Palestine Polytechnic University

Collage of Administrative Science and Informatics

Department of Information Technology

Computer Based Analysis of Electrophoresis Gel Images

Submitted by:

Bissan Abu-Sharar Mais Hantash

Madleen Barbarawi

Supervisors:

Dr. Rami Arafeh Dr. Hashem Tamimi

Project submitted in partial fulfillment of the introduction to project
2010

جامعة بوليتكنك فلسطين
Palestine Polytechnic University
(PPU)
The Library المكتبة
Acc. 24827 الرقم المتسلسل
Class. رقـم التصنيف

Abstract :

Determining the genetic diversity between individuals is one of the most important fields in biology. DNA marker systems have been used in diversity studies where markers are visualized on electrophoresis gel images. Nowadays analyzers convert electrophoresis gel images into (0/1) matrices manually, which causes human errors. Our project aims to overcome these errors by designing a software that converts electrophoresis gel images into (0/1) matrices with minimum errors.

Dedication

To all who stood by us during our work

To our parents

To our brothers and sister

To our teachers who supported us

To all those how care about us

To all of our friends who

support us

Thank you all for your support

Table of content

Table of figures

CHAPTER 1

Introduction

1.1 Project Benefits

1.2 The Organization of this Document

Chapter 1: Introduction

One of the key elements in studying biological diversity is to differentiate between genetically close individuals. During the last two decade, a huge number of literatures addressed the genetic diversity between individuals particularly (PCR) band techniques DNA images are obtained through many biological techniques that will be discussed later in this report; these images are one of the ways that are used to do differentiation between organisms. The resulting images are then converted into a (0/1) matrix where (0) represents the absence of a bands and (1) represents the presence of the bands. Figure 1.1 is an example of gel image.

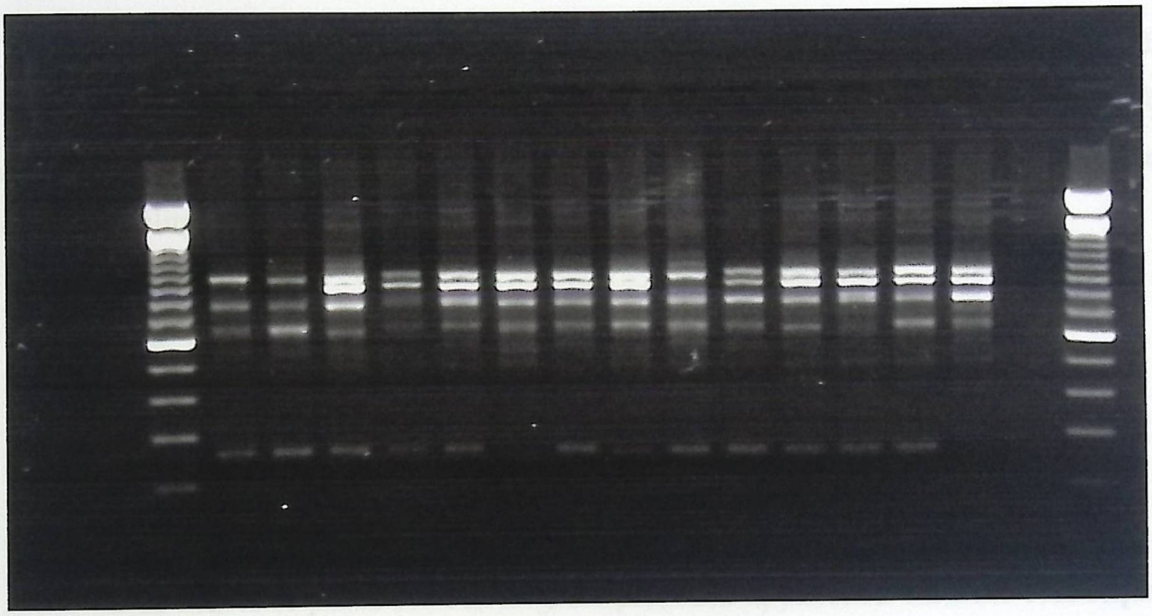


Figure 1.1: Gel image.

This particular conversion of the image into a 0/1 matrix with minimal human effort, error and interaction will be our projects' aim. After this conversion is done, further analysis is carried out by scientists based on the similarity of the resulting matrix. Image processing will be used intensively in order to obtain the (0/1) matrix.

1.1. Project benefits:

Usually gels of dominant markers are translated into a 0/1 matrix and then analyzed with different software available. Researchers and scientists have to convert the images into a (0/1) matrix manually. This has many disadvantages, some of them are:

- It is considered as a time and effort consuming process.
- The process is mostly subjective and this may cause many errors in analysis.

Errors in the data matrix would lead to errors which will affect the data analysis and interpretation. These disadvantages of manually converting the image into a matrix can be overcome by designing software generates the 0/1 matrix with minimal error, time and effort. But nevertheless, detection and image analysis still remain labor-intensive.

Although several commercial software packages (which will be mentioned later in this report) are now available in the market, DNA image analysis still requires some intervention by the user, and thus a certain level of image processing experience is required. In this project we present a fully automated software that is able to do the conversion of the image into a (0/1) matrix.

Image enhancement improves poor quality images that have faint DNA bands, experimental results show that the proposed method eliminates defects due to noise for good and average quality gel electrophoresis images, and it also improves the appearance of poor quality images.

To summarize the benefits of this software:

- It aims to help the researchers to speed up their analysis.
- They should be able to obtain more repeatable results in an easy way.
- The output would be obtained in short time.
- Less cost is required for building the proposed application than the cost require than buying a new software or application.
- More reliable and accurate results should be presented.

1.2. The organization of this document:

- **Chapter 1** – Introduction: this chapter addresses some biological concept and information about use of electrophoresis gel images. The idea, benefits, assumption and hypotheses of the project.

- **Chapter 2** – Background: this chapter identifies the DNA and the process that it passes throw until it reaches the final state that can be used to form the electrophoreses gel image. It also shows the image processing concept, and the filters that the project depends on. And the related work and other softwares that are used in this filed.

- **Chapter 3** – Methodology: this chapter views the collected data, the limitation and the steps that have been taken in consideration to overcome these limitation, how the images are converted into 0/1 matrix manually, and the solution algorithm.

- **Chapter 4** - Experiments and results: defines the approaches and specifications of the project, the data specification, and the results and discussion of the project results.

- **Chapter 5**- Conclusions and Future work: the project goals and limitations are reviewed, and what of these goals are achieved, and the limitations that are overcome, and what is suggested for future work.

HAPTER 2

Background

2.1 Biology

2.2 Image Processing

2.3 Related Work

2.4 The essentiality of the proposed application

Chapter 2: Background

This chapter will discuss two main parts the first one is about the biological background and the steps that the DNA passes throw to get the final status that can be used to produce the image that will be used as input for this software. The second part will shows the image processing filters, methodologies and algorithms that used to enhanced and analyze the images. And the last part will talk about some software that is used in this area and its advantages and drawbacks.

2.1. Biology:

Morphology is the structure and configuration of an organism which includes the form of the outer appearance such as shape and color, and form of the internal parts such as bones. Genetics is the study of how living organisms inherit some the outer and inner features from their parents, where it tries to specify which features of the organism is inherited and which are not. These features or traits are carried on the DNA which is made of sequences of simple units, the order of these units decide the instruction in the genetic code. The genetic information is transferred across generations, and gives the scientists explanations for many traits such as diseases resistance, tolerance to environment stresses, and this information has a significant impact on the improvement of organisms not only a valuable source of useful traits but also a bank of highly adapted genotypes.

The methods for detecting and assessing genetic diversity have extended. Several DNA-marker systems are now common and used in diversity studies of organisms. With these markers it is possible to exploit the diversity in DNA sequence. polymerase chain reaction (PCR) technology has become one of the most widely used techniques in molecular biology for many reasons, it is a rapid, inexpensive and simple mean of producing large numbers of copies of DNA molecules from small quantities of source DNA material. Random Amplified Polymorphic DNAs (RAPDs) and Inter Simple Sequence Repeats (ISSRs) are DNA markers provide an opportunity to characterize genotypes and to measure genetic relationships more precisely than other markers. RAPD and ISSR markers are a Polymerase Chain Reaction (PCR) based techniques for molecular analysis which use a random primer to amplification DNA, then the result of these markers are visualized on electrophoresis gel as bands that are present or absent.

2.1.1 PCR (Polymerase Chain Reaction):

Is a technique widely used in **molecular biology**. As PCR progresses, the DNA generated is used as a template for replication. This sets in motion a chain reaction in which the DNA template is amplified. With PCR it is possible to amplify a single or few copies of a piece of DNA, generating millions or more copies of the DNA piece. PCR can be extensively modified to perform a wide array of genetic manipulations (BARDAKCI, 2000) [1].

It is rapid, inexpensive and simple way of copying DNA. When the DNA reaction is entered into the PCR three cycles occurs as follows:

1. **Denaturation**: it is the first cycle, where the PCR heats the DNA reactions at high temperatures, which forces the DNA to separate from each other and this will make it possible for the primers to match with the DNA and start copying in the next step as in Figure 2.1

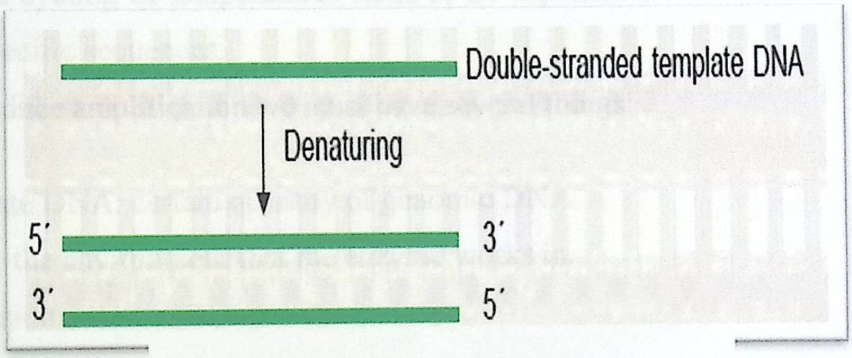


Figure 2.1: Denaturation step [2].

2. **Annealing**: in this step the PCR reduces the heat of the DNA reaction. The primers anneal to the complementary regions in the DNA templates strands and double strand are formed again between but this time between primers and complementary sequences of DNA, as in Figure2.2.

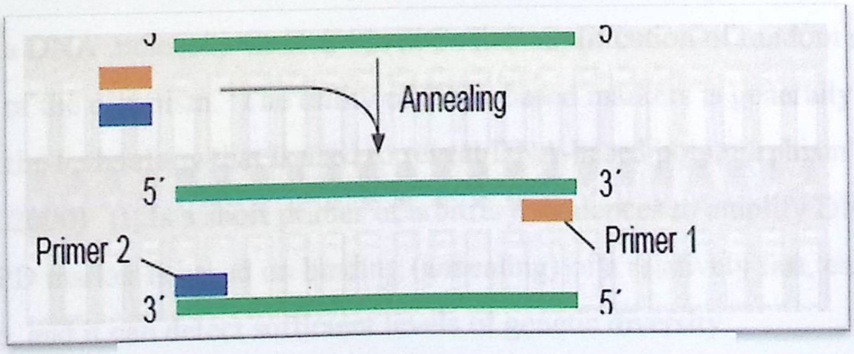


Figure 2.2: Annealing step [2].

3. **Extension:** after this the PCR increases again the heat in order to let primers start to copy the DNA. The enzyme reads the opposing strand sequence and extends the primers by adding nucleotides in the order in which they can pair. The primer copies, the DNA templates double every time as in Figure 2.3.

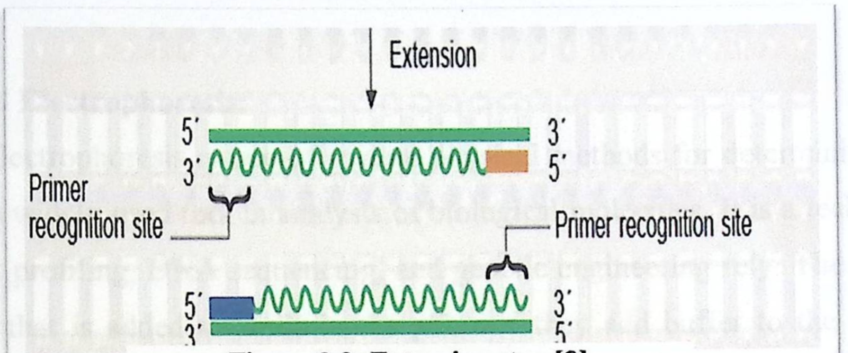


Figure 2.3: Extension step [2].

This cycling of temperatures leads to an exponential increase in the number of copies of specific sequences.

In order to produce amplification we must have several things:

- Template DNA: certain quantity of genomic DNA.
- Buffer: the environment that the enzyme works in.
- Magnesium.
- Enzyme Tag polymerase.
- Nucleotides (T,G,C,A): the subunits of DNA .
- Primer: usually consist of 10 nucleotides at a random order .
- H_2O: Interaction environment which all of these components will be added to.

2.1.2 RAPD (Random Amplified Polymorphic DNA):

RAPD a DNA molecular marker based on the amplification of random segments in the genome of the organism. 'The utility of DNA-based markers is generally determined by the technology that is used to reveal DNA-based polymorphism' (BARDAKCI, 2000) [1].Is a short primer of arbitrary sequences to amplify DNA segments. RAPD marker is based on binding (annealing) of a relatively fast, easy to perform, cheap, and it can detect sufficient levels of genetic diversity.

The RAPD reaction is relatively sensitive because of the length of a single and arbitrary primer used to amplify anonymous regions of a given genome. This reproductively problem is the case for lower intensity bands. The case of lower or higher intensity may be that the primer does not perfectly match the priming sequence, so amplifications in some cycles might not occur so bands remain fainter (BARDAKCI, 2000) [1] .

2.1.3 Agarose Gel Electrophoresis:

Agarose gel electrophoresis is one of several physical methods for determining the size of DNA. It is widely used tool in analysis of biological molecules. It is a technique upon which DNA profiling, DNA sequencing, and genetic engineering rely. The image shows a reaction that is added to wells in the gel, and they add buffer to the plastic casing then an electronic current is passed in the buffers.

In response to the electric current that will base through the buffers, the DNA is forced to migrate through highly cross-linked Agarose matrix the phosphates on the DNA are negatively charged and molecule will migrate to the positive pole.

Figure 2.4.a shows the plastic case that the gel will be lay in, Figure 2.4.b shows the process of loading DNA into the gel [3].

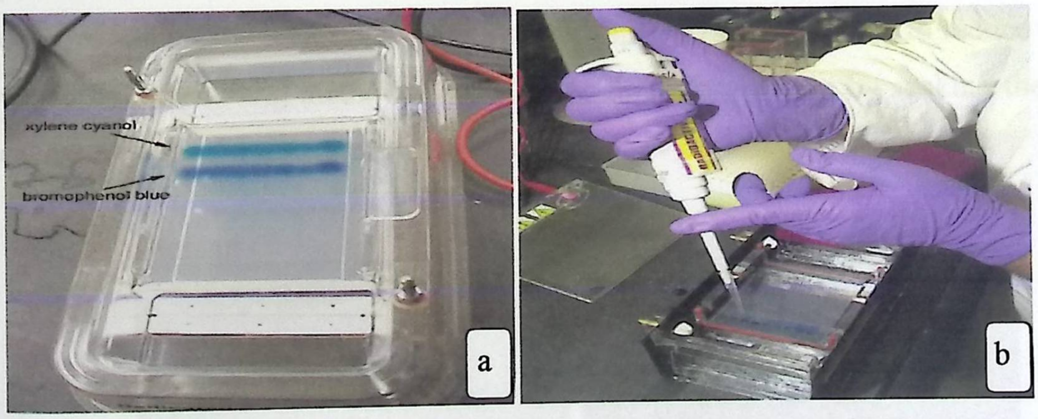


Figure 2.4: (a) Horizontal gel apparatus. (b) Loading process of the DNA in agarose gel [4].

As the fragments move out of the well and into the gel matrix they start to separate and move as separate bands these fragments moving from the top toward the Bottom of the gel, where the bands near the well are large and the farthest contain short fragments.

These separate bands are been close together in the beginning (near the well), and spread farther apart as they reach the end of the gel. Figure 2.5.

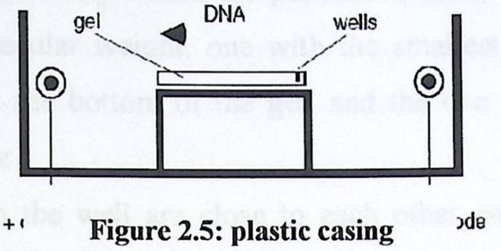


Figure 2.5: plastic casing

that will affect There are three factors the migration rate through the gel:

- DNA size
- DNA conformation
- Ionic strength of the running buffer.

If the DNA fragment is large it will migrate slower in the gel they have difficult time penetrating the gel and their migration does not have linear relationship to size and the entangled possibility in the matrix will increase, but if the fragment was small it will migrate more quickly than large fragment.

At the end of this process the DNA fragment is allocated in the gel. As explained previously sometimes the gel absorbs background of ethidium bromide which could obscure some bands if heavy. So the bands appear stained in order to reduce this effect we apply a distaining method that can help, distaining is done by soaking the gel in an excess of water for about an hour. Figure 2.6 show the effect of staining [3].

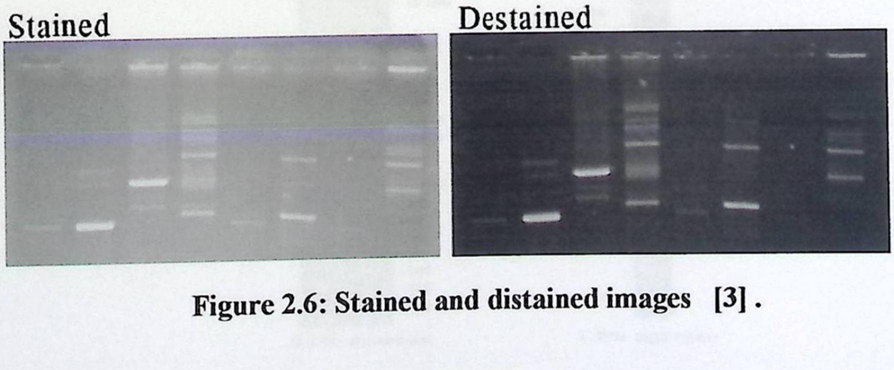


Figure 2.6: Stained and distained images [3] .

The gel is removed from the chamber and put on UV screen and has a digital camera on top of it to capture pictures for the gel. These pictures of gel give views of the DNA bands; the band that is present appears as white light in the image and the absence of this band appears to be black.

The analysis of these images and convert them into 0/1 matrices and use these matrices to know the diversity within the population under study. DNA runs in the gel depending on its molecular weight, one with the smallest molecular weight runs the fastest and appears at the bottom of the gel, and the one which has larger molecular weight migrates slower.

The bands next to the well are close to each other and the far are indistinct and sometimes missed. "For example, the band of ladder standard is always clear and distinguishable. Find this band on your gel and then count in both directions until you lose confidence in your ability to identify bands. Once you have identified the bands, enter the sizes onto your table of distances migrated. Now you can plot your standard graph."[3].

If the bands match the molecular weight in the ladder the band is read directly, but if it does not match the standard. The researchers should measure the distant of unknown bands and locate them on the scale. And now they can read the molecular weight directly off of the log scale. The scale for the same samples must match. But they must be run on the same gel because this distant between bands does not only represent the size of DNA but also represent the time that the gel was allowed to run. Therefore if the same DNA runs on two different gels will not have the same scale and will not be directly compared. Then pictures are compared and the (0/1) matrix is built. Researchers do these steps manually as in Figure 2.7.

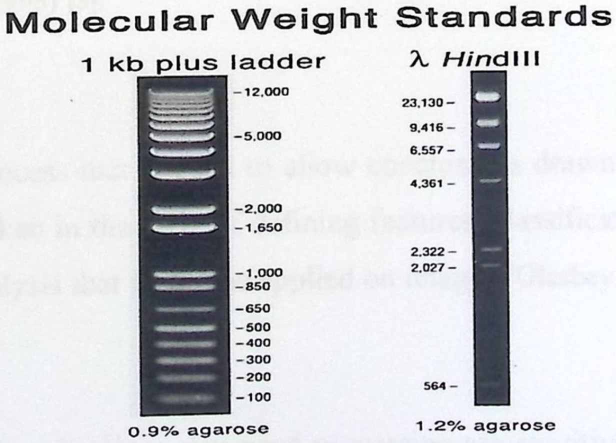


Figure 2.7: Molecular weight standards [3].

2.2. Image Processing:

Image processing is a growing field covering a wide range of techniques for the manipulation of digital images and describing images in all their forms, image characteristics, parameters that are related to the image. With low cost and high availability of digital cameras, more companies than ever before are able to utilize image processing software and reliable hardware in their research, as well as in their products.

These techniques can be divided into three main areas:

1. Encoding:

Which is a term that is used to describe the method by which an image can be described by using a series of binary digits, to overcome the limitation of storage space where storing images with high amount of data where encoding concentrates on reducing the size of the data that represents an image to be higher speed for transmission and lower requirements for storage (Glasbey, et al .1998) [5].

2. Transformation:

Which is a process of altering images to make them more suitable for any intended purpose by removing any noise or any incorrect features placed in the image to be ready to any presentation or any further analysis, so there are various methods and filters which are used to get the desired results, these filters will be described later in this report (Glasbey, et al .1998) [5].

3. Analysis:

Which is a process that is used to allow conclusions drawn from images; these conclusions can be taken in the form of defining features, classifications, measurements and any statistical analysis that would be applied on images (Glasbey, et al .1998) [5] .

2.3.1 Filters:

Some sorts of applications that need processing are sensitive to noise; the noise should be removed before going through the processing, and some when removing noise which can cause loss of some important detailed information in the image. Most noise removal processes are done by using image processing filters.Here there are many filters we are used for these project images and the result as below:

1. Morphological filter:

This type of filter is categorized as a subclass of non linear filters, it is based on minimum and maximum operations, and is used to enhance images by resolving many problems that images include such as noise removal, feature extraction, and estimate the background trends.

This type of filters includes many standards for morphological filters that are used to perform a lot of functions, like:

1.1 Dilation: this filter is used to "smooth small dark regions in an image, It is defined as the maximum of the sum of a local region of an image and a grayscale mask" [7]. Structuring elements (SE) are small images that are used to emphasize and de-emphasize elements in any image.

Performing dilation on the image can affect it as follows:

- If all the values in the structuring element are positive, the output image tends to be brighter than the input image.
- Dark elements within the image are reduced or eliminated, depending on how their shapes relate to the structuring element used.

These effects depend on the shapes and the values of structuring elements with the details of the image itself.

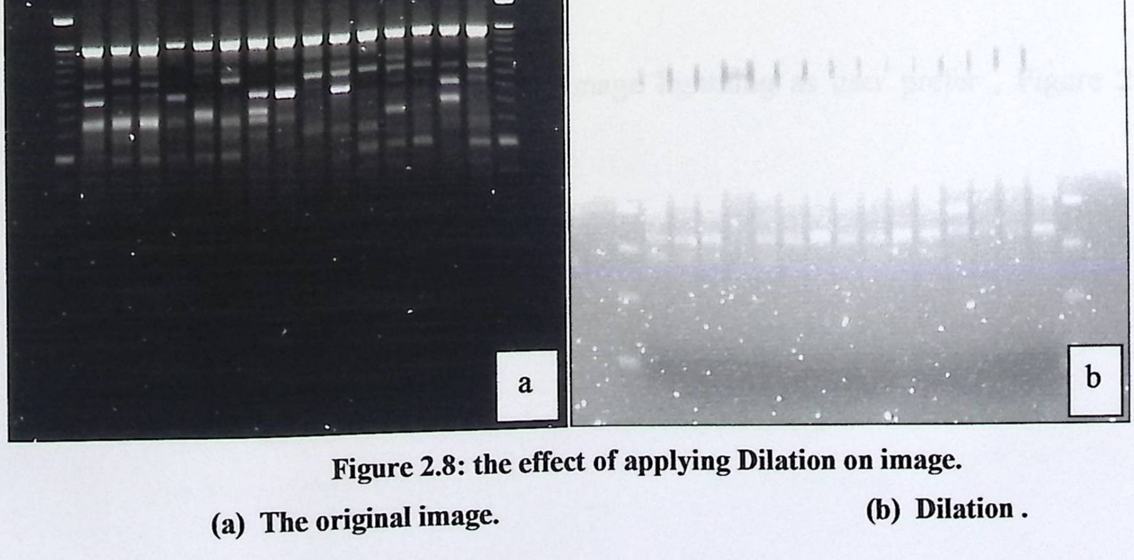


Figure 2.8: the effect of applying Dilation on image.

(a) The original image. **(b) Dilation .**

1.2. Erosion:

These filters are use to “smooth small light regions, it is defined as the minimum of the difference of a local region of an image and a grayscale mask” [7].

Performing Erosion on a grayscale image can affect them as follows:

If all the values in the structuring element are positive, the output image tends to be darker than the input.

Light elements within the image are reduced or eliminated, depending on how their shapes relate to the structuring element used.

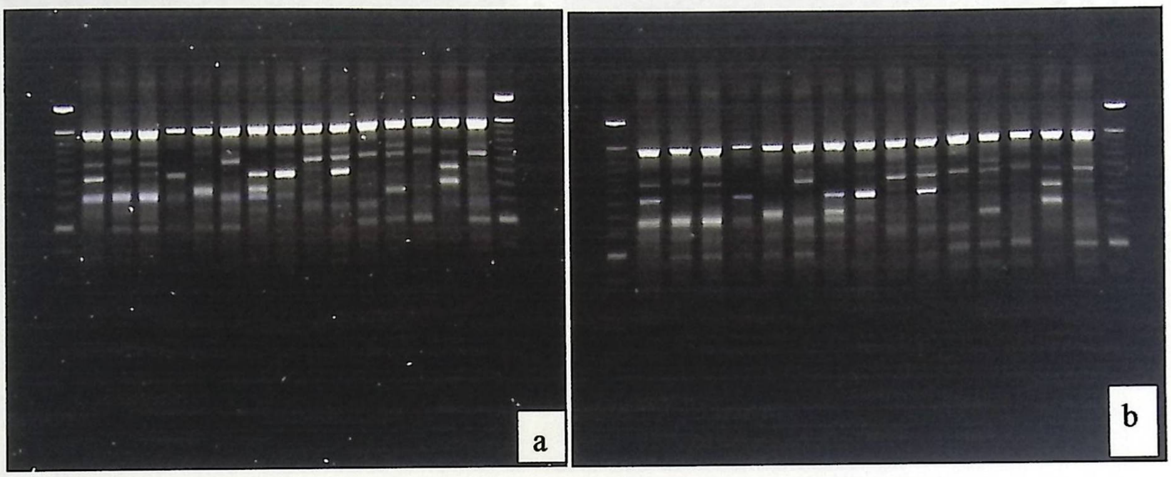


Figure 2.9: the effect of applying Erosion on image

(a) The original image. (b) Erosion .

The degree of erosion effectiveness depends on the shape, values of structuring elements with the details of the image itself.

Erosion filter is the filter that was used in the solution algorithm in order to remove the noise from the images before processing it.

2. Darkening and brightening image:

These filters are used to control image lightning as user prefer , Figure 2.10 shows the result of applying this filter :

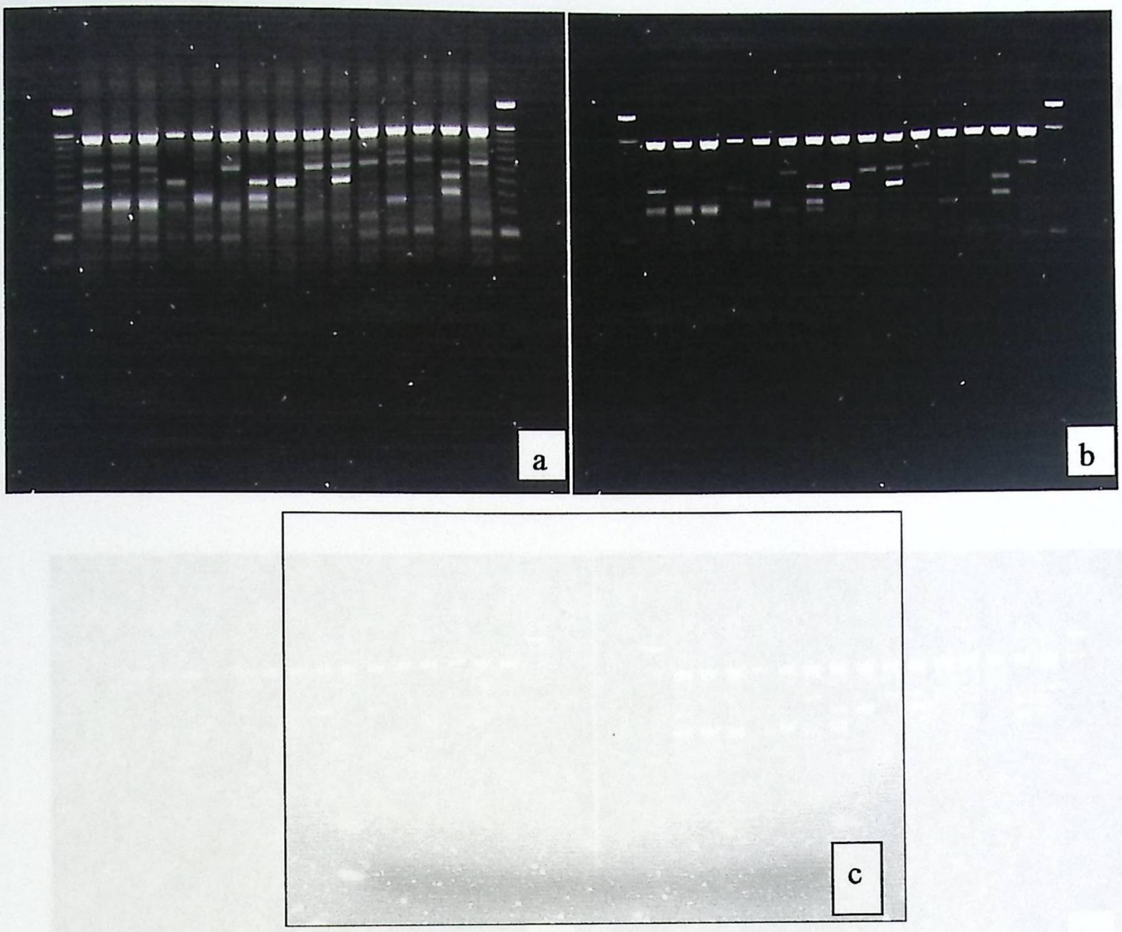


Figure 2.10: the effect of σ when applying darkening and brightening on image.
(a) The original image. (b) Drakining . (c) Brightening.

3. Embossing filter:

This type of filter is used to provide an optical illusion where some objects of image are closer or farther away than the background , so a 3D or embossed effect was provided on it . Figure 2.11 shows the effect of applying Embossing filter.

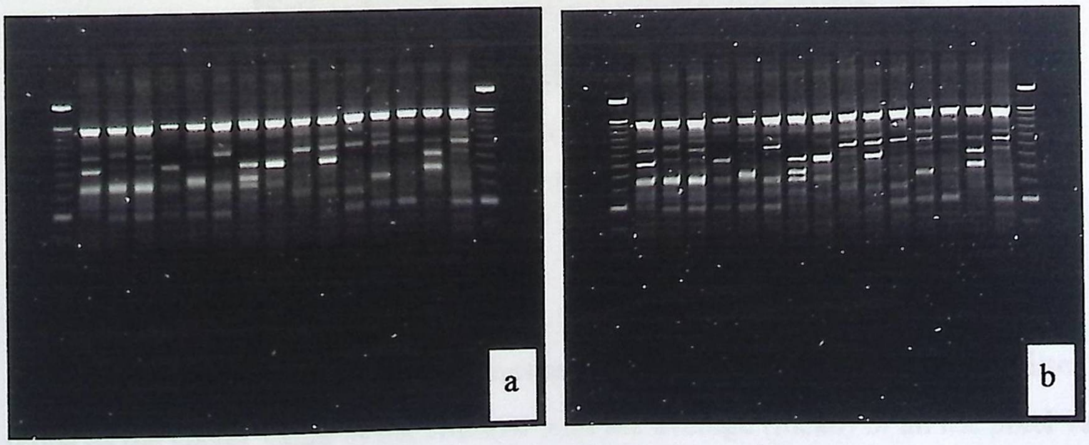


Figure 2.11: the effect of applying Embossing on image.
(a) The original image. (b) Embossing.

4. **Sobel Edge :**

Sobel Edge filter is use to detect edges based applying a horizontal and vertical filter matrix on image.

Horizontal filter

1	0	-1
2	0	-2
1	0	-1

Vertical filter

-1	-2	-1
0	0	0
1	2	1

Figure 2.12: the sobel edge detection kernel .

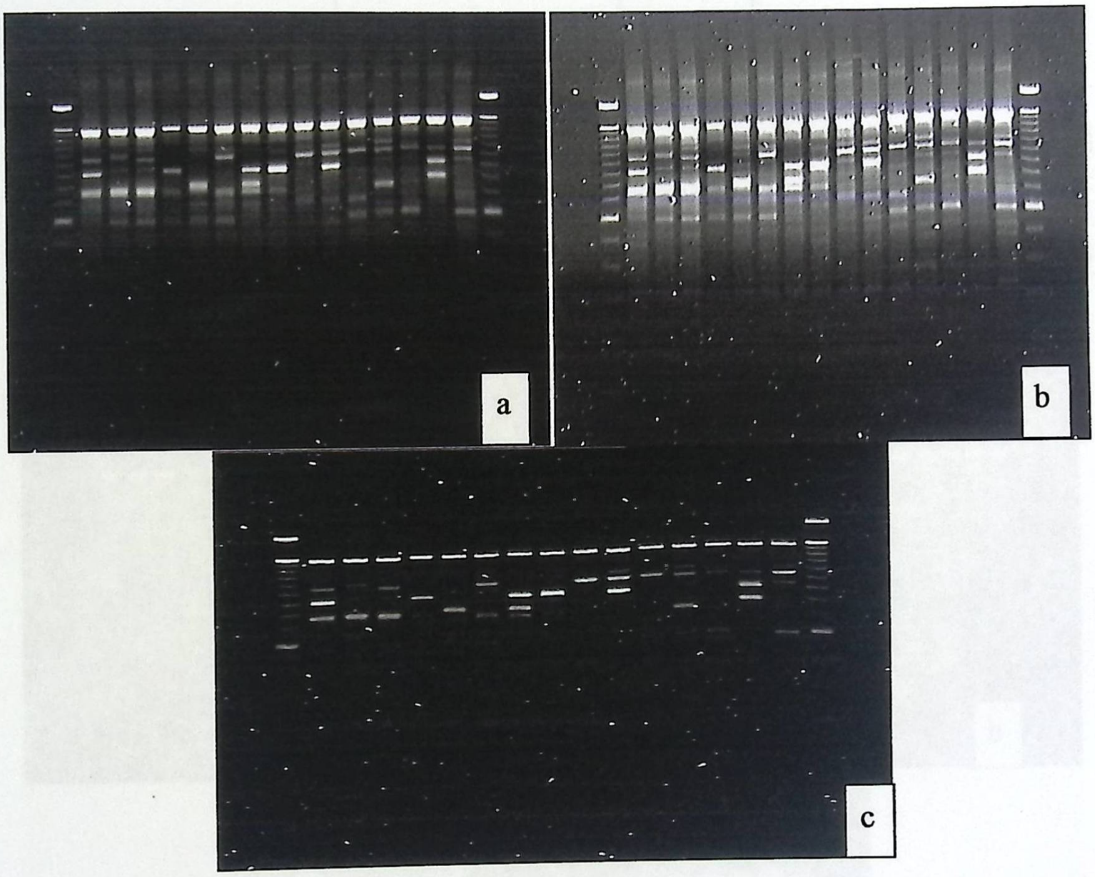


Figure 2.13: the effect of applying Sobel on image

(a) The original image. (b) Sobel(vertically) . (c) Sobel(horizontaly)

8. Mean filter:

This type of filter can be defines as "a simple sliding-window spatial filter that replaces the center value in the window with the average (mean) of all the pixel values in the window. The window, or kernel, is usually square but can be of any shape" (Wang, 2004) **[12]**.

It is used to reduce the noise and the amount of intensity variation between one pixel and the next [10].

This filter replaces each pixel value in an image with the mean value of its neighbors, including the pixel itself.

For example the mean can be calculated by multiplying each pixel in the image with the following 3×3 kernel in the figure 2.14:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 2.14: 3×3 averaging kernel often used in mean filtering.

A 3×3 kernel is almost always used, but if a more sever smoothing is needed to be done on the image then a larger kernel is used (5×5 kernel).

Figure 2.15: (b) shows an image before and after applying mean filter using 3×3 kernel and 5×5 kernel, as in Figure 2.15(c).

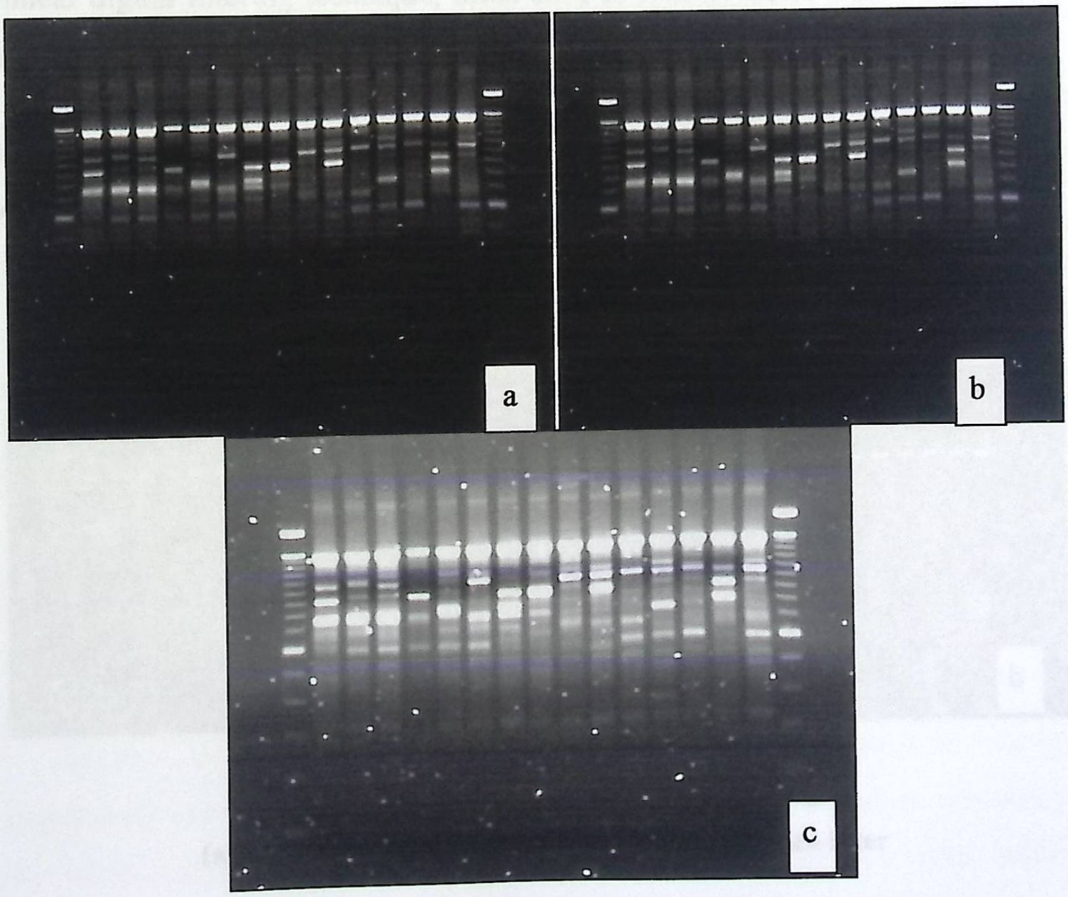


Figure 2.15: kernel used in mean filtering

(a) original image ,(b) 3×3 averaging kernel, (c) 5x5 averaging kernel

When using a larger kernel to multiply it by the values of the pixels in the image the result will not be as significant when using a smaller kernel.

This result leads us to list the problems that are associated with this filter as follows:

- A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighborhood.
- When the filter neighborhood straddles an edge, the filter will interpolate new values for pixels on the edge and so will blur that edge. This may be a problem if sharp edges are required in the output.

9. Median filter:

This type of filter can be defined as "a sliding-window spatial filter, but it replaces the center value in the window with the median of all the pixel values in the window. As for the mean filter, the kernel is usually square but can be of any shape"**[11].** And it is a non-linear digital filtering technique, often used to remove noise from images or other signals.

This filter is a more robust method than the traditional linear filtering, because it preserves the sharp edges (Wang, 2004) **[12].**

Figure 2.16 shows an image before and after applying median filter on it.

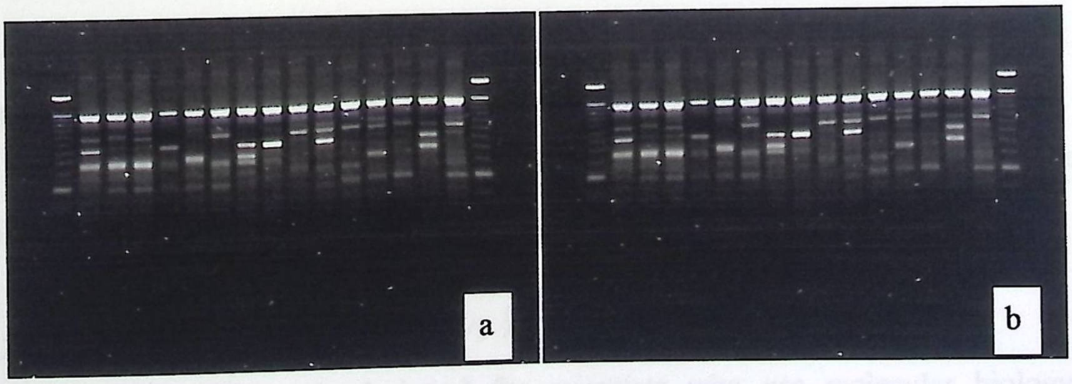
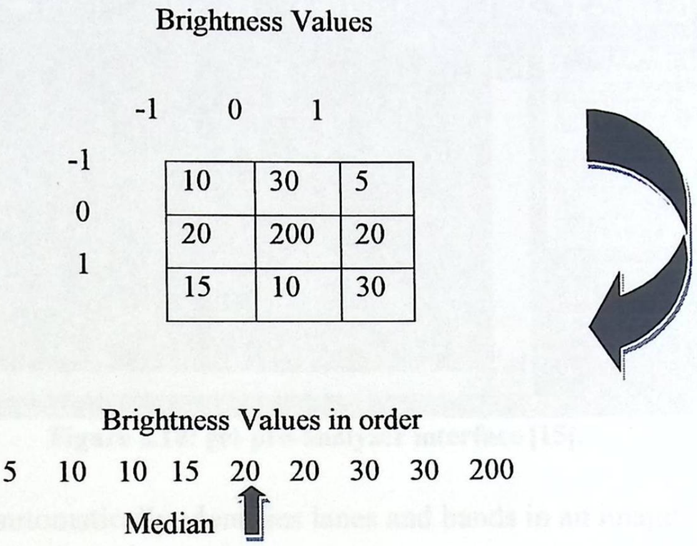


Figure 2.16: applying median filter on an image
(a)Original image (b) image after applying median filter

Median finds the new value of the pixel from the values of its neighbor pixels, because of that the new value of the pixel is more reliable and true.

Figure 2.17 shows an example of how to find the brightness value using median.



Brightness Values

	-1	0	1
-1	10	30	5
0	20	200	20
1	15	10	30

Brightness Values in order

5 10 10 15 20 20 30 30 200

Median

Figure 2.17: example of how to find the brightness value using median [13].

This type has many advantages over mean filter **[14]:**

- The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
- The median filter is much better at preserving sharp edges than the mean filter because the new value of the pixel is taken from the image which makes it more reliable and realistic.

2.3 Related work

There are a number of different softwares in the market that are considered similar to the software that we aim to design; some of these softwares are:

2.3.1 GEL-Pro Analyzer:

Gel-Pro Analyzer is designed for scientists who use molecular biology/gel electrophoresis techniques to improve solutions that increase laboratory efficiency and quality control when analyzing gels. "Gel-Pro Analyzer will increase laboratory productivity and efficiency by eliminating the time consuming steps involved in

Manually analyzing the DNA,RNA, and protein from electrophoresis gels and blots. From digitized gel images, data can be automatically analyzed, quantified, and compared in a reproducible fashion ensuring quality control in the laboratory" [16]. Figure 2.18 shows a snapshot of GEL-Pro Analyzer interface

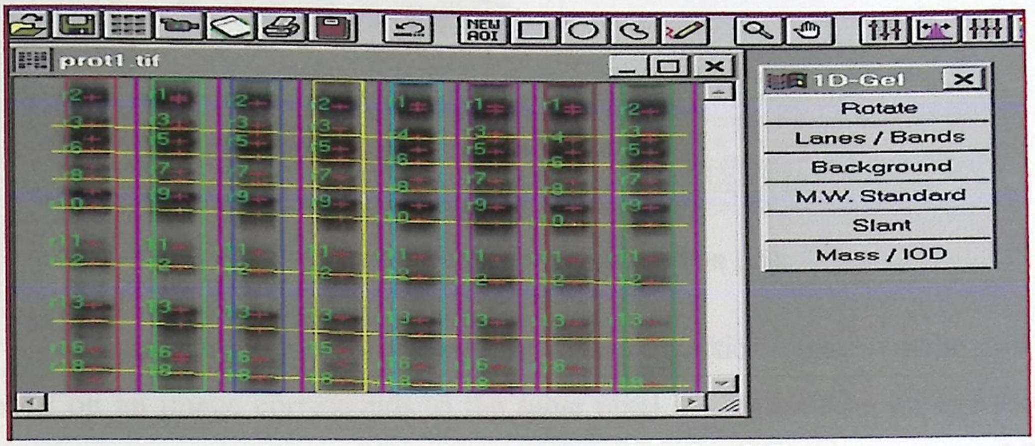


Figure 2.18: gel-pro-analyzer interface [15].

This software automatically identifies lanes and bands in an image. Shows another snapshot of GEL-Pro Analyzer.

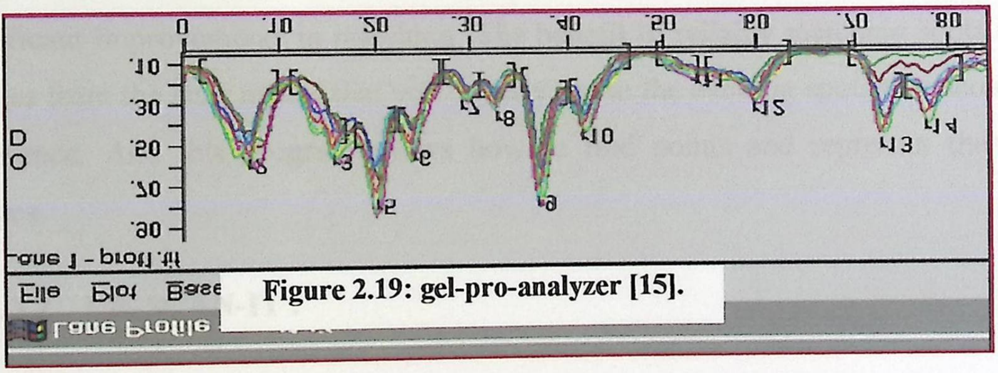


Figure 2.19: gel-pro-analyzer [15].

2.3.2Same Spots:

This software finds matches between gel images and finds the measurements and positions of spots in the images. A snapshot of Same Spots is shown in Figure 2.20.

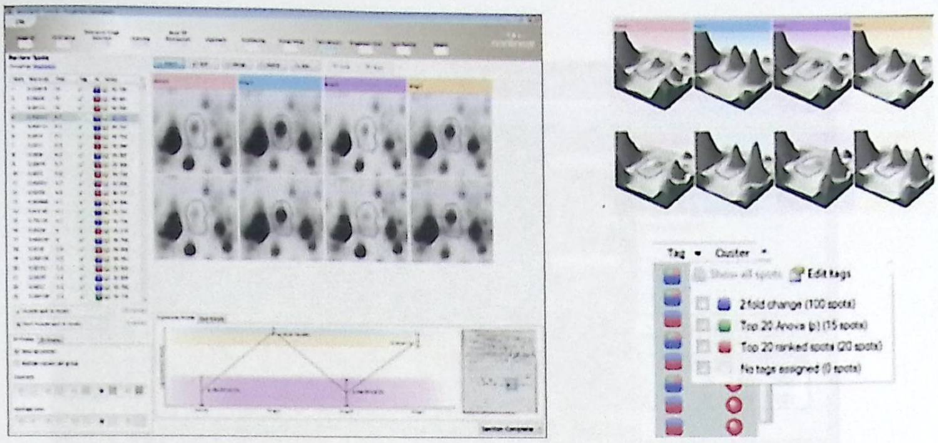

Tag ▾ Cluster ▾
Show all spots
Edit tags
□ 2 fold change (100 spots)
□ Top 20 Anova (p) (15 spots)
□ Top 20 ranked spots (20 spots)
□ No tags assigned (0 spots)

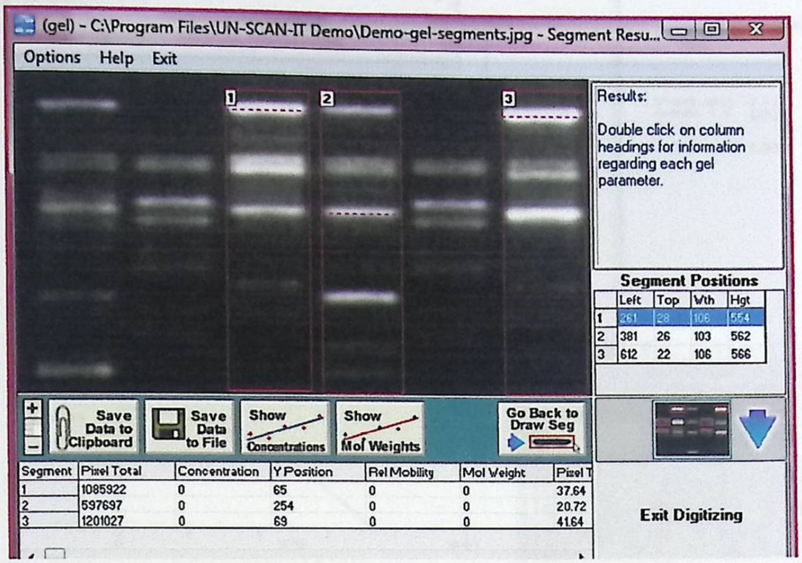
Figure 2.20: alignment correction [15].

By aligning the images before performing spot detection, you can make sure that each pixel on an image corresponds to the same pixel at that location on every other image through the experiment. Alignment is the key part of the samespots workflow, and is the most important to get the right result. The time that is needed for 12 images to be processed is less than 20 minutes [16].

Samespots delivers an alternative start point for 2D image analysis which offers significant improvements in matching. The benefit of reliably matching spots in all gel images from the start means that you quickly get to the exciting spots that show the real difference. And this program shows how to find points and represent them by 3D images.

2.3.2 UN-SCAN-IT :

The UN-SCAN-IT software allows us to automatically convert hard copy graphs to (x, y) ASCII data. Also integrates peak areas, smooth data, takes derivatives, re-scales graphs, and exports the (x, y) data. This softwear needs user interaction, the user has to select group of bands (a horizantal line) Then the softwer starts the analysis. It can be installed on both Widows and Mac environments; figure 2.21 shows a snapshot of UN-SCAN-IT.



(gel) - C:\Program Files\UN-SCAN-IT Demo\Demo-gel-segments.jpg - Segment Resu...

Results:

Double click on column headings for information regarding each gel parameter.

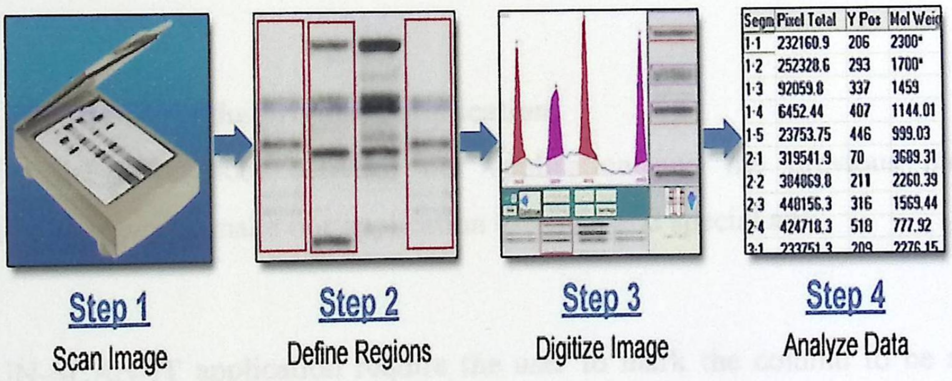
Segment Positions

	Left	Top	Wth	Hgt
1	261	28	106	554
2	381	26	103	562
3	612	22	106	566

Segment	Pixel Total	Concentration	Y Position	Rel Mobility	Mol Weight	Pixel T
1	1085922	0	65	0	0	37.64
2	597697	0	254	0	0	20.72
3	1201027	0	69	0	0	41.64

Figure 2.21: selection of the columns that include band manually [15].

Although UN-SCAN-IT gel has numerous options and features, the basic operation is simply converting an image into pixel density intensity values through simple steps as shown in Figure 2.22:



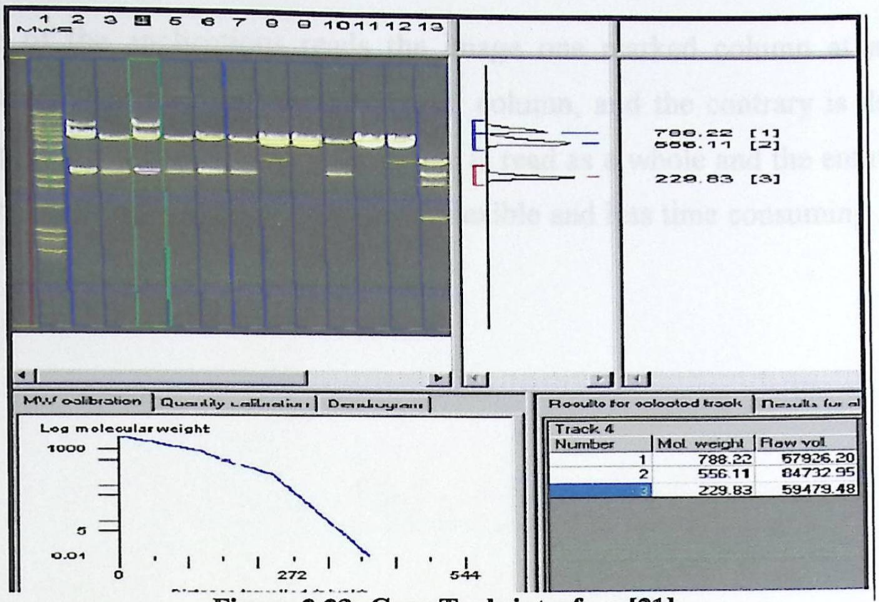
Segm	Pixel Total	Y Pos	Mol Weig
1·1	232160.9	206	2300*
1·2	252328.6	293	1700*
1·3	92059.8	337	1459
1·4	6452.44	407	1144.01
1·5	23753.75	446	999.03
2·1	319541.9	70	3689.31
2·2	384069.8	211	2260.39
2·3	448156.3	316	1569.44
2·4	424718.3	518	777.92
3·1	233751.3	209	2276.15

Figure 2.22: the steps of analyzing gel images using UN-SCAN-IT [15].

2.3.4 GeneTools:

Capturing an image of a DNA/RNA gel or plot with an image analysis system is only the start of determining what information that image has to yield. "The array of GeneTools applications has been developed by Syngene from many years' experience of producing integrated imaging equipment as well as consultation and feedback given by many international scientists using Syngene systems."[17].

Figure 2.23 shows a snapshot of genetools interface.



Track 4		
Number	Mol. weight	Raw vol
1	788.22	57926.20
2	556.11	84732.95
3	229.83	59479.48

Figure 2.23: Gene Tools interface [21]

GeneTools includes an automated multi-layer gel analysis capability, which automatically analyzes each row as if it was an individual gel. This means users can obtain separate results from several different gel layers using one large gel and a single electrophoresis run.

2.4 The essentiality of the proposed application:

As for the differences between our application and the previous mentioned applications that would make our application essential and special are:

1. UN-SCAN-IT application require the user to mark the column to be analyzed using the mouse cursor and further more mark each band in that chosen columns also using the mouse cursor, this is not required in the proposed application.

2. In our application if the user chooses to de-noise the image or not the effect are shown immediately on the image so that the user can see it, and also the effect of the chosen thresholds are shown on the image and on the matrix.

3. Sam Spots application display plots that are difficult to be understood by the user and this makes these applications not user friendly and some of them also require training from the user in order to understand what is done.

4. One of the applications reads the image one marked column at a time and displays the resulting matrix for each column, and the contrary is done by the proposed application where the image is read as a whole and the entire matrix is then displayed which is easier, more flexible and less time consuming .

Methodology

CHAPTER 3

3.1 Problem Definition

3.2 Data Collection and Matrix Analysis

3.3 Limitations and Constraints

3.4 Alternatives

3.5 Flowchart

3.6 Pseudocode

Chapter Three: Methodology

This chapter presents methodologies used in the project, and covers the data collection, it's limitations, and the steps that the project pass throw to get to the result .

3.1 Problem definition:

DNA marker systems have been used in diversity studies where markers are visualized on electrophoresis gel images. Nowadays analyzers convert electrophoresis gel images into (0/1) matrices manually, which causes human errors. Our project aims to overcome these errors by designing a software that converts electrophoresis gel images into (0/1) matrices with minimum errors, effort and time.

3.2 Data collection and matrix analysis:

We have collected some gel electrophoresis image samples from the biology labs of BioTRU at Palestine Polytechnic University; we categorized these images into a number of categories according to the problem that appears in each .We would classify the images into five main categories as follows:

3.2.1 The presence of different sorts of noise:

This category includes images that show white dots resulted from gel impurities another type could be resulted from losing the contrast (faint background) between the black color of the gel and the white color of the bands this noise will affect the result of the analysis that will be done on the images as in figure 3.1.a and, figure 3.1.b

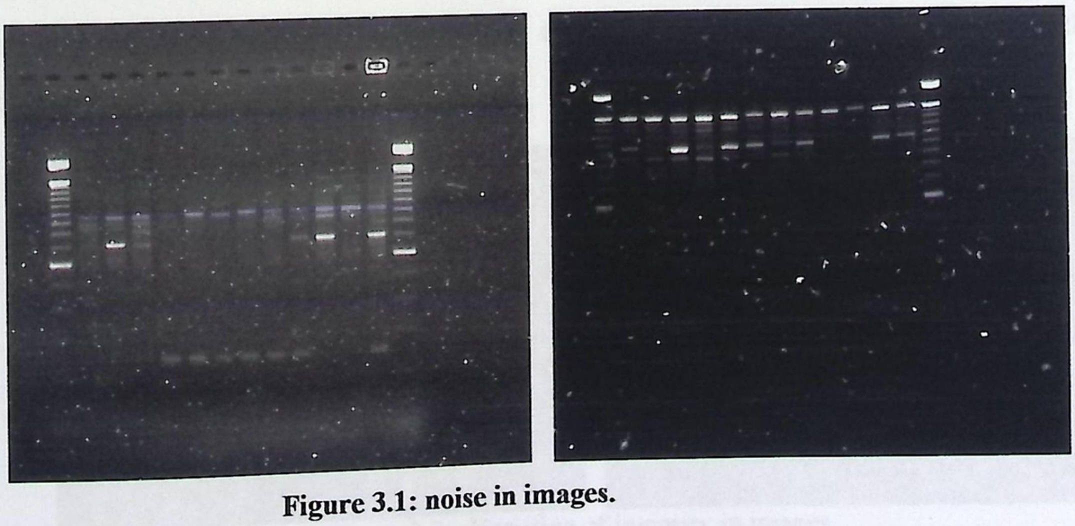


Figure 3.1: noise in images.

3.2.2 Bands Intensity in the image:

- ✓ **Image with low gel intensity :**

This category includes the images that vary in band intensity and size. There are images that have very low band intensity which makes them difficult to recognize, Figure 3.2.

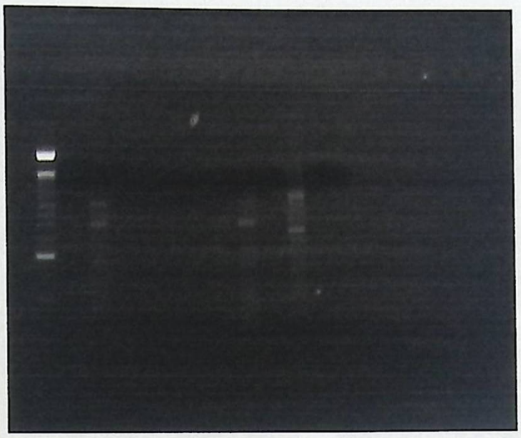


Figure 3.2: very low band intensity

- ✓ **Images with very high gel intensity:**

There are also some images that have bands of very high intensity. These high intensity bands could result in an analysis problem. The challenge that may face the person who analyzes gels with high intensity is that the bands could be misinterpreted or scored as double bands where in fact they belong to a single thick band, Figure 3.3.a.

In Figure 3.3.b we notice that it is hard to determine the position of each band because there is no separation between bands, they have a very close intensity and bands are not separated from each other, this is considered as a serious problem for the analyzer.

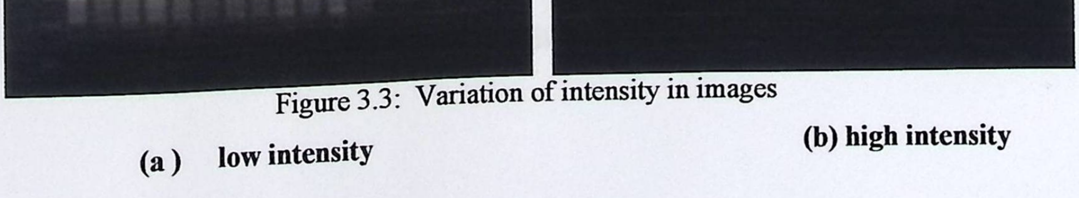


Figure 3.3: Variation of intensity in images

(a) low intensity **(b) high intensity**

3.2.3 Different image size

This category includes images that differ in size some of them are large and others are not completed Figure 3.4.a.and figure 3.4.b

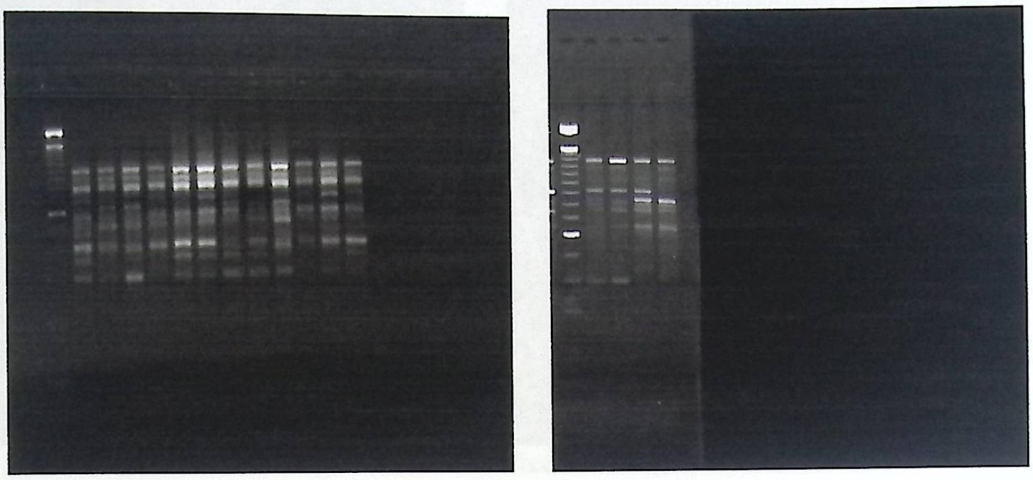


Figure 3.4: different size and layout of images

(a) large image size **(b)small image size**

3.2.4 Smiley images

This problem is a result of bad preparation of the gel. It happens when the gel is put in a plastic gel tray that has curves in it. The bands seem that they don't belong to the same x position (horizontal line), this problem also affects the analysis process, as in figure 3.5

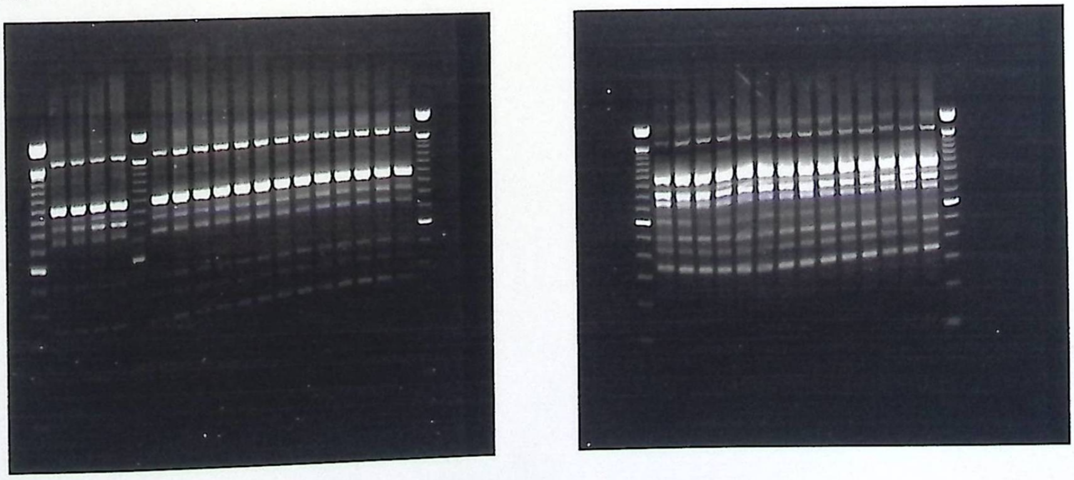


Figure 3.5: smiley images

3.2.5 Stained images:

This category includes the images that are stained. This problem occurs when the gel absorbs background of ethidium bromide which could obscure some bands that are heavy, so the bands appear stained as shown in Figure 3.6.

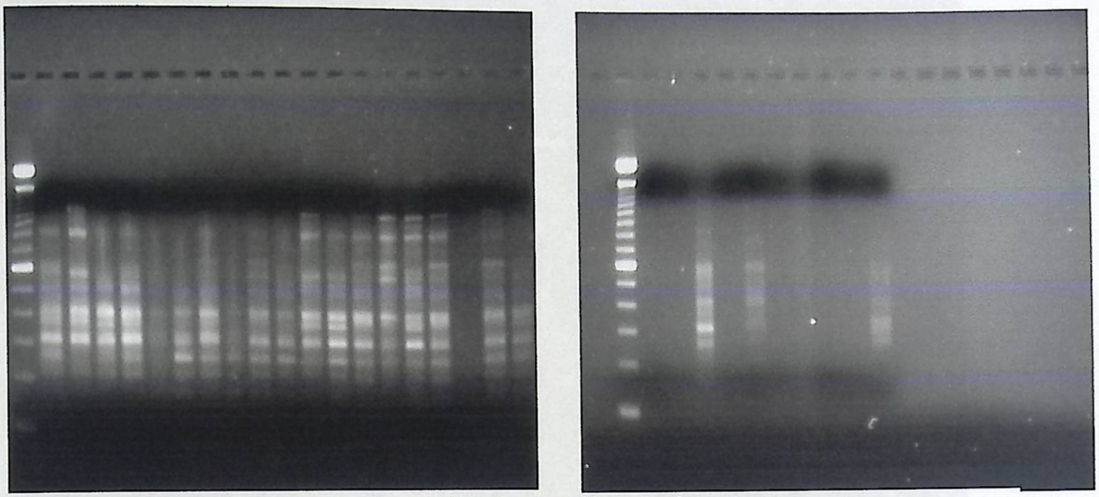


Figure 3.6: stain effect on images

✓ **Gel image (0/1) matrix analyzed manually :**

We analyzed a number of gel electrophoresis images into (0/1) matrices manually and here is one of them:

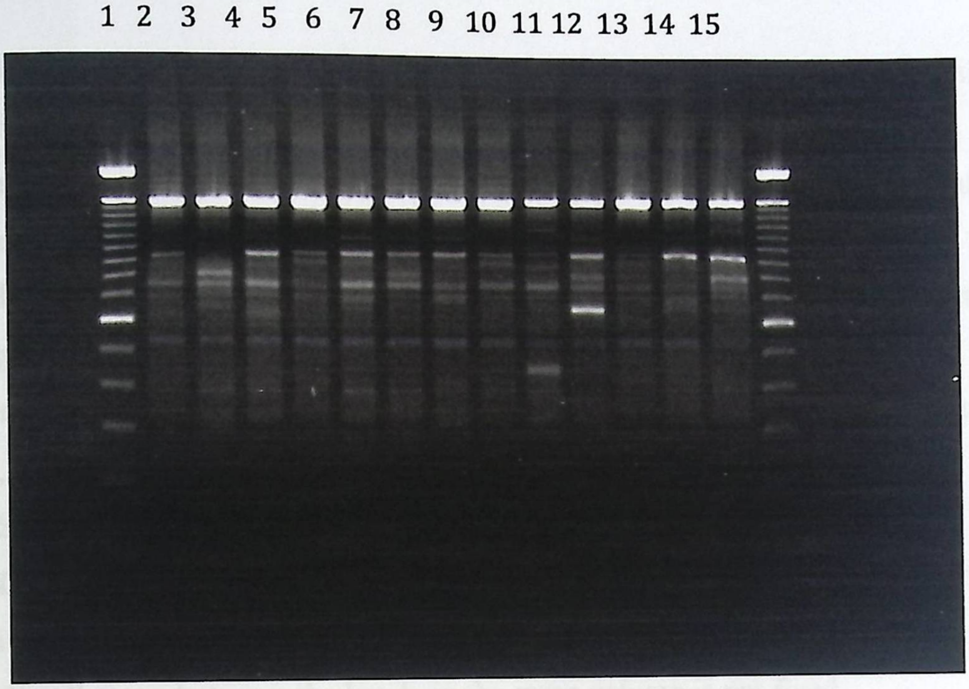


Figure 3.7:Gel image

Sample	Band	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	1	0	0	0	1
2	0	1	0	1	1	0	0	0	0	1
3	0	1	0	1	0	1	0	0	0	1
4	0	1	0	0	0	1	1	0	0	1
5	0	1	0	1	0	1	1	1	0	1
6	0	1	0	1	0	1	1	1	0	1
7	0	1	0	0	0	1	1	1	0	1
8	0	1	0	1	0	1	1	0	0	1
9	1	1	0	1	0	1	1	0	1	1
10	0	1	1	1	1	1	1	1	0	1
11	0	1	0	0	0	1	1	0	0	1
12	0	0	0	0	0	1	1	0	0	1
13	0	0	0	0	1	1	1	0	0	1

Table 3.1: 0/1 matrix for (gel image 1)

3.3 limitations and constraints:

We classified the images according to different limitations that faced us in processing these images; the following are those limitations and constraint:

1. Some images contain noise that will affect the result of the processing of the application, this can cause reliability problems.

2. Images sizes vary, where some images are scaled to be large, some are small and others could be incomplete.

3. The amount of information in each column is different from the others in the same image.

4. Smiley gel occurs when the gel is being produced in a plastic gel tray that has curves in it, so the resulting gel will cause a change in the shape of the bands to look curvy.

5. Some human made errors occur during the capture of the images.

6. Different band shapes: small bands, large bands, thick and thin bands in the same image.

7. The distance between the bands in the same column is not fixed.

8. The intensity of bands differs from the other bands in the same image.

9. The gel reflects background of the ethidium bromide which could makes the bands appear stained. Figure 3.8. Shows the ethidium bromide effect on the image.

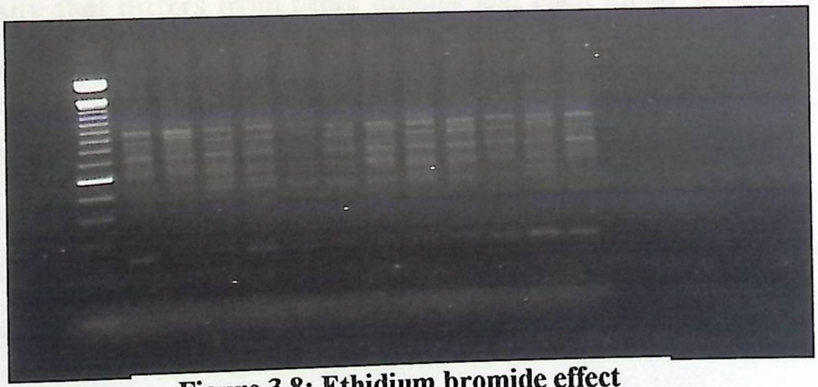


Figure 3.8: Ethidium bromide effect

The proposed application solves some of these constraints, and the following is a brief explanation of how the application solves each of the constraints:

1. As we mentioned earlier some images contain noise, the application solved this problem by using image processing filters; all the filters that we tried are described in the background chapter. The filter that we used in the application is the erosion filter.

2. The second constraint was the variation in images sizes, this problem was also solved in the application where it has the ability to read any image size and scale it so that it can be displayed in a proper manner on the screen of the application.

3. Another constraint was the difference in the amount of the information in each column in the image; the application solved this problem by dealing with each column at a time no matter what the amount of information in each column is.

4. The constraint of different band shapes does not affect the application because bands are dealt without having to take into consideration their shapes and sizes the band intensity is the only thing that matters.

5. As for the constraint of unfixed distance between the bands in the same column, it also does not affect the application because each band is read and analyzed without taking into consideration the distance between the read bands.

6. The different band intensities in the same image was also solved because all bands that have an intensity above the chosen threshold will be read and analyzed.

7. The constraint of stained images was solved, where each image has a total intensity that differs from other images and each image is solved depending on its total intensity.

3.4 Alternatives:

We have chosen our project idea from several alternatives described in details in the implementation of all these alternatives are mentioned briefly in the following:

1. Convert the image into a binary image using a certain threshold chosen by the user.
2. Convert the intensity of all the bands in the image to have the same intensity as the chosen band by the user.
3. Use summation of images columns and rows and thresholding.

3.5 Steps that present the used methodology:

The following steps describe and summarize the implementation of the methodology used in this project:

```
1. Read gray scale image.
2. Apply erosion filter on the image.
3. For each column in the image.
      Find sumX
   End for
4. Threshold sumX using &1 to obtain thre_sumX.
5. While (not last column in image)
      5. a.For each row in sumX
            Find sumY
      End for
      5.b. Threshold sumY using &2 to obtain thre_sumY.
      5.c. While thre_sumY (I) = = 0
            Skip.
            I = I + 1
      End while
      5.d. While thre_sumY (I) = = 1
            Counter1 = Counter 1 + 1
            I = I + 1
      End while
      5.e. If (Counter1 <= value1)
            Put a single 1 in the final matrix.
      Else
            Put a two 1s in the final matrix.
      End if
      5.f. While thre_sumY (I) = = 0
            Counter0 = Counter 0 + 1
            I = I + 1
      End while
      5.e. If (Counter0 <= value0)
```

```
                    Put a single 0 in the final matrix.
                Else
                    Put a two 0s in the final matrix.
End if
End while
```

6. Display the final matrix.

3.6 Pseudocode for each of the previous steps of the proposed methodology:

- After reading the image and de-noising it using erosion filter, a sum of the columns in the entire image is to be found:

  Input: a gel image.

  Output: a 1 dimensional (D) array containing the sum of the image columns.

  Variables: Sum_columns: 1 D array.

```
For each column in image
    For each row in image
        Sum_columns = sum of image values;
    End for
End for
```

Algorithm 3.1: Sum of image columns algorithm.

- Threshold the resulting array of sum of the image columns:

Input: 1D array (Sum_columns).

Output: 1 D array containing the threshold of the sum of the image's columns.

Variables:

a. Threshold_column: 1 dimensional array.

b. Index: an integer counter.

c. Threshold_val: the threshold value entered by the user.

```
For each columns in image
     If (Sum_columns > = threshold_val)
          threshold_column [index] = 1;
     Else
          threshold_column [index] = 0;

     index= index +1;
                              End if
End for
```

Algorithm 3.2: Apply thresholding over sum of columns array algorithm.

- The following algorithm finds the start and end of each column of ones in the resulting threshold_column array:

Input: 1D array (threshold_column).

Output:

a. 1D array containing the indexes of the beginning of each column containing information in the image.

b. 1D array containing the indexes of the beginning of each column not containing information in the image.

Variables:

a. col_index: integers counter for columns indexes.

b. tempO: integers counter for the index of ones.

c. tempZ: integers counter for the index of zeros.

d. Ones: 1D array contains indexes of the ones in threshold_column array.

e. Zeros: 1D array contains indexes of the zeros in threshold_column array.

```
For each columns in threshold_column array
      If (threshold_column = 1)
      Ones [tempO] = col_index;
      ELSE
              IF (threshold_column = 0)
                      Zeros [tempZ] = col_index;
              End if
      End if
End for
```

Algorithm 3.3: Find start and end of columns in columns threshold array algorithm.

- The following algorithm cuts the column containing information from the image and puts the column values in a 1D array:

  Input:

  a. 1D array (Ones).

  b. 1D array (Zeros).

  The image.

  Output: 2D array containing values of the column containing information.

  Variables:

  a. im: 2 dimensional array containing values of the column Containing information.

  b. i: an integer counter.

  c. Col_value: values of pixels in column.

```
FOR each column in image[Ones[0]][i]
    FOR        each row in image[Ones[0]][i]
          im = Col_value;
    End for
End for
```

Algorithm 3.4: Cutting column from image algorithm.

- The function that finds the sum o f the rows in the column that was cut:

  Input: 2D array (im).

  Output: 1 D array containing the sum of rows in the column.

Variables:

a. column_array: 2D array of the cut column.

b. sum_rows: 1D array containing the summation of rows in column.

```
FOR each column in column_array
            FOR each row in column_array
        sum_rows = sum of column_array values;
    End for
End for
```

- Threshold the resulting array of sum of the column rows:

Input: 1 D array (sum_rows).

Output: 1 D array containing the sum of rows in the column.

Variables:

a. threshold_row: 1D array.

b. Index: integers counter.

c. threshold_val: threshold value entered by the user.

```
For each row in image
    If (sum_rows > = threshold_val)
           threshold_row [index] = 1;
    Else
           threshold_row [index] = 0;
    end if
    index = index + 1;
end for
```

Algorithm 3.6: Apply thresholding over sum of column rows array algorithm.

- After getting the threshold of the sum of the rows in the spitted column further operations are conducted in order to get to the final matrix:

Input: 1 D array (threshold_row).

Output: final matrix of 0s and 1s.

Variables:

a. threshold_row: 1D array represents threshold_row.

b. I, Counter1, Counter2,m: integers counter.

c. Value1, value2: certain integer values.

d. Mat: final 0\1 matrix.

```
While ( thre_sumY [I] = = 0)
      Skip;
      I = I + 1;
End while
While (thre_sumY [I] = = 1)
      Counter1 = Counter 1 + 1;
      I = I + 1;
End while
If (Counter1 < = value1)
      Mat[m] = 1;
Else
      Mat[m] = 1;
      Mat[m + 1] = 1;
End if

While ( thre_sumY [I] = = 0)
      Counter0 = Counter 0 + 1;
      I = I + 1;
End while
If (Counter0 < = value0)
      Mat[m] = 0;
Else
      Mat[m] = 0;
Mat[m+1] = 0;
End if
```

Algorithm 3.7: The final matrix algorithm.

HAPTER 4

Implementation and Result

4.1 Approaches and Specifications

4.2 Data Specification

4.3 Results and discussion

HAPTER 4

Implementation and Result

4.1 Approaches and Specifications

4.2 Data Specification

4.3 Results and discussion

Chapter 4: Experiments and Results:

In this chapter some alternative experiments that we tried to use in order to solve the project problems will be discussed, the first idea is to apply thresholding on the image, the second is to convert all the bands' intensities depending on a chosen band and the third idea is to do thresholding on the sum of the columns in the image then to do another thresholding on the sum of rows in each column.

We'll also describe the data specification of the project which will include a detailed description of the steps used in the methodology of solving the project problem and an overview of the proposed application interface screen. we'll end the chapter with the results of the proposed application and a brief discussion about those results.

4.1. Approaches and Specifications:

As we were searching for an appropriate solution for the project problem we thought of several ideas, having to choose between all the alternatives to get to the best methodology that gives us the best results was not easy, so we used MATLAB program in order to set our minds on a particular methodology and test the results of it. But our application needed a certain amount of user interaction that was not possible to be built using MATLAB so we had to rebuild the application using Java Net Beans environment. Here are the alternatives that we experimented with a description of each alternative:

- ➢ The first experiment that we tried was to convert the image into a binary image where the user chooses a certain threshold and the values of the image pixels are turned into 0 or 255 (black or white), 255 for the pixel value that is larger than the chosen threshold value and 0 for the pixel value that is less than the chosen threshold value. This alternative was not successful because no matter what the chosen threshold was some bands will be lost and that was not acceptable of what so ever.

- ➢ The second experimented alternative was that the user gets to click on a certain band and the program calculates the intensity of the chosen band then makes all the bands in the image have the same intensity as the chosen band, this alternative was also not accepted for two reason:

1. Some bands will merge into each other!
2. Some weak bands that should not be read as a band in the final matrix will have a strong intensity by the conversion and will be read as a band in the matrix.

➢ The methodology that we used to solve the project problem is the third and last alternative, this approach finds the summation of the images' columns and puts the result in an array then finds the summation of the rows in each column in the array and puts the result in another array, after that the user chooses a threshold for the columns array and another threshold for the rows array, the final matrix is presented after a certain number of operations are done on the two arrays.

4.2. Data Specifications:

4.2.1. Project solution description:

Here is a detailed description of the steps used in the methodology of the proposed application:

1. Read the image using the image path and name taken from the brows widow after the user selects the image to be analyzed.
2. Get the image high and width.
3. Apply dilation filter on the image using a certain structure element.
4. Convert RGB to grayscale.
5. Put the conversion values in a 2 dimensional array of integers.
6. Find the summation of all the columns in the image and put the results in a one dimensional array that is the same size as the images' width.
7. Apply thresholding on the previous array where the threshold value is selected by the user, each value in the array higher than the threshold value is set to 1 and each value in the array lower than the threshold value is set to 0.
8. Using the previous array, find the indexes of the beginning of each ones area and save the indexes in another array (Os), using the same array find the indexes of the end of each ones area and save the indexes in another array (Oe).
9. Os and Oe arrays are used to find the beginning and end of each column that contains information and which will be cut for further operations.

10. Using Os [0]: contains the position of the first 1 in the column containing information and Oe [0]: contains the position of the last 1 in the column containing information, cut the column from the image starting from Os [0] to Oe [0] and save the column values in a one dimensional array of integers.
11. Find the summation of the rows in the previous column array and save the results in a one dimensional array of integers.
12. Apply thresholding on the previous array where the threshold value is selected by the user, each value in the array higher than the threshold value is set to 1 and each value in the array lower than the threshold value is set to 0.
13. Declare an array than will contain the 0s and 1s that represent the image.
14. Start counting the number of ones in the array until you get to the first zero if the number of ones is higher than 0 and less than or equal 30 put a single 1 in the final array, if the number of ones is higher than 30 put a two 1s in the final array.
15. Start counting the number of zeros in the array until you get to the first one if the number of zeros is higher than 0 and less than or equal 20 put a single 0 in the final array, if the number of zeros is higher than 20 put a two 0s in the final array.
16. Continue until all the columns in the image are analyzed and filled in the final matrix as 1s and 0s.

4.2.2. Detailed pseudocode for the project solution:

The following is a detailed pseudocode of the project solution methodology:

- Read a gel image and get the image height ,width and Apply image erosion filter:

Input: image path.

Output: de-noised image.

```
Read JPEG gel image;
Get image width;
Get image height;
Read structure element = {1, 1, 1, 1, 1, 1, 1, 1, 1};
Declare 3*3 Kernel
Add kernel (structure element);
Declare ParameterBlock;
Add image and kernel into ParameterBlock;
Declare RenderingHints;
Add RenderingHints into PlanarImage;
```

Algorithm 4.1: Applying erosion filter algorithm.

- Convert the image into 2 dimensional array:

  Input: gel image.

  Output: 2D array that contains the images' pixel values.

  Variables: Val: integer value of current pixel.

```
Declare a 2Darray as image high and width;
FOR each row in image
    FOR each column in image
        Val = RGB value from current pixel;
        Adjust val between 0 and 255;
        2Darray = val;
    End for
End for
```

Algorithm 4.2: Convert image into 2D array algorithm.

- Find the summation of the columns in the entire image:

Input: 2D array of the image.

Output: 1D array of sum of columns in image (sum_columns).

Variables:

a. sum_columns: 1D array.

b. Total: integer represents the total of sum.

c. I, j: integer counters

```
Declare sum_columns as 1D array;
FOR each column in image
    total = 0;
              FOR each row in image
        total = 2Darray + total;
    end for
sum_columns = total;
end for
```

Algorithm 4.3: Sum of image columns algorithm.

- Apply thresholding over the summation of the image columns:

Input:

a. 1D array (sum_column).

b. Threshold value entered by the user.

Output: 1D array (threshold_column).

Variables:

a. threshold_columns: 1D array.

b. index: integer counter.

c. The_val: Threshold value entered by the user.

```
Declare threshold_column as 1D array;
Declare index;
FOR each column in image
IF(sum_columns >= thr_val
            threshold_column [index] = 1;
       ELSE
            threshold_column [index] = 0;

index++;
End if
End for
```

Algorithm 4.4: Apply thresholding over sum_column array algorithm.

- The following code finds the start and end of each column in the resulting threshold of the sum of image columns:

Variables:

a. threshold_column, Ones, Zeros, Os, Oe, Zs, Ze: 1D array.

b. col_index, tempO, tempZ, tmp_zs, tmp_ze, tmp_os, tmp_oe: integer counter.

c. The_val: Threshold value entered by the user.

```
Declare  col_index;
FOR each column in threshold_column array
     IF (threshold_column = = 1)
                    Ones [tempo]  = col_index;
          tempo ++ ;
     end if
     ELSE
IF (threshold_column = = 0)
          Zeros [tempZ] = col_index;
          tempZ ++;

End if
End for
```

Algorithm 4.5: Find start and end of column containing information algorithm.

```
Os [0] = Ones [0];

FOR each column in Zeros
        IF(Zeros [0] > Os [0])
                Os [counter] = Zeros[counter];
                counter ++;
        end if
end for
Declare tmp_os;
Declare index;
FOR each column in Ones
        IF ( One[0]s > Oe [index])
                Os [tmp_os] = Ones [index];
                tmp_os ++;
        end if
end for
Declare tmp_oe;
FOR each column in Zeros
        IF(Zeros [0]> Os [index])
                Oe [tmp_oe] = Zeros [index];
                tmp_oe ++;
        end if
end for
Declare tmp_zs;
FOR each column in Zeros
        IF(Zeros [0] > Os [index])
        Zs [tmp_zs] = Zeros [index];
                tmp_zs ++;
Declare  tmp_ze;
FOR a number of Ones columns
        IF ( Ones [0] > oe [index])
                Ze [tmp_ze] = Ones [index];
                        tmp_ze ++;
        end if
end for
```

```
Declare tmp_zs;
FOR each column in Zeros
      IF(Zeros [0] > Os [index])
            Zs [tmp_zs] = Zeros [index];
            tmp_zs ++;
      end if
end for
Declare  tmp_ze;
FOR each column in Ones
      IF ( Ones [0] > oe [index])
             Ze [tmp_ze] = Ones [index];
                  tmp_ze ++;
      end if
end for
```

Algorithm 4.6: Find the indexes of the start and end of column algorithm.

- The following function cuts the columns from the image depending on Os and Oe:

  Input:

  a. Image.

  b. Os[0], Oe[0].

  Output: 2D array containing cut column values (im).

  Variables:

  a. col_value: value of pixel in cut column.

  b. Index, w1, i, j: integers values.

```
Declare w1 = (Oe [0] - Os [0]) + 1;
Declare im as 2D array;
FOR eah column i < = w1
      FOR        each row j < =h
              im    = col_value;
      end for
end for
```

Algorithm 4.7: Cutting image column algorithm.

NOTE : After doing certain amount of experiments w found that h = 316.

- The function that finds the summation o f the rows in the column that was cut:

Input: The previous resulting array (im).

Output: 1D array containing sum of the rows in im (sum_row).

Variables: Total, i, j: integers values.

```
declare sum_rows as 1D array;
FOR each column in im
    total = 0;
        FOR each row in im
            Total = total + 2Darray;
        End for
    sum_rows = total;
end for
```

Algorithm 4.8: Sum of row for cut column algorithm.

- Apply thresholding over the summation of the column rows:

Input: 1D array (sum_row), threshold value entered by the user (thr_val).

Output: 1D array (threshold_row).

Variables: index: integer counter.

```
Declare threshold_row as 1D array ;
Declare index;
FOR eah column in sum_rows
    IF(sum_rows [index] > = thr_val)
        threshold_row [index] = 1;
    ELSE
        threshold_row at index = 0;
    end if
    index ++;
end for
```

Algorithm 4.9: Apply thresholding over sum of column rows array algorithm.

- After getting the threshold of the sum of the rows in the spitted column we do the following operations in order to get to the final matrix:

  Input: threshold_rows.

  Output: final (0\1) matrix.

  Variables: index: integer counter.

```
Declare x as integer arraylist;
WHILE ( emptze = = 0)
Declare f as an integers arraylist;
x.add(f);
IF( threshold_row [0] = = 0)
    WHILE (threshold_row [I] = = 0 )
        I = I +1;
        A1 = I;
    End while
End if
WHILE ( A1 < (length_threshold_row -1 ))
Declare c;
WHILE (threshold_row [A1] = = 1 )
    IF( A1 > = (length_threshold_row -1 ))
        Break;
    End if

    c = c +1;
    A1=A1+1;
End while
IF  ((c <= 30) and (c > 0))
    f.add(1);
ELSE
    f.add(1);
    f.add(1);
end if
```

جامعة بوليتكنك فلسطين
Palestine Polytechnic University
(PPU)
The Library المكتبة
24827 الرقم المتسلسل
Acc.
Class. رقم التصنيف

```
WHILE (threshold_row [A1] = = 0 )
        IF (A1 > = (length_threshold_row -1))
                break;
        end if
        w1 ++;
        A1 ++;
End while
IF (w1 < = 20 and w1> 0)
        f.add(0);
Else
        f.add(0);
        f.add(0);
end if
A1=0;
End while
End while
End while
```

Algorithm 4.10: Calculate final (0/1) matrix algorithm.

4.2.3. Application libraries:

We used a number of libraries that we needed in order to build the proposed application to apply some processing operations on the image, here are the used libraries:

- javax.swing .*.
- java.awt .*.
- java.awt.image .* (needed for image objects like Image , bufferImage).
- javax.media.jai.JAI (needed for filters) .
- java.io .* (needed for File) .
- javax.imageio.*; (needed for ImageIO)

4.2.4. Application interface:

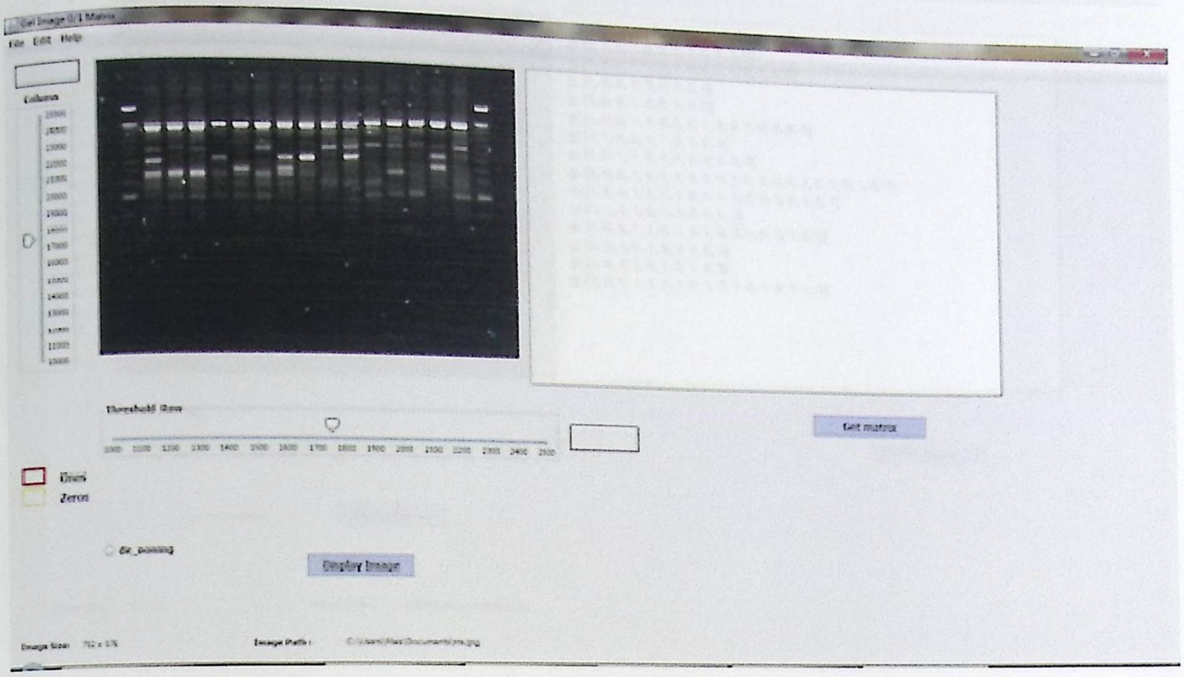


Figure 4.1 Snapshot of the application screen.

As for the application interface shown in figure 4.1, the user can browse his computer for a gel image in order to choose the image to be analyzed, after choosing the image it can be displayed on the screen by clicking on the display button, the user can choose to de-noise the image or not and see the effect of the de-noising on the image, then the user has to determine the threshold using a slider and see the effect of the chosen threshold on the matrix and on the image and by this the application is considered as user interactive (this certain amount of user interaction was asked to be done by the biology lab specialists), the interface is considered flexible where if the user doesn't like the effects of the chosen thresholds it can be changed any time without any problems, after determining the thresholds the user can click on the get matrix button to display the wanted matrix, finally by choosing save from the menu the user can save the matrix to notepad text file and it also saves the values of the chosen thresholds. Figure 4.2 show the interface of (0/1) matrix of gel image application.

4.2.4. Application interface:

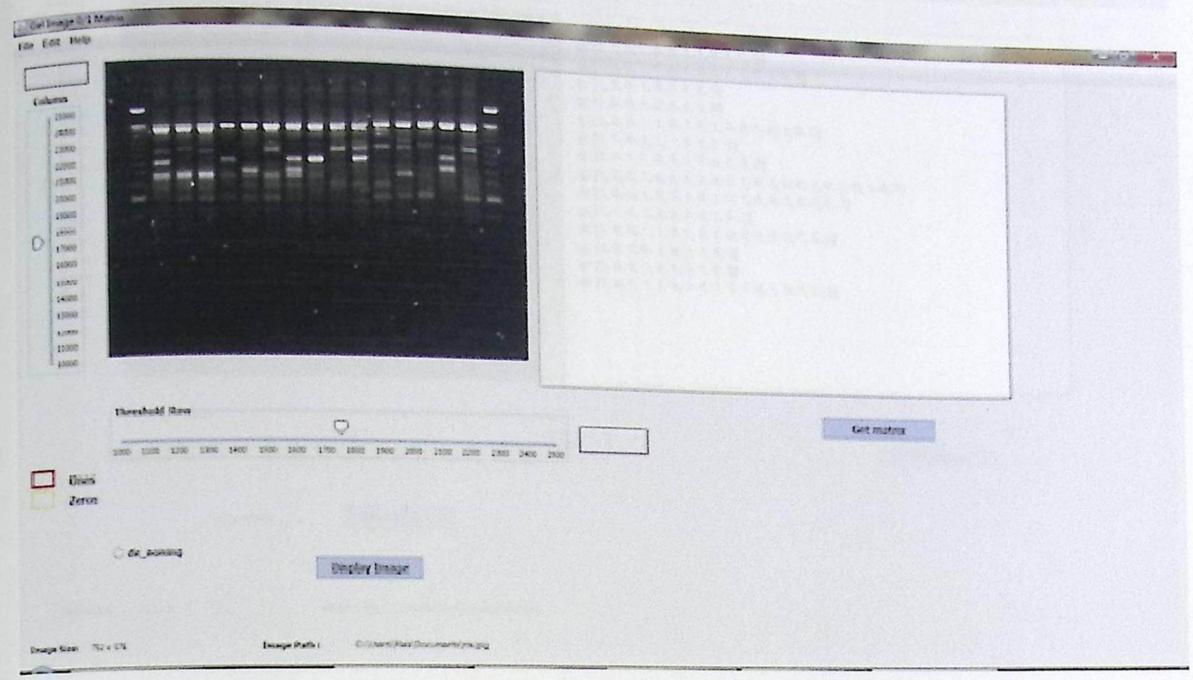
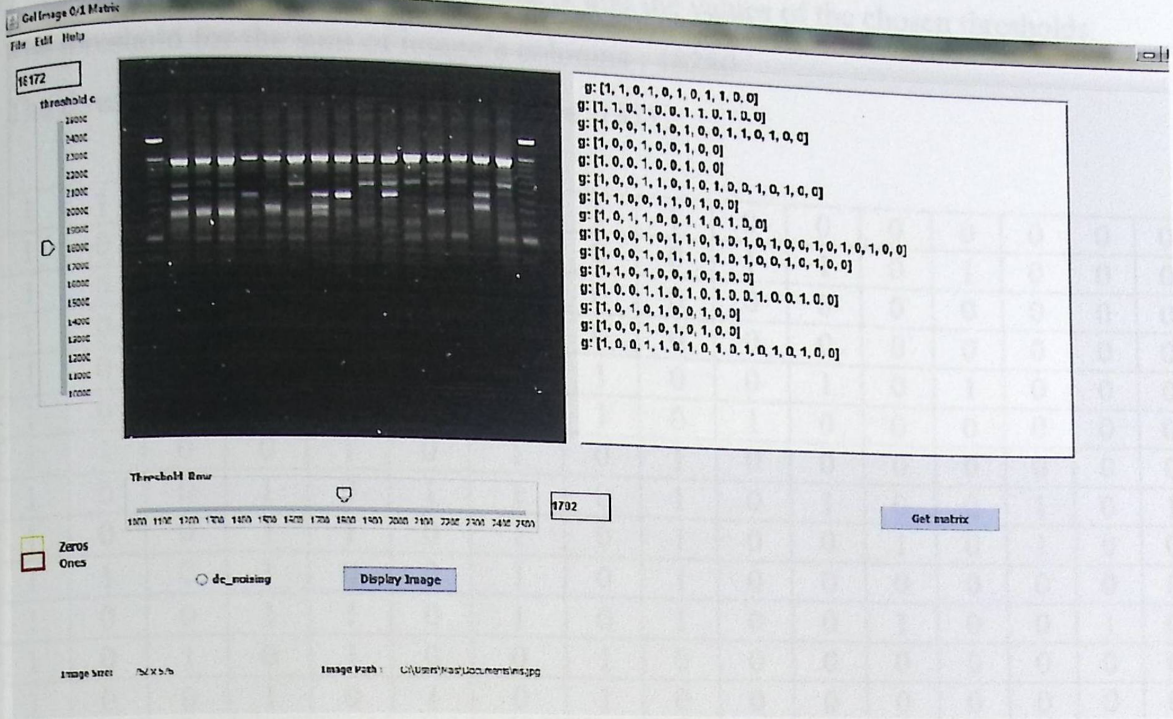


Figure 4.1 Snapshot of the application screen.

As for the application interface shown in figure 4.1, the user can browse his computer for a gel image in order to choose the image to be analyzed, after choosing the image it can be displayed on the screen by clicking on the display button, the user can choose to de-noise the image or not and see the effect of the de-noising on the image, then the user has to determine the threshold using a slider and see the effect of the chosen threshold on the matrix and on the image and by this the application is considered as user interactive (this certain amount of user interaction was asked to be done by the biology lab specialists), the interface is considered flexible where if the user doesn't like the effects of the chosen thresholds it can be changed any time without any problems, after determining the thresholds the user can click on the get matrix button to display the wanted matrix, finally by choosing save from the menu the user can save the matrix to notepad text file and it also saves the values of the chosen thresholds. Figure 4.2 show the interface of (0/1) matrix of gel image application.



16172

1702

752 X 575

C:\Users\Mas\Documents\ns.jpg

```
g: [1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0]
g: [1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 0, 0, 1, 0, 0]
g: [1, 0, 0, 1, 0, 0, 1, 0, 0]
g: [1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0]
g: [1, 1, 0, 0, 1, 1, 0, 1, 0, 0]
g: [1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0]
g: [1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0]
g: [1, 0, 1, 0, 1, 0, 0, 1, 0, 0]
g: [1, 0, 0, 1, 0, 1, 0, 1, 0, 0]
g: [1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0]
```

Figure 4.2 Snapshot of the application screen.

4.3. Results and Discussion:

4.3.1 The project results:

The result of the proposed project is an application that reads an electrophoresis image and analyzes it into a 0/1 matrix that shows the presence or absence of bands in the gel image, by this it is fair to say that the project purpose is met. The following is a gel image that will be analyzed through the application, the resulting 0/1 matrix, and a discussion of the result:

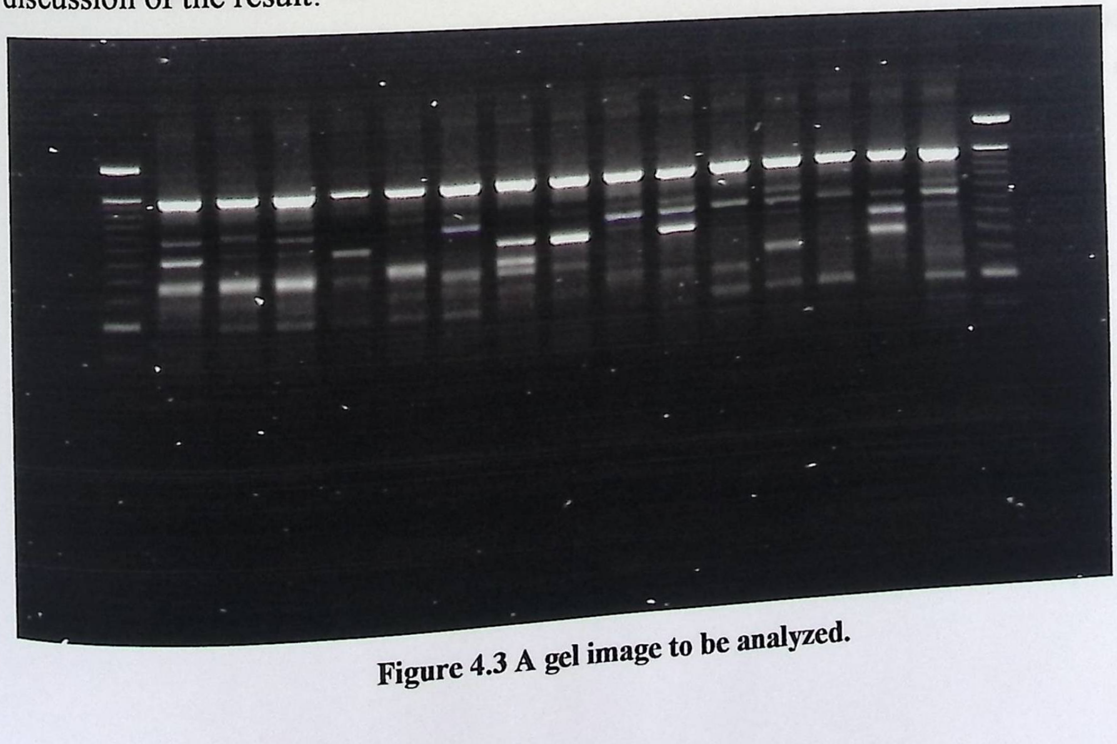


Figure 4.3 A gel image to be analyzed.

The following table is the resulting matrix and the values of the chosen thresholds:

The threshold for the sum of image's columns : 18280

The threshold for the sum of column rows: 1811

1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	1	0	0	1	0	1	0	0	0
1	0	1	1	0	0	1	1	0	1	0	0	0	0	0	0
1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	1	0	1	0	1	0	0	1	0	1
1	0	0	1	1	0	1	0	1	0	0	1	0	1	0	0
1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	1	0	0	1	0	0	1	0
1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0

Table 4.1: matrix resulting from application

4.3.2. Discussion:

The proposed application gives the results of analyzing the gel image according to the users' choice of whether to de-noise the image or not and on the chosen thresholds, so it is right to say that the error rate of the resulting matrix depends on the user choices.

Nevertheless we used MATLAB program to do the testing, where 100 images analyzed manually and the resulted matrices were entered to the program and compared with the matrix that results from the application analysis choosing the right thresholds the matrix resulting from the application has an error rate that is approximately 8.5 %.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

5.2 Future work

Chapter 5: Conclusions and Future Work

5.1 Conclusion:

The electrophoresis gel images produced by DNA marker systems are very important in the process of determining the genetic diversity between individuals, the studying of this field in biology required from the analyzers to convert the electrophoresis gel image into a 0/1 matrix manually which may contain errors and considered as time and effort consuming, in our project we aim to build an application that does the conversion automatically which solves the manually conversion problems. We faced several limitations while trying to solve the project problem, we solved of them, and here is a list of these limitations and a brief description of how they were solved:

1. Noise: was solved using erosion filter.
2. Different band sizes, intensities and distance between them in the same column, each band is read and analyzed separately from the others.
3. Amount of information in each column differs from the others: the application analyzes each column separately from the others.
4. Images sizes vary from one to another, the application deals with all images sizes where each image is scaled in order to be displayed properly on the screen.
5. The application reads the wanted columns only regarding to the missing columns in the image.
6. The application starts analyzing each column starting from the first band in the top of the column.
7. Each image has a different background color, but this does not affect the application analysis of the images where each image intensity differs and this is what matters the most.
8. After the resulting matrix is displayed for the user if any of the analyzed columns is not acceptable form the users' point of view, the chosen thresholds can be changed and so does the matrix.

After we have finished working on the project it is fair to say that the proposed application overcomes some of the limitations that we faced during our research and it meets the goals of the project, the goals that were met are the following:

1. Build software that does the image analysis automatically.
2. Reduce the time required from the analyzer to analyze the image.
3. Reduce the effort required from the analyzer to analyze the image.
4. Consistent image analysis in different times and by different analyzes.
5. The use of an application that is considered user friendly and user interactive.

5.2 Future work:

As for the future work suggested by our team can be in the following matters:

1. There is a constraint mentioned in methodology chapter that was not solved by this application and this constraint is the smiley gel images, future work can be to solve the smiley gel constraint.
2. Future work can also be conducted through connecting the proposed application with other analyzing softwares that do further analysis on the resulting matrices.

References

[1] BARDAKCI.F, 2000 'Random Amplified Polymorphic DNA (RAPD) Markers', T.BÜTAK, 25 (2001) 185-196

[2] Penchevsct , 2007, "polymerase chain reaction" < http://www.youtube.com/watch?v=v4L7rvmBXbY > (access date March- 13 -2009)

[[3] *Agarose Gel Electrophoresis*

< http://people.rit.edu/rhrsbi/GEPages/LabManualPDF5ed/15%20electrophoresis.pdf> (access date April -20-2009)

[4] http://delliss.people.cofc.edu/virtuallabbook/LoadingGel/Minigel2a.jpg (access date june- 12-2009)

[5] " C.A. Glasbey", " M. Berman" ," H.M. Hudson"," Encyclopaedia of Biostatistics: Image analysis and tomography ", (http://www.bioss.ac.uk/staff/chris/eob.pdf),1998. (Access date 15/5/2009).

[6] " A.Gasteratos","I.Andreadis ", "Ph.Tsalides ",http://robotics.pme.duth.gr/pubs/Journals/Realisation%20of%20soft%20morphological%20filters.pdf ", 1998 , (access date 29/5/2009) .

[7] http://www.atalasoft.com/Products/imgxcontrols/docs/MorphologicalFilters.html (access date 28/5/2009)

[8] "Robert E. Sheriff","What Is Deconvolution?",2004," http://www.searchanddiscovery.com/documents/2004/sheriff/images/sherrif.pdf ".(access date 29/5/2009)

[9] R. Fisher, S. Perkins, A. Walker and E. Wolfart.2003 http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm (access date 31/5/2009).

[10] "S.Kother Mohideen", " Dr. S. Arumuga Perumal" , " Dr. M.Mohamed Sathik" ,

"Image De-noising using Discrete Wavelettransform",http://paper.ijcsns.org/07_book/200801/20080129.pdf ,2008 (Access date 31/5/2009)

[11] R. Fisher, S. Perkins, A. Walker and E. Wolfart.2003 , http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm(access date 1/6/2009).

[12] Ruye Wang, 2004, http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node3.html (access date 1/6/2009).

[13] R. Fisher, S. Perkins, A. Walker and E. Wolfart.2003, http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm (access date 1/6/2009).

[14] http://tracer.lcc.uma.es/problems/mfp/mfp.html (access date 1/6/2009).

[15] Dr. Richard Grant, University of Washington, Medical Lake, Washington <http://helpdesk.agri.huji.ac.il/news/gppage.htm>(accessed: May 01, 2009)

[16] Will Dracup,CEO, Nonlinear Dynamics < http://www.nonlinear.com/products/progenesis/samespots/technology.asp>(accessed: May 01, 2009)

[17] application demo(snapshot for the program) <http://www.silkscientific.com/gel/index.htm> (accessed: Feb. 15, 2009) 18=17

[18] <www.ozyme.fr/_prod/gammes/syngene/pdf/ GeneTools%20for%20DNA%20Analysis. pdf>(accessed: May 03, 2009)

Abstract :

Determining the genetic diversity between individuals is one of the most important fields in biology. DNA marker systems have been used in diversity studies where markers are visualized on electrophoresis gel images. Nowadays analyzers convert electrophoresis gel images into (0/1) matrices manually, which causes human errors. Our project aims to overcome these errors by designing a software that converts electrophoresis gel images into (0/1) matrices with minimum errors.