance analysis is carried out to compare the results with some other state-of-the-art classification techniques such as Gaussian Mixture Modeling, Support Vector Machines, Back-Propagation Neural Networks and K-Nearest Neighbors [19].

In Chapter 3 and 4, the proposed method is applied to estimate acrylamide levels in digital color images of fried potato chips and roasted coffee beans, respectively. The relation between CIE a^* values and acrylamide levels of potato chips is investigated and it is observed that it is hard to define specific regions in the range of a^* values that point to acrylamide formation. A new method based on the segmentation of fried potato images into three regions is devised. Normalized-RGB color values are used as features for the segmentation of acrylamide contaminated areas in digital images. The changes of acrylamide levels and estimated values are tabulated and shown to follow almost the same trend. We also demonstrate that autocovariance estimates can be incorporated as additional statistical features into the feature set of fried potato images.

In Chapter 5, a feature domain post-processing method is developed to increase performance in the separation of empty hazelnuts from fully developed nuts by impact acoustics. The idea is inspired from the well-known median filtering approach. In addition to median filtering based post-processing, the results of an averaging filter are also examined.

The last chapter concludes the thesis with an elaborate summary of the results obtained in the previous chapters.

Chapter 2

Modified K-means based Classification

In this chapter, we provide a modification for the k-means algorithm which makes it more suitable for classifying data in a supervised manner within a training-followed-by-testing framework. K-means clustering algorithm [17, 18] is chosen because of its simplicity, high performance and fast implementation properties. In the following sections, we present a summary of the k-means clustering algorithm, our contribution to this algorithm, application of our method to some popular classification datasets and performance comparison with other widely used classification methods. The proposed method is generic in the sense that it can be applied to any classification dataset without much modification.

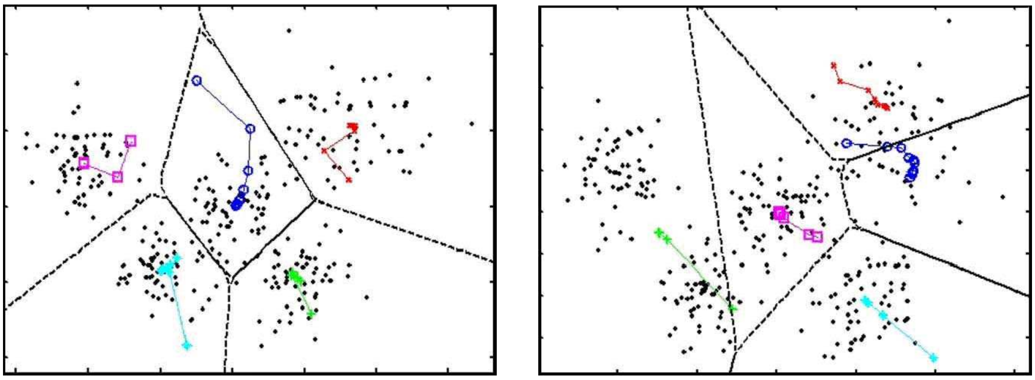
2.1 K-means Clustering

K-means clustering is one of the simplest unsupervised learning algorithms that is adopted to many problem domains as a result of its simple computation and accelerated convergence. It is also called Vector Quantization (VQ) in digital waveform coding literature [20]. It is a very popular method listed under the class of iterative optimization procedures. Given a dataset, this procedure provides an easy and simple way to partition the observation vectors in the dataset into k mutually exclusive clusters. It is computationally efficient and gives good results if the clusters are compact, hyperspherical in shape and well-separated in feature space. The main idea in k-means is to arrange the partitions in such a way that objects belonging to a certain cluster are as close to each other as possible and as far from the objects in other clusters as possible. Each object in the dataset is identified with the index of the cluster to which it belongs and the centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized.

Fixing the number of clusters a priori, the algorithm starts with the selection of k initial cluster centroids inside the space spanned by the observation vectors. The wise thing to do at this point is to select these initial points in such a way that they are separated as much as possible from each other inside the cloud of data points. At the next step, all points in the dataset are assigned to their nearest cluster centers with respect to a predefined distance measure. After all the points are processed, k new cluster centroids are calculated using the cluster bindings obtained in the previous step. Then a new iteration is initiated by reassigning points to their nearest cluster centroids resulting from the last iteration. Each iteration consists of these two consecutive steps of cluster assignment and centroid calculation. With this iterative approach, k centroids change their location step by step until no more changes are possible as illustrated in Figure 2.1(a). This results in the minimization of a criterion function, in this case the sum of point-to-centroid distances, summed over all k clusters. Depending on the kind of data being clustered, an educated selection can be made among a number of distance measures such as Minkowski, Euclidean, city-block and cosine distances.

Suppose that a dataset of n patterns is partitioned into k clusters $D_1, \ldots, D_k$ at any stage during the iterative procedure. Let n_i be the number of samples assigned to D_i and let $\mathbf{m_i}$ be the mean of those samples:

$$\mathbf{m_i} = \frac{1}{n_i} \sum_{x \in D_i} \mathbf{x}$$



(a) Good Initialization

(b) Bad Initialization

Figure 2.1: Effect of initial point selection on the performance of k-means algorithm. Here the same dataset is partitioned into five clusters; (a) yields the desired clustering while (b) gets trapped in a local minimum.

(Adapted from Selim Aksoy, Bilkent University)

Then, the minimization criterion function is defined as:

$$J_e = \sum_{i=1}^{k} \sum_{x \in D_i} \|\mathbf{x} - \mathbf{m_i}\|^2$$

and for a given cluster D_i, the mean vector $\mathbf{m_i}$ (centroid) is the best representative of the samples in D_i.

The algorithm is composed of the following steps:

1. Select an initial set of k cluster centroids.
2. Generate a new partition by assigning each pattern to its closest cluster centroid.
3. When all patterns are assigned, recalculate the positions of the k centroids.
4. Repeat steps 2 and 3 until either a local minimum of the criterion function is found or a predefined number of iterations is exceeded.

2.1.1 Major Drawbacks of the K-means Algorithm

This iterative procedure is guaranteed to converge but it does not necessarily find the most optimal configuration, corresponding to the global criterion function minimum (See Figure 2.1(b)). It is possible for the algorithm to reach a local minimum, where reassigning any one point to a new cluster would increase the total sum of point-to-centroid distances, but where a better solution does exist. To overcome this drawback, initial cluster centroids are often chosen by picking up k random points uniformly distributed from the range of the data or by randomly selecting k points from the data and running the algorithm several times. Another problem may occur when the set of observation vectors (patterns) closest to a cluster centroid is empty and as a result, this cluster center cannot be updated. One possible solution is to create a new cluster consisting of the one point furthest from its centroid and remove the empty cluster.

The distance measures mentioned in the previous paragraphs implicitly assign more weighting to features with large ranges than those with small ranges. This is likely to cause some trouble whenever there is a considerable amount of difference in the range of the data along different axes in a multidimensional space. A popular solution is to apply feature normalization (such as linear scaling to unit variance or unit range) so that features will have approximately the same effect in the distance computation. Selection of the optimal number of clusters for any given dataset also presents another difficulty. A simple way to overcome this problem is to run the clustering algorithm with several values of k and choose the one that best conforms to the requirements of the current situation. But one thing to be remembered is that increasing the value of k too much usually brings the risk of overfitting the dataset as a side effect.

2.1.1.1 Information Criterion Scoring for Estimation of the Optimal Number of Clusters

As mentioned earlier, it is not usually apparent to choose which value of k from the context of the problem. For this reason, a number of algorithms have been proposed in the literature to determine k automatically.

An information criterion function is composed of two main parts. The first part expresses the goodness of fit by the selected model. The second part is a penalty term for model complexity. Proposed models corresponding to different values of k are evaluated using this criterion function and the one producing the best score is selected.

Statistically, k-means is considered as a special case of Expectation-Maximization algorithm used in Gaussian Mixture Modeling. In k-means, equal mixture probabilities and identical spherical covariance matrices are assumed for all clusters. By assigning each sample point $\mathbf{x}_j$ to its closest centroid $\mathbf{u}_{k_j}$ obtained from k-means, the classification likelihood can be calculated as a measure of the goodness of fit as follows:

$$P(\mathbf{x}_j|\mathbf{M},\sigma^2) = \frac{1}{\sqrt{(2\pi)^d\sigma^{2d}}}\, exp\left(-\frac{\|\mathbf{x}_j - \mathbf{u}_{k_j}\|^2}{2\sigma^2}\right)$$

and the likelihood of the entire dataset $D = \{\mathbf{x}_j\}$ becomes:

$$P(D|\mathbf{M},\sigma^2) = \prod_j P(\mathbf{x}_j|\mathbf{M},\sigma^2)$$

where k_j is the cluster to which $\mathbf{x}_j$ is assigned, $\mathbf{M}$ is the tested model and d is the dimension (number of features). The maximum likelihood estimate (MLE) for variance, under identical spherical Gaussian assumption, is computed as follows:

$$\hat{\sigma}^2 = \frac{1}{Nd}\sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{u}_{k_j}\|^2$$

where N is the total number of sample points from all clusters.

Based on this observation, the following model criteria can be used to estimate the optimal number of clusters:

Akaike Information Criterion [21],

$$AIC(\mathbf{M}) = -\log P(D|\mathbf{M},\sigma^2) + (kd+1)$$

Bayesian Information Criterion [22],

$$BIC(\mathbf{M}) = -\log P(D|\mathbf{M}, \sigma^2) + \frac{(kd+1)}{2}\log N$$

Integrated Classification Likelihood [23],

$$ICL(\mathbf{M}) = -\log P(D|\mathbf{M}, \sigma^2) + \frac{(kd+1)}{2}\log N + \sum_{i=1}^{N}\log\left(i + \frac{k+2}{2}\right) - \sum_{i=1}^{k}\sum_{j=1}^{N_i}\log\,(j + \frac{3}{2})$$

where N_i is the number of sample points in cluster i, such that $\sum_i N_i = N$

Among these methods, AIC generally overestimates the number of clusters. On the other hand, BIC returns the true number of clusters under the assumption that the dataset is infinitely large. ICL tries to increase the performance of BIC by taking into account the internal membership of sample points to corresponding clusters.

2.1.2 Computational Complexity of the Algorithm

The computational complexity of the algorithm is $O(ndkT)$ where n is the number of patterns, d is the number of features, k is the desired number of clusters, and T is the number of iterations. In practice, the number of iterations is generally much less than the number of patterns.

2.1.3 Improvement of the Algorithm

The set of instructions explained in the beginning of Section 2.1 is defined as 'batch' updates in the literature. The values obtained at the end of this procedure can be accepted as the answer, or they can be used as starting points for more exact computations. In practice, classical k-means is often followed by an additional phase which is called 'online' updates. The distinction is that batch updates are applied to all points in the dataset at once during a single iteration.

However, online updates take over from the point where batch updates left and each point is individually reassigned if doing so will further reduce the value of the criterion function. Cluster centroids are recomputed after each reassignment. This is repeated for all points in the dataset which completes a single iteration for the second phase.

Following the introduction of basic k-means algorithm in the literature, many contributions have been proposed to improve its performance. Among these, we can mention two of them here:

2.1.3.1 Simulated Annealing

Simulated annealing gives a system the ability to escape from unfavorable local minima to which it might have been initialized [24, 25]. The simulated annealing method when applied to k-means algorithm is repeatedly executed in the following manner:

1. An initial partitioning of the given dataset is obtained by running the k-means algorithm until convergence.
2. The formed clustering is slightly modified to find a potentially better one.
3. If the resulting partitioning decreases the cost function, it is accepted; if the cost function is increased, it is accepted with some probability.

The modification mentioned in Step 2 involves the merging of two clusters and splitting of a cluster so that the total number of clusters remains the same. The selection of which clusters should be merged and which one should be split is made randomly, but there is a higher probability for a large cluster to be split and two closer clusters to be merged.

The probability with which a poor modification is accepted depends on a system parameter, called ‘temperature’. The algorithm starts with an initial temperature of T_0 and it drops as the algorithm proceeds. Lower temperatures result in higher rejection probabilities for unfavorable modifications. This procedure is continued repeatedly until the total number of acceptances and rejections exceeds a certain predetermined value or the system attains an acceptable value for the cost function.

2.1.3.2 Fuzzy K-means

While classical k-means procedure assumes that each sample point can be assigned to exactly one cluster, fuzzy k-means approach relaxes this condition and allows for each sample to have some graded or 'fuzzy' membership to a cluster [26, 27]. After deciding on the initial guesses for cluster centroids, the membership probabilities and cluster centroids are updated iteratively. The criterion function minimized during each iteration consists of the sum of distances from any given data point to a cluster center weighted by the data membership probability of that data point.

2.2 Our K-means based Classification Algorithm

In this section, we suggest a classification method based on the classical k-means algorithm. We demonstrate the efficiency of our method through several examples and provide figures to better explore its properties. Although this approach can be generalized for multi-class problems with arbitrary dimensionality[1], we select two-dimensional datasets with two-classes for easy visualization and illustrative purposes. We conclude this section with a performance evaluation of our method with some other state-of-the-art classification algorithms on three specifically selected datasets.

Like most of the supervised classification algorithms, our method is composed

[1] The following chapter on the application of our method for acrylamide concentration estimation carries these ideas into such a more complicated setting.

of two stages: *a*) training, and *b*) testing. Below, we explain the steps of each stage in detail.

Training Stage

Given a dataset with m classes in d dimensions, we first partition the whole dataset, irrespective of their class memberships, into k distinct clusters by running the classical k-means algorithm until convergence. It may be necessary to run the program a few times more with different initial centroids in order to prevent it from getting stuck in a local minimum. Optimal number of clusters can also be determined using information criterion techniques discussed in Section 2.1.1.1. It is also greatly advantageous to normalize the features of the given dataset beforehand to unit variance or unit range whenever certain distance metrics such as Euclidean, Minkowski and Manhattan are used in the k-means algorithm.

Let

- $C_1, C_2, \ldots, C_m$ denote the m classes present in the dataset,
- $D_1, D_2, \ldots, D_k$ denote the k clusters obtained as a result of running k-means successfully on the whole dataset,
- n_{ij} be the number of class i samples assigned to cluster j, and
- $\mathbf{m_{ij}}$ be the centroid location for class i inside cluster j.

Then, m_{ij} is calculated as follows:

$$\mathbf{m_{ij}} = \frac{1}{n_{ij}} \sum_{\substack{x \in C_i \\ x \in D_j}} \mathbf{x} \qquad \text{for all } i = 1, \ldots, m \text{ and } j = 1, \ldots, k$$

So we obtain, at most, k new centroid locations for each of the m classes. But usually not all the centroid locations are of important value in terms of their contribution to classification performance. For this reason, in our search for the representative vectors for each class of data, we define two threshold parameters to decide which $\mathbf{m_{ij}}$ values are appropriate for our purpose.

T$_1$: Let n_j be the number of samples belong to cluster j, obtained from classical k-means. This condition demands that

$$\frac{n_{ij}}{n_j} > T_1$$

where T_1 is a threshold.

Initial clusters are usually occupied by the sample points of more than one class near the boundaries. For this reason, it is necessary to check if a reasonable ratio is exceeded before starting to compute the representative vector for a certain class inside a cluster. Otherwise, a class centroid would be generated although the vicinity of that centroid is mainly occupied by the members of other classes.

T$_2$: Sometimes a cluster with too few elements may become generated and it may not be desirable to compute class centroids inside that cluster. This condition imposes $n_j > T_2$ and assures that such situations are handled in advance.

However, there may be conditions where our precautions explained in the previous paragraphs become too restrictive leading to loss of valuable class centroids. Since these thresholds are imposed globally for all clusters of the dataset, they can easily fail to perform desirably in the local scales of the dataset. Defining different thresholds for each cluster separately is not a handy solution and violates the automatic behavior of our method. Therefore, in order to make up what is possibly lost during thresholding stage we propose a brute force searching algorithm. All the class centroids discarded by user-selected thresholds are kept internally. Next, they are repeatedly introduced to the dataset under certain arrangements to see if they help to increase the classification accuracy. At each step, the arrangement that provides the highest contribution is selected. This iterative procedure is continued until no improvement is obtained by adding extra centroids to the dataset.

Below is a list of some of the arrangements we used in our experiments. At each iteration,

single centroid that provides the highest improvement on the overall classification accuracy is introduced,

m centroids (one from each class and all at once) that provide the highest improvement on the overall classification accuracy are introduced,

single or m centroids or any number of centroids in between whichever provides the highest improvement on the overall classification accuracy is introduced,

m closest centroids that provide the highest improvement on the overall classification accuracy is introduced.

It is also possible to force that additional centroids must increase the classification accuracy for each class in order to be included. But this may not be a feasible idea for datasets with large number of classes.

Test Stage

Using the algorithm outlined above in successive steps, we arrive at a number of representative vectors for each class. After applying feature normalization to the test set with the same parameters obtained in the training set, samples in the test set are ready to be classified as belonging to one of m classes. The classification is done by assigning the label of the class centroid that is closest to the test sample.

2.2.1 Properties of the Method and Examples

This section aims to give some insight into the properties related to parameter selection in our method. The advantages and disadvantages due to a particular choice of parameter values are discussed in the previous section. So, here we can directly pass to the specific examples. These examples are given for a 2-dimensional, 2-class dataset of 2000 objects, known as Lithuanian classes in the literature.

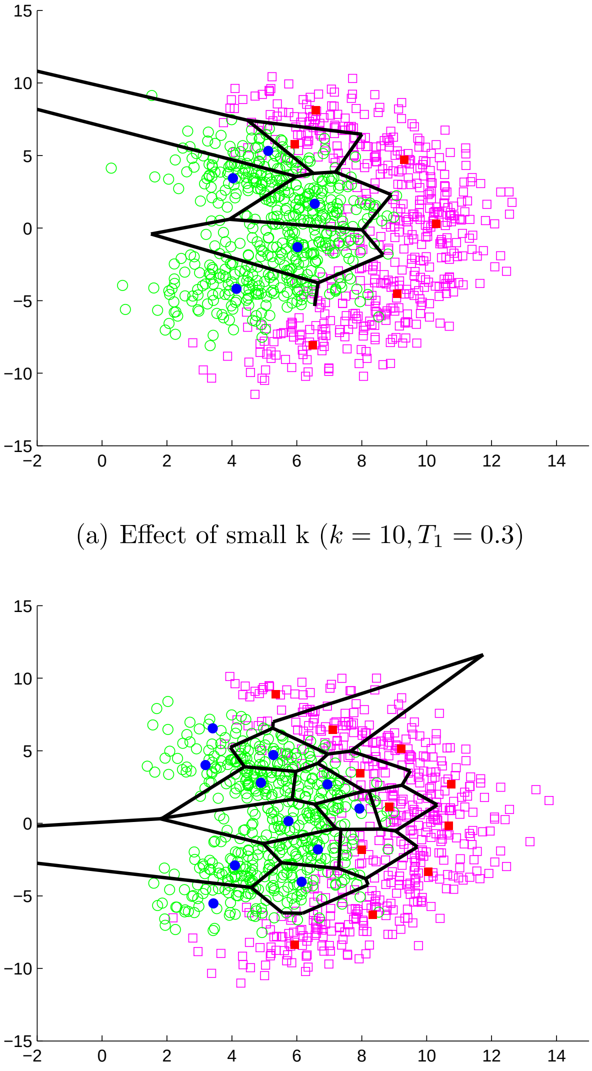
Figure 2.2 demonstrates the effect of increasing the number of initial clusters. As parameter k is increased from 10 to 20, we see that more class centroids are generated in the vicinity of the boundary between two classes. Enabling brute force searching with single centroid option allows one additional cluster center to be included for $k = 20$ case while none is included for $k = 10$ case. The results indicate a slight rise in the classification performance as shown in Table 2.1.

To analyze the effect of increasing T_1, we select a high value of initial clusters, $k = 15$. So, it is more likely that some class centroids will be discarded for greater values of T_1. This turns out to be a useful choice as depicted in Figure 2.3. While searching for additional class centroids that better describe the dataset, $T_1 = 0.5$ case finds a better partitioning of the sample space resulting in a higher classification accuracy as shown in Table 2.2.

Previously it was mentioned that brute force searching could provide a remedy for accidentally lost class centroids via thresholding. Figure 2.4 demonstrates this idea. By adding only two centroids at each iteration, this example run of the program incorporates 4 additional pairs of cluster centroids that provides a better representation of the data in training set as shown in Table 2.3. ‘2 closest centroids’ and ‘single or 2 centroids’ options listed in the previous section are also investigated in Figure 2.5 and Table 2.4. Combined with a suitable selection of thresholds they also help to construct a better model of the dataset. 3 pairs of closest centroids are added as shown in Figure 2.5(b), and a single and a pair of centroids are added as shown in Figure 2.5(d).

Table 2.1: Effect of increasing k on classification performance

	Classification Rate
$k = 10, T_1 = 0.3, T_2 = 20$	92.1
$k = 20, T_1 = 0.3, T_2 = 20$	92.5



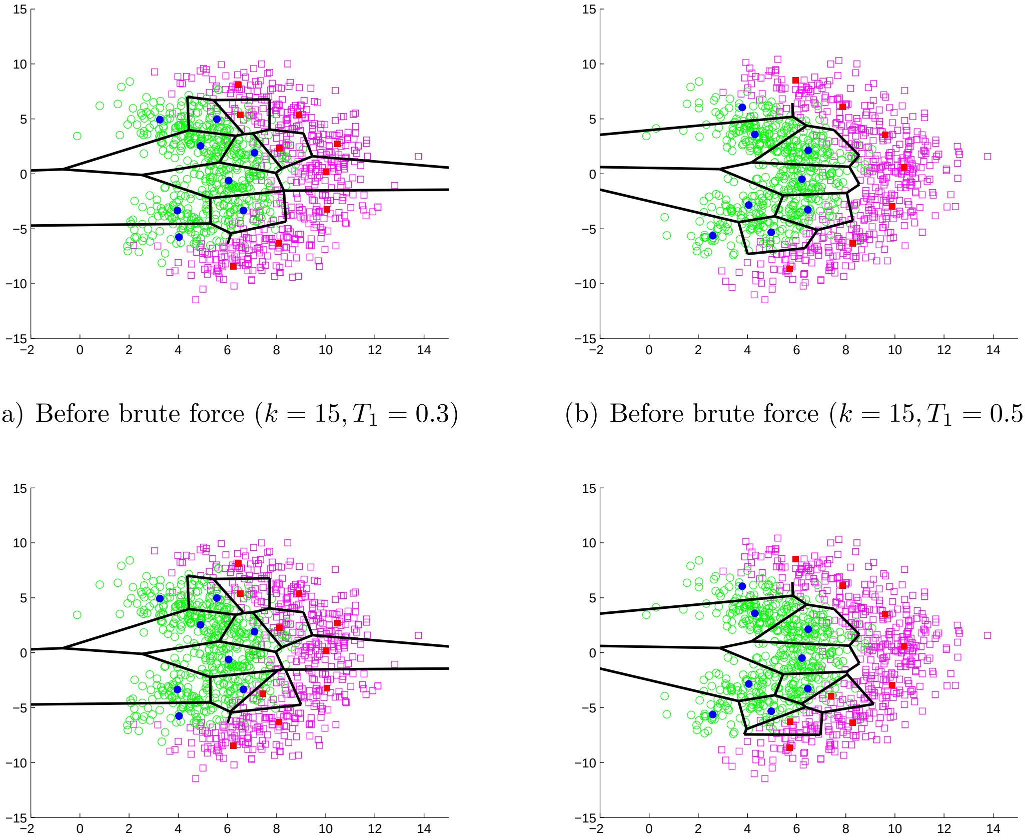
(a) Effect of small k ($k = 10, T_1 = 0.3$)

(b) Effect of large k ($k = 20, T_1 = 0.3$)

Figure 2.2: Effect of increasing k on the performance of our modified k-means algorithm. (b) shows that more cluster centers can be obtained in the vicinity of the boundary between two classes.

Table 2.2: Effect of increasing T_1 on classification performance

	Classification Rate (after brute force)
$k = 15, T_1 = 0.3, T_2 = 20$	92.3
$k = 15, T_1 = 0.5, T_2 = 20$	93.8



(a) Before brute force ($k = 15, T_1 = 0.3$)

(b) Before brute force ($k = 15, T_1 = 0.5$)

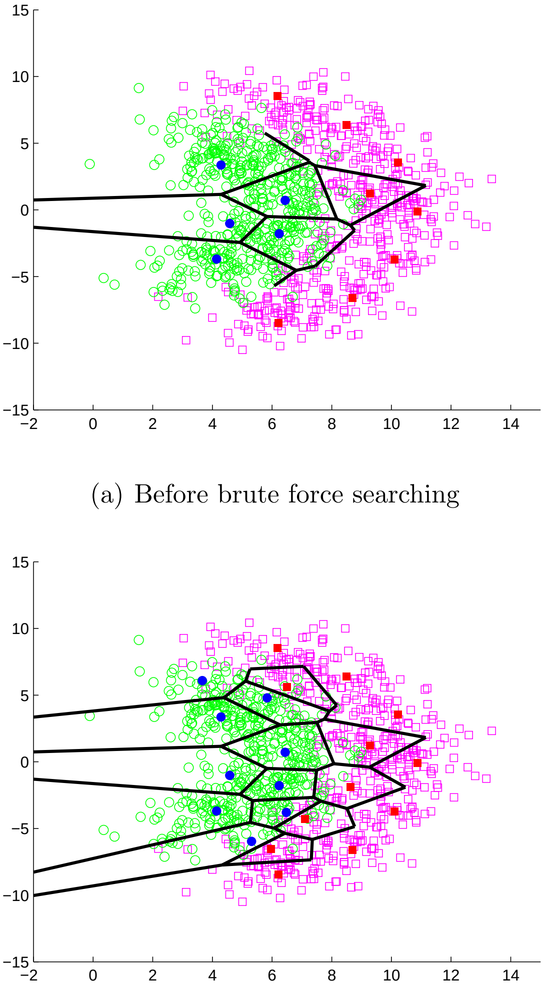
(c) After brute force ($k = 15, T_1 = 0.3$)

(d) After brute force ($k = 15, T_1 = 0.5$)

Figure 2.3: Effect of increasing T_1 on the performance of our modified k-means algorithm. Increasing T_1 results in ignoring more centroids formed near the boundary at the first stage of our algorithm (b). In some cases this may lead to a better partitioning when brute force searching of the second stage is applied (d).

Table 2.3: Effect of brute force searching on the representation of training set

	Classification Rate in Training Set
	$k = 20, T_1 = 0.8, T_2 = 30$
before brute force	92.3
after brute force	93.8



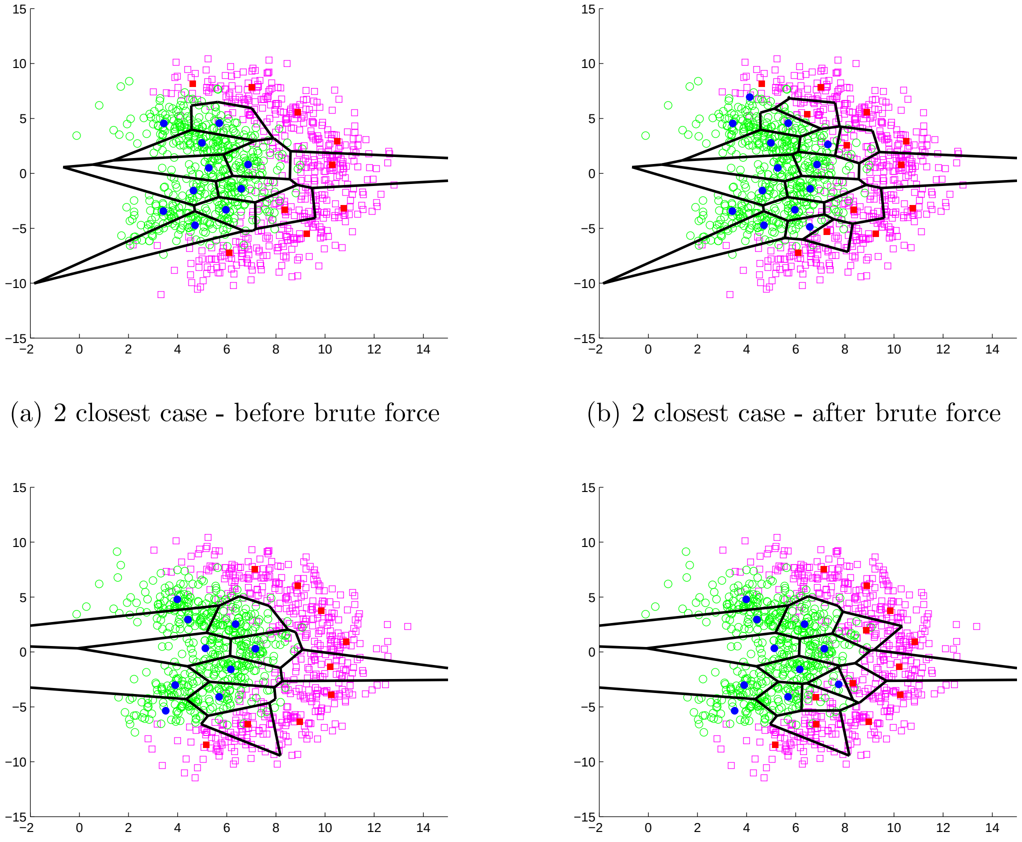
(a) Before brute force searching

(b) After brute force searching

Figure 2.4: Effect of brute force searching on the performance of our modified k-means algorithm. (b) depicts that a more accurate model can be constructed with intentionally selecting additional cluster centers that increase classification rate in the training set.

Table 2.4: Effect of other brute force searching methods on the representation of training set

	Classification Rate in Training Set $k = 25, T_2 = 30$ before brute force	 after brute force
‘2 closest’ case $T_1 = 0.6$	91.2	93.8
‘single or 2’ case $T_1 = 0.8$	93.7	94.4



(a) 2 closest case - before brute force

(b) 2 closest case - after brute force

(c) single or 2 case - before brute force

(d) single or 2 case - after brute force

Figure 2.5: Effect of different brute force searching methods on the performance of our modified k-means algorithm. In (b), two class centroids which are at minimum distance to each other are selected at every iteration of the brute force approach. In (d), selection is based on picking either one or two (i.e., one from each class) class centroids that increase the success rate most.

Table 2.5: Performance comparison of our modified k-means algorithm with other well-known classifiers

	Success Rate		
Classifiers	Lithuanian	Banana Set 1	Banana Set 2
our method	93.6	98.5	93.4
gaussian mixture modeling	94.0	98.4	94.4
support vector machines	93.5	98.2	93.1
back-propagation neural network	93.5	97.9	92.9
k-nearest neighbor	93.1	98.0	92.1
quadratic bayes normal	87.5	84.5	83.6

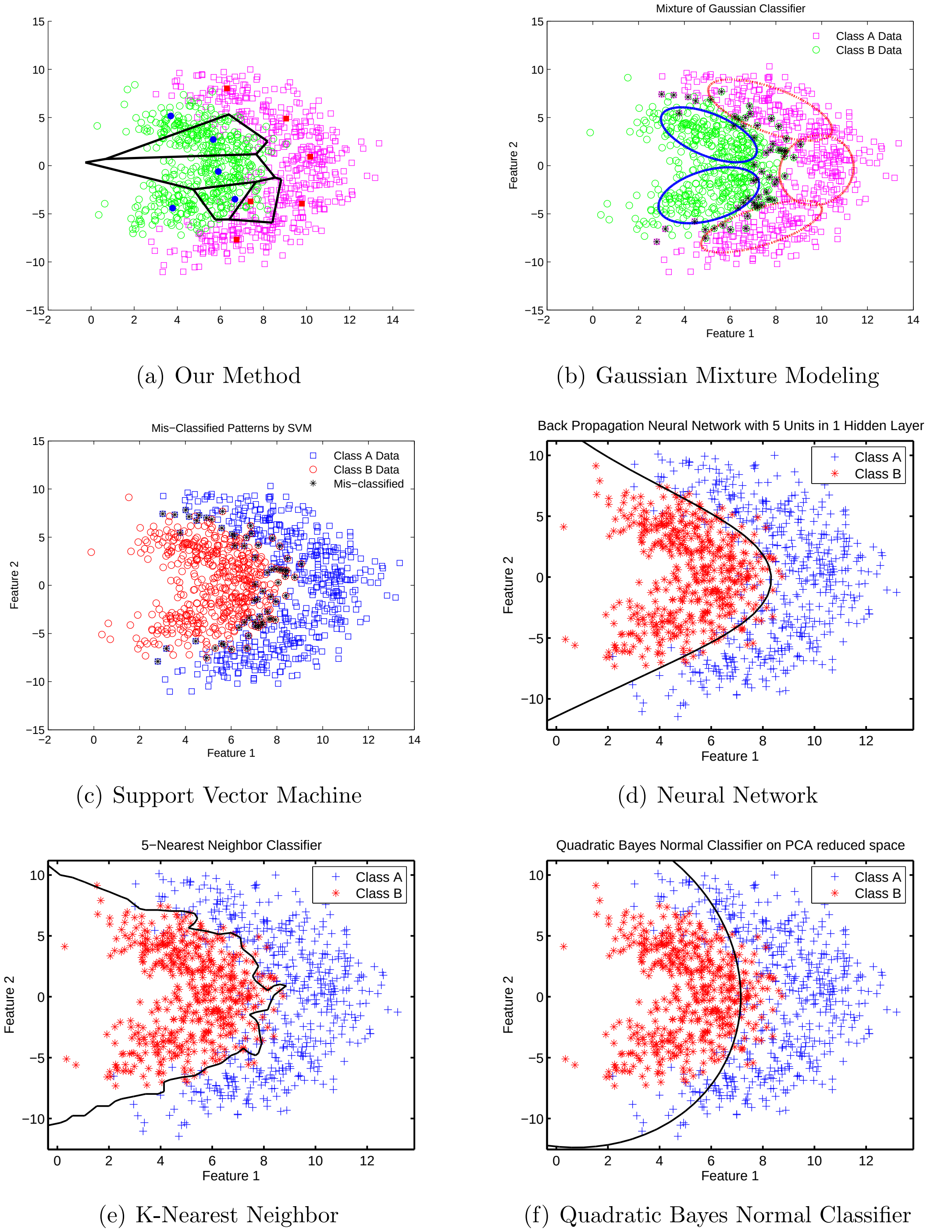
2.2.2 Performance Comparison

In this section, we compare the performance of our method with five other state-of-the-art classification algorithms. Two different datasets are provided for performance analysis. They are divided randomly into two equal halves: one forming the training set and the other forming the test set.

First one is a dataset of 2000 samples. The data is uniformly distributed along two sausages and is superimposed by a normal distribution with unit standard deviation in all directions. It is a 2-dimensional, 2-class dataset which is usually referred as Lithuanian dataset in the literature. Second dataset is the same size of the first one and contains samples with a banana shaped distribution. The data is uniformly distributed along the bananas and is again superimposed by a normal distribution with unit standard deviation in all directions. A third dataset is generated from the second one by increasing the standard deviation of the superimposed normal distribution to 1.5. Hence, more outliers are generated which will cause higher classification error rates.

Misclassified sample points, classification boundaries and details of the classifiers used are explained for all three datasets in Figures 2.6, 2.7 and 2.8, respectively. Success rates are given in Table 2.5. We observe that our classifier outperforms all except gaussian mixture modeling based classifier for two datasets.

In this part, libraries of LIBSVM are used for SVM classification [28]. It enables the automatic selection of optimal parameters by employing 10-fold cross validation and an exhaustive search on the parameter space. PRTools is used to optimize parameters, obtain classification errors and draw plots for k-nearest neighbor, back-propagation neural network and quadratic bayes normal classifiers [29]. All the code for classifiers and plots is written under Matlab 7.0 [30].



(a) Our Method

(b) Gaussian Mixture Modeling

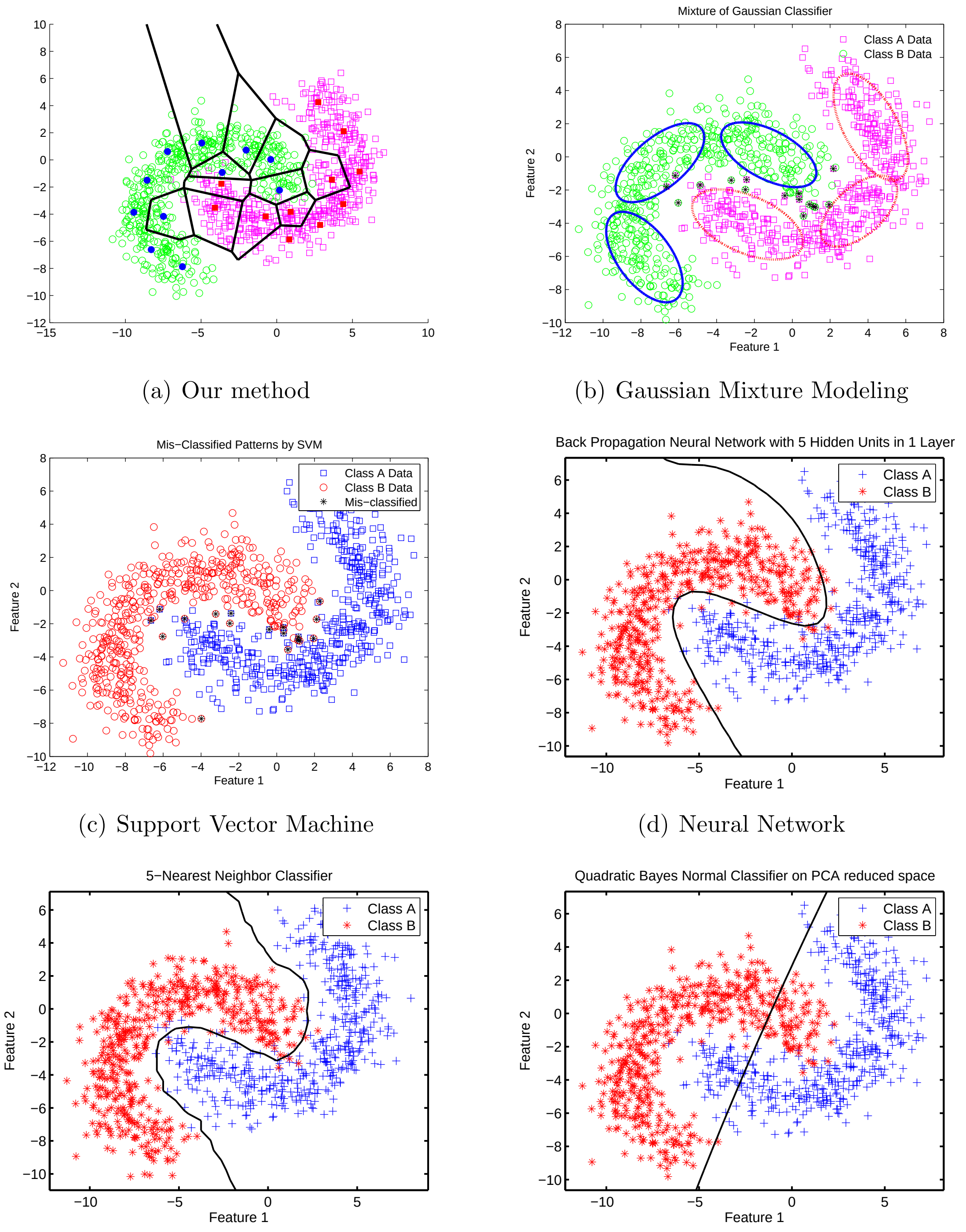
(c) Support Vector Machine

(d) Neural Network

(e) K-Nearest Neighbor

(f) Quadratic Bayes Normal Classifier

Figure 2.6: Performance comparison of our modified k-means algorithm with some state-of-the-art classifiers on Lithuanian dataset with 2 classes: (a) Our method ($k = 10$, $T_1 = 0.5$, $T_2 = 20$), (b) Gaussian Mixture Models using arbitrary covariance matrices, (c) SVM using radial basis function, (d) Back-propagation Neural Network with 5 units using 1 hidden layer, (e) K-Nearest Neighbors ($k = 5$), and (f) Quadratic Bayes Normal Classifier on PCA reduced space.



(a) Our method

(b) Gaussian Mixture Modeling

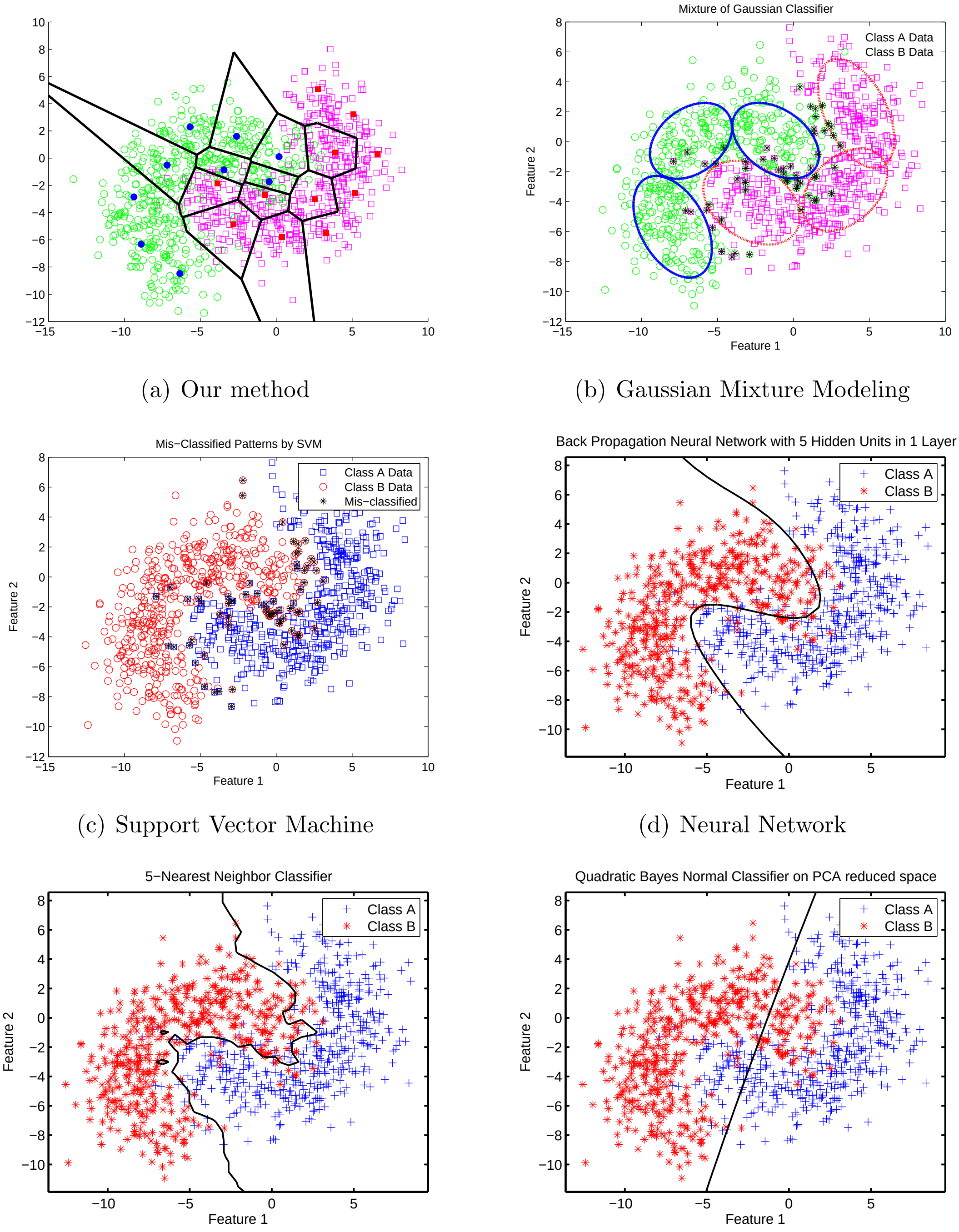
(c) Support Vector Machine

(d) Neural Network

(e) K-Nearest Neighbor

(f) Quadratic Bayes Normal Classifier

Figure 2.7: Performance comparison of our modified k-means algorithm with some state-of-the-art classifiers on banana-shaped dataset with 2 classes ($randomnessfactor = 1$): (a) Our method with minimum distance based brute forcing ($k = 25$, $T_1 = 0.6$, $T_2 = 30$), (b) Gaussian Mixture Models using arbitrary covariance matrices, (c) SVM using radial basis function, (d) Back-propagation Neural Network with 5 units using 1 hidden layer, (e) K-Nearest Neighbors ($k = 5$), and (f) Quadratic Bayes Normal Classifier on PCA reduced space.



(a) Our method

(b) Gaussian Mixture Modeling

(c) Support Vector Machine

(d) Neural Network

(e) K-Nearest Neighbor

(f) Quadratic Bayes Normal Classifier

Figure 2.8: Performance comparison of our modified k-means algorithm with some state-of-the-art classifiers on banana-shaped dataset with 2 classes ($randomnessfactor = 1.5$): (a) Our method with minimum distance based brute forcing ($k = 20$, $T_1 = 0.8$, $T_2 = 30$), (b) Gaussian Mixture Models using arbitrary covariance matrices, (c) SVM using radial basis function, (d) Back-propagation Neural Network with 5 units using 1 hidden layer, (e) K-Nearest Neighbors ($k = 5$), and (f) Quadratic Bayes Normal Classifier on PCA reduced space.

Chapter 3

Image Analysis of Potato Chips for Acrylamide Formation

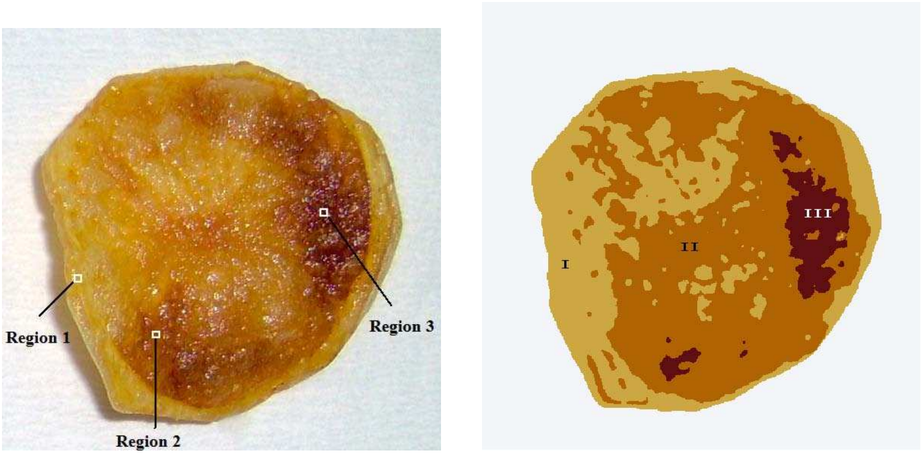
This chapter begins with a discussion on how to model acrylamide formation using digital images. It shows that the information contained in CIE a^* parameter is not sufficient for this purpose. The chapter continues with an implementation of our algorithm developed in Chapter 2 to estimate acrylamide levels in digital color images of fried potato chips. Normalized-RGB color values are selected as features for the segmentation of acrylamide contaminated areas in digital images. It is experimentally observed that the acrylamide levels in a fried potato chip can be estimated by determining the ratio of brownish yellow regions (obtained via our segmentation algorithm) to the total area of the chip image. We define this ratio as Normalized Area of the Brownish Yellow region (NABY) and linearly correlate it with the acrylamide levels of fried potatoes. The changes of acrylamide levels and NABY values are observed to follow almost the same trend during frying at 170 °C, which indicates a significant correlation between these two variables. We also demonstrate that autocovariance estimates can be incorporated as additional statistical features into the feature set because they provide a satisfactory level of discrimination.

3.1 Modeling Acrylamide Formation using Image Analysis

As explained in the Introduction part, CIE $L^*a^*b^*$ parameters used to measure non-homogenous surface color are not reliable predictors of acrylamide concentration in potato chips because the acrylamide concentrations are observed to be lower in darker regions of the potato images. Instead of seeking a linear correlation between CIE $L^*a^*b^*$ parameters and measured acrylamide concentrations, it may be advantageous to define a specific range of colors for acrylamide estimation.

Figure 1.1 shows that potato chips undergo certain color transitions as the frying proceeds. The initial pale soft yellow color of potato first turns to bright yellow, then to brownish yellow during 8 to 10 minutes of frying at 170 °C. After 10 minutes, browning in the surface becomes clearer reaching to a dark brown at the end of frying for 60 minutes. During the frying process, statistical texture and color properties of the digital photo image continuously change and different image regions appear in the given image.

Digital and analog cameras have built-in white-balancing systems modifying actual color values, therefore pixel values in an image captured by a camera of a machine vision system or a consumer camera may not correspond to true colors of imaged objects. In addition, CCD or CMOS imaging sensors of some cameras may not be calibrated during production. Nevertheless, after the frying process, one can clearly visualize three different regions (or equivalently three different kinds of pixels) in a fried potato chip image as shown in Figure 3.1(a): *a*) bright yellow (Region-1), *b*) brownish yellow (Region-2), and *c*) dark brown (Region-3). It is experimentally observed that Region-2 has a high probability of containing acrylamide. This provides us with the possibility of estimating acrylamide levels in a fried potato chip by determining the ratio of brownish yellow regions to the total area of the chip image. An automatic image analysis technique can segment pixels of a fried potato image into three sets and determine their area-wise ratios as shown in Figure 3.1(b). This idea is implemented successfully in the following section using normalized-RGB pixel values.



(a) (b)

Figure 3.1: (a) Original fried potato chip image with selected regions, and (b) Result of our segmentation algorithm described in Section 2.2

3.2 Acrylamide Analysis using CIE a^* Parameter

In this section, we demonstrate our analysis to define a specific range of colors for acrylamide estimation in CIE $L^*a^*b^*$ color space. In this study, we focus on CIE a^* parameter because previous work showed that L^* and b^* values did not show considerable changes as those shown by a^* during frying of potato chips [6].

3.2.1 Color Spaces

Long before the anatomical discovery of three color receptors (cones) in the human eye, it was proposed that color can be mathematically specified in terms of three variables and different colors can be obtained by mixing them in proper amounts. From then onwards, the idea has been studied extensively under the title of *trichromacy* and a number of important concepts have been introduced. Before moving to CIE standards, we briefly explain these concepts.

The sensation of color in human beings can be modeled as the projection of the visible region of electromagnetic spectrum onto the space spanned by three sensitivity functions. The properties of these functions are determined by the response of three types of cones present in the eye. Each cone is either sensitive to short, medium or long wavelengths such that the spectral sensitivities of these cones are linearly independent. To capture this sort of behavior, two important concepts are introduced: (a) Color Primaries, and (b) Color Matching Functions (CMF).

Color primaries are three colorimetrically independent light sources where each source is a collection of the visible electromagnetic spectra. Independence guarantees that the color of any primary cannot be visually matched by a linear combination of the remaining two primaries. These primaries are used to obtain a nonsingular linear transformation of the sensitivities of the three cones in the eye, defined as a CMF [31]. This, in turn, allows us to the represent the color of a visible spectrum in terms of *tristimulus* values (obtained via a color matching transformation) instead of actual cone sensitivity values. In order to prevent any confusion, it is necessary to specify with respect to which CMF the tristimulus values are computed. *CIE*, *International Commission on Illumination*, developed a number of standards to serve this purpose.

After careful studies on human color perception, two equivalent sets of CMFs are first defined by the *CIE* in 1931: (1) CIE *Red-Green-Blue (RGB)*, and (2) CIE *XYZ*. They are based on direct measurements of the human eye. The three monochromatic primaries used in the first set are at wavelengths of 700 nm (red), 546.1 nm (green) and 435.8 nm (blue). The second set of CMFs is obtained by a linear transformation of the CIE *RGB* CMFs and the following relation exists between the tristimulus values in *CIE XYZ* and *CIE RGB* spaces.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.488718 & 0.310680 & 0.200602 \\ 0.176204 & 0.812985 & 0.0108109 \\ 0.000000 & 0.0102048 & 0.989795 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

where $r = R^{2.2}$, $g = G^{2.2}$, $b = B^{2.2}$ and RGB values are scaled to unit range $[0.0, 1.0]$. The output values are also in the unit range.

CIE *xy chromaticity* space is derived from X,Y,Z tristimulus values in the CIE XYZ space according to the equations below:

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}$$
$$z = \frac{Z}{X+Y+Z}$$

3.2.2 Color Differences and CIE $L^*a^*b^*$ Color Space

Perceptual uniformity is an important property for a variety of industrial applications. It requires that equal perceived color differences should correspond to equal Euclidean distances in the tristimulus color space. However, the color spaces we mentioned so far are perceptually nonuniform. For this reason, special emphasis was given to develop a device-independent, perceptually uniform representation of all the colors visible to the human eye. The concept of *Just Noticeable Difference (JND)* was introduced to quantify small color changes and a distance metric based on MacAdam ellipse was used. MacAdam ellipse defines a region on a chromaticity diagram such that all colors which are indistinguishable to the average human eye from the color at the center of the ellipse are grouped together.

In an attempt to linearize the perceptibility of color differences, CIE recommended $L^*a^*b^*$ color space in 1976. In $L^*a^*b^*$ space, L^* represents the luminance of the color and ranges in the interval $[0, 100]$, 0 indicating black and 100 indicating white. As a^* changes from negative values to positive values, the position of the color moves from green to red. Similarly, b represents the position of the color between blue and yellow, negative values yielding blue and positive values yielding yellow. A reference illumination parameter, called *whitepoint* is also used to provide a crude approximation for eye's adaptation to white color under different lighting conditions.

3.2.3 Conversion from CIE XYZ to CIE $L^*a^*b^*$

CIE $L^*a^*b^*$ space is defined with the following nonlinear transformation from CIE XYZ tristimulus color space:

$$\begin{aligned} L^* &= 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b^* &= 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned}$$

where

$$f(x) = \begin{cases} x^{1/3} & x > 0.008856 \\ 7.787x + \frac{16}{116} & x \leq 0.008856 \end{cases}$$

and X_n, Y_n, Z_n are the tristimuli of the white stimulus. Under this transformation, a *JND* corresponds to an Euclidean distance of 2.3.

3.2.4 Practical Issues

An *RGB* image is described with red, green and blue pixel values but these values are not standardized and do not have precise definitions. *RGB* is not an absolute, device-independent color space like CIE XYZ or $L^*a^*b^*$, and a direct conversion formulae between *RGB* and $L^*a^*b^*$ spaces have no meaning. Therefore, an *RGB* image may look considerably different from one monitor to another.

To overcome this effect, a standard has been adopted recently by major manufacturers to characterize the behavior of an average CRT monitor.[1] All non-CRT hardware, such as LCD screens, digital cameras and printers are also built with additional circuitry or software to obey this standard. For this reason, it is in general safe to assume that an image file with 8 bits per channel is in *sRGB* space and a meaningful conversion can be defined between *sRGB* and $L^*a^*b^*$ spaces. *sRGB* values are first transformed into CIE XYZ space as follows:

[1] More information on *sRGB* is available at `http://www.w3.org/Graphics/Color/sRGB`

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412424 & 0.357579 & 0.180464 \\ 0.212656 & 0.715158 & 0.0721856 \\ 0.0193324 & 0.119193 & 0.950444 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

where

$$r = \begin{cases} R/12.92 & R \leq 0.04045 \\ ((R + 0.055)/1.055)^{2.4} & R > 0.04045 \end{cases}$$

$$g = \begin{cases} G/12.92 & G \leq 0.04045 \\ ((G + 0.055)/1.055)^{2.4} & G > 0.04045 \end{cases}$$

$$r = \begin{cases} B/12.92 & B \leq 0.04045 \\ ((B + 0.055)/1.055)^{2.4} & B > 0.04045 \end{cases}$$

Input RGB values are scaled to unit range. Output XYZ values are also in unit range $[0.0, 1.0]$. CIE XYZ values are then nonlinearly mapped into CIE $L^*a^*b^*$ space using the same method discussed above. D65 daylight illumination is used as reference white in these calculations. A more rigorous discussion about conversion operations can be found in the following references [32, 33, 34].

3.2.5 Results and Discussion

Gokmen et al. measured CIE a^* values for potato chips shown in Figure 3.2 using a Minolta CM-3600d model spectrophotometer. Potato chips were fried at 170 °C with sampling at $1, 3, 5, 8, 10, 15, 30$ and 60 minutes. The results are shown in

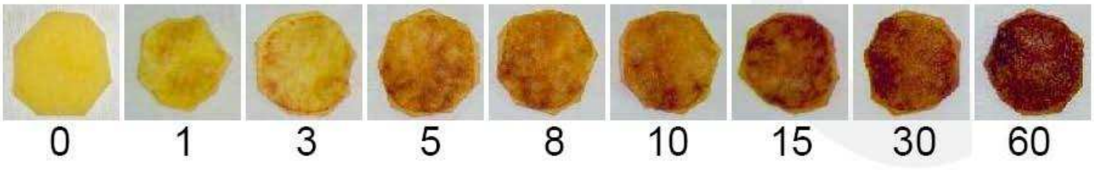


Figure 3.2: Potato chip images used for acrylamide analysis aligned according to frying time

Table 3.1: Measured acrylamide concentration, measured and estimated CIE a^* values for potato chips

t, min	AA, ng/g	Measured CIE a^*	Estimated CIE a^*
1	582	1.96	5.013
3	2554	5.35	8.6375
5	9519	11.73	20.6559
8	10963	12.47	22.6269
10	10500	12.78	23.0435
15	8198	12.92	27.7742
30	5119	13.86	30.6953
60	4987	13.55	30.9508

Table 3.1. From these results, it is not possible to define a specific range of CIE a^* values for acrylamide estimation because a^* values are very close to each other for brownish yellow and dark brown colored potatoes.

We estimate average CIE a^* values from RGB images of fried potato chips following the formulation described in the previous sections. Using Matlab 7.0 built-in functions for color conversions (*makecform,applycform*), potato chip images (See Figure 3.2) are transformed from RGB into CIE $L^*a^*b^*$ color space. An average CIE a^* value is calculated for each potato chip image by taking the mean of all extracted a^* values. As shown in Figure 3.3 and Table 3.1, high acrylamide concentrations are observed for intermediate values of a^* parameter. Lower values of a^* indicate a decrease in measured acrylamide concentration. However, higher values of a^* do not indicate such a clear decrease and defining a specific range responsible for acrylamide formation from estimated CIE a^* values is not possible, either. Hence, we turn our attention to a new set of features explained in the following section.

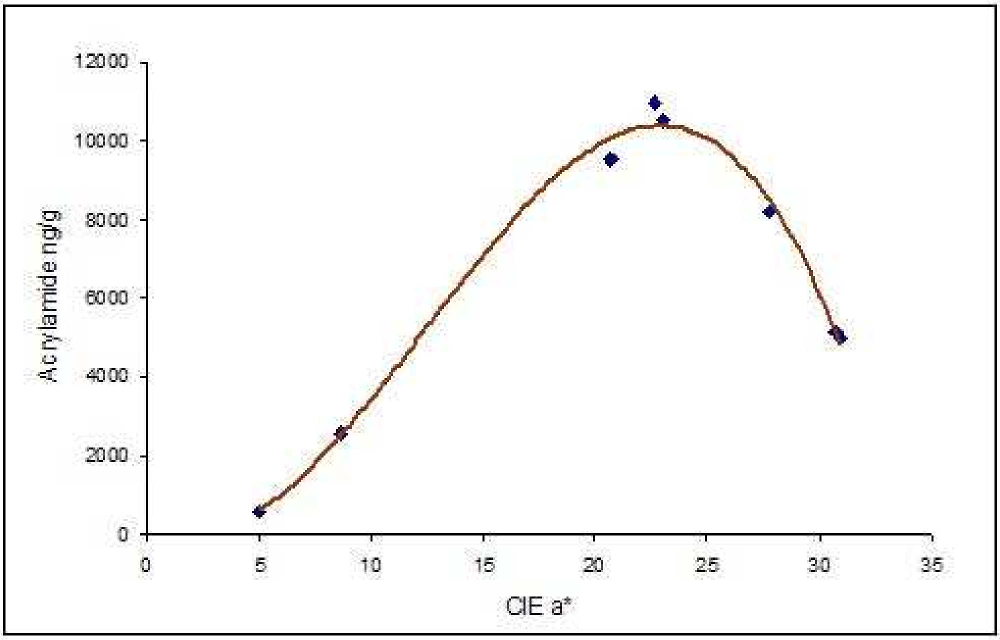


Figure 3.3: Change of acrylamide concentration and estimated CIE redness parameter a^* in potato chips during frying at 170 °C

3.3 K-means Clustering based Segmentation for Acrylamide Analysis in Potato Chip Images

As mentioned in Chapter 3.1, the segmentation of fried potato images into three regions can provide us the necessary information to estimate acrylamide levels. Several state-of-the-art image segmentation and pattern classification algorithms are available in the literature each having its own advantages and disadvantages. For a complete discussion of these algorithms, the reader may refer to any of the following references [35, 36, 37, 19, 38].

In this section, we analyze digital color images of fried potatoes to estimate acrylamide levels using the method developed in the previous chapter. We show that acrylamide levels in a fried potato image can be estimated by determining the ratio of brownish yellow regions to the total area of a given potato chip, abbreviated as NABY. Three different regions corresponding to bright yellow, brownish yellow and dark brown are extracted using our k-means based classifier and their corresponding area-wise ratio is calculated using the segmentation results obtained from the classifier.

3.3.1 Selected Features

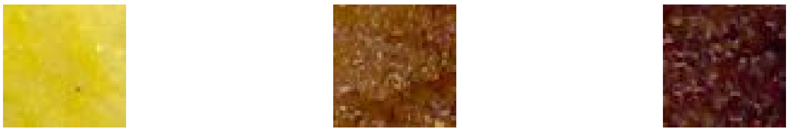
A typical image captured by a digital camera consists of an array of vectors called pixels. Each pixel $x[n,m]$ has red, green and blue color values:

$$x[n,m] = \begin{bmatrix} x_r(n,m) \\ x_g(n,m) \\ x_b(n,m) \end{bmatrix}$$

where $x_r(n,m)$, $x_g(n,m)$ and $x_b(n,m)$ are the values of red, green and blue components of the $(n,m)^{\text{th}}$ pixel $x[n,m]$, respectively. In digital images, x_r, x_g and x_b color components are represented in 8 bits, i.e., they are allowed to take integer values between 0 and $255(= 2^8 - 1)$ [37]. Digital and analog cameras have built-in white balancing systems modifying actual color values, therefore pixel values in an image captured by a camera of a machine vision system or a consumer camera may not correspond to true colors of imaged objects. In addition, CCD or CMOS imaging sensors of some cameras may not be calibrated during production. To reduce such variations due to lighting conditions and white-balancing scheme of digital cameras, normalized image pixel color values are used as features for our classification system.

Figure 3.5 plots the distribution of the normalized color values obtained from the regions shown in Figure 3.4. From this figure, we deduce that a set of features containing only normalized color values can be enough to provide a reasonable separation of the dataset using our k-means based classifier. They are computed as follows:

$$\overline{x}_r(n,m) = \frac{x_r(n,m)}{x_r(n,m)+x_g(n,m)+x_b(n,m)}$$
$$\overline{x}_g(n,m) = \frac{x_g(n,m)}{x_r(n,m)+x_g(n,m)+x_b(n,m)}$$
$$\overline{x}_b(n,m) = \frac{x_b(n,m)}{x_r(n,m)+x_g(n,m)+x_b(n,m)}$$



(a) Bright Yellow (b) Brownish Yellow (c) Dark Brown

Figure 3.4: Potato regions used in feature distribution and autocovariance estimation plots (a) Region 1, (b) Region 2, and (c) Region 3

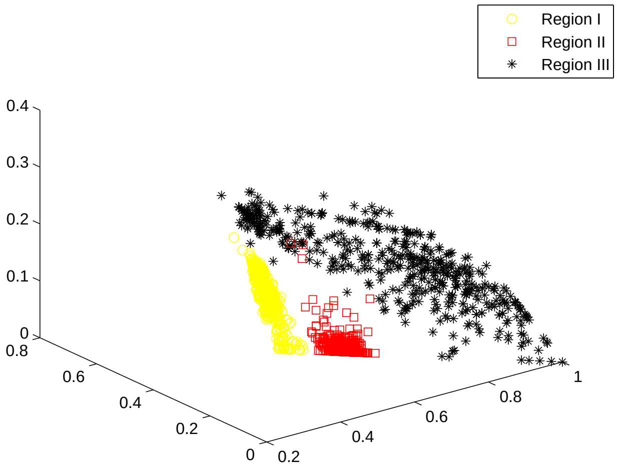


Figure 3.5: Distribution of the features used for acrylamide level estimation in normalized RGB space.

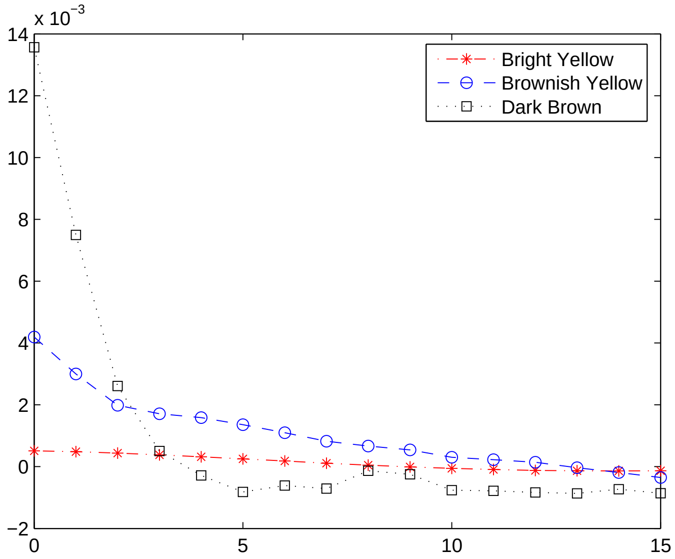


Figure 3.6: Row-wise unbiased autocovariance estimates from normalized red pixels

In this study, we also demonstrate another set of useful features that can further increase the classification accuracy of our system by incorporating statistical properties of three different colored regions. Unbiased estimates of the autocovariance values for bright yellow, brownish yellow and dark brown regions are obtained in windows of size $(N \times M)$ horizontally and vertically, using the following formulas:

$$c_{H,i}(k) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-k-1} \left(\overline{x}_i(n,m) - \frac{1}{M} \sum_{l=0}^{M-1} \overline{x}_i(n,l) \right)$$

$$\times \left(\overline{x}_i(n,m+k) - \frac{1}{M} \sum_{l=0}^{M-1} \overline{x}_i(n,l) \right)$$

$$c_{V,i}(k) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-k-1} \left(\overline{x}_i(n,m) - \frac{1}{N} \sum_{l=0}^{N-1} \overline{x}_i(l,m) \right)$$

$$\times \left(\overline{x}_i(n+k,m) - \frac{1}{N} \sum_{l=0}^{N-1} \overline{x}_i(l,m) \right)$$

where $i = r, g, b$; $k = 0, 1, 2, \ldots$; N and M are the row and column number of pixels over which estimation is carried out. Figure 3.6 plots the corresponding

Table 3.2: Measured acrylamide concentration and estimated NABY value for potato chips

t,min	AA,ng/g	NABY
0	0	0
1	582	0.0485
3	2554	0.2249
5	9519	0.9147
8	10963	0.9503
10	10500	0.9209
15	8198	0.7574
30	5119	0.5539
60	4987	0.4136

autocovariance values estimated from the red pixels of (50 × 50) regions shown in Figure 3.4. The first two autocovariance values, $c_{H,r}(0)$ and $c_{H,r}(1)$ are very different from each other in Region 1, Region 2 and Region 3. Hence, they can be included into our feature vector as additional elements and provide a better segmentation of the potato images for acrylamide analysis.

3.3.2 Classification Results

The proposed acrylamide estimation method was implemented using Matlab programming environment and tested on a set of images containing potato chips fried at 170 °C with sampling at $1, 3, 5, 8, 10, 15, 30$ and 60 minutes (See Figure 3.2). Prior to feature extraction and segmentation, potato chip images are convolved with a $[5 \times 5]$ median filter to remove oil sparks. A morphological erosion operation is applied at the boundaries of potato chips to remove shadowing effects.

As illustrated in Figure 3.7 and Table 3.2, changes of acrylamide levels and NABY values follow approximately the same trend during the frying of potato chips at 170 °C. The results indicate a linear regression coefficient as high as 0.9868. The correlation between measured acrylamide levels and NABY values is demonstrated in Figure 3.8.

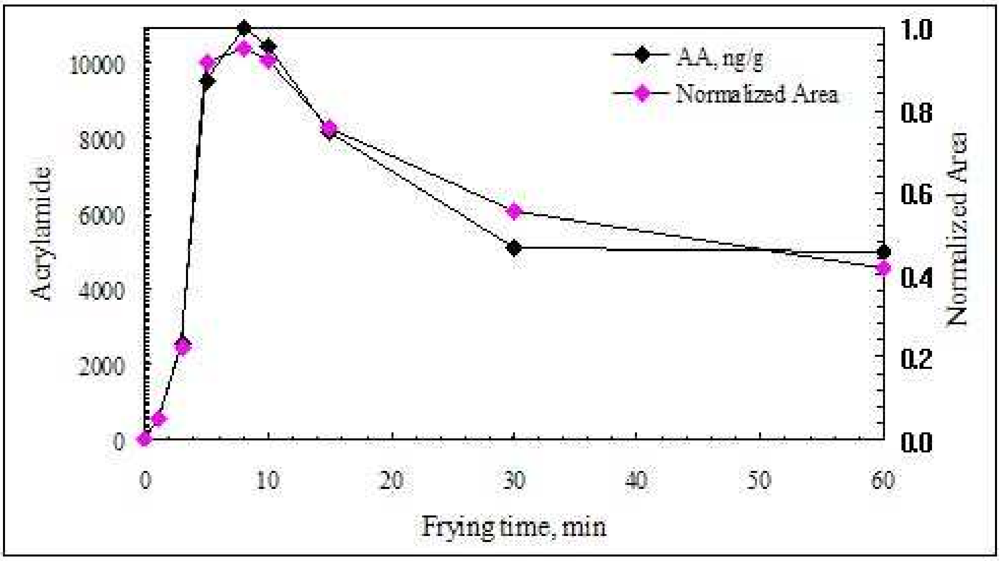


Figure 3.7: Change of acrylamide level and NABY value in potato chips during frying at 170 °C.

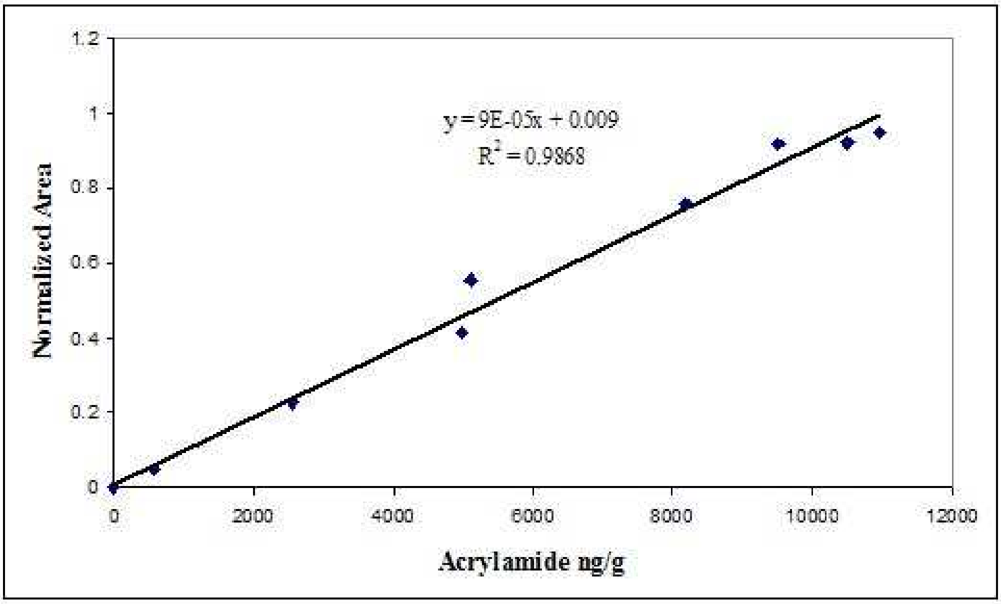


Figure 3.8: Correlation between acrylamide level and NABY value in potato chips fried at 170 °C.

Chapter 4

Image Analysis of Coffee Beans for Acrylamide Formation

Coffee is a highly consumed beverage in many countries. Significant levels of acrylamide can be present in coffee due to roasting of coffee beans during manufacturing process. The effect of various roasting conditions on acrylamide formation and color changes was analyzed by the following researchers [39, 40, 41, 42].

Motivated with the results obtained from potato chip images, same sort of analysis is carried out on a set of green coffee images. These images correspond to the same coffee samples previously studied by Gokmen et al. [39]. Coffee samples are roasted at $150, 200$ and 225°C with sampling at $5, 10, 15, 20$ and 30 minutes (See Figure 4.1). Gokmen et al. reported that the amount of acrylamide measured increased rapidly at the onset of roasting, reaching an apparent maximum, and then decreasing exponentially as the rate of degradation exceeded the rate of formation at 200 and 225 °C. However, the amount of acrylamide measured continued to increase during roasting at 150 °C.

Gokmen et al. also showed that dark colored coffee may contain much lower amounts of acrylamide than light colored coffee. This is consistent with the results of Chapter 3 in the sense that acrylamide concentration decreases at the later stages of cooking. Using these observations, pixels of coffee images are classified

Table 4.1: Low-pass filter coefficients

h[0]	**h[1]**	**h[2]**	**h[3]**	**h[4]**	**h[5]**	**h[6]**	**h[7]**	**h[8]**	**h[9]**	**h[10]**
0.0378	0.0141	-0.0035	-0.0294	-0.0488	-0.0437	-0.0030	0.0694	0.1538	0.2215	0.2473
h[11]	**h[12]**	**h[13]**	**h[14]**	**h[15]**	**h[16]**	**h[17]**	**h[18]**	**h[19]**	**h[20]**	–
0.2215	0.1538	0.0694	-0.0030	-0.0437	-0.0488	-0.0294	-0.0035	0.0141	0.0378	-

into 4 regions. First 3 regions are the same as the ones proposed in the previous chapter. A fourth region is added to distinguish dark colored coffee grains from discontinuity parts between the boundaries of the coffee grains. NABY values are again calculated using only first 3 regions. Prior to feature extraction, coffee images are also convolved with a 21-tap low-pass FIR filter to further reduce boundary effects. The filter is designed with a cut-off frequency of $\pi/4$ using the Remez routine in Matlab [30] (See Table 4.1 and Figure 4.2). Results are shown in Tables 4.2, 4.3 and 4.4, and Figures 4.3 and 4.4.

These results indicate that NABY value can be an approximate predictor of acrylamide level for roasted coffee beans. However, it should also be noted that experimentation with a large database of roasted coffee images is necessary to establish more accurate relations.

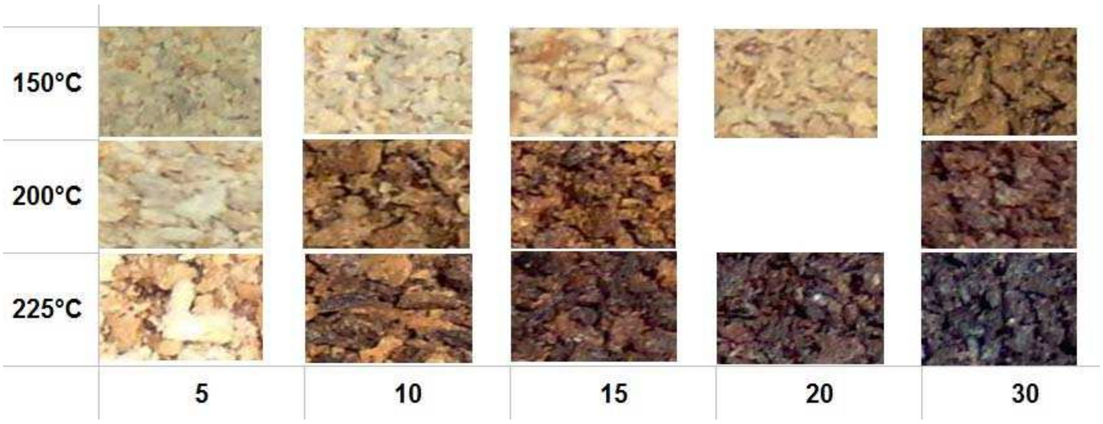


Figure 4.1: Coffee images used for acrylamide analysis aligned according to temperature (vertically) and frying time (horizontally)

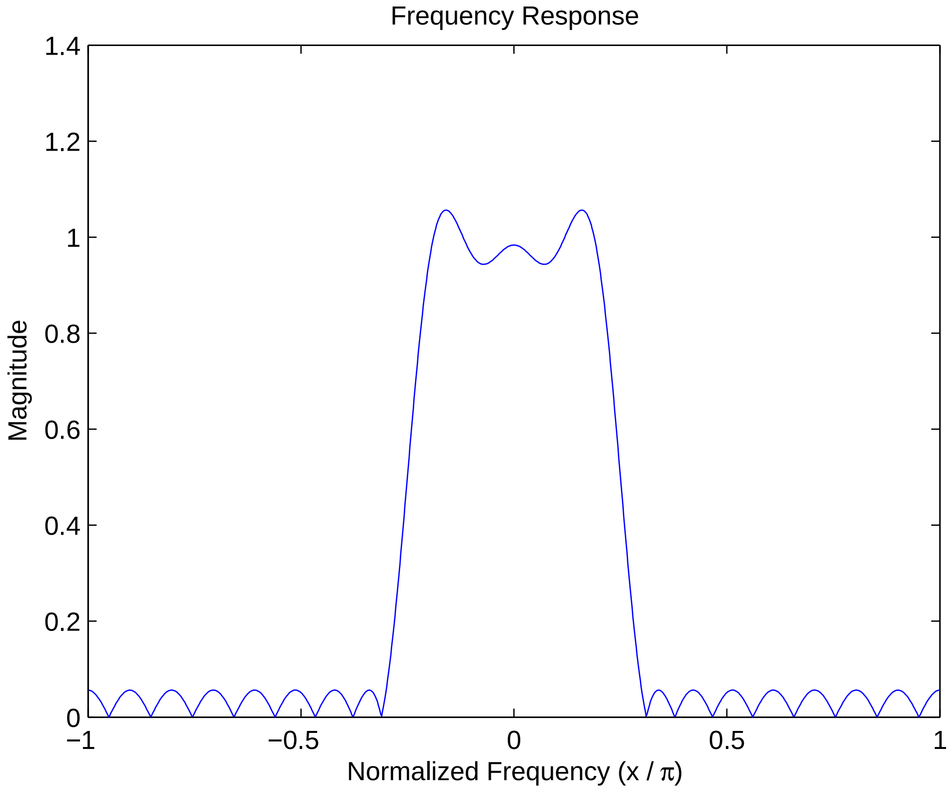


Figure 4.2: Frequency response of the low-pass filter used in pre-processing

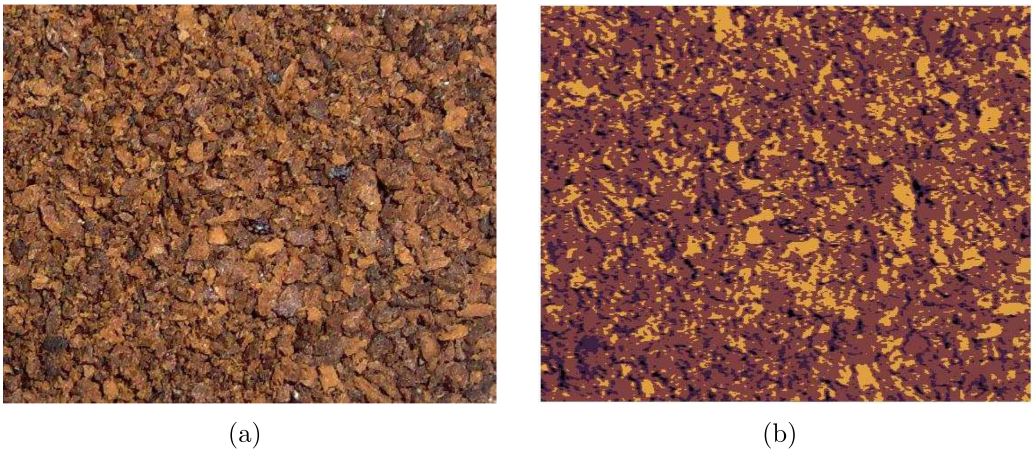
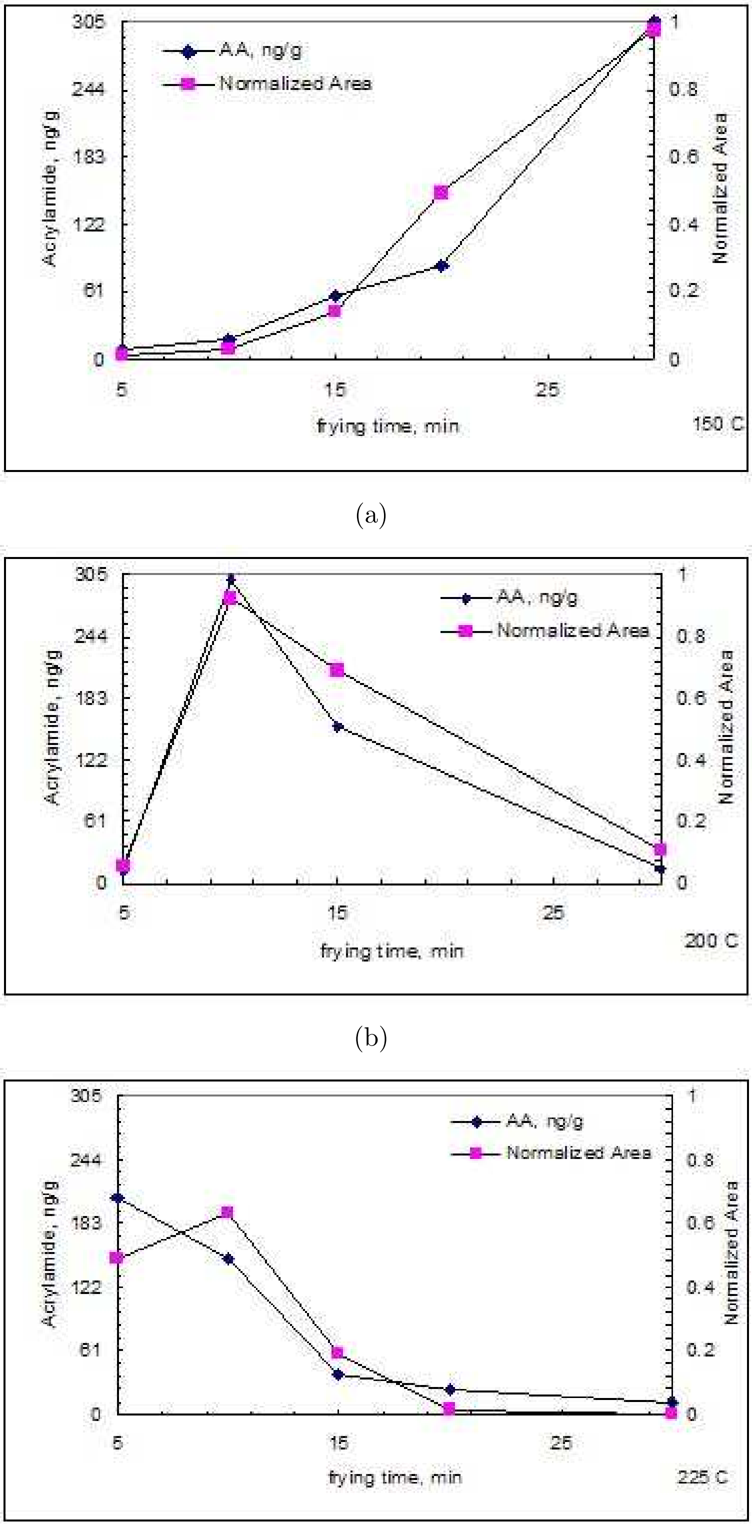


Figure 4.3: (a) Original coffee image roasted at 200 °C for 15 minutes, and (b) Segmented coffee image



(a)

(b)

(c)

Figure 4.4: Change of acrylamide level and NABY value in coffee during roasting at (a) 150, (b) 200 and (c) 225 °C.

Table 4.2: Measured acrylamide concentration and estimated NABY value for coffee roasted at 150 °C

t,min	AA,ng/g	NABY
5	8	0.0134
10	18	0.0292
15	57	0.1422
20	85	0.4930
30	305	0.9780

Table 4.3: Measured acrylamide concentration and estimated NABY value for coffee roasted at 200 °C

t,min	AA,ng/g	NABY
5	13	0.0584
10	300	0.9225
15	155	0.6890
30	15	0.1075

Table 4.4: Measured acrylamide concentration and estimated NABY value for coffee roasted at 225 °C

t,min	AA,ng/g	NABY
5	208	0.4892
10	150	0.6326
15	38	0.1886
20	23	0.0147
30	12	0.0001

Chapter 5

A Feature Domain Post-processing Method to Increase Performance for Hazelnut Classification

In this chapter, a feature domain post-processing method is developed to increase performance in the separation of empty hazelnuts from fully developed nuts by impact acoustics. The use of signal processing techniques for the detection of empty hazelnuts from fully developed nuts is investigated in [16]. It is observed that the classification accuracy can be further increased by applying our method on the features extracted from impact sounds of hazelnuts before feeding them into the classifier. The idea is inspired from the well-known median filtering approach, which is mainly used to reduce noise due to outliers while preserving useful detail in an image [43]. In addition to median filtering based post-processing, the results of an averaging filter are also examined.

The motivation behind empty hazelnut detection is discussed in the Introduction part. Here, we follow by a summary of the proposed solution presented in our paper [16]. It is later shown that a better training set can be obtained by removing outliers in the feature space with the help of median and mean post-processing methods. With an appropriate selection of the corresponding method parameters, a more accurate model for the dataset can be obtained, thus leading to a better classification performance.

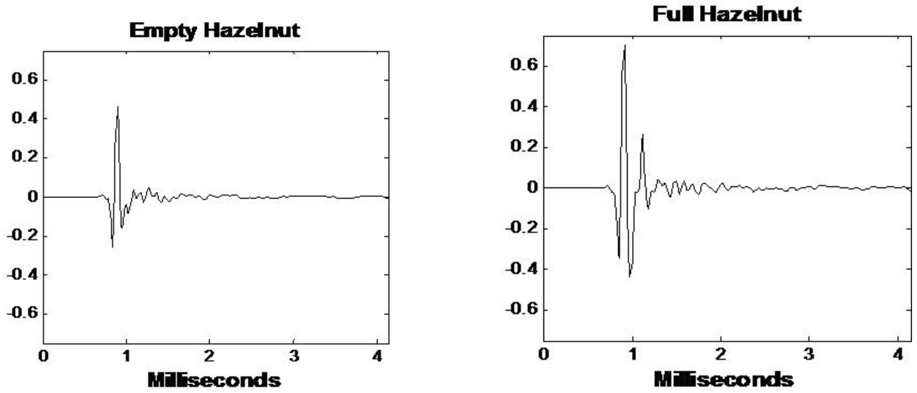
5.1 Experimental Setup and Dataset

Since the aim is to assess the applicability of proposed signal processing algorithms, mechanical part of the setup discussed in Section 1.2.2 is simplified into an impact plate, a chute through which hazelnuts are dropped and a microphone which is sensitive to frequencies up to 20kHz. Meanwhile, signal processor part is fully preserved. A heavy polished block of stainless steel ($7.5 \times 15 \times 2$ cm^3) is chosen as impact surface to minimize the interference from internal vibrations of the plate.

In this study, dataset is composed of the features extracted from impact sounds of 'Levant' type hazelnuts from Akçakoca, Düzce region of Turkey. There are a total of 492 impact sounds obtained from 231 empty and 261 full hazelnuts.

5.2 Signal Processing

Features are extracted from the recorded impact sounds of empty and full hazelnuts. Subsequently, test and training sets are constructed by randomly dividing each group into two halves. Mean and median filtering based post-processing is applied on the feature space spanned by the training data. Lastly the classification is performed using Support Vector Machines [28].



(a) Empty Hazelnuts

(b) Full Hazelnuts

Figure 5.1: Typical impact sound signals from an empty hazelnut and a full hazelnut. The extremum of a full hazelnut is usually higher than an empty hazelnut.

5.2.1 Feature Extraction

5.2.1.1 Time Domain Signal Modeling

Figure 5.1 shows example time domain signals for empty and full hazelnuts. In order to fully capture the differences between two waveforms, a smoothed envelope of each signal (from which Weibull function parameters and coefficient of multiple regression[1] are estimated) is computed as follows:

1. rectify the signal by taking the absolute value at all points,
2. non-linearly filter the signal by replacing the center data point with the maximum value in a 7-point window,
3. estimate the four parameters of the Weibull function, given by the following equation:

[1] Coefficient of multiple regression(R^2): A statistic that measures how successful the fit is in explaining the variation of the data.

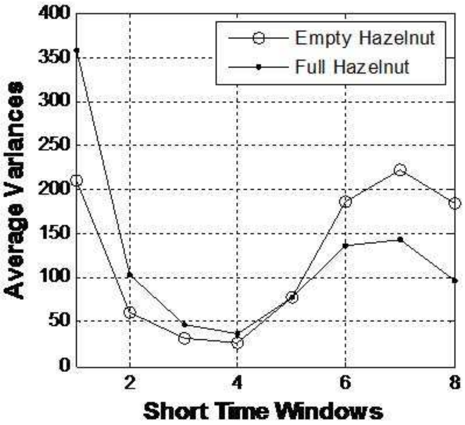


Figure 5.2: Average variances from short time windows of time domain signals

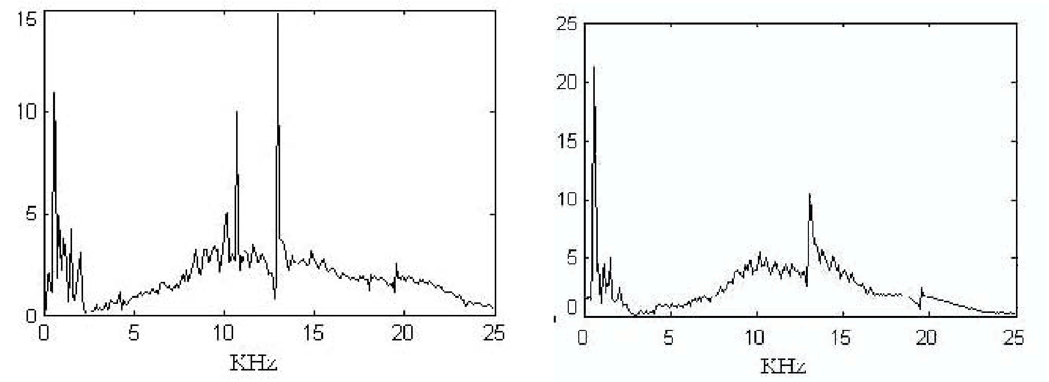
$$Y(t) = \begin{cases} \frac{bc}{a}\left[\frac{(t-t_0)}{a}\right]^{(b-1)}\left\{e^{-\left[\frac{(t-t_0)}{a}\right]^{b}}\right\} & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases}$$

5.2.1.2 Short Time Variances in Frames of Data

In addition to modeling global behavior of the impact signal with Weibull function, local time domain variations are captured by computing variances in short time windows. Short time windows are 50 samples in duration and each windows overlaps with the previous and next window by 20 samples. A total of 8 short time windows are used to compute variances and the first window begins 40 samples before the sample location corresponding to the maximum amplitude. After all variances are calculated, they are normalized by the sum of all 8 variances as follows:

$$\sigma_{ni}^2 = \frac{\sigma_i^2}{\sum_{i=1}^{8} \sigma_i^2}$$

where σ_{ni}^2 and σ_i^2 are the normalized and computed variances from window i with $i = 1$ being the first and $i = 8$ being the last. This method captures the increased duration of signals from empty hazelnuts in the last three windows as shown in Figure 5.2.



(a) Empty Hazelnuts

(b) Full Hazelnuts

Figure 5.3: Example frequency spectra magnitudes for empty and full hazelnuts

5.2.1.3 Extrema in Short Time Windows

Beginning from the 30$^{\text{th}}$ sample, time domain signal is divided into 11 non-overlapping windows, each having a size of 15 samples. The extremum value of each window is selected as a feature value.

5.2.1.4 Frequency Domain Processing

Beginning from 80 samples before the signal maximum slope, a 256-point DFT (Discrete Fourier Transform) is computed using a Hamming window for each impact sound. Magnitude of the computed spectra is then low-pass filtered using a 20-tap FIR filter with cut-off frequency equal to $\pi/4$ in the normalized DFT domain to remove jagged spikes. Then the frequency corresponding to the peak magnitude in the spectra is saved as a discriminating feature. In addition, 15 magnitude values before and after the peak are also preserved after being normalized by the peak magnitude. Figure 5.3 shows the corresponding spectra for empty and full hazelnuts.

5.2.1.5 Line Spectral Frequencies

Linear predictive modeling techniques are widely used in various speech coding, synthesis and recognition applications [44, 45]. Linear Minimum Mean Square Error prediction based data analysis is equivalent to Auto-Regressive modeling of the data. Line Spectral Frequency (LSF) representation of Linear Prediction (LP) filter was introduced by Itakura [46] and extensively used in GSM and MELP speech coding systems.

In LMMSE analysis, it is assumed that the sound data can be modeled using an *m-th* order linear predictor, i.e., $x_p[n] = a_1x[n-1]+a_2x[n-2]+\ldots+a_mx[n-m]$ where $x[n-k]$ is the sound sample at time instant $(n-k)T_s$ and $x_p[n]$ is the estimated sound sample at time instant nT_s (T_s is the sampling period). Let the prediction error filter $\Lambda_m(z)$,

$$\Lambda_m(z) = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \ldots + \alpha_m z^{-m}$$

be obtained by LP analysis of the impact sound, ($\alpha_i = -a_i$). The corresponding all-pole synthesis filter is $1/\Lambda_m(z)$. A minimum phase prediction error filter (i.e., one with all its roots within the unit circle) has a corresponding synthesis filter which is stable. The LSF polynomials $P(z)$ and $Q(z)$ are formed as follows:

$$P(z) = \Lambda_m(z) + z^{-(m+1)}\Lambda_m(z^{-1})$$

$$Q(z) = \Lambda_m(z) - z^{-(m+1)}\Lambda_m(z^{-1})$$

The roots of these two auxiliary polynomials determine Line Spectral Frequencies. It is shown in [47] that if $\Lambda_m(z)$ is minimum phase, then

- the roots of $P(z)$ and $Q(z)$ are on the unit circle, and
- the roots are interlaced.

If the underlying process is truly Auto-Regressive, phase angles of LSFs concentrate around spectrum peaks. Thus, they provide a compact way of representing the spectrum of the impact sound under AR assumption.

5.3 Mean and Median Filtering Based Post-Processing

In classification problems, it is usually the case that dataset is corrupted with outliers due to various sources of noise sneaking into the system during data acquisition process. In a system where model parameters are estimated from a subset of the dataset (called training set), same sort of noise is also inherited. This may have a negative effect on the system performance when the model is evaluated on the test set.

Another problem in classification occurs when the samples belonging to different classes are not completely separable. Sometimes, part of the feature space is heavily polluted by samples of different classes simultaneously (occlusion). This, in turn, increases the complexity of the classifier considerably for just a small gain in the system performance. In other cases, the increase in the complexity may not even provide any improvement. Under such circumstances, it may be desirable to modify training samples locally in small groups without disturbing their overall distribution noticeably. It may be possible to give sample points a more organized look in finer scale within the areas of occlusion, and hence, increase the separability among the sample points belonging to different classes.

In this section, a remedy based on mean and median filtering ideas is proposed to overcome these problems. In the following sections, corresponding results for hazelnut dataset are compared with the previously obtained results [16].

5.3.1 Algorithm

Given a sample point and values for the parameters r and T,

- a hypersphere of radius r is drawn such that the sample point lies in the center of the hypersphere,

- if the number of points inside the hypersphere belonging to the same class as the sample point exceed T, the sample point is replaced with either the mean or median of those points; otherwise the sample point is deleted from the set.

This algorithm is repeated for all sample points in the dataset. Since a hypersphere is used, it is crucial to normalize each feature to have zero mean and unit variance before running the algorithm. The coefficients used in normalization must be preserved in order to apply the same linear scaling to test set before the samples are sent to the classifier.

As mentioned earlier, selection of r parameter is important. If it is too large, filtered output will be a very trivial set and most of the information contained inside the training set will be lost. If it is too small, a lot of points will be classified as outliers and this destroys the benefits that we expect from mean/median filtering. It is usually helpful to investigate the distribution of the data in places where occlusion occurs and decide the values of T and r parameters accordingly. As a consequence, it may become a necessity to select different values of r for different classes whenever the variability of classes are not close to each other.

Filtering can be carried out in two ways:

Mean Filtering: Output sample point is obtained by taking the average of all sample points inside the hypersphere that belong to the same class as the selected sample point,

Vector Median Filtering: Output sample point is chosen as the sample point whose sum of Euclidean distances to other points belonging to the same class is a minimum.

Table 5.1: Average classification results obtained for banana shaped classes

	Success Percentages		
	`Mean`	`Median`	`Without filtering`
Back propagation NN with 5 hidden units	98.5	98.6	98.3
K-nearest neighbor ($k = 5$)	98.0	98.0	97.9
GMM with 3 mixtures	98.5	98.4	98.3
SVM with Radial Basis Function	98.2	98.3	98.3

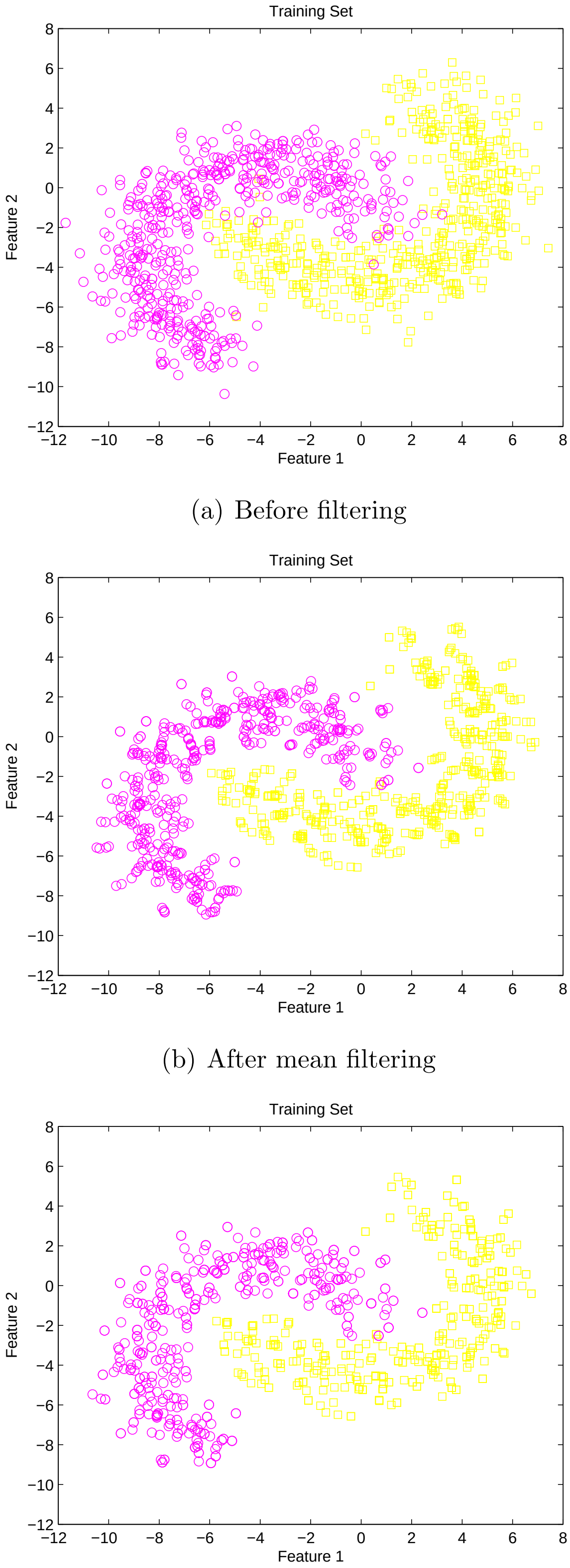
5.3.2 Performance Analysis

In order to assess the validity of our approach, described filtering algorithm is applied to an artificially generated dataset of banana-shaped classes similar to the one used in Chapter 2 (Banana Set 1). Each one of the two classes consists of 1000 samples. Classification is repeated 5 times by randomly reordering training and test set features, and final results are taken to be the average of these 5 experiments. Filtering parameters are chosen empirically as $r = 0.54$, $T = 1$ for both classes and same values are used for all classifiers. During the filtering process, 24.0 and 21.7 samples are discarded on the average from training set of each banana class. Due to randomization and averaging, non-filtered results differ slightly with respect to previous findings. Corresponding results are tabulated in Table 5.1. Effect of filtering on the training set is demonstrated on Figure 5.4.

5.4 Support Vector Machine Classifier

Support Vector Machine classifiers operate on the principle of defining a linear boundary between classes such that the margin of separation between samples from different classes that lie next to each other is maximized [48]. Support vectors lie on the margin and carry all the relevant information about the classification problem.

This approach is generalized to non-linear case by mapping the original feature space into some other space using a mapping function and performing optimal



(a) Before filtering

(b) After mean filtering

(c) After median filtering

Figure 5.4: Effect of filtering on training set (a) before filtering, (b) after mean filtering, (c) after median filtering

hyperplane algorithm in this dimensionally increased space. In the original feature space, the hyperplane corresponds to a non-linear decision function whose form is determined by the mapping kernel.

Results presented in the next chapter for hazelnut classification are obtained using a two-class SVM classifier with radial basis function (RBF) as kernel. RBF kernels are computed according to the formula:

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$$

LIBSVM package provides the necessary quadratic programming routines to carry out classification [28]. It also normalizes each feature by linearly scaling it to the range $[-1, 1]$.

5.5 Classification and Comparison of Results

In this section, classification results obtained by using all extracted features are tabulated for the detection of empty hazelnuts from fully developed nuts. It is also shown that these results can be further increased by employing post-processing techniques explained in Section 5.3.1. In order to eliminate noise effects, SVM classification is repeated 5 times by shuffling feature vectors. The final results are taken to be the average of these 5 experiments.

During the filtering process, 19.8 and 16.6 samples are discarded on the average from training sets of empty and full hazelnuts, respectively. r parameter is set to different values for each class such that 5.5 sample points of the same class lie in each hypersphere on the average. If there is no sample in the hypersphere except the point in the center, that point is deleted ($T = 1$). As depicted in Table 5.2, overall classification performance is increased with the filtering approach. The gain in system performance is greater for the case of mean filtering.

Table 5.2: Average classification results obtained with/without mean/median filtering based post-processing by using all features

	Success Percentages		
	`Mean`	`Median`	`Without filtering`
Empty	97.8	95.9	96.5
Full	96.6	97.4	96.5
Overall	97.2	96.7	96.5

Chapter 6

Conclusions

In this work, we present signal and image processing algorithms for two specific agricultural applications: (a) Estimation of acrylamide levels in fried potato chips and roasted coffee beans using digital images, and (b) detection of empty hazelnuts from fully developed nuts using impact acoustics.

During the frying of potato chips, statistical texture and color properties of the corresponding digital photo image continuously change and different image regions appear. However, it is difficult to establish a direct correlation between CIE a^* parameter and measured acrylamide concentration. After the frying process, three different regions corresponding to bright yellow, brownish yellow and dark brown areas become visible in a given potato chip image. It is experimentally observed that brownish yellow pixels have a high probability of containing acrylamide. For this reason, the ratio of brownish yellow regions to the total area of the chip image is chosen as an estimator of the acrylamide level in fried potato chips. To reduce illumination effects, normalized-RGB color values are selected as features. Using our method, we segment each potato chip image into three corresponding regions. Results indicate that a linear regression coefficient as high as 0.9868 is achieved for potato chips fried at 170 °C. A similar image analysis method is applied to the images of roasted coffee beans and satisfactory results are also obtained. In this part, we also show that autocovariance estimates can be incorporated as additional statistical features to predict acrylamide levels in potato chips.

There is currently little information about, and poor understanding of, how acrylamide forms in foods. But it is known that acrylamide forms as an intermediate product during frying and its concentration begins to decrease as the rate of degradation exceeds the rate of formation during heating. In our analysis, we assume that formation of dark colored regions on the surface of the potatoes corresponds to this fact. This assumption has proven its validity throughout our experiments.

In order to estimate acrylamide levels from a given potato chip or coffee image, an automatic classification method based on the classical k-means algorithm is proposed. This supervised method consists of training and testing stages, and proceeds as follows: Given a dataset with a number of classes in arbitrary dimension, the whole dataset is first partitioned into k distinct clusters by running the classical k-means algorithm until convergence. Inside each cluster, a representative vector is calculated for each class by averaging the sample points of that class assigned to the specified cluster.

This simple approach may fail to perform efficiently due to the large degree of variability in real-world datasets. This is because all the representative vectors calculated as described above may not possess the same degree of importance in terms of their contribution to classification performance and unnecessary centroids may be generated due to noise present in the dataset. For this reason, two threshold parameters are introduced to decide which representative vectors should be kept at the end. The first threshold requires that the ratio of the sample points belonging to each class inside a cluster must exceed some predetermined value. The second thresholds aims at preventing the calculation of class centroids inside clusters with too few sample points. However, these conditions may sometimes become too restrictive. To compensate for such an effect, all class centroids discarded by user-determined thresholds are kept internally and presented repeatedly under certain arrangements to see if they help to increase the recognition accuracy. Using this approach step by step, a number of representative vectors are estimated for each class from the training set. In the test set, the classification is performed by assigning the label of the class centroid that is closest to the test sample.

The effect of each parameter on classification performance is analyzed using Lithuanian classes. The applicability of our approach is tested on both Lithuanian and banana-shaped classes. The results indicate comparable performance with some other state-of-the-art classification techniques such as Gaussian Mixture Modeling, Support Vector Machines, Back-Propagation Neural Networks and K-Nearest Neighbors. The proposed method is generic in the sense that it can be applied to any classification dataset without much modification.

Information criterion techniques are applied to obtain an initial estimate of the number of clusters present in the feature space at the beginning of the program. This is useful to check whether our assumption about three different regions existing in a potato image is valid. However, information scores follow a monotonically decreasing behavior within the range of acceptable k values – pointing to larger values of k as optimal. This is mainly attributed to the inflexible 'identical spherical Gaussian assumption' used in model selection. The data coming from three main regions of a potato chip image is highly non-Gaussian. This forces the model selection process to move towards larger values of k by modeling non-Gaussian clusters using a number of smaller identical-spherical Gaussian clusters. Hence, the density of the distribution is modeled better but resulting estimate for the number of clusters does not make any sense for our purposes.

Our k-means based image analysis system seems to be a promising approach for the prediction of acrylamide levels in fried potatoes and roasted coffee beans. A linear regression equation obtained from a correlation curve, similar to one that is plotted in Figure 3.8, can be used for this prediction. Since higher NABY values indicate higher acrylamide levels, products exceeding a predefined critical value of NABY may be simply sorted out in a processing line based on this principle. In such systems, cameras can be installed in the packaging lines and digital images can be analyzed in real-time and those products with high NABY values can be removed. For example, if a provisional maximum permitted concentration of acrylamide in the finished product is established, the fried potatoes or roasted coffee beans exceeding the corresponding NABY value (to be obtained from the linear regression plot) can be removed by the machine vision system. It should be noted here that the calibration curve presented here should be modified according to the results that will be obtained for a wide range of potato/coffee cultivars and frying/roasting conditions.

Another contribution of this thesis is the development of a feature domain post-processing method to increase performance for hazelnut classification. A prototype system was previously proposed to detect empty hazelnuts using impact acoustics. In that study, a number of feature vectors describing time and frequency nature of the impact sounds were extracted from the acoustic signals and classified using support vector machines. We show that a better training set can be obtained by filtering the data and removing outliers in the feature space with an appropriate selection of the filtering parameters. Vector median and mean filtering techniques are used in the post-processing step and the resulting training features are fed into the SVM classifier. The results indicate a slight gain in the system performance. The validity of the approach is assessed by applying the method to artificially generated banana-shaped classes.

As noted earlier, the performance of the proposed filtering approach is completely determined by a particular choice of filter parameters. Therefore, it is necessary to examine the distribution of the data in the training set in order to obtain a promising set of filter parameters. After deciding on which samples should be considered as outliers, a value for the radius of the hypersphere can be determined by looking at the average distance of the points considered as outliers to the nearby points considered as representatives of their class.

Bibliography

[1] D. S. Mottram, B. L. Wedzicha, and A. T. Dodson, “Acrylamide is formed in the maiilard reaction,” *Nature*, no. 419, pp. 448–449, 2002.

[2] R. H. Stadler, I. Blank, N. Varga, F. Robert, J. Hau, P. A. Guy, M. C. Robert, and S. Riediker, “Acrylamide from maillard reaction products,” *Nature*, no. 419, pp. 449–450, 2002.

[3] M. Friedman, “Chemistry, biochemistry and safety of acrylamide,” *Journal of Agricultural and Food Chemistry*, vol. 51, pp. 4504–4526, 2003.

[4] V. A. Yaylayan, A. Wnorowski, and C. P. Locas, “Why asparagine needs carbohyrades to generate acrylamide,” *Journal of Agricultural and Food Chemistry*, vol. 51, pp. 1753–1757, 2003.

[5] G. Márquez and M. C. Aňón, “Influence of reducing sugars and amino acids in the color development of fried potatoes,” *Journal of Food Science*, vol. 51, pp. 157–160, 1986.

[6] F. Pedreci, P. Moyano, P. Kaack, and K. Granby, “Color changes and acrylamide formation in fried potato slices,” *Food Research International*, vol. 38, pp. 1–9, 2005.

[7] V. Gökmen, H. Z. Şenyuva, J. Acar, and K. Sarıoğlu, “Determination of acrylamide in potato chips and crisps by high-performance liquid chromatography,” *Journal of Chromatography A*, 2005. (doi:10.1016/j.chroma.2004.10.094).

[8] S. I. F. S. Martins and M. A. J. S. van Boekel, "Melanoidin's extinction coefficient in the glucose/glycine maillard reaction," *Food Chemistry*, vol. 83, no. 1, pp. 135–142, 2003.

[9] S. E. Papadakis, S. Abdul-Malek, R. E. Kandem, and K. Yam, "A versatile and inexpensive technique for measuring color of foods," *Food Technology*, vol. 54, no. 12, pp. 48–51, 2000.

[10] T. M. Amrein, B. SchönBächler, F. Escher, and R. Amado, "Acrylamide in ginderbread: critical factors for formation and possible ways for reduction," *Journal of Agricultural and Food Chemistry*, vol. 52, pp. 4282–4288, 2004.

[11] N. Surdyk, J. Rosén, R. Andersson, and P. Åman, "Effects of asparagine, fructose and baking conditions on acrylamide content in yeast-leavened wheat bread," *Journal of Agricultural and Food Chemistry*, vol. 52, pp. 2047–2051, 2004.

[12] D. Taubert, S. Harlfinger, S. Henkes, L. Berkels, and E. Schömig, "Influence of processing parameters on acrylamide formation during frying of potatoes," *Journal of Agricultural and Food Chemistry*, vol. 52, pp. 2735–2739, 2004.

[13] T. C. Pearson, "Detection of pistachio nuts with closed shells using impact acoustics," *Applied Engineering in Agriculture*, vol. 17, no. 2, pp. 249–253, 2001.

[14] A. E. Cetin, T. C. Pearson, and A. Tewfik, "Classification of closed and open shell pistachio nuts using voice recognition technology," *Transactions of ASAE*, vol. 47, no. 2, pp. 659–664, 2004.

[15] A. E. Cetin, T. C. Pearson, and A. H. Tewfik, "Classification of closed and open shell pistachio nuts using impact acoustical analysis," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 17 of *2*, pp. 249–253, 2004.

[16] I. Onaran, B. Dulek, T. C. Pearson, Y. Yardimci, and A. E. Cetin, "Detection of empty hazelnuts from fully developed nuts by impact acoustics," in *Proceedings of 13th European Signal Processing Conference*, 2005.

[17] J. Hartigan, *Clustering Algorithms.* New York, NY: John Wiley & Sons, 1975.

[18] J. Hartigan and M. Wong, *Applied Statistics*, ch. Algorithm AS136: A k-means clustering algorithm, pp. 100–108. 1979.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* John Wiley & Sons, second ed., 2000.

[20] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice Hall Professional Technical Reference, 1990.

[21] G. Schwartz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[22] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[23] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated classification likelihood," Rapport de Recherche 3521, INRIA, 1998.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[25] A. E. Cetin and V. Weerackody, "Design vector quantizers using simulated annealing," *IEEE Transactions on Circuits and Systems*, vol. 35, p. 1550, December 1988.

[26] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[27] J. J. deGruijter and A. B. McBratney, *Classification and Related Methods of Data Analysis*, ch. A modified fuzzy k means for predictive classification, pp. 97–104. Amsterdam: Elsevier Science, 1988.

[28] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[29] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D. Tax, *PRTools: A Matlab Toolbox for Pattern Recognition*, 2004.

[30] "Mathworks Inc. Matlab: The language of technical computing," 1984-. `http://www.mathworks.com`.

[31] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Transactions on Image Processing*, vol. 6, pp. 901–932, July 1997.

[32] *$L^*a^*b^*$ Color Space.* Information at `http://en.wikipedia.org/wiki/Lab_color_space`.

[33] *XYZ Color Space.* Information at `http://en.wikipedia.org/wiki/CIE_XYZ_color_space`.

[34] *Color Space Transformations.* Information at `http://www.brucelindbloom.com`.

[35] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1994.

[36] R. H. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 100-132, 1985.

[37] R. C. Gonzales and R. E. Woods, *Digital Image Processing.* Prentice-Hall, 2002.

[38] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[39] H. Z. Şenyuva and V. Gökmen, "Study of acrylamide in coffee using an improved liquid chromatography mass spectrometry method: Investigation of colour changes and acrylamide formation in coffee during roasting," *Food Additives and Contaminants*, vol. 22, pp. 214–220, March 2005.

[40] D. Taeymans, J. Wood, P. Ashby, I. Blank, A. Studer, R. H. Stadler, P. Gonde, P. van Eijvk, S. Laljie, H. L. H, M. Lindblom, R. Matissek,

D. Müller, D. Tallmadge, J. OBrien, S. Thompson, D. Silvani, and T. Whitmore, “A review of acrylamide: An industry perspective on research, analysis, formation and control,” *Critical Reviews in Food Science and Nutrition*, vol. 44, pp. 323–347, 2004.

[41] K. Granby and S. Fagt, “Analysis of acrylamide in coffee and dietary exposure to acrylamide from coffee,” *Analytica Chimica Acta*, vol. 520, pp. 177–182, 2004.

[42] D. V. Zyzak, R. A. Sanders, M. Stojanovich, D. H. Tallmadge, B. L. Eberhart, D. K. Ewald, D. C. Gruber, T. R. Morsch, M. A. Strorthers, G. P. Rizzi, and M. D. Villagran, “Acrylamide formation in heated foods,” *Journal of Agricultural and Food Chemistry*, vol. 51, pp. 4782–4787, 2003.

[43] J. S. Lim, *Two-Dimensional Signal and Image Processing*, pp. 469–476. Englewood Cliffs, NJ: Prentice Hall, 1990.

[44] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* Prentice-Hall, 1993.

[45] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Prentice-Hall, 1978.

[46] F. Itakura, “Line spectrum representation of linear predictive coefficients of speech signal,” *J. Acoust. Soc. Amer.*, vol. 57, p. S35, April 1975.

[47] F. K. Soong and B. W. Juang, “Line spectrum pair (lsp) and speech data compression,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, (San Diego, CA), pp. 1.10.1 – 1.10.4, March 1984.

[48] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods: Support Vector Learning.* Cambridge, MA: MIT Press, 1999.