The Islamic University - Gaza

Deanery of Higher Studies

Faculty of Information Technology

Optimum Automated Procedures for Detecting and Mining Microscopic Urine Particles

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Information Technology

by

Mohamed D. Almadhoun

Supervisor
Dr. Alaa Alhalees

1433 H (June, 2012)

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Optimum Automated Procedures for Detecting and Mining Microscopic Urine Particles

Mohamed D. Almadhoun

Abstract

Urine analysis reveals the presence of many problems and diseases in human body. Manual microscopic urine analysis is time consuming, subjective to human observation, and causes mistakes. Computer aided automatic microscopic analysis can overcome these problems.

This research introduces a comprehensive approach for automating procedures for detecting and recognition of microscopic urine particles. Samples of RBC, WBC, epithelial cell, calcium oxalate, triple phosphate, and other undefined images were used in experiments. Experiment was applied in two tracks, first considered vague particles which have light boundaries, and second considered the solid particles which have strong boundaries.

In first experiment, images were segmented, textural features were extracted, features' selection was applied, and five classifiers were tested to get the best results where accuracy of 90.16% was got. In second experiment, image processing functions and segmentation were applied, shape and textural features were extracted, and five classifiers were tested to get the best results where accuracy of 96.41% was got. Repeated experiments were done for adjusting factors to produce the best evaluation results. A very good performance was achieved compared with many related works.

Keywords

Microscopic urine analysis, computer aided medical analysis, automatic recognition, image preprocessing, image segmentation, feature extraction/selection, data mining, classification.

الإجراءات الآلية الأمثل لاستخراج وإدراك جسيمات البول المجهرية

محمد داود المدهون

الملخص

تحليل البول يكشف العديد من المشاكل والأمراض في جسم الانسان . يعتبر تحليل البول المجهري يدويا بواسطة الانسان مضيع ة للوقت، يتبع خبرة الشخص الفاحص، وكذلك يسبب الأخطاء . أما التحليل الذي يتم بواسطة الحاسوب فيمكنه تجاوز هذه المشاكل.

يقدم هذا البحث نهج اً متكاملاً لجعل إجراءات الكشف والتعرف على جسيمات البول المجهرية آليا . وقد تم استخدام صور لعينات مختلفة من كرات الدم الحمراء، كرات الدم البيضاء، خلايا القشور، أملاح الكالسيوم، وأملاح الفوسفات. كما تم إجراء التجربة على طريقتين، الأولى تأخذ بالحسبان الجسيمات غير الواضحة المعالم ذات الحدود الباهتة، والثانية تراعي الجسيمات الواضحة المعالم ذات الحدود البارزة.

في الطريقة الأولى، تمت تجزئة الصور، استخراج الملامح النسيجية، انتقاء بعض الملامح، و تجريب خمس مصنفات للحصول على أفضل النتائج . في الطريقة الثانية، تم تنفيذ عمليات معالجة وتجزئة الصور، استخراج ملامح الشكل والملامح النسيجية، ومن ثم ت جريب خمس مصنفات للحصول على أفضل النتائج . كما تم تكرار العديد من التجارب مع تغيير العوامل في سبيل انتاج أفضل النتائج من التقييم. تم تحقيق أداء جيد جدا بالمقارنة مع الأعمال ذات العلاقة.

الكلمات المفتاحية

تحليل البول المجهري، التحاليل الطبية بمساعدة الحاسوب، الإدراك الآلي، معالجة الصور، تجزئة الصور، استخراج وانتقاء الملامح، تنقيب البيانات، التصنيف.

Dedication

To my family

To my teachers

To my friends

To my colleagues

To Palestine

Acknowledgment

No thank better than thanking Allah who created me and gave me a brain to think, to learn, and to produce, an eye to see, arms to hold, ears to hear, and everything to be a human.

Foremost, I would like to express my sincere gratitude to my advisor Dr. Alaa Alhalees his continuous support of my master study and research, for his patience, motivation, enthusiasm, and knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master thesis.

My sincere thanks also go to University College of Applied Sciences and Alamal Medical Lab for offering the use of their medical equipments and getting urine specimens for capturing.

Last but not the least; I would like to thank my family: my parents, my wife, brother, and sisters for supporting me.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RBC	Red Blood Cell
WBC	White Blood Cell
EP	Epithelial Cell
UTI	Urinary Tract Infection
GLCM	Gray Level Co-occurrence Matrix
ANN	Artificial Neural Network
KNN	K Nearest Neighbors
SVM	Support Vector Machine
CFS	Correlation Feature Selection
AODE	Averaged one-dependence estimators
SVR	Support Vector Regression
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
IPL	Image Processing Library
BLOB	Binary Large Object

Chapter 1. Introduction

This research introduces a comprehensive approach for automating procedures for detecting and recognition of microscopic urine particles. A comprehensive approach will pass through all steps related with recognition systems of urine samples. Automating procedures are those procedures done by computer hardware and software tools to replace the manual actions. Detecting urine particles occurs through specifying those regions that contain target objects and extract them. Recognition process is used for defining each detected particle to which group of urine particles it belongs to.

By this chapter a set of basic ideas and concepts related to thesis field of study will be introduced, which consists of: urine analysis definition and approaching to computer aided medical analysis, motivations behind this research, studies presented in the same domain, objectives of thesis, main idea of research methodology, scope and limitations, and thesis structure.

1.1. Urine Analysis

Microscopic urinalysis detects the presence of elements that often provide early diagnostic information concerning dysfunction, infection, or inflammation of the kidneys and urinary tract. The traditional urine sediment test can be applied by ocular inspection conducted on the smear of the urine sediment. It aims to obtain quantity information of particles in the urine sediment which is mainly composed by red blood cells (RBC), white blood cells (WBC), epithelial cells (EP), yeast-like cells, hyaline or pathological casts, crystals and mucus. This test is always subject to the experience, skills and visual differences of those who conduct the test and thus very subjective [3] [4] [56] . Usually, the results are estimated numbers because the quantity of checked samples is very large and thus mistakes occur [54] . Also, routine manual way is tedious and time-consuming for operators [56] . In addition, it causes heavy workload, slow pace of identification [49] . A better way is to use computer aided analysis.

1.2. Computer Aided Urine Analysis

With the application of computer technology, image processing, and pattern recognition to the field of medicine, it is possible to establish an automatic recognition system of urine sediment, which can help pathologist cumulate diagnosis experiences, and get the benefits of computer's visual resolution factor and the characteristics of flexibility [49] . Computer aided medical analysis is using computer hardware and software tools to apply medical analysis instead of human manual procedures. It became one of the most attractive systems used in medical labs [1] . Computer aided urine analysis is carrying out urine tests by using computer hardware and software tools. Digital microscopy is the comprehensive integration of light microscopy and digital imaging. Technically, digital microscopy technology combines optical, electronic, mechanical, digital imaging, image processing, image analysis, artificial intelligence, automated control, networking, and computer technologies [2] [54] .

1.3. Research Motivation

To make computer aided urine analysis systems effective and highly required in medical labs, they should be trusted, and this can be achieved by increasing the rate of successful recognition of specimen objects. In addition, to address variation between using different classifiers, a comparison is needed to locate best suitable classifiers for this field. Also, noisy and undefined particles that appear in lots of urine specimens need a special treatment; in the mean time they were not considered by other researches. This research will investigate best techniques that can be used together to increase success rate, compare different classifiers, and consider undefined particles in the study. These techniques will pass through preprocessing, object extraction, feature extraction, object classification, evaluation, analysis, and testing methods and results.

1.4. Domain Studies

A set of published researches studied this problem from several points of view. Some of them focused on segmentation as a very well segmented image will affect the general accuracy rate of recognition, others focused on proving better recognition methods, and a group of researchers studied the whole process starting from image acquisition reaching to evaluation of counted particle inside input images. However, by investigation it's noticed that these researches suffered from a set of problems which are shown in related works chapter, such as studying just one particle, not considering undefined particles, low performance rates, and problems related with their methodology.

On the other side, this research will go though solving these problems to overcome the problem of incomplete or undependable solutions.

1.5. Problem Statement

Problems of automatic detecting and mining particles of microscopic urine analysis are the unstable success rates and differentiation between researchers on choosing better classifier to recognize particles. This caused weaknesses in computer aided urine analysis applications.

1.6. Objectives

1.6.1. Main objective

Main objective is to investigate the best approach for applying accurate computer aided microscopic urine analysis.

1.6.2. Specific objectives

Specific objectives are:

- Capture many original images with good resolution for different particles of urine samples.
- Investigate best method for preprocessing & segmenting of microscopic urine sample images.
- Extract and select best features of detected objects that produce highest classification rates.
- Investigating best classification model for classifying urine particles.
- Evaluate approaches with respect to accuracy, precision, recall, and f-measure and apply analysis on results.

1.7. Research Methodology

Methodology of this research followed next steps

1. Collecting a big set of images of urine samples captured using a microscope.
2. Image enhancement will be applied to improve the perception of information inside images.

3. Segmentation will be applied to simplify the representation of an image into something that is more meaningful and easier to analyze and to locate objects for better object extraction.
4. Evaluation on different segmentation adjustments will show best approach that maximizes well extraction of particles.
5. Shape and texture features will be extracted to reduce information and speed up the recognition process.
6. Correlation based feature selection will be used to remove redundant features and different experiments will be applied to adjust best correlation.
7. Different classification models with different adjustments will be applied on extracted features.
8. Finally, evaluation techniques will be used for assessing results and to choose the best classification that gets out the best performance.

Figure 1.1 summarizes the past overall recognition process.

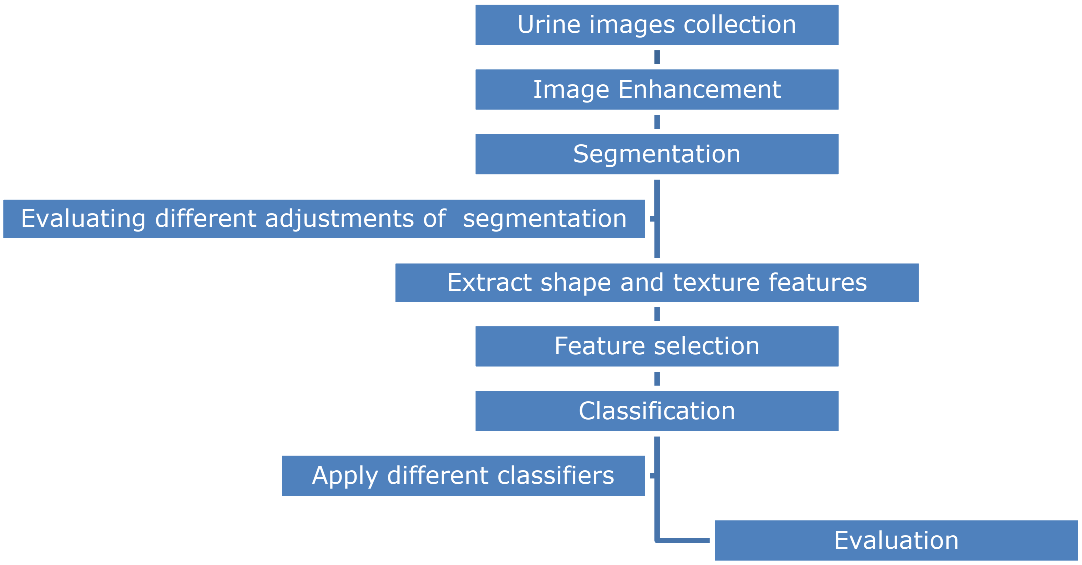

Urine images collection
Image Enhancement
Segmentation
Evaluating different adjustments of segmentation
Extract shape and texture features
Feature selection
Classification
Apply different classifiers
Evaluation

Figure 1.1 Overall recognition process

1.8. Scope and Limitations

Scope of this research is that it will detect and recognize microscopic urine particles.

Limitations of research are:

1. Inability to collect all expected urine particles, where some particles did not appear in any image and some other particles were little to be studied. So, a part of urine particles will be studied by this research.
2. It will not cover the recognition of moving objects, like bacteria.
3. Images' capturing will be done manually without automation neither by auto-focusing nor auto-slide moving.
4. Overlapped and touching particles that can't be split by a normal segmentation process will not be solved by this research.

5. Evaluation methods of vague particles experiment will not count the number of epithelial particles in classification process, but will count square regions that contain epithelial particles or parts of them.

1.9. Significance of the thesis

This study will be a significant endeavor in

- Promoting good improvement on the work environment in the workplace of medical lab technicians.
- Adding a new approach of strong segmentation for urine specimen images.
- Putting an end to differentiation between using classifiers by comparing different classifiers and evaluating results.
- Providing beneficial steps to overcome problem of recognition of noisy and undefined particles inside urine specimen, which was not studied before.

1.10. Thesis Structure

The rest of research is organized as follows: chapter 2 is literature review; chapter 3 is about related works; chapter 4 presents experiment and results; and chapter 5 is the conclusion and future work.

Chapter 2. Literature Review

This chapter aims to review the points of knowledge and concepts that were used by thesis experiments. It starts by defining the computer vision definition and its relation with medical field, and it shows some facts about microscopic medical analysis. After that, image processing concepts that were used are displayed in some details. In addition, it presents image segmentation techniques and dimensionality reduction methods that were used to prepare input data for data mining processes. Finally, it discusses data mining tasks.

2.1. Computer Vision

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images. In general, it works with high-dimensional data from the real world in order to produce numerical or symbolic information. Computer vision is concerned with the theory behind artificial systems that extract information from images [5] [6] .

Following are typical functions which are found in many computer vision systems

1. Image acquisition: it acquires image by one or several image sensors.
2. Pre-processing: it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method, like noise reduction in order to assure that sensor noise does not introduce false information, or contrast enhancement to assure that relevant information can be detected.
3. Feature extraction: image features are extracted from the image data. Typical examples of such features are texture, shape or motion.

4. Detection/segmentation: at some point in the processing a decision is made about which image points or regions of the image are relevant for further processing, like selection of a specific set of interest points.
5. High-level processing: at this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object (like blob extraction or ROI cropping) to be processed in later operations like classifying a detected object into different categories.
6. Decision making: making the final decision required for the application using evaluation methods.

   [8]

Typical tasks of computer vision are recognition, motion analysis, scene reconstruction, and image restoration.

This research works on recognition of medical image which means determining whether or not the image data contains some specific object, feature, or activity.

2.2. Medical Computer Vision

Medical imaging is the technique and process used to create images of the human body (or parts) for clinical purposes (medical procedures seeking to reveal, diagnose or examine disease) or medical science (including the study of normal anatomy and physiology) [9] .

One of the most prominent application fields is medical computer vision or medical image processing. This area is characterized by the extraction of information from image data for the purpose of making a medical diagnosis of a patient [7] .

2.3. Microscopic Urine Analysis

Urine analysis is one of the most famous medical tests. It was defined by [10] as “A urinalysis is an array of tests performed on urine, and one of the most common methods of medical diagnosis. A complete urinalysis includes physical, chemical, and microscopic examinations”.

Urine analysis has a very important role in the diagnosis of urologic conditions such as calculi, urinary tract infection (UTI), and malignancy, and alert to the presence of systemic disease affecting the kidneys. Microscopic examination is an indispensable part of urinalysis; the identification of casts, RBC and WBC cells, crystals, yeast, parasites, and bacteria aids in the diagnosis of a variety of conditions [10] .

To prepare urine sample, well-mixed urine (usually 10-15 ml) is centrifuged in a test tube at relatively low speed (about 2-3,000 rpm) for 5-10 minutes. The supernate is decanted and a volume of 0.2 to 0.5 ml is left inside the tube. A drop of suspended sediment is poured onto a glass slide and coverslipped by a glass cover [11] . Figure 2.1 shows specimen parts

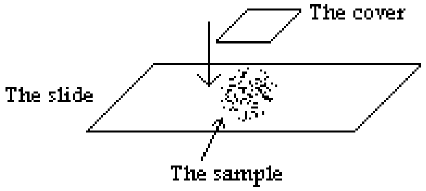


Figure 2.1. Specimen parts

Following is a list of particles' images that appear inside urine specimen under microscope.

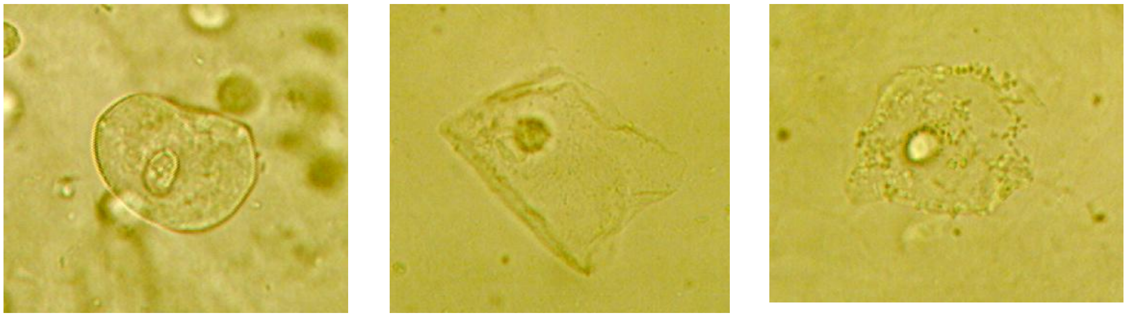


Figure 2.2. Epithelial cells

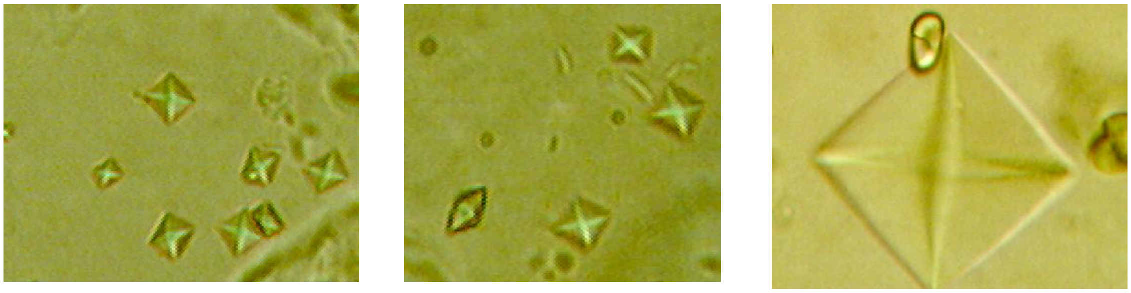


Figure 2.3. Calcium oxalate crystals

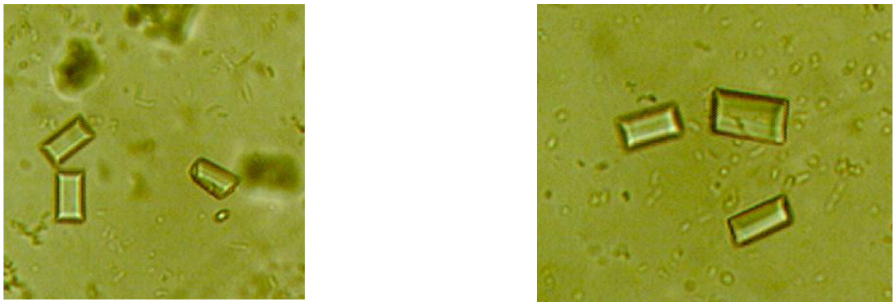
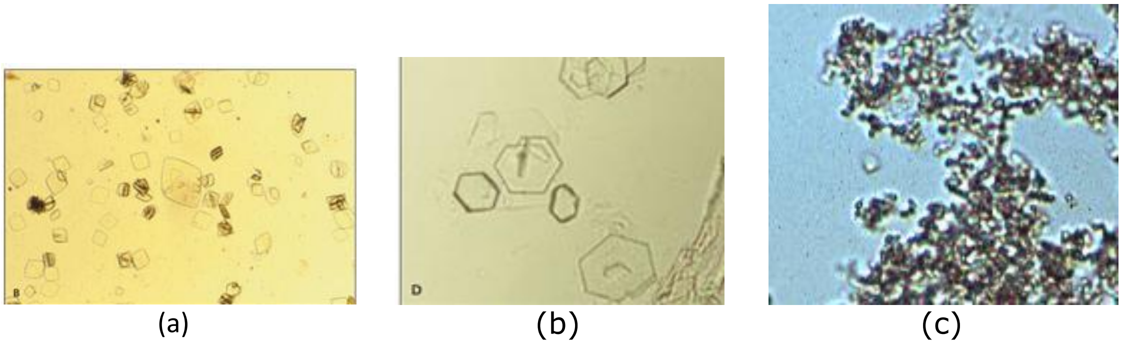


Figure 2.4. Triple phosphate crystals



(a) (b) (c)

Figure 2.5. (a) Uric acid crystals. (b) Cystine crystals. (c) Amorphous crystals.

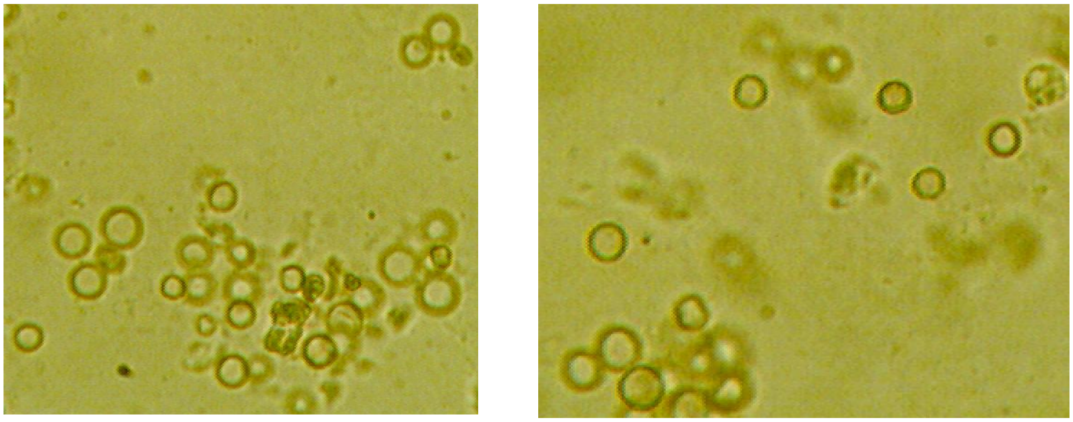


Figure 2.6. Red blood cells

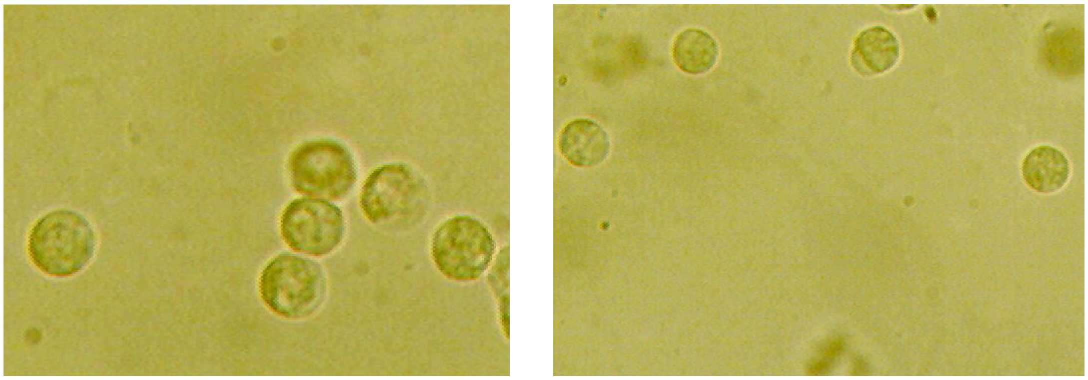
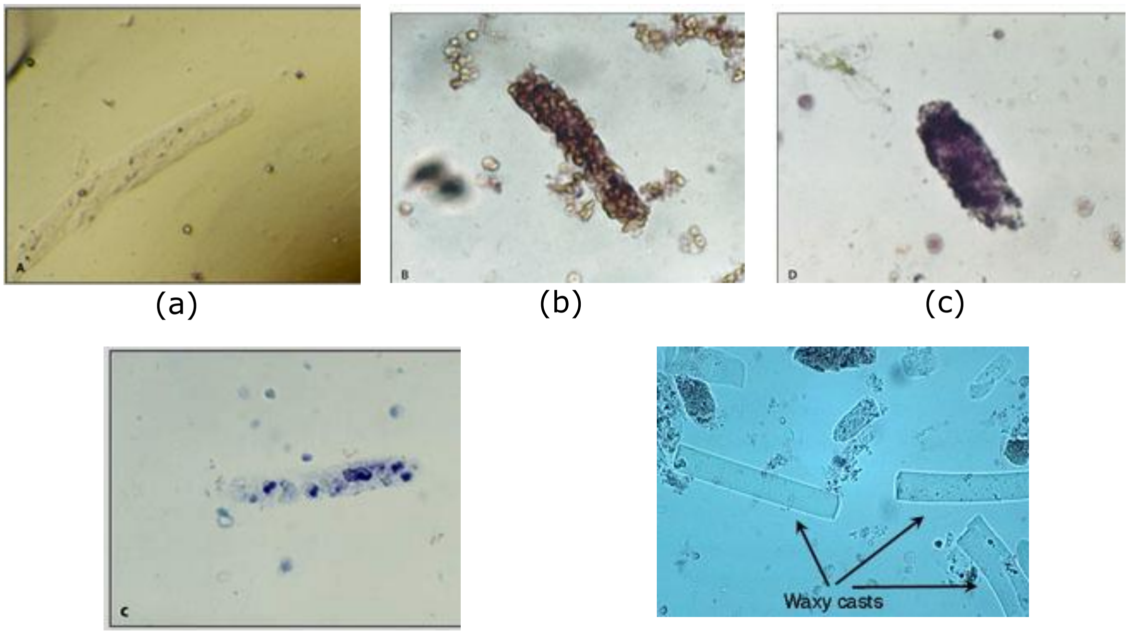


Figure 2.7. White blood cells



(a) (b) (c)

(d) (e)

Figure 2.8. (a) Hyaline cast. (b) Erythrocyte cast. (c) Granular cast. (d) Leukocyte cast. (e) Waxy casts.

2.4. Image Processing

Image processing is techniques by which the data from an image are digitized and various mathematical operations are applied to image data, in order to enhance image to be more useful or interesting to a human observer, or to perform some of the interpretation and recognition tasks [12] .

The most important concepts of image processing used by this research are: Gaussian blue and curvature computing. Following subsections show summarized texts about main ideas of their implementation.

2.4.1. Gaussian Blur

A Gaussian blur (Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, to reduce image noise and reduce detail. Mathematically, applying a Gaussian blur to an image occur by convolving this image with a Gaussian function. Applying a Gaussian blur has the effect of reducing the image's high-frequency components; a Gaussian blur is thus a low pass filter [13] .

The Gaussian blur filter uses a Gaussian function for calculating the transformation to be applied to each pixel in the image. The equation of a Gaussian function in two dimensions is as follows [13] :

$$G(x,y) = \frac{1}{2\prod\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis (x and y are the coordinates of a point inside the convolution matrix), and σ is the standard deviation of the Gaussian distribution. When applied, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases [13] [14] .

2.4.2. Principal Curvatures

Structural image features such as lines, edges, and curves define interest points or regions in an image. Intensity-based interest operators and the object recognition systems based on them often fail to identify discriminative features. The structural features tend to be more robust to express intensities, and pose variations [15] . Curvatures inside images means rate of change in edge direction. Object recognition can be improved by emphasizing object boundaries; extracting principal curvatures can enhance objects' boundaries inside an image.

The used plugin in this research for computing the Principal Curvatures of images works as follows:

1. Compute Gaussian image with input sigma.
2. loop on all image pixels
    a. Compute Hessian matrix for the smoothed image by using the same sigma value.
    b. Compute Eigen values for the calculated Hessian matrix.
3. Output image (curvature map) is a matrix of Eigen values which correspond to the Principal Curvatures.

2.5. Image Segmentation

Segmentation is the process of grouping together pixels that have similar attributes, or it is partitioning a digital image into multiple meaningful regions (groups of pixels). Image segmentation is typically used to locate objects and boundaries in images. Pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture [16] [5] .

Segmentation techniques can be classified as either contextual or non-contextual. Non-contextual techniques ignore the relationships that exist between features in an image, but contextual techniques, on the other hand, exploit the relationships between image features [16] .

Global image threshold operation is a non-contextual segmentation technique because it ignores relationships between image pixels, but pixel connectivity which extracts the adjacent pixels into a set of distinct groups is considered a contextual segmentation because it considered the spatial relationship between pixels.

This research used segmentation by threshold and 8-connectivity blob extraction. Next subsections will show threshold and pixel connectivity in more details.

2.5.1. Threshold

Image threshold is a value that classifies pixels into two categories, first contain pixels with intensity that falls below a threshold, and second contain pixels with intensity that equals or exceeds the threshold. When gradient falls below the threshold output a value is commonly 0, commonly 1 if gradient matches or exceeds the threshold [16] .

Simplest grey level threshold applies to every pixel the following rule [16]

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{otherwise} \end{cases} \qquad (2.2)$$

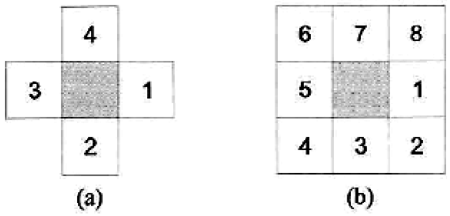
If g(x, y) is a thresholded version of f(x, y) at some global threshold T.

This was the fixed threshold, but another way of applying threshold is called histogram-driven threshold which is chosen from the brightness histogram of the region or image that we wish to segment [17] . Huang [41] , Intermodes [42] , Li [43] , MaxEntropy [44] , Mean [45] , and IsoData [46] thresholds that are used by this research are all global histogram-driven threshold techniques. Fixed and histogram-driven thresholds are considered global thresholds because they uses a global threshold for all pixels, but adaptive threshold (or dynamic, or local) changes the threshold dynamically over the image where threshold has to be calculated for each pixel in the image [18] .

2.5.2. Blob Extraction

Blob extraction is a process to detect connected regions in binary images.

There are two types of neighborhood surrounding a pixel: first is a 4-neighbourhood which contains pixels from above, below, to the left and to the right of the target pixel, second is an 8-neighbourhood which contains all 4-neighbourhood pixels and four diagonal neighbors.



(a) The 4-neighbours of a pixel. (b) The 8-neighbours of a pixel.

Figure 2.9 . Neighborhood of pixels [16]

A 4-connected path from a pixel p_1, to another pixel p_n, is the sequence of pixels {p_1 , p_2, . . . , p_n}, where p_{i+1}, is a 4-neighhour of p_i for all i = 1, . . . , n-1. The path is said to be 8-connected if p_{i+1}, is an 8-neighhour of p_i [16] .

The distinction between 4-connectivity and 8-connectivity can be clear in Figure 2.10 where we can get two 8-connected regions from image part 1 or as six 4-connected regions from image part 2.

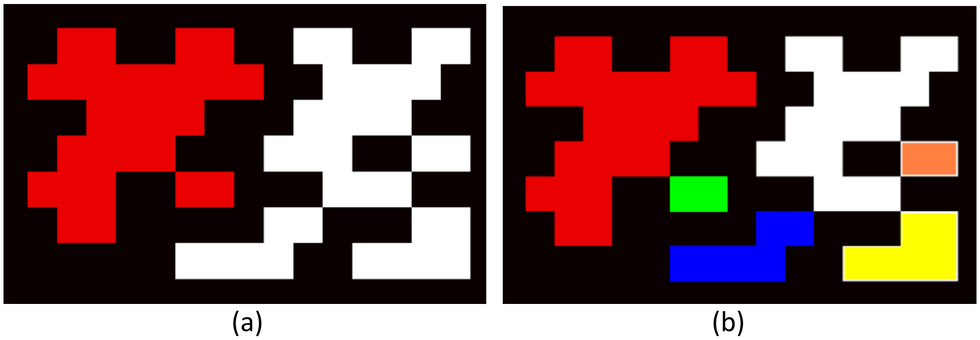


Figure 2.10. A set of connected pixels. (a) Two 8-connected regions. (b) Six 4-connected regions

2.6. Dimensionality Reduction

Dimensionality Reduction is the mapping of data to a lower dimensional space such that uninformative variance in the data is discarded, and can be divided into feature selection and feature extraction [47] .

2.6.1. Feature Extraction

Because processing costs increases when using the whole data of input object, a process of reduction is needed. Feature extraction means transforming the input data into the set of features that represent the original data. This set of features is called feature vector. Shape, histogram, and gray level co-occurrence matrix features are shown next.

2.6.1.1. Shape features

One of most used features in recognition is Hu invariant moments which were introduced by Hu. in [19] , Hu derived six absolute invariants and one skew invariant, these features are independent of position, size and orientation [20] . A set of parameters called moments [21] , moments of order p+q of a region represented by the bitmap $b_{n,m}$ are:

$$M_{p,q} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} n^p m^q b_{n,m} = \sum_{n,m\in\text{region}} n^p m^q \qquad (2.3)$$

a bitmap $b_{n,m}$ means sequence of pixels for some object in the image or may be a contour, n can be x values of pixels, m can be y values of pixels, and $b_{n,m}$ is replaced with the grey level image pixel [21] .

First order moments $M_{0\ 1}$, and $M_{1\ 0}$, are related to the balance point ($\bar{x}$, $\bar{y}$) of the region:

$$\bar{x} = M_{1,0}/M_{0,0} \qquad \text{and} \qquad \bar{y} = M_{0,1}/M_{0,0} \qquad (2.4)$$

The point ($\bar{x}$, $\bar{y}$) is called the centre of gravity, or centroid.

In order to make features' values independent on position, moments can be re-calculated with respect to the centroid. The results are then called central moments (second order moments) which can be calculated by formula (2.5) [21]

$$\mu_{p,q} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} (n-\bar{x})^p (m-\bar{y})^q b_{n,m} = \sum_{n,m\in\text{region}} (n-\bar{x})^p (m-\bar{y})^q \qquad (2.5)$$

Then we can calculate the principal moments by the following formulas

$$\lambda_{\max} = \tfrac{1}{2}\left(\mu_{2,0} + \mu_{0,2}\right) + \tfrac{1}{2}\sqrt{\mu_{2,0}^2 + \mu_{0,2}^2 - 2\mu_{0,2}\mu_{2,0} + 4\mu_{1,1}^2} \qquad (2.6)$$

$$\lambda_{\min} = \tfrac{1}{2}\left(\mu_{2,0} + \mu_{0,2}\right) - \tfrac{1}{2}\sqrt{\mu_{2,0}^2 + \mu_{0,2}^2 - 2\mu_{0,2}\mu_{2,0} + 4\mu_{1,1}^2} \qquad (2.7)$$

Now this leads us to deduce principal component analysis for the image object, see Figure 2.11

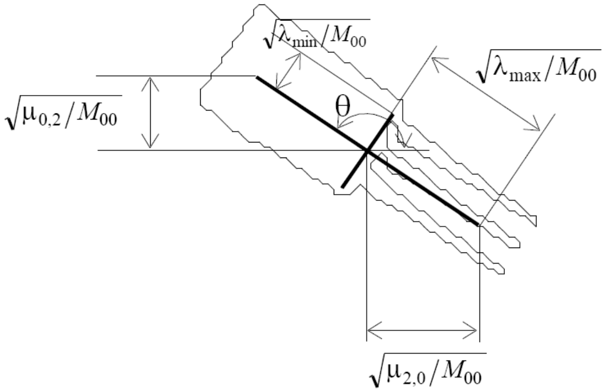

$\sqrt{\lambda_{\min}/M_{00}}$
$\sqrt{\lambda_{\max}/M_{00}}$
$\sqrt{\mu_{0,2}/M_{00}}$
θ
$\sqrt{\mu_{2,0}/M_{00}}$

Figure 2.11. Principal component analysis [21]

Now we can use formula (2.8) to calculate **orientation** of a region from the direction of the largest principal moment

$$\theta = \tan^{-1}\left(\frac{\lambda_{\max} - \mu_{2,0}}{\mu_{1,1}}\right) \tag{2.8}$$

Also, **eccentricity** of a region which can be defined as the ratio between square roots of the two principal moments is calculated by the following formula

$$eccentricity = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \tag{2.9}$$

In addition, $M_{0\ 0}$ is the number of pixels inside the region which is **area** of the region.

To make feature’s values independent on size, the normalized central moments will be calculated using the following formula

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\alpha}} \qquad \text{with:} \qquad \alpha = \frac{p+q}{2} + 1 \tag{2.10}$$

From above, a set of parameters that depend neither on position, nor size, nor orientation was established by **Hu**.

$$
\begin{aligned}
&h_1 = \eta_{2,0} + \eta_{0,2} \qquad (2.11)\\
&h_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2\\
&h_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2\\
&h_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{0,3} + \eta_{2,1})^2\\
&h_5 = (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})\left((\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2\right) +\\
&(3\eta_{2,1} - \eta_{0,3})(\eta_{0,3} + \eta_{2,1})\left(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2\right)\\
&h_6 = (\eta_{2,0} - \eta_{0,2})\left((\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2\right) + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{0,3} + \eta_{2,1})\\
&h_7 = (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})\left((\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2\right) -\\
&(\eta_{3,0} - 3\eta_{1,2})(\eta_{0,3} + \eta_{2,1})\left(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2\right)
\end{aligned}
$$

2.6.1.2. Histogram features

An image histogram is a type of histogram that plots the number of pixels for each intensity value. Histogram reflects information about the intensities distribution. A set of statistical values can be calculated on image histogram vector such as mean, variance, skewness, and kurtosis.

The arithmetic **mean** is the "standard" average which is the sum of the values divided by the number of values, denoted by $\bar{x}$, often simply called the "mean" [22] .

$$Mean = \bar{x} = \frac{1}{n} . \sum_{i=1}^{n} x_i \qquad (2.12)$$

The **variance** is a measure of how far a set of numbers is spread out describing how far the numbers lie from the mean [23] .

In general, the population variance of a finite population of size n is given by

$$Variance = \frac{1}{n}.\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad Where\ \bar{x}\ is\ the\ population\ mean \tag{2.13}$$

Skewness is a measure of the asymmetry of the probability distribution. It can be positive or negative, or even undefined. A negative skew indicates that the majority of values lie to the right of the mean. A positive skew indicates that indicates that the majority of values lie to the left of the mean. A zero value indicates that the values are evenly distributed on both sides of the mean [24] .

For a sample of n values the sample skewness is

$$Skewness = \frac{1}{n}.\sum_{i=0}^{n-1}\left(\frac{x_i - \bar{x}}{\sqrt{Variance}}\right)^3 \tag{2.14}$$

Kurtosis is any measure of the peakedness of the probability distribution[25] .

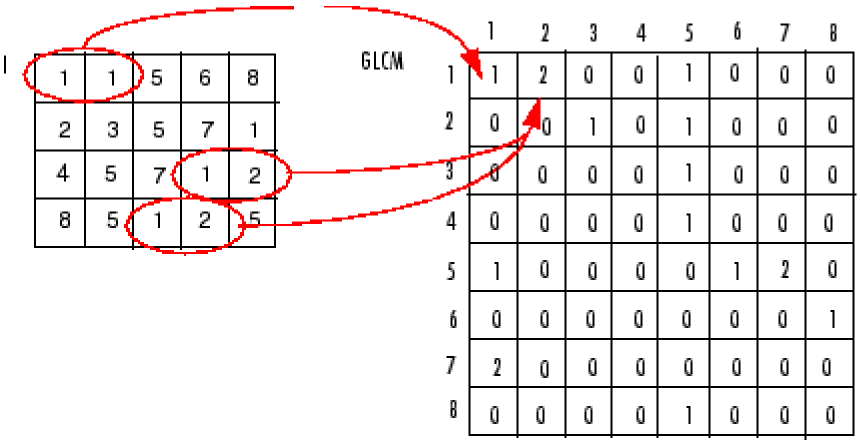
For a sample of n values the sample excess kurtosis is

$$Kurtosis = \frac{1}{n}.\sum_{i=0}^{n-1}\left(\frac{x_i - \bar{x}}{\sqrt{Variance}}\right)^4 - 3 \tag{2.15}$$

2.6.1.3. GLCM features

Shapiro L. and Stockman G. in [5] defined image texture as a set of metrics calculated in image processing to quantify the perceived texture of an image which gives information about the spatial arrangement of color or intensities in an image. Haralick et al. in [26] described texture as one of the important characteristics to identify objects or regions in images.

Gray level co-occurrence matrix (GLCM) is created by calculating how often a pixel with some gray-level (grayscale intensity) value *i* occurs horizontally adjacent to a pixel with gray-level value *j*. Figure 2.12 represents an example of how GLCM can be instantiated from given 4-by-5 image *I* [27] .



1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figure 2.12 . GLCM of input image *I* supposing that image *I* is quantized to 8 gray levels [27]

A main parameter that should be specified and effect GLCM calculation is the offset, which is used to specify other pixel spatial relationships through specifying the distance between the pixel of interest and its neighbor. It consists of two elements, row_offset which is the number of rows between the pixel-of-interest and its neighbor, and col_offset which is the number of columns between the pixel-of-interest and its neighbor. By this parameter we can specify the inter-pixel distance (δ) and orientation (θ). Figure 2.13 shows an example of offset setting [27] .

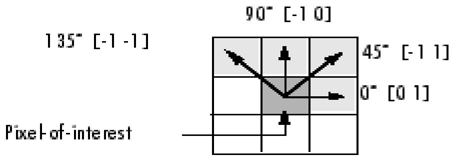


Figure 2.13. GLCM Offsets [27]

Statistics were applied to the co-occurrence probabilities to generate the texture features. These probabilities represent all pair wise combinations of grey levels in the spatial window of interest given two parameters: inter-pixel distance (δ) and orientation (θ).

The probability measure can be defined as:

$$\Pr(x) = \{C_{ij} \mid (\delta, \theta)\} \tag{2.16}$$

where C_{ij} (the co-occurrence probability between grey levels i and j) is defined as:

$$C_{ij} = \frac{P_{ij}}{\sum_{i,j=1}^{G} P_{ij}} \tag{2.17}$$

where P_{ij} represents the number of occurrences of grey levels i and j within the given window (P_{ij}=(i,j)th entry in GLCM), given a certain ((δ),(θ)) pair; and G is the quantized number of grey levels [28] .

A set of statistical features can be calculated from GLCM. Clausi in [28] used a set of textural features in his study. Let C_{ij} be the (i,j)th entry in a normalized GLCM. Soh et. al. in [29] mentioned how mean and standard deviations for rows and columns of the GLCM can be calculated

$$mean_x = \mu_x = \sum_i \sum_j i\,.\,C_{ij} \tag{2.18}$$

$$mean_y = \mu_y = \sum_i \sum_j j\,.\,C_{ij} \tag{2.19}$$

$$Standard\ deviation_x = \sigma_x = \sum_i \sum_j (i - \mu_x)^2\,.\,C_{ij} \tag{2.20}$$

$$Standard\ deviation_y = \sigma_y = \sum_i \sum_j (j - \mu_y)^2 . C_{ij} \tag{2.21}$$

Some of the textural features used by Clausi in [28] are the following

Uniformity (UNI)
$$\sum {C_{ij}}^2 \tag{2.22}$$

Inverse difference (INV)
$$\sum \frac{C_{ij}}{1 + |i - j|} \tag{2.23}$$

Inverse difference moment (IDM)
$$\sum \frac{C_{ij}}{1 + (i - j)^2} \tag{2.24}$$

Correlation (COR)
$$\sum \frac{(i - \mu_x)(j - \mu_y)C_{ij}}{\sigma_x \sigma_y} \tag{2.25}$$

Following are ten textural features used by Soh et. al. in [29] , but they used the symbolic expression *p(i,j)* instead of C_{ij}

Energy:

$$f_1 = \sum_i \sum_j p(i, j)^2. \tag{2.26}$$

Contrast:

$$f_2 = \sum_{n=0}^{N_g - 1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \middle| |i - j| = n \right\}. \tag{2.27}$$

where N_g is the number of gray levels in the quantized image

Correlation:

$$f_3 \frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}. \tag{2.28}$$

Homogeneity:

$$f_4 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j). \tag{2.29}$$

Entropy:

$$f_5 = -\sum_i \sum_j p(i, j) \log(p(i, j)). \tag{2.30}$$

Autocorrelation:

$$f_6 = \sum_i \sum_j (ij)p(i, j). \tag{2.31}$$

Dissimilarity:

$$f_7 = \sum_i \sum_j |i - j| \cdot p(i, j). \tag{2.32}$$

Cluster Shade:

$$f_8 = \sum_i \sum_j (i + j - \mu_x - \mu_y)^3 p(i, j). \tag{2.33}$$

Cluster Prominence:

$$f_9 = \sum_i \sum_j (i + j - \mu_x - \mu_y)^4 p(i, j). \tag{2.34}$$

Maximum Probability:

$$f_{10} = \underset{i,j}{\text{MAX}}\, p(i, j). \tag{2.35}$$

Also, another set of features were used by Santosa et. al. in [58] , but they used the symbolic expression *p(z$_i$,z$_j$)* instead of C_{ij}. Some of them are listed below

$$\textbf{Second order angular moment : } \mathit{AngScMom} = \sum_{z_i=0}^{N-1} \sum_{z_i=0}^{N-1} \left[p\left(z_i, z_j\right)\right]^2 \tag{2.36}$$

$$\textbf{Sum of squares : } \mathit{SumOfSqs} = \sum_{z_i=0}^{N-1} \sum_{z_j=0}^{N-1} (z_i - \mu_x)^2\, p(z_i, z_j) \tag{2.37}$$

$$\textbf{Inverse difference moment : } \mathit{InvDfMom} = \sum_{z_i=0}^{N-1} \sum_{z_j=0}^{N-1} \frac{p(z_i, z_j)}{1 + (z_i - z_j)^2} \tag{2.38}$$

$$\textbf{Sum Average : } \mathit{SumAverg} = \sum_{z_i=0}^{2(N-1)} z_i\, p_{x+y}(z_i)$$

$$p_{x+y}(k) = \sum_{z_i=0}^{N-1} \sum_{z_j=0}^{N-1} p(z_i, z_j) \text{ where } z_i+z_j=k=0,1,2,\ldots,2(N-1) \tag{2.39}$$

$$\textbf{Sum variance : } \mathit{SumVariance} = \sum_{z_i=0}^{2(N-1)} (z_i - \mathit{SumAverg})\, p_{x+y}(z_i) \tag{2.40}$$

$$\textbf{Sum Entropy : } \mathit{SumEntrp} = -\sum_{z_i=0}^{2(N-1)} p_{x+y}(z_i) \log(p_{x+y}(z_i)) \tag{2.41}$$

$$\textbf{Difference variance : } \mathit{DifVarnc} = \sum_{z_i=0}^{2(N-1)} (z_i - \mu_{x-y})^2\, p_{x-y}(z_i)$$

$$p_{x-y}(k) = \sum_{z_i=0}^{N-1} \sum_{z_j=0}^{N-1} p(z_i, z_j) \quad |z_i\text{-}z_j| = k=0,1,2,\ldots,N\text{-}1 \tag{2.42}$$

$$\textbf{Entropy of difference : } \mathit{DifEntrp} = -\sum_{z_i=0}^{2(N-1)} p_{x-y}(z_i) \log(p_{x-y}(z_i)) \tag{2.43}$$

In these expressions, *N* is the number of gray levels, *z$_i$,z$_j$* are the different gray levels (rows and columns in GLCM), *p(z$_i$,z$_j$)* is the value of the GCM at point *(i,j)*, μ_x is the mean value of GCM values accumulated in the *x* direction and $\mu_{x\text{-}y}$ is the mean value of the distribution $p_{x\text{-}y}$.

Also, Haralick et. al. in [22] added the following two features

$$Information\ measure\ of\ correlation1 = \frac{HXY - HXY1}{\max\{HX, HY\}} \tag{2.44}$$

$$Information\ measure\ of\ correlation2 = (1 - \exp[-2.0(HXY2 - HXY])])^{1/2} \tag{2.45}$$

where HX and HY are entropies of p_x and p_y, and

$$HXY = -\sum_i \sum_j p(i,j) \log(p(i,j))$$

$$HXY1 = -\sum_i \sum_j p(i,j) \log(p_x(i) p_y(j))$$

$$HXY2 = -\sum_i \sum_j p_x(i) p_y(j) \log(p_x(i) p_y(j))$$

$$p_x(i) = \sum_{j=1}^{N_g} p(i,j)$$

$$p_y(j) = \sum_{i=1}^{N_g} p(i,j)$$

But, in matlab code which was used by experiment of this research, there is more two features were modified by matlab library, which are:

$$correlation\ (matlab) \frac{\sum_i \sum_j (ij) p(i,j)}{\sigma_x \sigma_y} \tag{2.46}$$

$$Homogeneity\ (matlab) \sum_i \sum_j \frac{1}{1 + |i - j|} p(i,j) \tag{2.47}$$

So, in experiment chapter the term "*Correlation (matlab)*" means that modified correlation by matlab, and the term "*Correlation (paper)*" means that correlation computed by Haralick et. al. in [22].

2.6.2. Feature Selection

Feature selection is a technique for selecting a subset of relevant features from the N-dimensional measurement vector that is most suitable for building robust learning models, by removing the most irrelevant and redundant features from the data [21] [30] .

The Correlation Feature Selection (CFS) measure evaluates subsets of features on the basis of the following hypothesis: "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other" [31] .

The most common measure of correlation is the Pearson Product Moment Correlation. Pearson's correlation reflects the degree of linear relationship between two variables. It ranges from +1 to -1. A correlation of +1 means that there is a perfect positive linear relationship between variables. A correlation of -1 means that there is a perfect negative linear relationship between variables. A correlation of 0 means there is no linear relationship between the two variables. The formula for Pearson's correlation takes on many forms. A commonly used formula is shown below

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}} \qquad (2.48)$$

Where N is the number of samples, X and Y are field values of the two attributes which we want to calculate correlation between each other [32] [33]

2.7. Data mining

Han and Kamber in [34] defined data mining as "Data mining refers to extracting or "mining" knowledge from large amounts of data". According to the Gartner Group, "Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques" [35] . Common stages of Knowledge Discovery in Databases contain selection, pre-processing, transformation, data mining, and evaluation. Data mining involves six common classes of tasks: outlier detection, association rule learning, clustering, classification, regression, and summarization[48] . This research passes through preprocessing, data mining classification, and evaluation. Next subsections show these three concepts in more details.

2.7.1. Preprocessing

As the data may be incomplete (e.g. missing attribute values), noisy (e.g. containing errors, or outlier values that deviate results from the expected), or inconsistent [34] , A preprocessing is needed to modify, remove, or complete data before entering pattern recognition methods.

2.7.1.1. Missing Values

Missing values in database bias result of recognition, so it's important to get rid of it. Following methods are used for clearing data of missing values.

1. Ignore the instance of missing value.
2. Fill in the missing value manually.
3. Use a global constant to fill in the missing value.
4. Use the attribute mean to fill in the missing value.
5. Use the mean of the attribute of missing value for all samples belonging to the same class.
6. Use the most probable value to fill in the missing value.

   [34]

2.7.1.2. Data Reduction

“Data reduction techniques can be applied to obtain a reduced representation of the data set with producing the same (or almost the same) analytical results” [34] . Some of these techniques are

1. Attribute subset selection, where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.
2. Dimensionality reduction or feature selection.
3. Discretization and concept hierarchy generation, where raw data values for attributes are replaced by ranges or higher conceptual levels.

2.7.1.2.1. Discretization

Attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median. In binning operation, values are sorted and then partitioned into equal-frequency bins of some size. In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries and each bin value is then replaced by the closest boundary value [34] .

2.7.1.2.2. Sampling

Sampling allows a large data set to be represented by a much smaller sample (or subset) of the data[34] . One of the most used sampling techniques is stratified sampling which specifies a fixed ratio and selects data instances from each set of instances that belongs to one class according to this ratio, so sampling will cover all class labels without discarding small sized classes.

2.7.1.3. Data Transformation

In data transformation, the data are transformed into forms appropriate for mining. Data transformation can involve smoothing, aggregation, generalization, attribute construction, and normalization [34] .

This research used normalization which makes attribute data to be scaled so as to fall within a small specified range such as -1.0 to 1.0, or 0.0 to 1.0.

2.7.1.3.1. Min–Max Normalization

Min-max normalization works by calculating the difference between field value and the most minimum value of that field attribute, and scaling this difference by the range [35] , formula (2.49) is used in calculating min-max normalization between 0 and 1

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{2.49}$$

2.7.2. Classification

Classification is a data mining task used to predict group membership for data instances [36] . Classification process consists of two steps, first step is the learning or training phase in which classifications algorithm uses the training data set to build classifier (model) by analyzing training data set and their associated class labels. Second step will use the built model for classifying input testing data set; accuracy of classifier can be estimated by comparing the original class labels with the predicted class labels, if the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data instances [34] . This research used the most common classifiers which are naïve byes, k-nearest neighbor, decision tree, neural network, and rule induction. Next subsections show each of them in more details.

2.7.2.1. Naïve Byes

Naïve Bayes classifier is statistical classifier. It can predict class membership probabilities, such as predicting the probability that a given instance belongs to some class. Bayesian classification is based on Bayes' theorem. It's simple and easy to implement, but dependencies between attributes can't be modeled by it [34] .

Following is a simple description of how algorithm works

1. Let D be a training set of instances
2. Class probabilities may be estimated by $P(C_i)=|C_{i,D}|/|D|$,where $|C_{i,D}|$ is the number of training instances of class C_i in D
3. $$\begin{aligned} P(\boldsymbol{X}|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i) \end{aligned}$$, x_k refers to the value of attribute A_k for instance X (k is the number of attributes).
4. In order to predict the class label of $\boldsymbol{X}$, $P(\boldsymbol{X}|C_i)P(C_i)$ is evaluated for each class Ci.

The classifier predicts that the class label of instance $\boldsymbol{X}$ is the class C_i if and only if $P(\boldsymbol{X}|C_i)P(C_i) > P(\boldsymbol{X}|C_j)P(C_j)$
In other words, the predicted class label is the class C_i for which $P(\boldsymbol{X}|C_i)P(C_i)$ is the maximum [34] .

2.7.2.1.1. W-AODE byes classifier

Averaged one-dependence estimators (AODE) is a probabilistic classification learning technique. It was developed to address the attribute-independence problem of the popular naive bayes classifier [39] .

AODE seeks to estimate the probability of each class y given a specified set of features X={x_1, ... x_n}, P(y | x_1, ... x_n) using the following formula

$$\hat{P}(y \mid X) = \frac{\sum_{i:1\leq i\leq n\wedge F(x_i)\geq m} \hat{P}(y, x_i)\prod_{j=1}^{n}\hat{P}(x_j \mid y, x_i)}{\sum_{y'\in Y}\sum_{i:1\leq i\leq n\wedge F(x_i)\geq m}\hat{P}(y', x_i)\prod_{j=1}^{n}\hat{P}(x_j \mid y', x_i)} \qquad (2.50)$$

where F(xi) is a count of the number of training examples having attribute-value xi and m is a user specified minimum frequency with which the term must appear in order to be used in the outer summation.

$\hat{P}(\cdot)$ denotes an estimate of $P(\cdot)$ where AODE provide probability estimates by normalizing the numerator [39] .

2.7.2.2. K nearest Neighbors

K nearest neighbors algorithm is the most used for classification. K nearest neighbors is an example of instance based learning because classifying a new unclassified instance depends on stored training data set [35] . Algorithm of K nearest neighbors works by comparing the unclassified instance to all training data set and finds the most similar k instances, then assigns this instance to the class label that have the majority between the k nearest neighbors. The k nearest neighbors are detected by calculating distance between the input candidate instance and all training data set instances using distance measures such as Euclidean distance or Canberra distance.

Following is a simple description of how algorithm works

1. Determine the value of K
2. Calculate distance between the input instance and all training data set instances

The Canberra distance, d^{CAD}, between two vectors p,q in an n-dimensional vector space is given as follows:

$$d^{CAD}(p,q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{2.51}$$

where $p=(p_1, p_{2,}..., p_n)$ and $q=(q_1, q_{2,}..., q_n)$ are vectors [40] .

3. Sort training data set instances according to their distances, and specify the k instances that have the least distances.
4. Count occurrence times for each class, and detect the majority class that achieved the highest number, and this class will be assigned to the input instance

   [35] .

Main drawback of KNN is choosing the best K that produces the best classification, this problem makes us apply several experiments with changing k to adjust the best k that result best accuracy rate.

2.7.2.3. Decision Tree

"A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label" [34] . The most well-known algorithm for building decision trees is C4.5 which is an extension of Quinlan's earlier ID3 algorithm. C4.5 algorithm uses the concept of information gain or entropy reduction to select optimal split[35] .

Method of classifying instances starts at the root node and traverses the tree on the basis of attribute values of that instance [34] .

Following is a simple description of how algorithm works

1. In the beginning, all training examples are at the root.
2. Recursive partitioning process by choosing attribute each time

    1- Find the best attribute to install at the root node

    Best attribute results the smallest tree, which produces the greatest information gain (Information before split – Information after split)

    2- Split data by applying the root test

    3- At each new node, find the best attribute to install at.

    4- repeat until:

    i. Data can’t be split any more (purity rate reached).

    ii. Tree reached a predetermined max depth.

    iii. There are no remaining attributes on which instances may be partitioned.

    [34]

2.7.2.4. Neural Network

A neural network is a set of connected input/output units in which each connection has a weight associated with it. Backpropagation is a neural network learning algorithm in which the network learns by adjusting weights for prediction. The backpropagation algorithm performs learning on a multilayer feed-forward neural network. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer like the Figure 2.14 [34] .

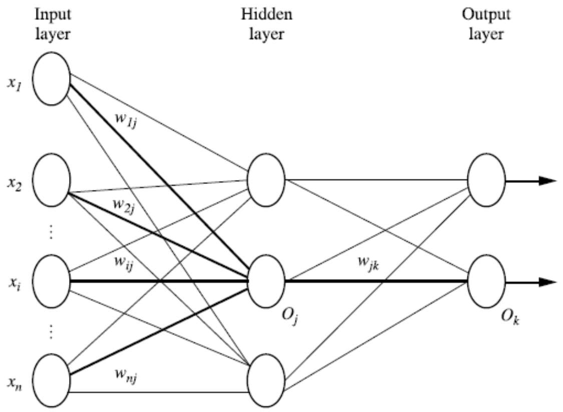


Figure 2.14 . A multilayer feed-forward neural network.

Each layer is made up of units. The inputs to the network correspond to the attributes measured for each training instance. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer of units which is the hidden layer. The outputs of the hidden layer units can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, but usually only one is used. The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given instances. The network is feed-forward in that none of the weights cycles back to a unit of a previous layer [34] . As for number of hidden layers in rapidminer (the tool used in this research for classification tasks), If the user did not specify any hidden layers, a default hidden layer with size of (number of attributes + number of classes) / 2 + 1 will be created and added to the network.

Neural networks have been criticized for their poor interpretability. It is difficult for humans to interpret the symbolic meaning behind the learned weights and hidden units in the network. However, include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained [34] .

2.7.2.5. Rule Induction

A rule-based classifier extracts a set of rules that implies relationships between attributes of a dataset and class label. It produces a model of a set of IF-THEN rules to be used for classification. A rule's coverage is the percentage of instances that are covered by the rule. A rule's accuracy is the percentage of correctly classified instances from those that the rule covers. Coverage and accuracy are defined as

$$coverage(R) = \frac{n_{covers}}{|D|} \tag{2.52}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}} . \tag{2.53}$$

Where D is the number of instances in Data set

The algorithm works by starting from an empty rule, then grow a rule using the Learn-One-Rule function, it then removes training records covered by the rule and repeat this operation until stopping criterion is met [34] .

Another way of rule extraction is from a Decision Tree, by extracting IF-THEN rules from a decision tree. One rule is created for each path from the root to a leaf node. Each splitting criterion along some path in the tree is converted to logical "AND" operation inside the "IF" part. The leaf node holds the class prediction to form the "THEN" part [34] .

Another way of rule induction is by using a Sequential Covering Algorithm, in which Rules are learned one at a time. Each time a rule is learned, the instances covered by that rule are removed, and the process repeats on the remaining instances. Popular sequential covering algorithms include AQ, CN2, and RIPPER [34] .

2.7.3. Evaluation

Evaluation is the key to making real progress in data mining. It determines which method to use on a particular problem and to compare one with another [38] . A set of performance measurement techniques are used for this purpose such as: measuring accuracy, recall, precision, and f-measure. Also a statistical arrangement is called confusion matrix is used to express measurements. In addition, using cross validation as a method of assessing the success labeling of testing data.

2.7.3.1. Accuracy

Results of classification can’t be generalized to new examples if we just rely on training data, but an effective way is to hide some data for testing. This will prevent unexpected poor results and help in extracting best performance of the system. By using a special part of data for testing we can compare predicted answer with the actual label. If the two match, there is no error. If they do not match, then an error has occurred. This is what we call performance measurement which is number of errors divided by the number of examples [37] .

$$Error\ rate = \frac{number\ of\ errors}{number\ of\ examples} \tag{2.54}$$

But in terms of accuracy, we can use the relation obtained by dividing the correct number of classifications by the total number of cases as overall classification accuracy [66] .

2.7.3.2. Precision, Recall, and F Measure

Let's define four useful major terms when analyzing a classifier's ability.

1. True positives refer to the positive instances (instances of the main class of interest) that were correctly labeled by the classifier
2. True negatives are the negative instances that were correctly labeled by the classifier.
3. False positives are the negative instances that were incorrectly labeled.
4. False negatives are the positive instances that were incorrectly labeled [34] .

Precision is the percentage of correctly classified instances of some class between all instances of that class. Recall is the percentage of correctly classified instances of some class between all predicted instances to that class.

F measure is defined as the harmonic mean of precision and recall. It is often used to measure the performance of a system when a single number is preferred. The formal definitions are following [37]

$$\text{precision} = \text{TP}/(\text{TP}+\text{FP}) \qquad (2.55)$$

$$\text{recall} = \text{TP}/(\text{TP}+\text{FN}) \qquad (2.56)$$

$$F \text{ measure} = \frac{2}{1/\text{precision} + 1/\text{recall}} \qquad (2.57)$$

2.7.3.3. Confusion matrix

The confusion matrix is a useful tool for analyzing how well your classifier can recognize instances of different classes [34] .

Given *m* classes, a confusion matrix *CM* is a table of at least size *m* by *m*. An entry, $CM_{i,j}$ in the first *m* rows and *m* columns indicates the number of instances of class *i* that were labeled by the classifier as class *j*. Figure 2.15 shows a sample form of confusion matrix [34] .

	Observed Class A	Observed Class B	Observed Class C
Predicted Class A	15	1	0
Predicted Class B	0	64	3
Predicted Class C	2	0	45

Figure 2.15 . A confusion matrix

For a classifier to have good accuracy, most of instances would be represented along the diagonal of the confusion matrix, from entry $CM_{1,1}$ to entry $CM_{m,m}$ [34] .

2.7.3.4. Cross-validation

In practical terms, the holdout method reserves one-third of the data for testing and use the remaining two-thirds for training. In general, we can't tell whether a sample used for training (or testing) is representative or not. However, an important simple form of statistical technique is called cross-validation. In cross-validation, we decide on a fixed number of folds, or partitions of the data. Suppose we use three. Then the data is split into three approximately equal partitions and each in turn is used for testing and the remainder is used for training, this will repeat the procedure three times. Extensive tests on numerous datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error [38] .

2.8. Summary

In summary, this chapter introduced an overview of the most important used methods, and algorithms in the thesis. The main concepts that it show were: A definition of computer vision and its relationship with medical applications, benefits of microscopic urine analysis and common found particles inside urine specimens, image processing as a good tool for image enhancement and preparation for successful segmentation with details about Gaussian blur and curvature computing, segmentation for isolating the target regions inside an image from other unneeded ones by the aid of threshold and blob extraction, dimensionality reduction using feature extraction and feature selection, and finally data mining by passing through tasks of preprocessing, classification, and evaluation. By total, this forms an overall look at the process of recognition which is the top mission of this thesis.

Chapter 3. Related Works

The thesis focuses on microscopic computer vision from urine samples. Microscopic computer vision is a hard research track and collection of its input data is a hard operation. In addition, available commercial applications that work on that field are little because of its sensitive task and unstable success rates. Despite that, there are number of researches in this area, but still not enough to cover all problems of this issue.

This chapter reflects a number of researches that worked on microscopic computer vision issues especially those who worked on urine samples. Research in this field can be categorized into four tracks:

1. Researches applied all phases starting from data preparation until recognition, and they considered multi-particles in recognition phases.
2. Researches applied all phases starting from data preparation until recognition, but they considered only one particle in recognition phases.
3. Researches applied only some of phases, and they considered multi-particles in recognition phases.
4. Researches applied only some of phases, and they considered only one particle in recognition phases.

The following are researches in each task:

3.1. Apply all phases on multi-particles

The following researches applied all phases starting from data preparation until recognition, and they considered multi-particles in recognition phases.

First, Zhou et. al. in [49] proposed a comprehensive description of the principles and process to develop automatic recognition and count system for urine-sediment visual components. With the help of image processing technology, image segmentation, representation and description, geometric features and texture features have been extracted. They used Gaussian filter, min max threshold, and edge connector in image segmentation. They used neural network classifier based on genetic algorithm for micro image automatic recognition and count system. They stated that neural network can be effectively improved by combining genetic algorithms with proper feature selection. Accuracy rates ranged from 77% for waxy casts to 96% for leukocytes.

The main drawback of this research is that it did not consider the undefined particles in their classification and did not test how the classification model will treat when input data instances have undefined particles. Also they tested only neural network without trying other classifiers. Their results show that they considered all crystals particles are one type and one accuracy rate was for crystals without considering different types of crystals, this causes unstable recognition accuracy when different crystal forms are tested. In addition, our research will improve accuracy of epithelial cells rate more than 6%. They did not consider precision or recall so their evaluation is not guaranteed. They did not show the rate of hold out samples for testing and this affects accuracy too much.

Second, Ranzato et. al. in [50] , proposed a general-purpose system to recognize biological particles. Their system was developed to classify 12 categories of particles found in human urine. Promising locations in the image are detected and small regions containing interesting samples are extracted. The detection algorithm extracts a square bounding box around a particle, and then an invariant feature vector is computed without segmenting the particle from the background of the patch. Bayesian classifier was used with input dataset of 500 images per class. 90% of samples were used for training and 10% for testing. Classification was applied 10 times in each time the 10% of testing samples was chosen randomly, and result was an average of the 10 results. The average error rate was 6.8%. The same experiment was applied also on a set of images of airborne pollen with resulting error rate of 22%, and 35.1% when unknown particles were considered.

The main weaknesses of this research are that their accuracy is not stable and not expressive because they used random selection of the 10% testing samples in the 10 experiments in classification process, and this causes duplication in using the same sample different times, so using cross-validation will be better. They did not consider the unknown particles in classification process of urine samples and did not show how their system may deal with this issue. Also they did not try classifiers other than Bayesian to get the best classifier.

Third, Song et. al. in [51] , proposed a technique for incorporating contextual information into object classification applied to microscopic urinalysis image recognition. They proposed a framework that incorporates partial context at a linear cost unlike previous techniques that cost exponential computation. The automated urinalysis system they developed consists of three steps: image processing and feature extraction, learning and pattern recognition, and context incorporation. Features are fed into neural network with 16 inputs. They stated that augmenting these features by incorporating context into the classification process yielded significant benefits. The algorithm using partial context was tested on a database of 83 urine specimens, containing a total of 20,276 analyte images. Four classes were considered: bacteria, red blood cells, white blood cells and amorphous crystals. The diagnosis for a specimen is either normal or abnormal. A significant improvement of recognition rate achieved by decreasing the error rate from 44.48% before using the partial context to 36.66% after.

The weakness in this research, success rates are low, where the best success rate was 63.34%. Also they did not work on counting the inner particles, but their system diagnoses the specimen by just telling if it's normal or not.

Forth, Hans et. al. in [60] , presented a novel method for urine particle classification. Spectral images were captured which are three dimensional cubes of data comprised of a series of 2D images of different wavelengths. On the bases of spectral images, DAISY descriptors have been used to capture the textural characteristics of each particle and subjected to dimensionality reduction across three linear subspaces which are PCA, ISO, and NPE. Classification is based on a similarity metric to labeled examples in each subspace and the decisions are fused by using support vector regression (SVR). 60% of the images were randomly sampled to generate a training dataset, 25% for validation, and 15% for testing. Analyzed urine particles included a dataset of 452 images for crystals, casts and blood cells, and the results obtained show an average classification accuracy of 92.6% for 6 classes of urinary particles. The use of multispectral data improved classification performance over 11.0% when compared to brightfield data.

The shortcomings in this research are that they did not consider the undefined particles. They did not try different classifiers to get the best. Their segmentation was simple with no preprocessing to get rid of noisy background. They used local adaptive threshold and did not try other threshold techniques. As for accuracy there are two problems in their evaluation, first is splitting images for training, validation, and testing with small number of images like 26 uric acid images, 32 cast images or 47 triple phosphate images, second is that there is no balancing between recall and precision and most values of recall and precision are under 90% which makes classification accuracy unsatisfied.

3.2. Apply all phases on one particle

The following researches applied all phases, but they considered only one particle in recognition phases

First, Li et. al. in [56] , proposed an approach for casts detecting and recognition in urine sediment images, they followed three stages. Firstly, 4-direction variance mapping image was acquired from gray scale image. Secondly, they obtained binary image by applying an improved adaptive bi-threshold segmentation algorithm to the mapping image. Connected domain searching and filling holes technique is applied to the image. In the last stage, five texture and shape characteristics of casts were extracted from both gray scale image and binary image. A decision-tree classifier was developed to distinguish casts from other particles in the image. Total casts number was 232, 161 of them was truly assigned. So, resulting precision in recognizing casts in the urine sediment images was 69.3%.

We can criticize this research in that authors did not try classifiers other than decision tree. They did not evaluate their segmentation, so accuracy rates of segmentation are not available to assure segmentation efficiency. Extracted texture features are not enough to describe particles. As for accuracy rate, 69.3% is far from the accepted range of medial analysis accuracy.

Second, Cao et. al. in [57] , worked in research on detection of red blood cells in urine image captured under microscope. They used an improved Sobel operator for image preprocessing to smooth noise by using eight directions instead of four and edge tracking through sensing similarity between edge points' intensities unlike noisy points. Red blood cells were localized using Hough Transform. Features were extracted and selected by dimension reduction using Principal Component Analysis (PCA). Finally, values of dimensionality reduction were classified by LDA (Linear Discriminant Analysis). The experiment adopted 90 images, 720 images as experimental samples. Select 55 original images as training set (440 samples) and 35 as test set (280 samples) randomly. Best result with the 440 training was 94%, and with the 280 test samples was 91%. Classification rates in the process of training and test changed when PCA energy intensity changes from 0.1 to 0.99.

What we can criticize in this research is that authors did not evaluate their improved Sobel operator and what are improvement success rates. Also, they did not show how Hough Transform localization treats with other particles. There is no consideration for images that contain other particles like WBC which have a similar geometrical circular shape to RBC and test how classifier treats with this issue. In addition, they did not try other classifiers to test which is better to choose. As for accuracy, our research will show an improvement over their accuracy rate of more than 5%.

Third, Santos et. al. in [58] , developed an automatic system for scoring in vitro cultures of renal cells through discrimination and quantification of alive and dead cells microscopy images. Their solution based on texture analysis. 21 images were chosen for training, 222 regions of interest were extracted from them each of which is 60x60 pixel window, and regions were manually classified to alive cells, dead cells or background. Texture parameters were computed from the histogram and gray level co-occurrence matrix. 229 texture parameters were computed. Discriminant analysis were used for feature selection, 12 parameters were selected. Resulting accuracy of classifying regions was 94%.

Critique on this research is that it was good in choosing the textural feature to discriminate regions of renal cells, like our research which will test this approach on epithelial cells inside urine samples of microscopic images. But they did not show what the classifier which was used is, and they did not try different classifiers to test which is better to use. As for accuracy it's not comparable with our accuracy because the field is different.

3.3. Apply some of phases on multi-particles

The following researches applied only some of phases, and they considered multi-particles in recognition phases.

First, Mendez et. al. in [59] , presented a study for classification of urine sediments using neural networks and fractal geometry. An automated microscopy system was used, which is controlled through the computer. Their system improved the precision of the results reported by other authors, using neural networks, from 94%-95% up to 98%.

Deduced shortcomings are that they did not show how image was segmented and how particles were extracted. They show just three extracted features using fractal geometrical methods. They did not mention any thing about input examples and number of particles for each type and percentage of training to testing. They did not consider the undefined particles. As for accuracy they did not mention what was their accuracy, they just mentioned precision which is not expressive alone.

Second, CHEN et. al. in [61] , proposed a method of texture feature extraction using the distance mapping based on a set of local gray value invariants and the feature is robust to the shift and rotation. They reduced the high dimensional feature into a lower dimensional space from feature vector dimension of 432 to 50 using PCA. A multiclass SVM was applied to classify 5 categories of particles cells including red blood cells, white blood cells, epithelial cells, impurity and mucus. Finally the experiment results achieved an average of accuracy of 90.02% and a F1 value of 90.44%. To build a multiclass SVM, they use a package called LIBSVM.

Also the same weaknesses appear in this research as they did not consider the undefined particles. They did not try different classifiers to get the best. They did not show how particles were segmented from original images. As for accuracy our research will add an improvement of more than 6% on their accuracy and F measure.

Third, Mei-li and Rui in [62] , proposed a system to recognize six kinds of urine sediment components: red blood cells, white blood cells, cast, epithelial cells, crystalline, and pus cells. They extracted Harr features, and used AdaBoost to select the best Harr features that distinguish between positive and negative samples. The most optimal classifier was computed by adjusting SVM kernel function and some parameters and using cross-validation method. They got recognition rate of 90%.

Problem of Mei-li and Rui paper is that they did not show how particles were segmented. Also, they did not consider undefined particles. They did not show precision and recall of each particle. As for accuracy our research will add an improvement of more than 6% on their accuracy.

Forth, Calva et. al. in [63] , presented two examples of automated microscopy, a system for urine sediment analysis and a system for analyzing and finding parasites in fecal material. Their target particles were epithelial cells, crystals (triple phosphate, Uric Acid, Calcium Carbonate and oxalate), red blood cells, casts (hyaline, granular, red blood cell, white blood cell, waxy), yeast and bacteria. Feature parameters were extracted from fractal dimension and entropies. Neural networks classifier was used. Best results were obtained with the tests performed when using the NeuroSolution software, they programmed an ANN multilayer perceptron trained with static backpropagation. They stated that the precision increased to 99.2% for the case of urinary sediment, and 92.0% in the case of parasites.

Problems in this research are that they did not mention any thing about dataset and input examples and what is percentage of training to testing. They did not consider the undefined particles. No segmentation was mentioned and how features were extracted. They did not test their methods on noisy backgrounds. They did not list precision of different particles. As for accuracy they did not mention what was their accuracy, they just mentioned precision which is not expressive alone.

Fifth, Li and Zeng in [64] , presented a strategy for segmenting urinary sediment based on wavelet, morphology and combination method. Firstly, they used wavelet transform to remove the effect of defocusing and morphology processing to locate elements, finish the coarse segmentation, and get the subimages that include the particles. Then based on the characteristics of subimages; edge detection and adaptive thresholding are employed adaptively to precisely segment each subimage. Finally, a simplified watershed algorithm to solve the overlapping particles was used. Correct segmentation results were 227 correct of 235 red cells, 33 correct of 34 white cells, 16 correct of 16 casts, 24 correct of 24 epitheliums.

Weaknesses in this research are that they did not show how their algorithm deals with noisy background. They did not try different threshold algorithms to detect the best. Number of tested samples was small.

3.4. Apply some of phases on one particle

The following researches applied some of phases, and considered only one particle in recognition phases.

First, Li et. al. in [54] , proposed an approach for image segmentation and particles detection using Gabor-based feature combining iterative method applied on image to enhance the edge information. Threshold was determined from the average intensity of high gradient pixels. Following an edge-linking procedure. They focused on extracting particles such as long casts in microscopic images. Experiments were applied on 30 visual images including gray images. They stated that the proposed Gabor-based algorithm generated much more accurately bounded regions of objects.

This research tested the proposed segmentation approach on just one type of particles which is casts; they did not try other particles. They did not test how the proposed algorithm treats with noisy regions. They did not show segmentation success rates when using different auto-threshold mechanisms. As for accuracy this research did not use any evaluation technique and no accuracy or performance numbers were mentioned, so it’s not dependable.

Second, Luo et. al. in [55] , proposed a segmentation model based on Mumford-Shah (a tool for image segmentation introduced in [65]). They added a new algorithm to optimize the computational efficiency. The experimental results for a microscopic red cell image of urinary sediment 115x115 pixels reached to promising point after running 240 iterations which made curve stops at the edge of the RBC cell.

Weak points in this research are that they did not show how the algorithm treats with particles other than RBCs. They did not show how the algorithm deals with noisy regions. As for accuracy they did not use any evaluation technique to assess segmentation rates and no accuracy or performance numbers were mentioned. So, it’s not a general case to be used.

3.5. Conclusion

To sum up the problems of related works, we can be concluded that some of them did not study multi particles of urine samples but studied just one, most of them did not consider the undefined particles, most of them did not try different classifiers to compare between them and detect the best. Others who worked on segmentation did not assess success rates of their proposed segmentation, some of them did not apply feature selection, most of them did not consider noisy background in their segmentation, and researches which applied auto-threshold did not try different algorithms and compare. Some of researches’ accuracy was low, others did not evaluate their results well, and some of them did not evaluate their work at all.

In this research, a comprehensive study for the problem of microscopic urine sample is provided through image preprocessing and segmentation, particle localization, feature extraction, and recognition. Different particles will be considered in classification, undefined particles will be entered during classification to assess classifier behavior, different classifiers will be tested to detect the best, feature selection is used to decrease redundant input data to classifier which decreases processing time, noisy background is overcome under proposed segmentation, and different auto threshold algorithms are tested to reach top success rates of segmentation.

Chapter 4. Experiment and Results

This chapter presents the experimental procedures and results of research to prove my research approach. section one displays dataset characteristics and collection, and discussion of treatment tracks that could be there upon data nature; section two lists all details and steps related to vague particles recognition; section three lists all details and steps related to solid particles recognition, and section four presents the problem of undefined particles, and to involve them in proposed solution.

Data images entered a set of processing and data preparation steps to be ready for recognition.

Figure 4.1 shows the work flow of overall experiment, each step matches more details in next sections.

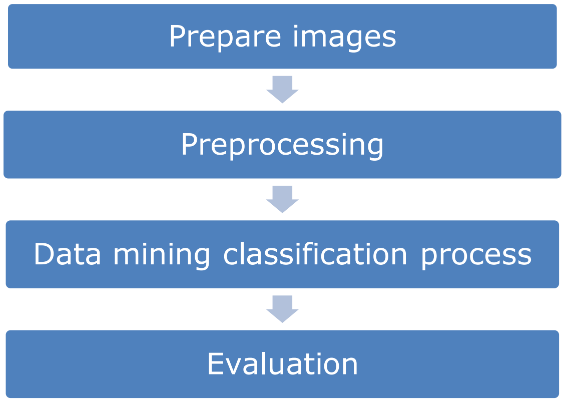


Figure 4.1 . Experiment work flow

In Figure 4.1, preparing images starts by capturing images, and then every image was assigned a unique number and added to its appropriate group. Preprocessing started by image enhancement and processing operations then blob extraction, and manual labeling, and finally features extraction. Data mining classification process used data instances with selected features to build classification models and test them. Evaluation was handled by calculating performance vectors and success and error rates.

4.1. Collected datasets

Experiment depends totally on data images, so as more clear and good images, more accurate results will be there. First step of experiment was capturing images of microscopic urine samples using digital eyepiece camera (MSR350 Digital Eyepiece), microscope, laptop, and the urine sediment sample.

Images were captured in two different places, first was University College of Applied Sciences medical lab (October, 2011), and second was Alamal medical lab (which is private lab and located in Gaza, March 2012).

Images format is PNG with resolution of 1280x960 and RGB color. Number of captured images was 2193. Every image was renamed to the form "Image (X).PNG" where X is the number of image, so every image has a unique number that can be tracked in later analysis, especially to track its records in database. Table 4.1 lists number of images that contain different particles.

After reviewing dataset images, we can notice that not all types of particles can have the same features, such as epithelial cells which have a vague body with very light boundaries which can be lost during preprocessing. This makes it impossible to get geometric features for that type of particles, and its experiment work flow process will be different something.

The change is that epithelial cells did not enter a preprocessing step like other particles, hence we have two tracks of experiment: first is for vague particles, and second for solid particles.

As for applications that will work on this basis, they should be adaptive by applying the two experimental tracks to discover each of vague and solid particles.

Table 4.1 . Number of images in each group

Urine Particle	Number of Images
RBC	340
WBC	296
Epithelial cells	451
Calcium Oxalate	220
Triple Phosphate	152
Undefined	1435
Total	**2894**

4.2. Vague particles experiments

Up to the nature of vague body with light boundaries of epithelial cells, there is no way to get blob of this particle in the binary image, but in contrast to other particles, epithelial cells have a special content with small granules that make it distinguishable about the background and also it has usually a big size which may be many times of other particles size. So it’s better to study the texture of these objects.

Next subsections will show experiment steps to create a model of epithelial cells recognition process.

4.2.1. Prepare Images

Following steps show procedures for preparing images for preprocessing operation

1. Choose a set of images that contain epithelial cells.
2. Resize Image from 1280x960 to 640x480 to minimize processing time required.
3. Split each image into 9 parts, each part of 320x240 as shown in the Figure 4.2

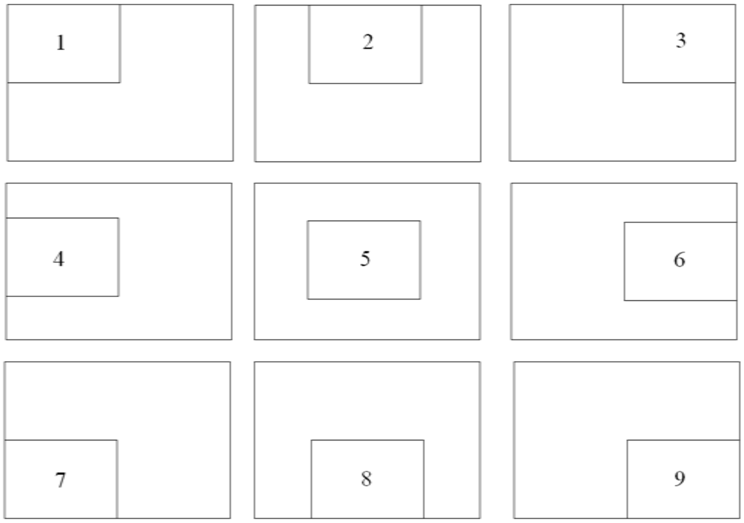


Figure 4.2. Image splitting

This was applied by a written code using Matlab[1] software

4. One part or more (of the nine parts) was selected from each image that contains epithelial cells.
5. Each selected part was divided into 12 images each of which is 80x80 as shown in Figure 4.3

[1] MATLAB is a programming environment for algorithm development, data analysis, visualization, and numerical computation for signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology.

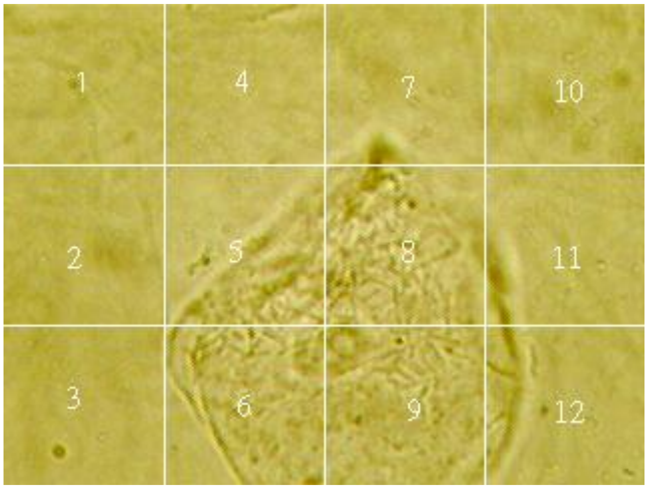


Figure 4.3. Image of size 320x240 with 12 divisions

This was applied by a written code using Matlab software

6. Manually each 80x80 part was labeled as containing epithelial cell or not (So, there are two classes: Yes and No, from perspective of containing epithelial cell)

   To help speed up manual labeling, a matlab code was written to add a mask of 80x80 divisions on those images of size 320x240 and add number for each division like Figure 4.4

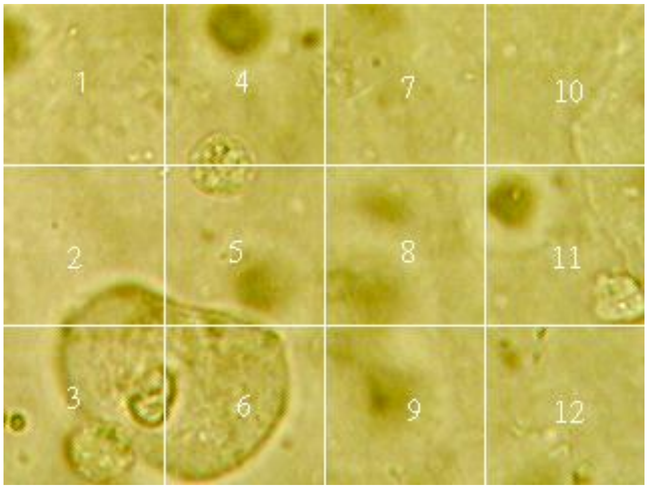


Figure 4.4. Each 80x80 division in image has a serial number

In addition to another matlab code was written to move each one of these 12 divisions to its appropriate class folder.

Class folders were four: much (labeled as *Yes*), moderate (labeled as *Yes*), little (labeled as *No*), and none (labeled as *No*) up to percentage of epithelial cell in 80x80 image part.

7. Another 80x80 images were extracted from samples images that contain other objects like RBCs, WBCs, and Crystals in order to make the class "No" contains images parts from all expected environments (not only samples of epithelial cells)

4.2.2. Preprocessing

A matlab code was written to read selected image parts and extract its textural features on excel file, features were as follows

4.2.2.1. Co-occurrence matrix parameters

Table 4.2 lists all extracted co-occurrence matrix parameters

Table 4.2 . Co-occurrence matrix parameters

GLCM parameters	GLCM parameters
Autocorrelation	Maximum probability
Contrast	Sum of squares
Correlation (matlab)	Sum average
Correlation (paper)	Sum variance
Cluster Prominence	Sum entropy
Cluster Shade	Difference variance
Dissimilarity	Difference entropy
Energy	Information measure of correlation1
Entropy	Information measure of correlation2
Homogeneity (matlab)	Inverse difference normalized (INN)
Homogeneity (paper)	Inverse difference moment normalized

These parameters were calculated for each of the following offsets:

0 1, 0 3, 0 5, 0 7 (Angle 0) [e.g. 0 1 is called offset]

1 1, 3 3, 5 5, 7 7 (Angle -45)

1 0, 3 0, 5 0, 7 0 (Angle -90)

-1 1, -3 3, -5 5, -7 7 (Angle 45)

All these offsets were calculated to get GLCM (Gray level co-occurrence matrix) features in all directions shown in Figure 4.5

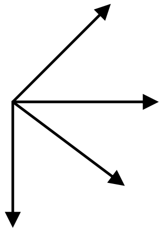


Figure 4.5. GLCM feature offset directions

4.2.2.2. Histogram features

Four parameters were extracted from histogram of images, which are:

- Mean
- Variance
- Skewness
- kurtosis

These parameters were calculated for each of channels: Red, Green, Blue, and Gray [e.g. image matrix of red values of pixels].

Total number of regular attributes was 368.

4.2.3. Data mining classification process

A classification process was built on rapidminer[1] software to apply classification operations and get the best model

Process steps are shown in Figure 4.6

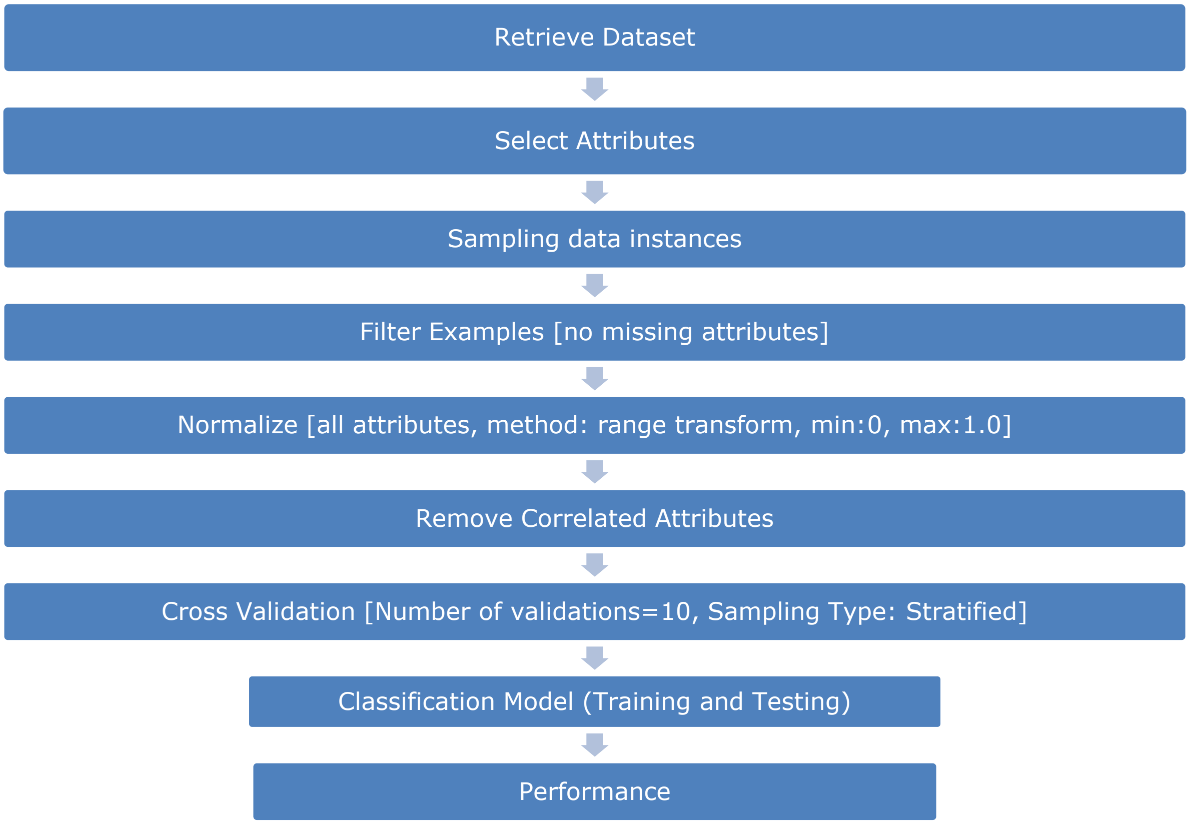

Retrieve Dataset
Select Attributes
Sampling data instances
Filter Examples [no missing attributes]
Normalize [all attributes, method: range transform, min:0, max:1.0]
Remove Correlated Attributes
Cross Validation [Number of validations=10, Sampling Type: Stratified]
Classification Model (Training and Testing)
Performance

Figure 4.6. Epithelial cells classification process in rapidminer

Figure 4.7 is a snapshot of rapidminer process. This figure shows the three levels of process, first level starts by retrieving the input dataset which consists of 2407 instances, 545 of them contain Epithelial Cells and labeled *Class=Yes*, and 1862 of them does not contain Epithelial Cells and labeled *Class=No*.

Next operator of first level is *Select Attributes* which selects the histogram features and GLCM features of offsets for one of the four directions. Next

[1] RapidMiner is open-source system for data mining. It is available as a stand-alone application for data analysis and as a data mining engine for the integration into own products. URL: http://rapid-i.com/content/view/181/190/

operator is a loop which is a sub-process that will iterate on changing parameters' values for parameter adjustment, and this is considered the second level.

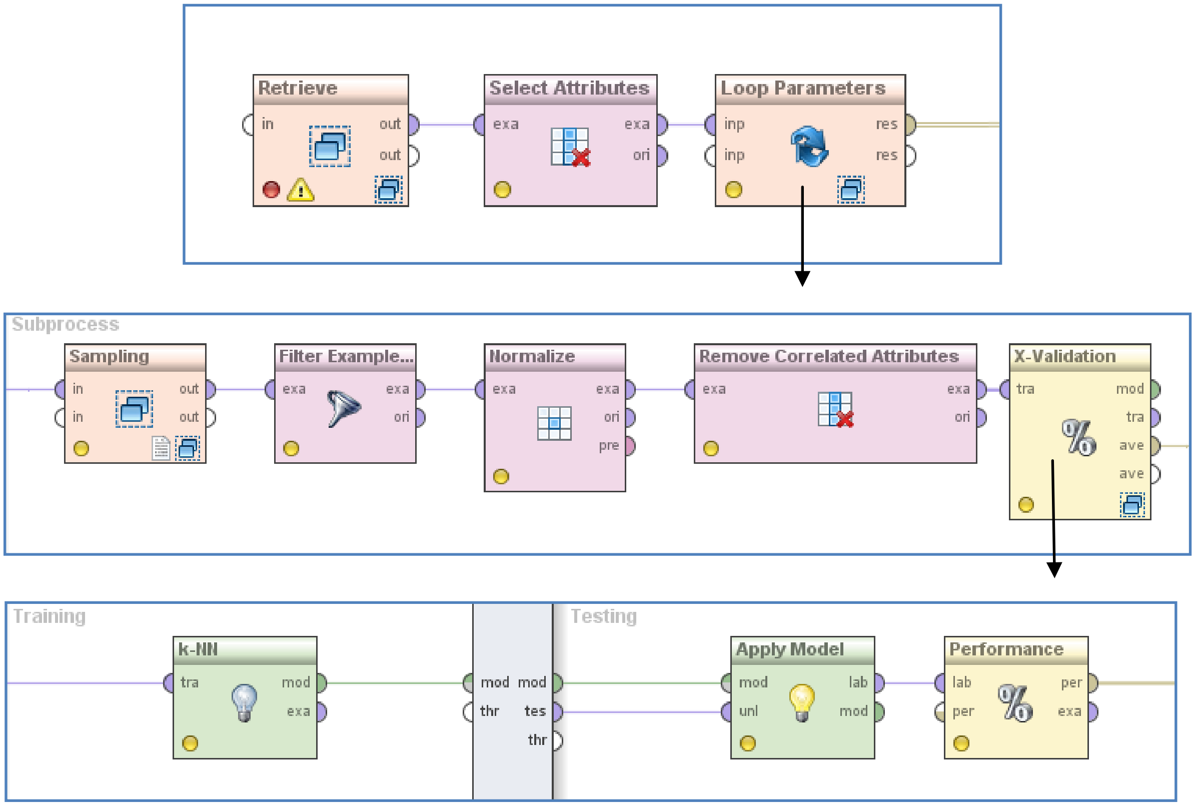


Figure 4.7. Epithelial Cells classification process in rapidminer

Second level starts by sampling data instances which select a part of input instances of class label "No" that does not contain epithelial cells. This sampling comes because number of instances will affect the built classification model and change its results.

Next operator is *Filter Examples*, to remove all examples that contain attributes with missing values.

Next operator is *Normalize,* to perform normalization between minimum and maximum values. It was used to make data consistent as there were negative values.

Next operator in second level is *Remove Correlated Attributes*, which helps in minimizing the input redundant attributes; it removes one of two features if their correlation exceeds correlation value. The used correlation function is the Pearson correlation.

Next operator is *X-Validation,* which encapsulates a cross-validation in order to estimate the performance of a learning operator. Number of validations (Number of subsets for the cross validation) is 10. The inner sub-processes are applied 10 times using S_i as the test set (input of the Testing sub-process) and $\{S\}$-S_i as training set (input of the Training sub-process). Inside this operator there is two parts, one for training and contains a learning model operator, and the other is for testing which contains an operator for applying the learning model and operator for calculating performance vector. Output of X-Validation operator is performance vector which contains accuracy, recall, and precision.

The experiment was enumerated on different changing parameters to adjust parameter values and get the best adjustment.

Changing parameters are:

1. GLCM features: there are 4 different sets of offsets up to the four directions shown in Figure 4.5.

2. Remove Correlated Attributes operator: changing the value of correlation changes input attributes to the x-validation, which causes changes in the learning model and performance.
3. Sampling: because the number of input data instances of label “No” is large, we can use a part of them to make data balanced and produce a balanced precision and recall values, as increasing the “No” instances makes learning model knows “No” examples more than “Yes” ones and this produces a low precision and recall rates of “yes” class. Sampling ratio was set to 8 values which are 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.
4. Classification model: five models were used to investigate best resulting performance vector, which are K-NN, Neural Network, Naïve Bays, Decision Tree, and Rule Induction, but a special changing parameter were tested for k-NN which is K [The used number of nearest neighbors].

4.2.3.1. Using K-NN model

Firstly, experiment was applied using K-NN model with Canberra distance using

1. GLCM features for offsets 0 1, 0 3, 0 5, 0 7 which are of angle 0 (one enumeration).
2. Remove Correlated Attributes operator: with changing correlations 0.8, 0.85, 0.9, 0.95, and 1 (5 enumerations).
3. Sample operator: with changing ratios 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 (8 enumerations).
4. Cross validation with 10 validations and stratified sampling.
5. K-NN model with changing k between 5 and 15 (11 enumeration).

This experiment consists of 440 enumerations. Part of results is shown in Table 4.3.

Table 4.3 . Epith Cells classification experiment performance results when changing sample ratio of instances of class label "No", changing the correlation parameter of "Remove correlated attributes" operator, and changing k of K-NN model

K	Sample Ratio	Correlation	Accuracy	F-Measure
..	..	..	..	..
12	0.9	0.9	90.73	81.74
12	0.9	0.95	90.51	81.34
12	0.9	1	90.66	81.70
12	1	0.8	91.12	81.15
12	1	0.85	90.57	79.85
12	1	0.9	90.69	80.00
12	1	0.95	90.9	80.37
12	1	1	90.14	78.83
13	0.3	0.8	86.78	87.38
13	0.3	0.85	87.92	88.46
13	0.3	0.9	88.22	88.51
13	0.3	0.95	87.28	87.53
13	0.3	1	85.49	86.19
13	0.4	0.8	90.16	88.79
13	0.4	0.85	88.63	87.07
13	0.4	0.9	88.55	86.83
13	0.4	0.95	87.63	85.84
13	0.4	1	86.35	84.54
13	0.5	0.8	88.61	85.25
13	0.5	0.85	88.8	85.41
13	0.5	0.9	88.53	85.05
13	0.5	0.95	89.34	86.05
13	0.5	1	87.4	83.62
..	..	..	..	..

In Table 4.3, first column lists the changing values of k (threshold of nearest neighbors) of K-NN model, second column lists number of data instances of class label "No" which belongs to those images that do not have epithelial cells, changing quantity of input instances of "No" class makes model change which affects success rates.

Third column in Table 4.3 lists tested values of "correlation" parameter in the "Remove correlated attributes" process. Best adjustment that resulted the best accuracy, and f-measure (of positive class, labeled “Yes”) rates was by using k=13, 0.4 of input "No" instances, correlation = 0.8 (shaded row in Table 4.3).

4.2.3.2. Using Decision Tree

Secondly, experiment was applied using Decision Tree model using

1. GLCM features for offsets 0 1, 0 3, 0 5, 0 7 which are of angle 0 (one enumeration).
2. Remove Correlated Attributes operator: with changing correlations 0.8, 0.85, 0.9, 0.95, and 1 (5 enumerations).
3. Sample operator: with changing ratios 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 (8 enumerations).
4. Cross validation with 10 validations and stratified sampling.

This experiment consists of 40 enumerations. Results are shown in Table 4.4.

In Table 4.4 Best adjustment that resulted the best accuracy and f-measure rates was achieved by using all input “No” instances and correlation=0.95 (shaded row in Table 4.4), but f-measure of “Yes” class is low, so it can be deduced that K-NN is better to use than Decision Tree.

4.2.3.3. Using Naïve Byes

Experiment was applied using Naïve Byes model using the same inputs and parameters used in Decision Tree

This experiment consists of 40 enumerations. Results are shown in Table 4.5.

Table 4.4 . Epith Cells classification experiment performance results when changing sample ratio of instances of class label "No", changing the correlation parameter of "Remove correlated attributes" operator, using Decision Tree model

Sample Ratio	Correlation	Accuracy	F-measure
0.3	0.8	80.62	82.55
0.3	0.85	81.88	83.49
0.3	0.9	80.28	82.08
0.3	0.95	83.14	84.02
0.3	1	83.18	84.21
0.4	0.8	85.55	83.73
0.4	0.85	83.64	81.79
0.4	0.9	79.26	79.11
0.4	0.95	83.77	82.02
0.4	1	84.47	82.91
0.5	0.8	84.95	80.91
0.5	0.85	81.51	77.96
0.5	0.9	83.78	80.14
0.5	0.95	85.38	81.72
0.5	1	83.36	79.46
0.6	0.8	86.82	80.37
0.6	0.85	85.61	79.51
0.6	0.9	85.34	78.57
0.6	0.95	87.62	81.67
0.6	1	85.24	78.07
0.7	0.8	85.86	76.98
0.7	0.85	84.17	74.42
0.7	0.9	85.13	74.88
0.7	0.95	87.06	79.47
0.7	1	88.08	81.09
0.8	0.8	88.99	79.85
0.8	0.85	87.17	77.06
0.8	0.9	86.54	76.20
0.8	0.95	87.39	78.12
0.8	1	88.77	79.60
0.9	0.8	88.96	77.65
0.9	0.85	88.08	76.77
0.9	0.9	87.34	74.89
0.9	0.95	88.43	77.24
0.9	1	88.54	77.55
1	0.8	88.48	75.78
1	0.85	86.99	72.58
1	0.9	87.85	74.65
1	0.95	89.46	78.21
1	1	89.29	77.54

In Table 4.5, no one of enumerations achieved accuracy more than 80%, Best adjustment that resulted the best accuracy, and f-measure rates was by using 0.3 of input "No" instances, correlation = 0.8 (shaded row in Table 4.5). So, it can be deduced that K-NN still the better to use than Decision Tree and Naïve Byes.

4.2.3.4. Using Rule Induction

Experiment was applied using Rule Induction model using the same inputs and parameters used in Decision Tree.

Table 4.5 . Epith Cells classification experiment performance results when changing sample ratio of instances of class label "No", changing the correlation parameter of "Remove correlated attributes" operator, using Naïve Byes model

Sample Ratio	Correlation	Accuracy	F-measure
0.3	0.8	78.33	80.62
0.3	0.85	70.93	76.24
0.3	0.9	75.86	79.11
0.3	0.95	77.35	80.00
0.3	1	76.55	76.10
0.4	0.8	76.62	76.09
0.4	0.85	67.94	71.45
0.4	0.9	74.05	74.87
0.4	0.95	73.83	74.77
0.4	1	76.68	72.99
0.5	0.8	68.45	68.77
0.5	0.85	64.87	66.58
0.5	0.9	77.64	72.86
0.5	0.95	76.28	73.82
0.5	1	76.55	70.01
0.6	0.8	63.24	62.59
0.6	0.85	75.03	69.75
0.6	0.9	73.56	69.06
0.6	0.95	74.44	70.06
0.6	1	77.95	69.13
0.7	0.8	77.41	69.22
0.7	0.85	63.64	60.30
0.7	0.9	73.58	66.90
0.7	0.95	73.4	66.85
0.7	1	78.06	67.75
0.8	0.8	62.45	57.34
0.8	0.85	74.65	64.91
0.8	0.9	72.98	64.23
0.8	0.95	74.28	65.68
0.8	1	78.1	64.79
0.9	0.8	62.95	55.70
0.9	0.85	63.1	55.20
0.9	0.9	71.09	60.90
0.9	0.95	73.17	62.35
0.9	1	78.23	63.25
1	0.8	62.52	53.33
1	0.85	61.42	52.56
1	0.9	72.25	59.62
1	0.95	73.52	60.84
1	1	78.45	61.62

This experiment consists of 40 enumerations. Results are shown in Table 4.6.

In Table 4.6, it’s noted that most f-measure values of “Yes” class are less than 80%. Best adjustment that resulted the best accuracy and f-measure rates was by using 0.7 of input "No" instances, correlation = 0.8 (shaded row in Table 4.6). There is no result here better than that which was found by K-NN, so it can be deduced that K-NN still the better to use than Decision Tree, Naïve Byes, and Rule Induction.

Table 4.6 . Epith Cells classification experiment performance results when changing sample ratio of instances of class label “No”, changing the correlation parameter of “Remove correlated attributes” operator, using Rule Induction model

Sample Ratio	**Correlation**	**Accuracy**	**F-measure**
0.3	0.8	86.13	86.24
0.3	0.85	85.87	85.77
0.3	0.9	86.06	85.95
0.3	0.95	87.44	87.29
0.3	1	85.69	85.87
0.4	0.8	85.76	83.00
0.4	0.85	85.42	82.56
0.4	0.9	87.69	85.37
0.4	0.95	84.78	82.39
0.4	1	85.42	82.43
0.5	0.8	86.41	81.00
0.5	0.85	86.85	81.73
0.5	0.9	86.14	80.54
0.5	0.95	86.52	81.12
0.5	1	86.11	80.77
0.6	0.8	86.85	79.30
0.6	0.85	86.69	78.79
0.6	0.9	86.17	78.55
0.6	0.95	87.33	80.15
0.6	1	86.79	78.86
0.7	0.8	88.18	79.02
0.7	0.85	87.32	78.17
0.7	0.9	88.56	79.84
0.7	0.95	88.05	78.70
0.7	1	86.06	75.53
0.8	0.8	88.23	77.24
0.8	0.85	87.85	76.64
0.8	0.9	88.57	77.55
0.8	0.95	87.83	76.69
0.8	1	87.87	76.62
0.9	0.8	88.72	76.42
0.9	0.85	88.75	75.70
0.9	0.9	88.06	74.73
0.9	0.95	88.86	76.73
0.9	1	88.8	76.92
1	0.8	89.46	75.78
1	0.85	88.78	73.86
1	0.9	89.04	74.51
1	0.95	89.33	75.70
1	1	89.8	75.95

4.2.3.5. Using Neural Network

Experiment was applied using Neural Network model using the same inputs and parameters used in Decision Tree.

This experiment consists of 40 enumerations. Results are shown in Table 4.7.

Table 4.7 . Epith Cells classification experiment performance results when changing sample ratio of instances of class label "No", changing the correlation parameter of "Remove correlated attributes" operator, using Neural Network model

Sample Ratio	Correlation	Accuracy	F-measure
0.3	0.8	87.51	87.62
0.3	0.85	88.35	88.31
0.3	0.9	86.39	86.27
0.3	0.95	86.69	86.90
0.3	1	85.92	86.20
0.4	0.8	88.49	86.46
0.4	0.85	87.41	85.26
0.4	0.9	88.04	86.11
0.4	0.95	87.63	85.34
0.4	1	87.8	85.42
0.5	0.8	88.43	84.53
0.5	0.85	87.69	83.46
0.5	0.9	88.81	85.27
0.5	0.95	89.14	85.53
0.5	1	88.31	84.23
0.6	0.8	89.66	84.10
0.6	0.85	89.81	84.57
0.6	0.9	89.66	84.06
0.6	0.95	88.87	82.94
0.6	1	88.37	82.64
0.7	0.8	90.08	82.73
0.7	0.85	90.25	83.73
0.7	0.9	90.45	83.44
0.7	0.95	89.95	83.12
0.7	1	89.94	82.96
0.8	0.8	90.52	82.25
0.8	0.85	91.02	83.16
0.8	0.9	90.75	83.12
0.8	0.95	90.72	82.95
0.8	1	90.44	82.01
0.9	0.8	91.71	82.66
0.9	0.85	90.55	80.61
0.9	0.9	91.66	82.78
0.9	0.95	90.88	81.50
0.9	1	89.87	79.63
1	0.8	90.86	79.47
1	0.85	91.63	81.74
1	0.9	91.63	81.78
1	0.95	91.41	81.19
1	1	90.95	79.89

In Table 4.7, it's noted that most f-measure values of "Yes" class are less than 85%. Best adjustment that resulted the best accuracy and f-measure rates was by using 0.3 of input "No" instances, correlation = 0.85 (shaded row in

Table 4.7). There is no result here better than that which was found by K-NN, so it can be deduced that K-NN still the best between all tested classification models.

Table 8 compares between best performance results got be different classifiers

Table 8 . Comparison between best performance for different classifiers

Classification Model	Sample Ratio	Correlation	Accuracy	F-measure
KNN	0.4	0.8	90.16	88.79
Decision Tree	1	0.95	89.46	78.21
Naïve Bayes	0.3	0.8	78.33	80.62
Rule Induction	0.7	0.8	88.18	79.02
Neural Network	0.3	0.85	88.35	88.31

4.2.3.6. Using other GLCM offsets

As mentioned above, GLCM features were extracted for the four directions

1. 0 1, 0 3, 0 5, 0 7 (Angle 0)
2. 1 1, 3 3, 5 5, 7 7 (Angle -45)
3. 1 0, 3 0, 5 0, 7 0 (Angle -90)
4. -1 1, -3 3, -5 5, -7 7 (Angle 45)

And features of offsets of angle 0 only were used in previous experiments. To know which is better to use between those sets of offsets, best adjustment (mentioned in K-NN experiment) was used with each set of features for offsets of each angle particularly.

Best adjustment is to use K-NN with k=13, correlation of *Remove Correlated Attributes* operator=0.8, and sampling ratio of class “No” instances is 0.4. This adjustment was deduced using GLCM features of offsets 0 1, 0 3, 0 5, 0 7 (Angle 0).

Testing other GLCM features of offsets of other angles occurred using the best adjustments, as follows:

1. Applying process using GLCM features of offsets 1 1, 3 3, 5 5, and 7 7 which acts as texture on Angle -45

   Resulted performance
   Accuracy: 87.26% +/- 2.34%
   F-measure: 85.59%

2. Applying process using GLCM features of offsets 1 0, 3 0, 5 0, and 7 0 which acts as texture on Angle -90

   Resulted performance vector
   Accuracy: 89.11% +/- 2.03%
   F-measure: 87.59%

3. Applying process using GLCM features of offsets -1 1, -3 3, -5 5, and -7 7 which acts as texture on Angle 45

   Resulted performance vector
   Accuracy: 88.10% +/- 3.52%
   F-measure: 86.41%

4. Applying process using GLCM features of all offsets

   Resulted performance vector
   Accuracy: 87.83% +/- 2.88%
   F-measure: 86.1%3

4.2.3.7. Best approach

It’s clear that there is no big difference between using GLCM features of offsets of angle 0, -45, -90, or 45, and even using all of them, but any how features of offsets of angle 0 produced the best results, so now to make better recognition of epithelial cells following adjustments should be followed

1. Use k-NN with k=13.
2. Sample input "No" instances=0.4, which means 745 instances.
3. Use Remove Correlated Attributes with correlation = 0.8.
4. Use histogram features and GLCM features of offsets 0 1, 0 3, 0 5, and 0 7.

This adjustment resulted in the following performance vector
Accuracy: 90.16% +/- 3.20%
F-measure: 88.79%

Table 4.9 . Performance vector of best adjustment for classification of Epith Cells

	true Yes	true No	class precision
pred. Yes	491	71	87.37%
pred. No	53	645	92.41%
class recall	90.26%	90.08%	

As for attribute selection, the "Remove correlated attributes" operator resulted in removing 91 attributes, the remaining 13 attributes which are the most un-correlated attributes that best differentiate the two classes (Yes and No) are listed in Table 4.10.

From analyzing the performance results table of K-NN, we find that the degradation of accuracy and f-measure starts after increasing the sample ratio of instances with class "No" over 0.4 and increasing the correlation factor over 0.8. Figure 4.8 shows how the gap between the two curves of accuracy and f-measure increases as sample ratio increases, best value is marked by up going arrow which has the top two peaks together.

Table 4.10 . Selected features in best approach

Histogram features	GLCM features
1. Mean in red channel	1. Contrast in offset 0 1
2. Mean in blue channel	2. Correlation (matlab) in offset 0 1
3. Variance in red channel	3. Cluster Prominence in offset 0 1
4. Skewness in red channel	4. Information measure of correlation2 in offset 0 1
5. Skewness in blue channel	5. Information measure of correlation1 in offset 0 3
6. Kurtosis in red channel	6. Correlation (paper) in offset 0 7
	7. Inverse difference moment normalized in offset 0 7

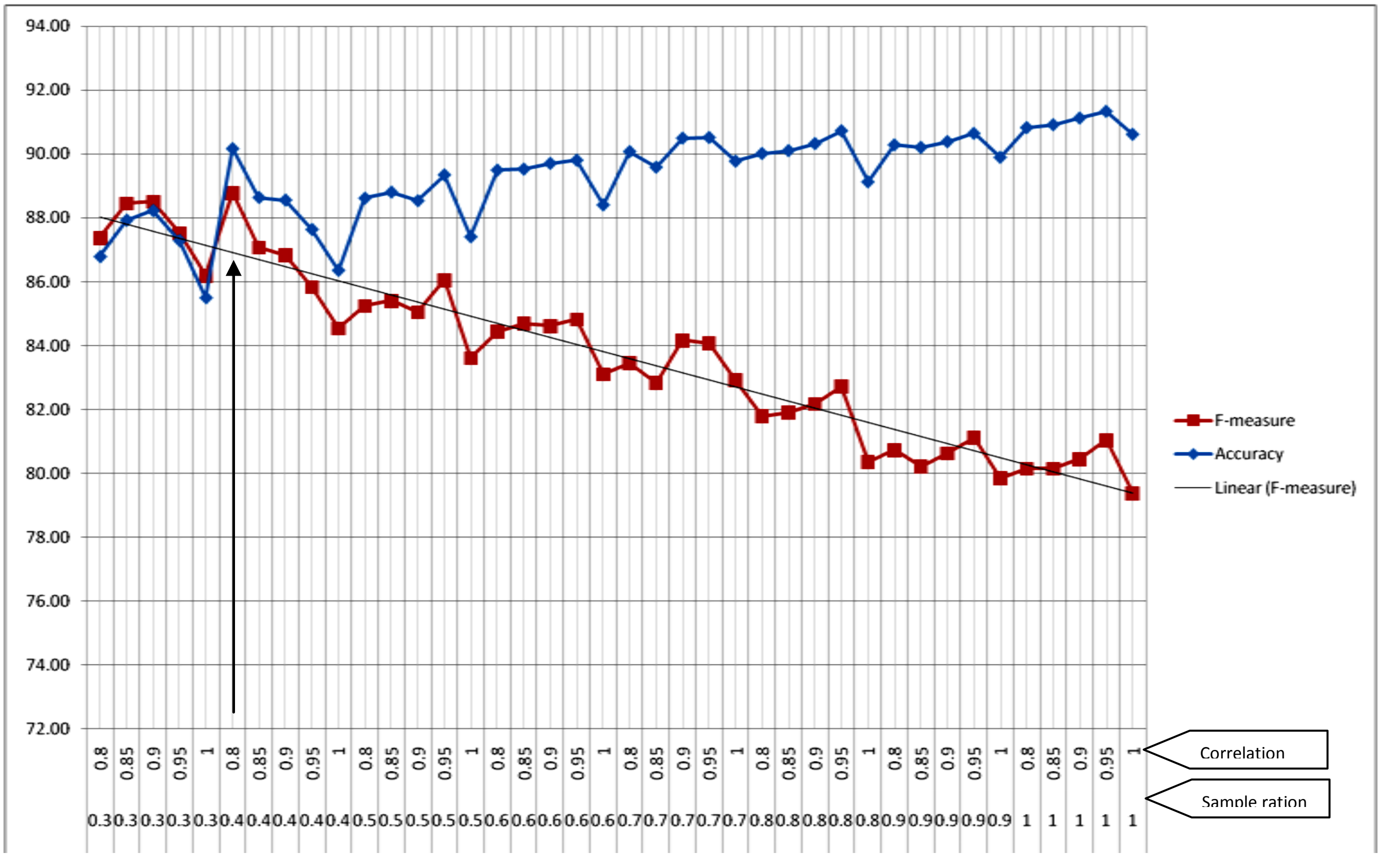


Figure 4.8. Accuracy and f-measure response to variation of sample ratio of instances of class "No" and correlation factor, when k=13.

We find that classifier is sensitive to redundant attributes, because the curve direction of accuracy and f-measure in Figure 4.8 is going down in all cases of correlation=1 (in which most attributes enters training process).

4.3. Solid particles experiments

The nature of solid particles is that they have solid boundaries that make it prominent from background, this formation can help us in extracting the binary shape of these particles and hence shape descriptors can be extracted. Next subsections will show experiment steps to create a model of recognizing solid particles in urine sample image:

4.3.1. Prepare Images

Choose a set of images that contain RBCs, WBCs, Crystals – Calcium Oxalate, and Crystals – Triple Phosphate.

4.3.2. Preprocessing

4.3.2.1. Image preprocessing

Preprocessing was applied on each image by following steps in Figure 4.9

Figure 4.9 started by converting image from 32-bit to 16-bit gray image as this was required by the third step (curvatures computing). Next is applying Gaussian blur filter with sigma=2.5 where this step makes image smoother and decreases number of curvatures produced by the third step. Next step applies principal curvatures computing on image to produce image of surface curves. Next is running the Default auto-threshold which is a modified version of isoData algorithm (By decreasing the highest tone of histogram to be 1.5 times of the tone with second order of arrangement, but this occur just when the highest tone was more than twice the second tone and second tone is not zero. This modification was done by ImageJ[1] developers) to segment image data. Next is an operation to convert image to binary in order to make it ready for blob extraction. Finally image was resized to 640x480 which is half of original size, to make image data less and to decrease size of processing. Figure 4.10 shows the effect of applying preprocessing steps on image containing calcium oxalate crystals.

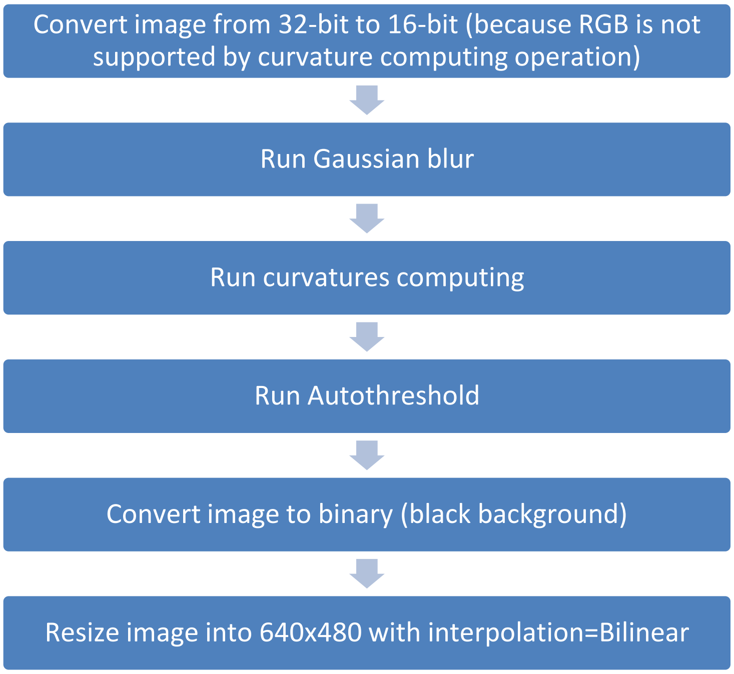

Convert image from 32-bit to 16-bit (because RGB is not supported by curvature computing operation)
Run Gaussian blur
Run curvatures computing
Run Autothreshold
Convert image to binary (black background)
Resize image into 640x480 with interpolation=Bilinear

Figure 4.9. Preprocessing steps

These steps were applied using Fiji [2] system (an image processing package) using a batch process that automatically gets an input folder and applies this process on all images in it and saves output images to another folder.

A set of preprocessing scenarios and different steps were tested by using Fiji software to get the best preprocessing results that can produce the best segmentation for urine particles.

[1] ImageJ is a public domain Java image processing program, URL: http://rsbweb.nih.gov/ij/docs/intro.html

[2] Fiji is an image processing package. It can be described as a distribution of ImageJ together with Java, Java 3D and a lot of plugins organized into a coherent menu structure, URL: http://fiji.sc/wiki/index.php/Fiji

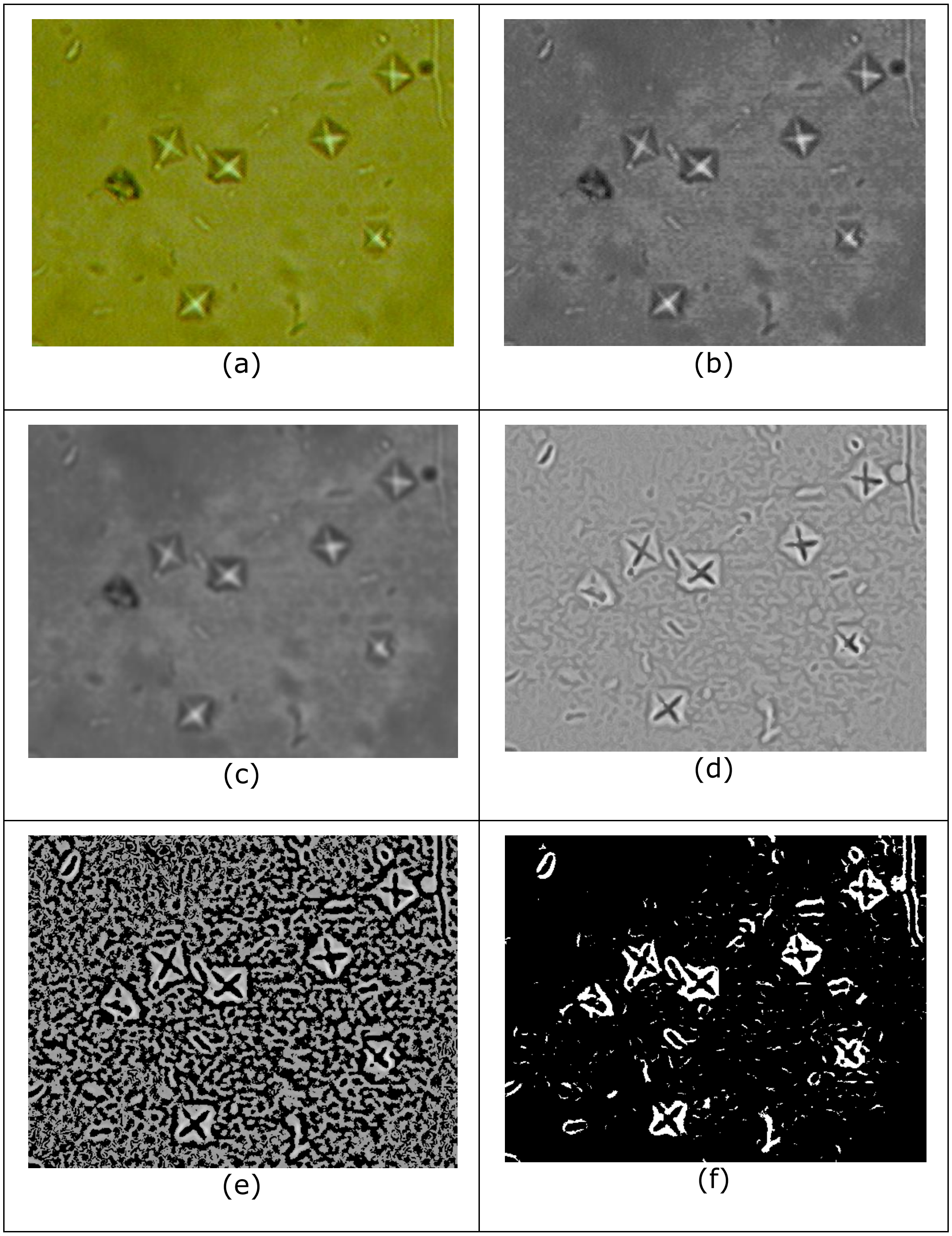

(a) (b) (c) (d) (e) (f)

Figure 4.10. Example of image under preprocessing. (a) Original image. (b) After converting image to 16 bit. (c) Result of Gaussian blur. (d) By applying curvature computing. (e) after auto-threshold. (f) Converted to binary.

The process of Figure 4.9 was applied with changing three values to get the most suitable preprocessing for input images. Process was applied on 16 images captured from microscope for the four types, 5 images for calcium oxalate crystals, 5 images for triple phosphate crystals, 2 images for RBCs, 4 images for WBCs. The 16 images entered the preprocessing steps and a blob extraction was applied to extract binary blobs, then extracted binary blobs were counted for each type. Table 4.11 shows resulted counts for each adjustment.

In Table 4.11, first column shows changeability of sigma factor of Gaussian blur step, second column lists tested sigma values of curvature computing operation, third column lists tested auto-threshold algorithms, and the next four columns are number of counted particles for each type after blob extraction.

Table 4.11 . Preprocessing experiments with different parameters

Gaussian Blur sigma	Compute Curvatures sigma	Auto-Threshold algorithm	# of oxalates	# of RBCs	# of WBCs	# of phosphates
1	1	Default	21	62	16	15
2.5	1	Default	37	90	26	18
3.5	1	Default	31	88	28	15
2.5	2	Default	29	69	28	15
2.5	3	Default	44	60	30	16
2.5	1	Huang dark	42	56	25	15
2.5	1	Intermodes	28	75	13	16
2.5	1	Li	29	72	16	18
2.5	1	MaxEntropy	8	11	1	1
2.5	1	Mean	27	80	20	16
2.5	1	IsoData	34	83	19	22

After blob extraction there is a set of particles are lost for breaking their parts or losing borders or touching with background. The best segmentation adjustment should get out the most numbers of particles. So, we can deduce that the best adjustment can be achieved by using:

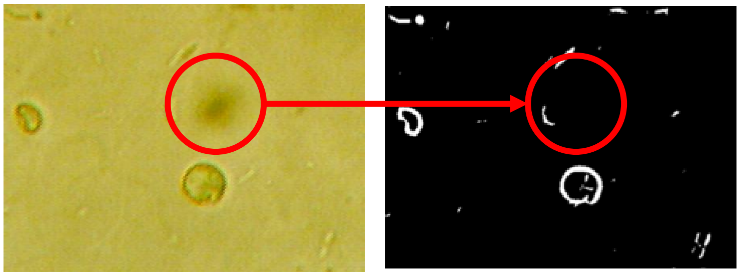
1. Gaussian Blur sigma=2.5
2. Compute Curvatures sigma=1
3. Using the Default auto-threshold algorithm

Default auto-threshold is a variation of the IsoData algorithm used by ImageJ 1.41 and earlier (Public Java image processing program).

4.3.2.1.1. Solved problems by image preprocessing

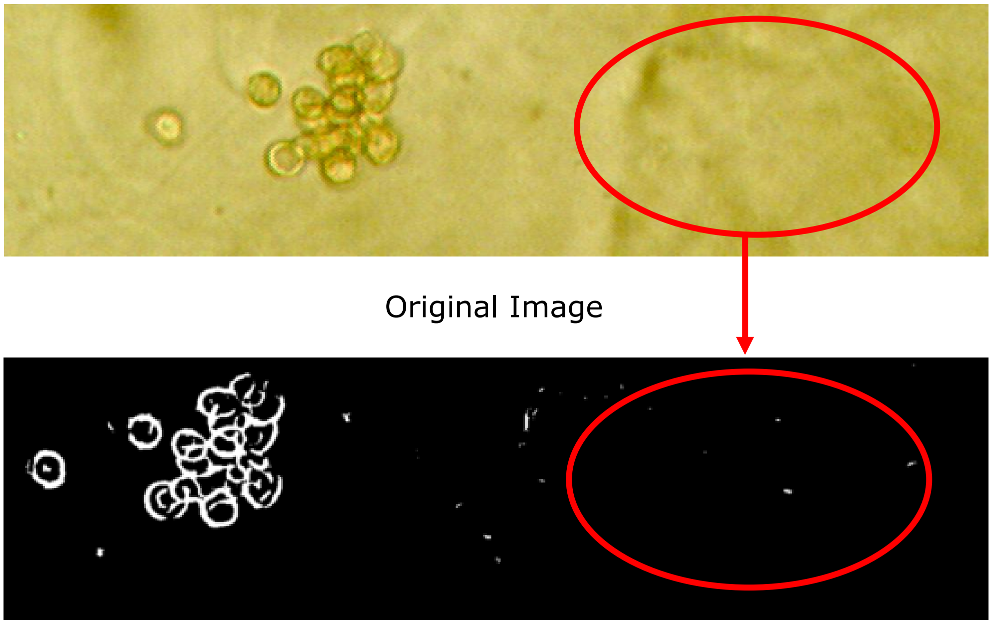
The proposed preprocessing scenario with the best adjustments could overcome different problems like:

1. Dark areas that do not have a solid form are removed like next figures



Original Image Preprocessed Image

Figure 4.11. Preprocessed image is clean of noisy dark areas.

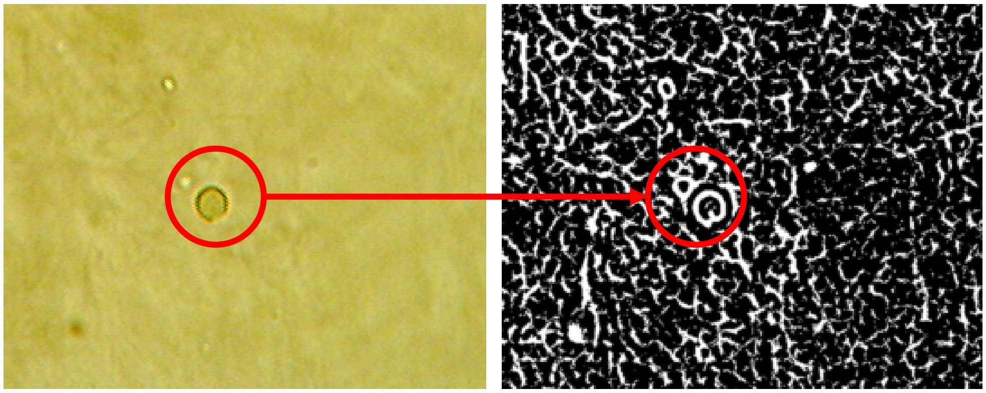


Original Image

Preprocessed Image

Figure 4.12. Preprocessed image is clean of noisy dark areas.

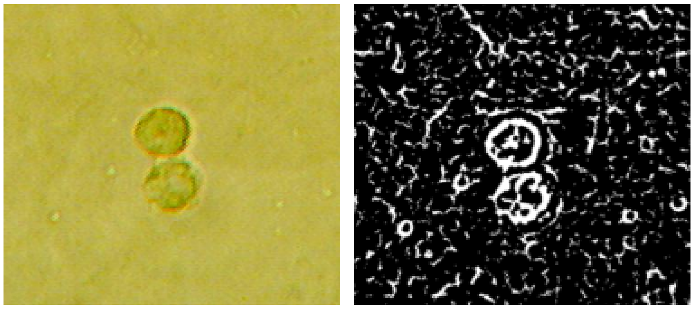
2. Noisy background does not affect particle boundaries and an empty halo can be noticed surrounding the particle in the preprocessed image of Figure 4.13



Original Image Preprocessed Image

Figure 4.13. Noisy background does not affect solid boundaries

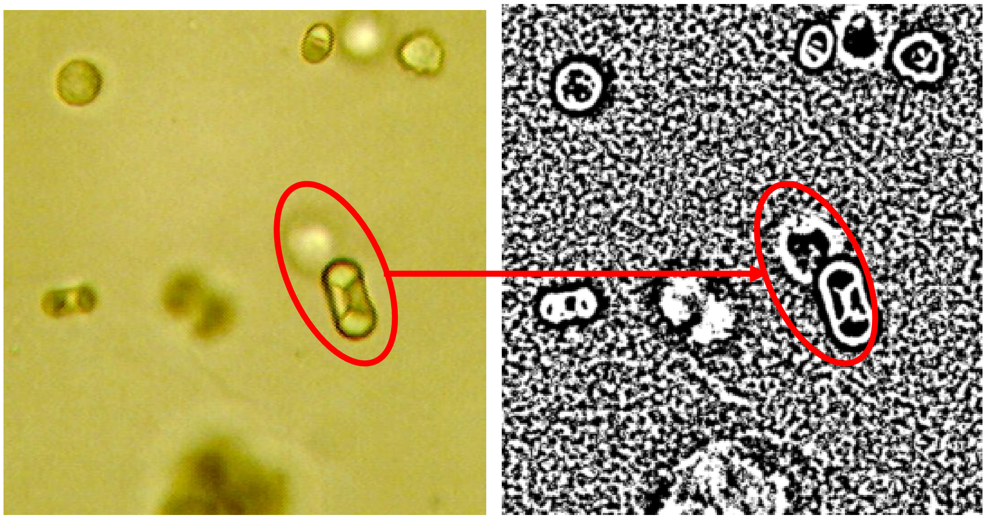
3. Very closed solid particles can be separated in the preprocessed image of Figure 4.14 because of variation of curvatures between them, so they were not touching each other in the preprocessed image



Original Image Preprocessed Image

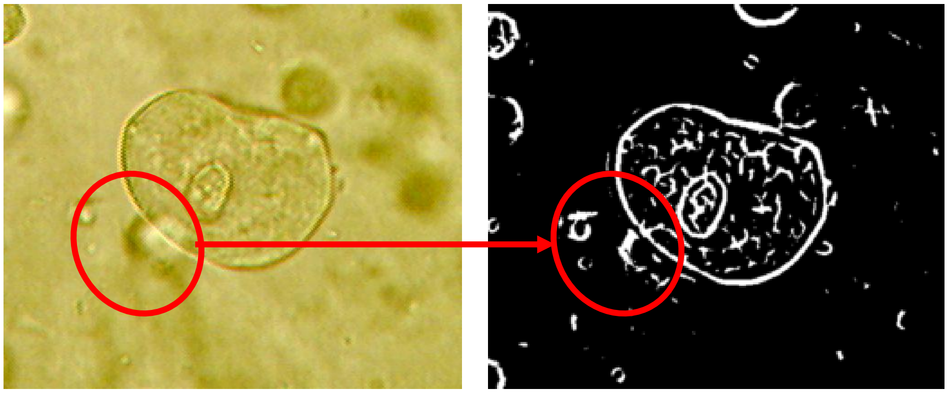
Figure 4.14. Separated closed particles

4. Solid particles were not affected by crossing blurry particles in the background, like next figures



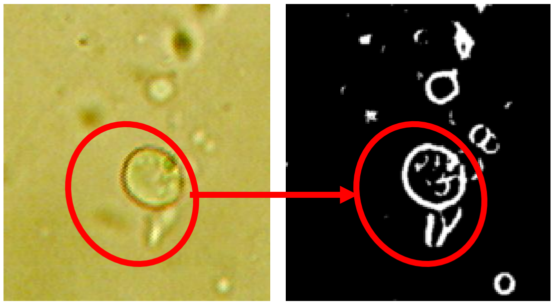
Original Image Preprocessed Image

Figure 4.15. Clear of cross with blurry particles



Original Image Preprocessed Image

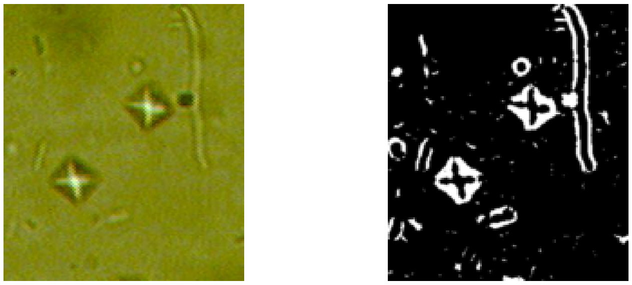
Figure 4.16. Clear of cross with blurry particles



Original Image Preprocessed Image

Figure 4.17. Clear of cross with blurry particles

5. Crystals in Figure 4.18 do not have a solid boundaries but its preprocessed form is expressive and reflects the X form of this particle.



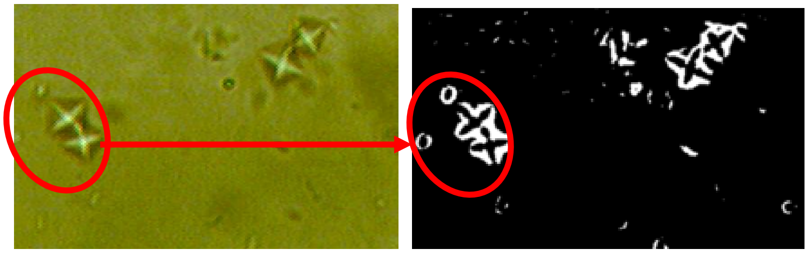
Original Image Preprocessed Image

Figure 4.18. Expressive crystals form

4.3.2.1.2. Not solved problems by image preprocessing

Some problems that appeared in different examples and preprocessing could not recover them like:

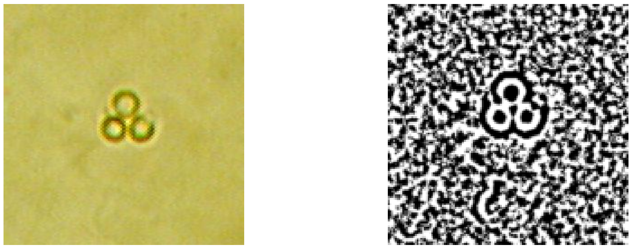
1. Overlapping particles resulted as a one particle in the preprocessed image



Original Image Preprocessed Image

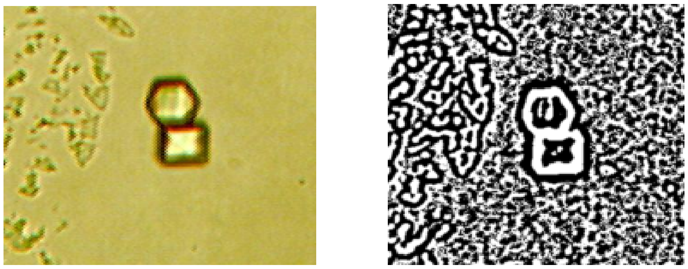
Figure 4.19. Overlapped particles

2. Touching particles resulted as a one particle in the preprocessed image like next figures



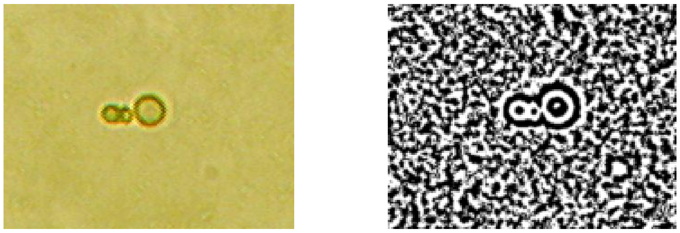
Original Image Preprocessed Image

Figure 4.20. Touching particles



Original Image Preprocessed Image

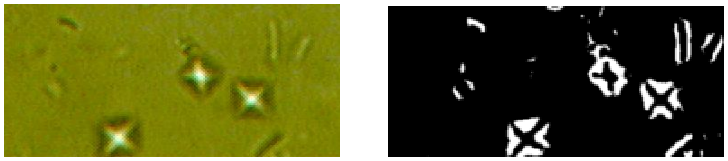
Figure 4.21. Touching particles



Original Image Preprocessed Image

Figure 4.22. Touching particles

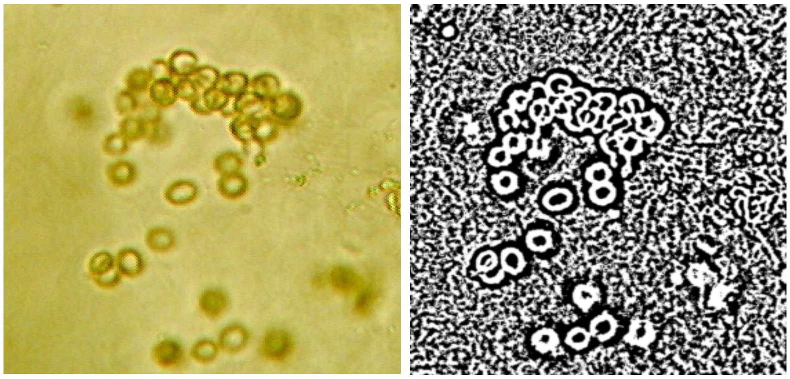
3. In some calcium oxalate crystals at some focus degree, the particle appeared as broken into four parts. This problem can be overcome by adjusting a good focus when capturing the image.



Original Image Preprocessed Image

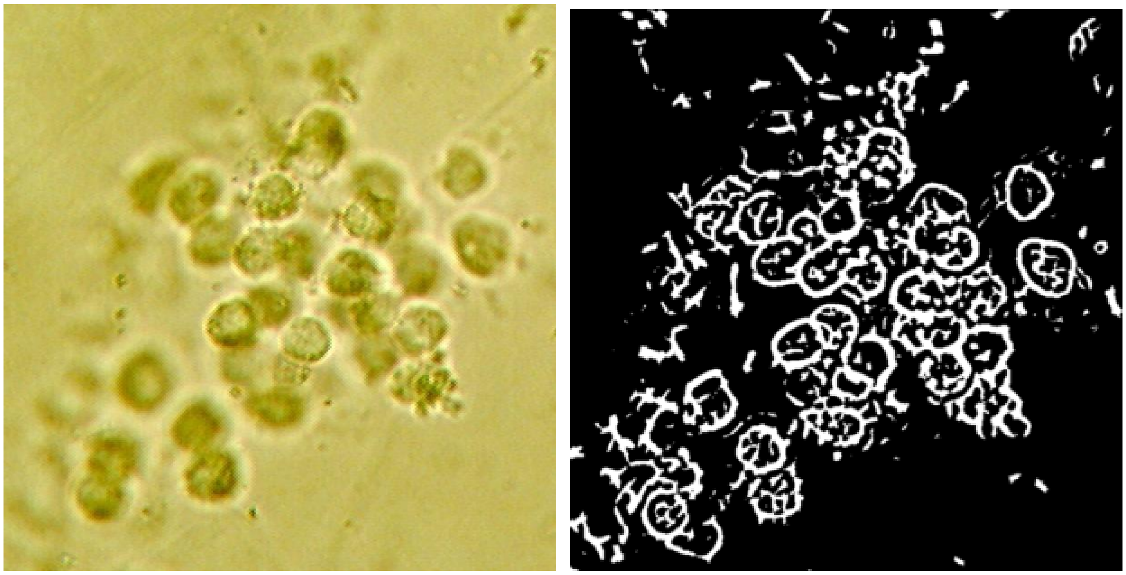
Figure 4.23. Broken particle

4. Overlapped collection of particles like next figures can’t be segmented in this preprocessing process and needs a special recognition, but it was not studied by this research for the little number of samples.



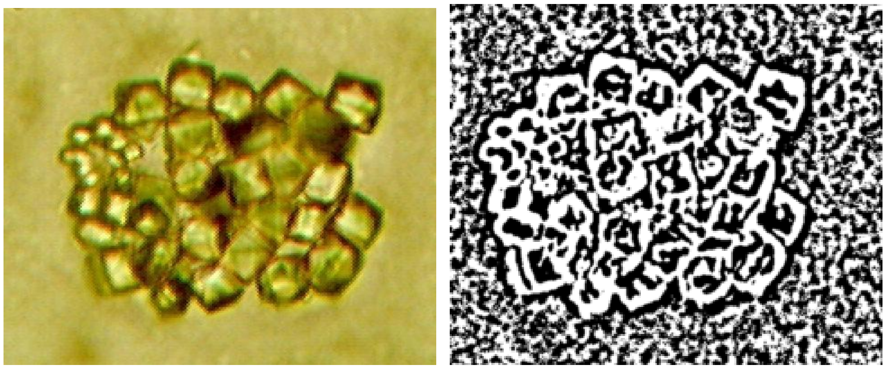
Original Image Preprocessed Image

Figure 4.24. Overlapped collection of particles



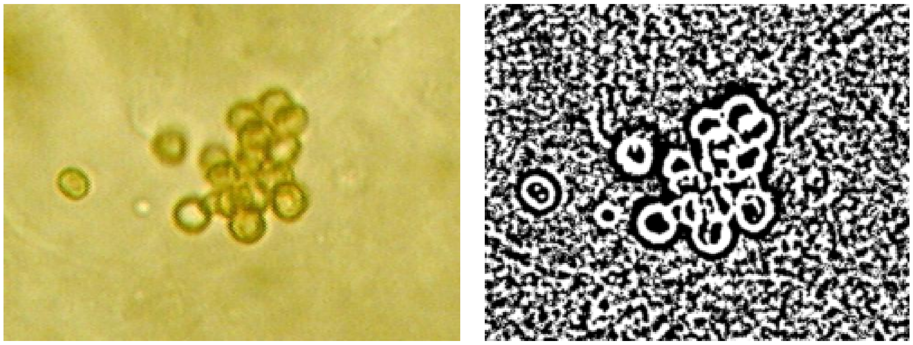
Original Image Preprocessed Image

Figure 4.25. Overlapped collection of particles



Original Image Preprocessed Image

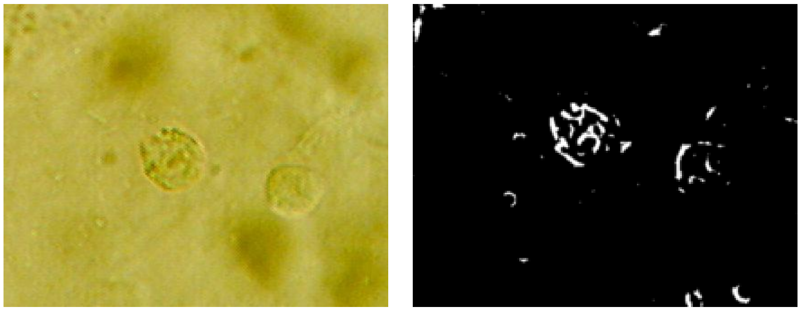
Figure 4.26. Overlapped collection of particles



Original Image Preprocessed Image

Figure 4.27. Overlapped collection of particles

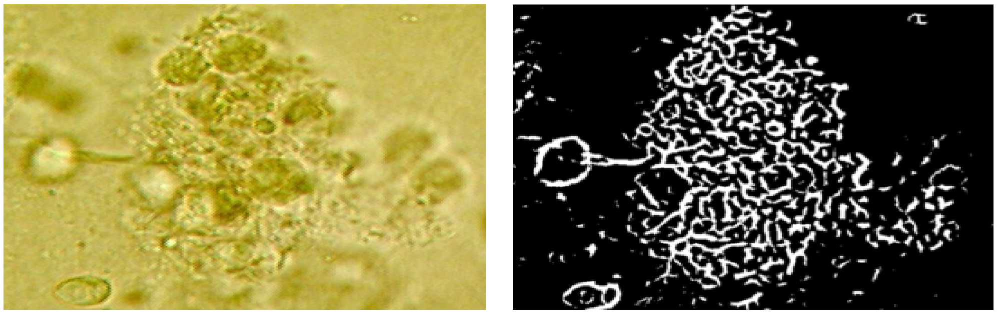
5. In some cases of WBCs, the particle was broken into small parts for the very similarity between it and background, like Figure 4.28



Original Image Preprocessed Image

Figure 4.28. Broken particle

6. In some cases of WBCs, particles merged with background for the very tough background that made WBC particle boundaries very light to be distinguished by preprocessing operation, like Figure 4.29



Original Image Preprocessed Image

Figure 4.29. Merged particles with background

4.3.2.2. Blob extraction and manual labeling

1. Blob Extraction process is to extract the 8-connected objects in the binary image and save each blob into particular image file, and extracts its position and saves it into database.

This process was applied using a code written in VC++ and IPL98 [1] library (Image Processing Library 98), and data was recorded in MSAcess database file.

The code started by segmenting the image pixels into groups up to eight-connectivity method with discarding those connected pixels groups that contain fewer than 100 pixels to decrease the number of unneeded noisy blobs as shown in Figure 4.30.

Each pixels group was saved into image file and it's coordinates in the original image were recorded in database by getting positioning values which are left, right, width, and height.

2. Manual labeling was applied to extract the candidate blobs images (RBCs, WBCs, and Crystals) for feature extraction and classification.

4.3.2.3. Features extraction

Feature extraction process which grabs all blobs images and extracts a set of features for shape and texture. Shape descriptors are extracted from the binary image and texture descriptors are extracted from the original image. These features are saved to excel file.

A matlab code was written to apply this functionality.

[1] The Image Processing Library 98 (IPL98) is a platform independent image manipulating C/C++ library. URL: http://ipl98.sourceforge.net/

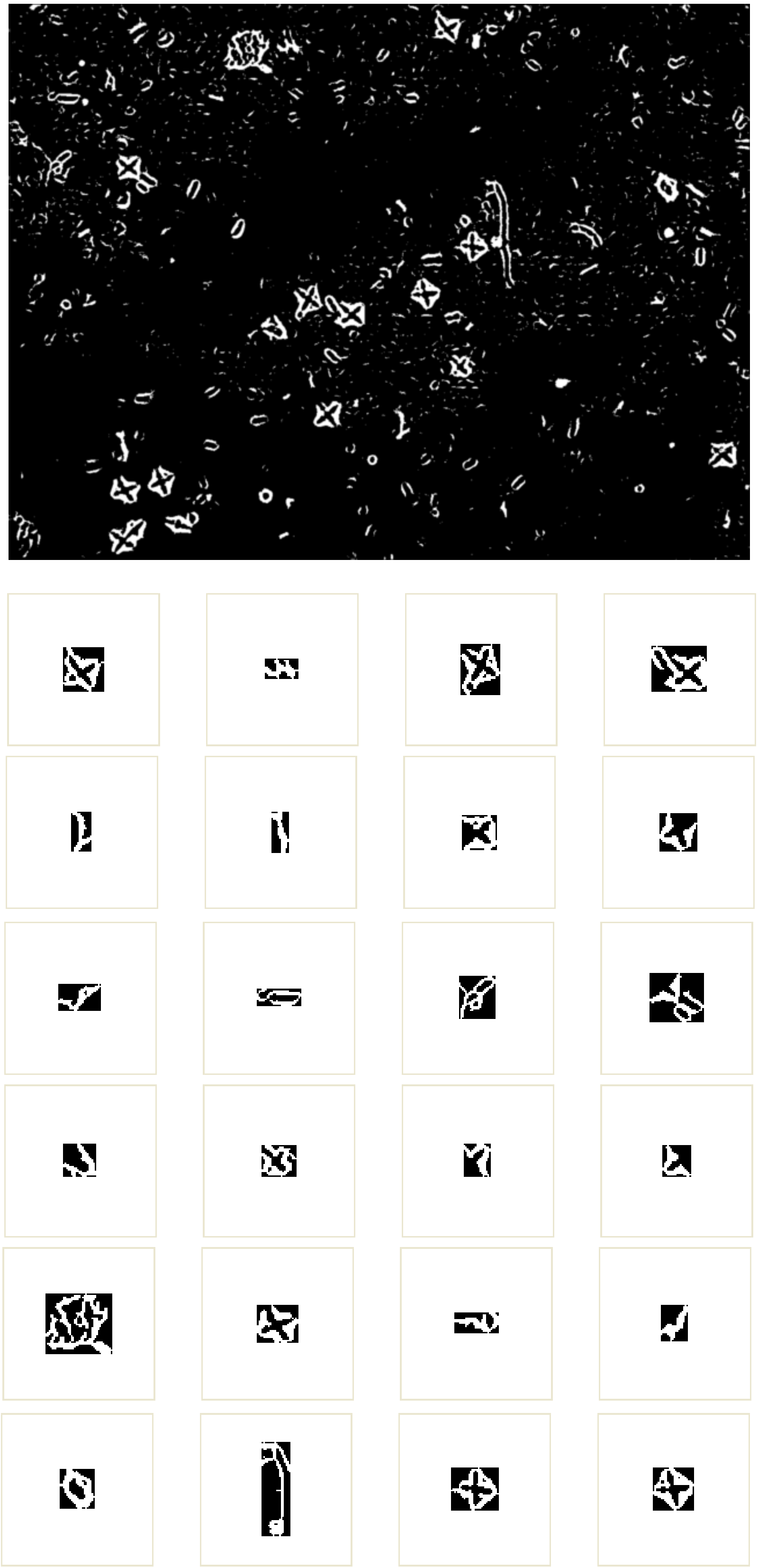


Figure 4.30. Blob Extraction and Saving into Image File

Shape descriptors

Table 4.12 shows extracted features from geometrical shape of particles

Table 4.12 . Shape descriptors

Shape features	
Area	Hu1
Orientation	Hu2
Perimeter	Hu3
ConvexArea	Hu4
FilledArea	Hu5
Solidity	Hu6
Eccentricity	Hu7
EquivDiameter	Roundness

Texture descriptors

- 16 Histogram features (Mean, variance, skewness, and kurtosis. For each of channels: Red, Green, Blue, and Gray)
- 22 Co-occurrence matrix parameters (Table 4.2)

  These parameters were calculated for offset 0 3

To extract texture features, each blob positioning values were extracted from the MSAcess database file and used to crop the particle image from its original image.

4.3.3. Data mining classification process

Classification process was built on rapidminer software to apply classification operations and get the best model

Process steps are shown in Figure 4.31

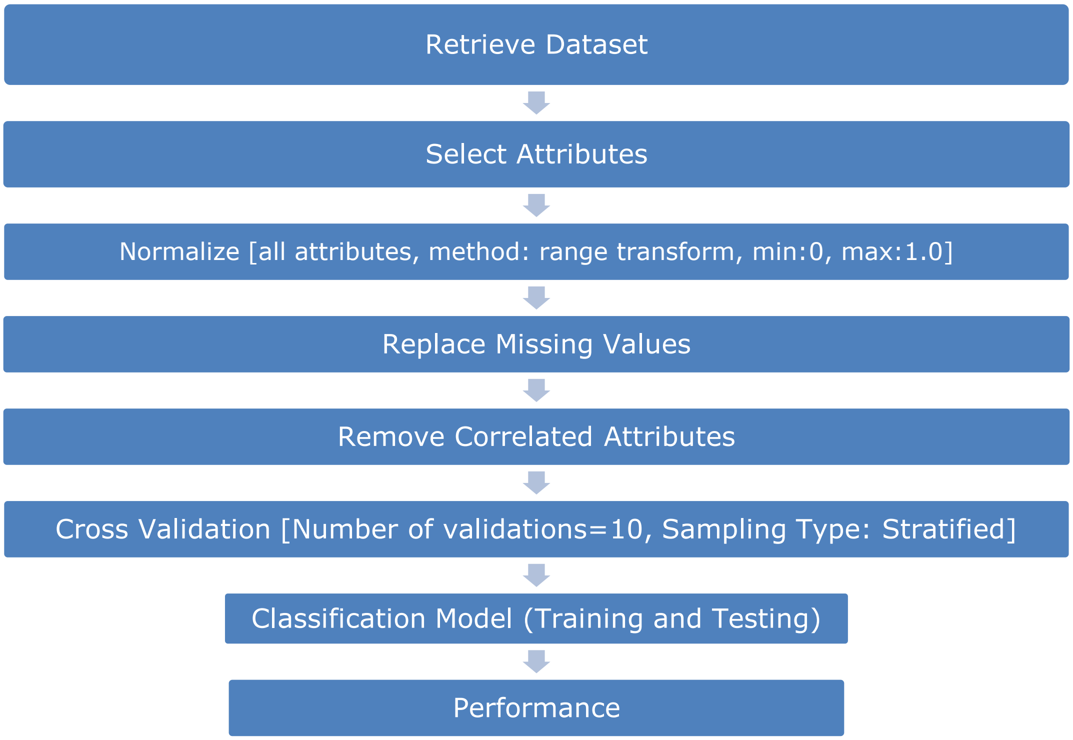

Retrieve Dataset
Select Attributes
Normalize [all attributes, method: range transform, min:0, max:1.0]
Replace Missing Values
Remove Correlated Attributes
Cross Validation [Number of validations=10, Sampling Type: Stratified]
Classification Model (Training and Testing)
Performance

Figure 4.31. Solid particles classification process in rapidminer

Figure 4.32 is a snapshot of rapidminer process. This figure shows the three levels of process, first level starts by retrieving the input dataset which consists of 807 instances, 268 of them for white blood cells and labeled "*Class=WBC"*, 188 for red blood cells and labeled "*Class=RBC",* 201 for calcium oxalate crystals and labeled "*Class=Crystals - Calcium Oxalate"*, and 150 for triple phosphate crystals and labeled "*Class=Crystals-Triple Phosphate"*.

Next operator of first level is *Select Attributes* which selects all input attributes except identities like *Image Name*.

Next operator is *Normalize,* to perform normalization between minimum and maximum values. It was used to make data consistent as there were negative values.

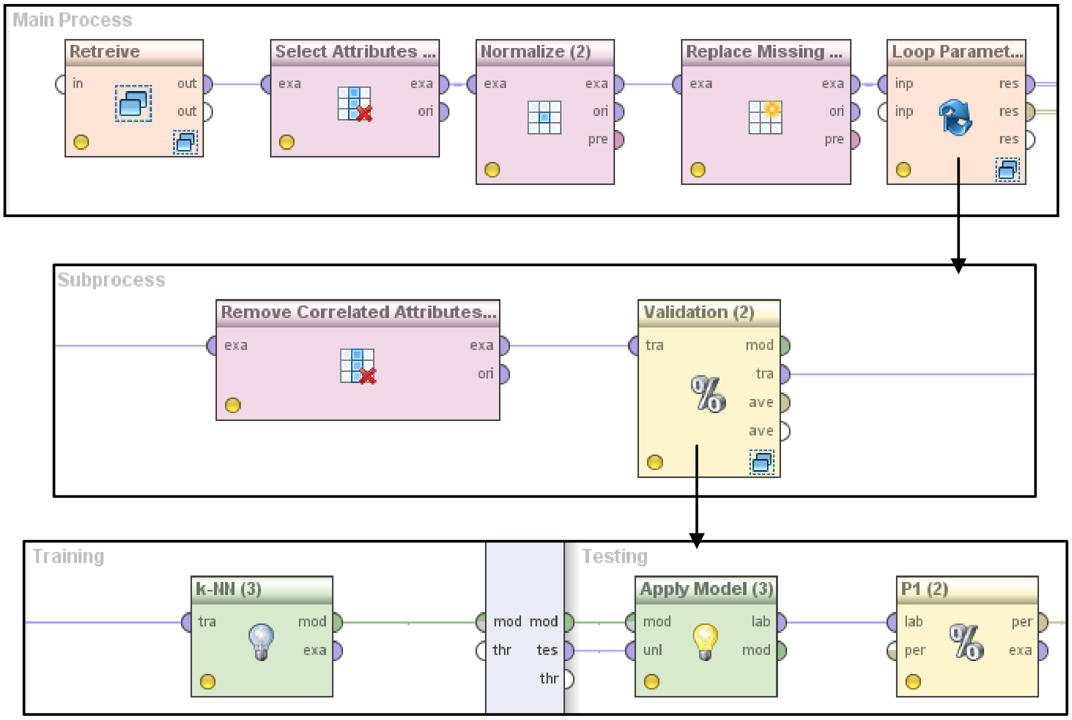


Figure 4.32. Solid particles classification process in rapidminer

Next operator is *Replace Missing Values* operator to replace missing values in examples. If a value is missing, it is replaced by average value of that attribute of other examples of the same class label.

Next operator is a loop which is a sub-process that will iterate on changing parameters' values for parameter adjustment and this is the second level.

Second level starts by *Remove Correlated Attributes* operator which helps in minimizing the input redundant attributes; it removes one of two features if their correlation exceeds correlation value.

Next operator is *X-Validation* which encapsulates a cross-validation in order to estimate the performance of a learning operator. Number of validations (Number of subsets for the cross validation) is 10. The inner sub-processes are applied 10 times using S_i as the test set (input of the Testing sub-process) and

$\{S\}-S_i$ as training set (input of the Training sub-process). Inside this operator there are two parts, one for training and contains a learning model operator, and the other is for testing which contains an operator for applying the learning model and operator for calculating performance vector. Output of X-Validation operator is performance vector which contains accuracy, recall, and precision.

The experiment was enumerated on different changing parameters to adjust parameter values and get the best adjustment.

Changing parameters are:

1. Remove correlated attributes operator: changing the value of correlation changes input attributes to the x-validation which changes the learning model and performance.
2. Classification model: five models were used to test resulting performance vector which are K-NN, Neural network, Naïve bays, Decision tree, and rule induction, but a special changing parameter were tested for k-NN which is K [The used number of nearest neighbors].

4.3.3.1. Using K-NN model

Firstly experiment was applied using K-NN model with Canberra distance using

1. Remove correlated attributes operator: with changing correlations 0.8, 0.85, 0.9, 0.95, and 1 (5 enumerations).
2. Cross validation with 10 validations and stratified sampling.
3. K-NN model with changing k between 5 and 15 (11 enumeration).

Table 4.13 . Solid particles classification experiment performance results when changing correlation parameter of "Remove correlated attributes" operator, and k of K-NN model

K	Correlation	Accuracy	Oxalate Recall	Oxalate Precision	RBC Recall	RBC Precision	WBC Recall	WBC Precision	Phosphate Recall	Phosphate Precision
..	..	..	..	..	..	..	..	..	..	..
10	0.8	90.83	80.60	92.57	93.09	89.29	98.13	87.96	88.67	97.08
10	0.85	91.95	83.58	94.92	91.49	91.01	98.51	88.89	92.00	95.83
10	0.9	91.44	83.08	93.82	94.15	89.39	98.51	89.19	86.67	96.30
10	0.95	92.07	84.58	92.90	93.62	91.67	98.51	89.49	88.67	97.08
10	1	92.18	84.58	92.90	92.02	91.53	98.88	90.14	90.67	96.45
11	0.8	90.84	81.59	92.66	93.09	88.38	98.13	88.26	87.33	97.76
11	0.85	91.83	83.58	94.92	92.02	91.05	98.88	88.63	90.00	95.74
11	0.9	91.83	83.58	94.38	94.68	90.36	98.51	89.19	87.33	96.32
11	0.95	92.45	86.57	94.05	93.62	91.67	98.51	89.80	88.00	97.06
11	1	93.06	87.06	93.09	93.09	93.58	98.88	90.44	90.67	97.84
12	0.8	90.21	79.10	93.53	92.55	87.88	98.51	86.56	87.33	97.76
12	0.85	90.83	82.09	94.29	89.89	89.42	98.51	87.42	90.00	95.74
12	0.9	91.94	85.07	95.00	92.55	90.16	98.88	89.53	88.00	95.65
12	0.95	92.07	85.57	92.97	92.55	92.06	98.51	89.49	88.67	96.38
12	1	92.69	85.07	94.48	93.09	92.59	98.88	89.83	91.33	96.48
13	0.8	89.96	80.10	92.53	91.49	88.66	98.13	86.80	86.67	95.59
13	0.85	91.95	84.08	94.94	92.02	92.02	98.51	88.29	90.67	95.77
13	0.9	91.33	82.09	94.29	92.55	89.23	98.88	88.63	88.67	96.38
13	0.95	91.82	84.58	93.92	93.09	90.21	98.51	89.19	88.00	97.06
13	1	92.44	84.58	93.92	92.55	92.06	98.88	90.14	91.33	95.80
..	..	..	..	..	..	..	..	..	..	..

This experiment consists of 55 enumerations. Results are shown in Table 4.13.

In Table 4.13, first column lists the changing values of k (threshold of nearest neighbors) of K-NN model; second column lists tested values of "correlation" parameter in the "Remove correlated attributes" process. Best adjustment that resulted the best accuracy, recall, and precision rates was by using k=11, correlation = 1 (shaded row in Table 4.13).

4.3.3.2. Using Decision Tree

Secondly, experiment was applied using Decision Tree model using

1. Remove correlated attributes operator: with changing correlations 0.8, 0.85, 0.9, 0.95, and 1 (5 enumerations).
2. Cross validation with 10 validations and stratified sampling.

This experiment consists of 5 enumerations. Results are shown in Table 4.14.

Table 4.14 . Solid particles classification experiment performance results when changing correlation parameter of "Remove correlated attributes" operator, using Decision Tree model

Correlation	Accuracy	Oxalate Recall	Oxalate Precision	RBC Recall	RBC Precision	WBC Recall	WBC Precision	Phosphate Recall	Phosphate Precision
0.8	90.09	87.06	82.55	88.83	89.78	92.16	94.27	92.00	93.88
0.85	89.47	83.08	85.64	93.09	84.13	91.04	93.85	90.67	94.44
0.9	89.96	83.08	84.77	89.89	86.22	93.28	93.98	93.33	94.59
0.95	90.33	81.09	86.24	93.09	88.38	93.66	91.94	93.33	95.24
1	90.09	85.57	81.52	90.96	90.00	91.79	93.89	92.00	95.83

In Table 4.14, best accuracy achieved by using correlation=0.95 (shaded row), but recall and precision values of are less than those produced by K-NN, so it can be deduced that K-NN is better to use than Decision Tree.

4.3.3.3. Using Naïve Byes

Experiment was applied using Naïve Byes model using the same inputs and parameters used in Decision Tree.

This experiment consists of 5 enumerations. Results are shown in Table 4.15.

In Table 4.15, best accuracy achieved by using correlation=1 (shaded row), no one of enumerations achieved accuracy more than 90%, so it can be deduced that K-NN still the better to use than Decision Tree and Naïve Byes.

Table 4.15 . Solid particles classification experiment performance results when changing correlation parameter of "Remove correlated attributes" operator, using Naïve Byes model

Correlation	Accuracy	Oxalate Recall	Oxalate Precision	RBC Recall	RBC Precision	WBC Recall	WBC Precision	Phosphate Recall	Phosphate Precision
0.8	69.39	95.02	46.36	56.38	97.25	52.99	97.26	80.67	86.43
0.85	78.44	91.54	57.68	84.04	91.33	64.55	97.19	78.67	86.13
0.9	81.66	94.03	63.00	82.45	92.81	73.13	96.55	79.33	86.86
0.95	83.4	91.54	67.15	85.11	89.39	78.36	96.33	79.33	87.50
1	85.38	85.07	76.68	80.85	87.86	91.04	89.05	81.33	89.05

4.3.3.4. Using Rule Induction

Experiment was applied using Rule Induction model using the same inputs and parameters used in Decision Tree

This experiment consists of 5 enumerations. Results are shown in Table 4.16.

Table 4.16 . Solid particles classification experiment performance results when changing correlation parameter of "Remove correlated attributes" operator, using Rule Induction model

Correlation	Accuracy	Oxalate Recall	Oxalate Precision	RBC Recall	RBC Precision	WBC Recall	WBC Precision	Phosphate Recall	Phosphate Precision
0.8	88.83	80.10	86.10	86.70	91.57	94.40	89.40	93.33	88.05
0.85	90.21	83.08	88.36	87.77	93.75	95.90	91.46	92.67	86.34
0.9	90.21	84.08	85.79	88.83	93.82	95.15	92.73	91.33	87.26
0.95	90.99	83.56	86.32	91.63	92.08	95.49	91.06	91.46	94.94
1	90.21	87.06	85.78	90.96	91.44	92.54	92.88	89.33	89.93

In Table 4.16, best accuracy achieved by using correlation=0.95 (shaded row), there is no result here better than that which was found by K-NN, so it can be deduced that K-NN still the better to use than Decision Tree, Naïve Byes, and Rule Induction.

4.3.3.5. Using Neural Network

Experiment was applied using Neural Network model using the same inputs and parameters used in Decision Tree.

This experiment consists of 40 enumerations. Results are shown in Table 4.17.

Table 4.17 . Solid particles classification experiment performance results when changing correlation parameter of "Remove correlated attributes" operator, using Neural Network model

Correlation	Accuracy	Oxalate Recall	Oxalate Precision	RBC Recall	RBC Precision	WBC Recall	WBC Precision	Phosphate Recall	Phosphate Precision
0.8	95.29	94.03	91.30	96.28	96.28	95.52	96.24	95.33	97.95
0.85	95.30	93.03	91.67	94.68	95.70	96.27	96.27	97.33	97.99
0.9	95.79	93.53	93.53	96.81	94.30	97.01	96.65	95.33	99.31
0.95	95.54	92.54	94.42	96.28	96.28	95.90	95.54	98.00	96.08
1	96.41	94.53	93.14	97.34	96.32	96.64	97.74	97.33	98.65

In Table 4.17, best accuracy achieved by using correlation=1 (shaded row), it's noted that most accuracy, recall, and precision values produced by neural network are better than those produced by k-NN, so it can be deduced that Neural Network classifier is the best between all tested classification models.

4.3.3.6. Best approach

To make better recognition of solid particles following adjustments should be followed

1. Use Neural Network.
2. Use Remove Correlated Attributes with correlation = 1

This adjustment resulted performance vector shown in Table 4.18

Accuracy: 96.41% +/- 1.51%

Table 4.18 . Performance vector of best adjustment for classification of solid particles

	true Crystals - Calcium Oxalate	true RBC	true WBC	true Crystals-Triple Phosphate	**class precision**
pred. Crystals - Calcium Oxalate	190	4	6	4	93.14
pred. RBC	5	183	2	0	96.32
pred. WBC	5	1	259	0	97.74
pred. Crystals-Triple Phosphate	1	0	1	146	98.65
class recall	94.53	97.34	96.64	97.33	

As for attribute selection, the “Remove correlated attributes” operator resulted in removing just one attribute which is Correlation (paper) extracted from GLCM features of offset 0 3.

4.4. Undefined particles

Some other particles that were not listed above; these particles can be called “undefined” or unknown. These are noisy particles that should be trained in the classifier to be filtered out of results.

To consider filtering undefined particles out of results, another process was built to testify how the classification models will treat with that situation.

Two cascade classifiers were used, first was used to remove as can as possible, the undefined particles from input dataset, second classifier was used to classify the remaining instances of input dataset to one of classes: WBC, RBC, Crystals-Calcium Oxalate, Crystals-Triple Phosphate, and Undefined.

First classifier was AODE (Averaged one-dependence estimators) classifier which classifies instances to “undefined” or not. This classifier was trained on 1435 samples of only undefined particles. Then classifier model was applied on input testing particles (input dataset) to remove the undefined particles and keep other particles on output to enter the second classifier. Second classifier was trained on all dataset instances of WBC, RBC, Crystals-Calcium Oxalate, Crystals-Triple Phosphate, and Undefined, and was applied on those particles which got out from the first classifier. Using the cascade classifiers helped in minimizing error rate.

To act as the real system, a set of testing images were selected to enter the process of classification, these images will be processed to extract particles from them, these particles will be of the five types: WBC, RBC, Crystals-Calcium Oxalate, Crystals-Triple Phosphate, and Undefined. The classification process should count how many particles there are in each image for each of the five types.

Following points show process steps to count number of particles inside input images to measure success and error rates.

1. 32 images were captured for testing

   (11 Calcium Oxalate, 10 Triple Phosphate, 5 RBC, 6 WBC)

2. Preprocessing, blob extraction, feature extraction steps were applied on these 32 images.

   Output of blob extraction was 1223 images of extracted blobs from the 32 images; feature extraction process extracted the features of the 1223 blobs’ images.

3. Testing particles' images of each type entered the first classifier (filter of “undefined”). Images of each type entered alone without other types (like to enter particles of RBCs only, to test success rates for RBCs).

   e.g. In the input dataset, only instances of extracted blobs from those 11 images of calcium oxalate were entered to the classification process

4. First classifier was AODE.

5. Training dataset of first classifier contained 1435 samples of undefined particles with label “undefined”.

6. Result of first classifier is a dataset with class label of “undefined” or “?”

7. A filter will enter only other particles which have the label “?” to the second classifier.
8. K-NN, Decision tree, Neural Network, Rule Induction, and Naïve Byes were exchanged to act as second classifier in order to test which of them produce the best success rates.
9. Training dataset of second classifier was the same training dataset used previously (in the process shown in Figure 4.32) in addition to the undefined particles instances.
10. Output of this process is the number of particles for each type (WBC, RBC, Crystals-Calcium Oxalate, Crystals-Triple Phosphate, and Undefined).
11. Measurements were applied to calculate success and error rates.

Figure 4.33 shows the process snapshot from rapidminer

Following steps show details about process steps and parameter values in Figure 4.33.

Step 1. Retrieve operator to get dataset of all training particles, this dataset contains 2242 instances, 268 of them for white blood cells and labeled “*Class=WBC”*, 188 for red blood cells and labeled “*Class=RBC”,* 201 for calcium oxalate crystals and labeled “*Class=Crystals - Calcium Oxalate”*, 150 for triple phosphate crystals and labeled *“Class=Crystals-Triple Phosphate”, and* 1435 for undefined and labeled “*Undefined*”.

Step 2. Select all attributes except “ImageName” as it’s not a feature.

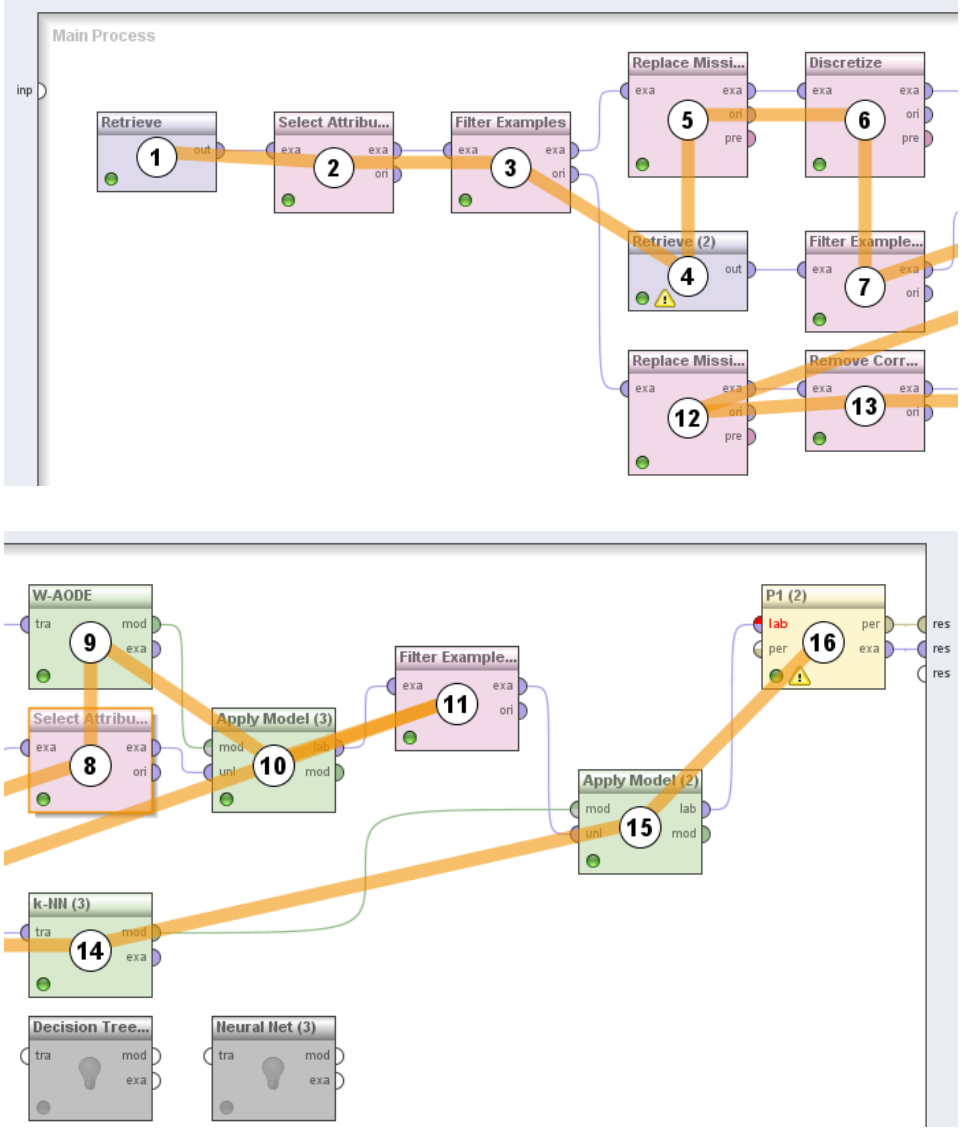


Figure 4.33. Classification process in rapidminer

Step 3. Filter "undefined" examples, this operator has two output, first is "exa" (as shown in Figure 4.33) which gets out only instances with label "undefined", and second output is "ori" which gets out all dataset instances.

Step 4. Retrieve operator to get dataset instances of extracted particles from the testing 32 images.

Step 5. Replace Missing Values operator to replace missing values in input instances (come from step 3 output "exa"). If a value of some attribute for some instance is missing, it is replaced by the average value of that attribute for all other instances with the same class label.

Step 6. Discretize by binning with number of bins: 20, this operator discretizes all numeric attributes in the dataset into nominal attributes.

Step 7. Filter range of input testing examples to enter classification operation, this filter will assign only instances of one particle type (e.g. like filtering examples in range between ID=1 and ID=333 to enter only instances of the blobs extracted from the 11 images of calcium oxalate, to count how many calcium oxalates there are in these 11 images and how many other types there are).

Step 8. Select all attributes except "ImageName" as it's not a feature.

Step 9. W-AODE byes classifier which acts as the first classifier.

Step 10. Apply model of W-AODE classifier (which trained on 1435 undefined particles) to detect which instances (from input testing dataset=output of step7) can be labeled as "Undefined".

Step 11. Filter out all examples except those whose label is "undefined". Now output of this operator is a dataset with minimized number of undefined particles, so that it's easier to detect and count other particles.

Step 12. Replace Missing Values operator to replace missing values in input instances (coming from step 3 output "ori").

Step 13. Remove Correlated Attributes, with correlation=0.85 when using K-NN, Decision Tree, and Rule Induction, and correlation=0.98 when using Neural Network. But this operator was not used by Naïve byes for an error occurred because of different attributes between training and testing data. (Values of correlation were adjusted up to several tests of varying correlation value).

Step 14. Classification model which is one of these classifiers: K-NN, Decision tree, Neural Network, Rule Induction, and Naïve Byes. This model acts as the second classifier which will be trained on all input dataset that was retrieved by step 1.

Step 15. Apply the built model in step 14 on input testing dataset retrieved by step 11.

Step 16. Calculate counts of particles of each label using performance operator.

Following tables show the counted particles inside input images

Table 4.19 . Counted particles by KNN classifier

KNN, k=5 , remove correlated=0.85, Descretize bins=20			**pred. Undefined**	**pred. Crystals - Calcium Oxalate**	**pred. Crystals-Triple Phosphate**	**pred. RBC**	**pred. WBC**	**Success Rate**	**Error Rate**
	Input Images	**Calcium Oxalate (11 images)**	196	43	10	12	13	89.58	12.77
		Triple Phosphate (10 images)	124	1	45	1	1	75.00	1.74
		RBC (5 images)	181	20	25	79	41	52.67	24.86
		WBC (6 images)	164	4	2	7	34	80.95	6.16

In Table 4.19, for calcium oxalate, input testing images were 11, these images contains calcium oxalate particles and some other undefined particles inside them, these images were captured from the same specimen slide for one patient, each image passed through preprocessing, blob extraction, and feature extraction steps, and finally all features' instances of all extracted particles from the 11 images were entered to rapidminer software to apply classification and count how many particles are there for each type. Results for calcium oxalate are shown in the shaded row, it's noticed that 196 object were not classified to any of the four class labels, 43 objects were labeled as calcium oxalate, but the real number is 48 (were counted manually), so success rate can be calculated by

$$\frac{\text{Number of labeled objects as "calcium oxalate "}}{\text{Number of real existing "calcium oxalate "objects}} = \frac{43}{48}=89.58\%$$

As for RBC, WBC, and triple phosphate, these counts are considered error rate because they are not found in the 11 images of calcium oxalate samples. Error rate can be calculated by

$$\frac{\text{Number of error objects}}{\text{Number of all objects}} = \frac{12+13+10}{196+43+12+13+10}=12.77\%$$

Other input images of RBCs, WBCs, and triple phosphate were calculated by the same way. Next tables show results of applying the same experiment and calculations using other classification models.

Table 4.24 . Counted particles by Decision Tree classifier

Decision tree, gain ratio , remove correlated=0.85			pred. Undefined	pred. Crystals - Calcium Oxalate	pred. Crystals-Triple Phosphate	pred. RBC	pred. WBC	Success Rate	Error Rate
	Input Images	Calcium Oxalate (11 images)	209	49	12	3	1	97.92	5.84
		Triple Phosphate (10 images)	133	5	34	0	0	94.44	2.91
		RBC (5 images)	199	23	26	98	0	65.33	14.16
		WBC (6 images)	187	5	6	8	5	11.90	9.00

Table 4.20 . Counted particles by Neural Network classifier

Neural , remove correlated=0.98			pred. Undefined	pred. Crystals - Calcium Oxalate	pred. Crystals- Triple Phosphate	pred. RBC	pred. WBC	Success Rate	Error Rate
	Input Images	**Calcium Oxalate (11 images)**	201	54	18	1	0	97.92	6.93
		Triple Phosphate (10 images)	124	2	45	1	0	94.44	1.74
		RBC (5 images)	164	0	33	145	4	96.67	10.69
		WBC (6 images)	194	1	1	8	7	16.67	4.74

Table 4.21 . Counted particles by Rule Induction classifier

Rule Induction , remove correlated=0.85			pred. Undefined	pred. Crystals - Calcium Oxalate	pred. Crystals- Triple Phosphate	pred. RBC	pred. WBC	Success Rate	Error Rate
	Input Images	**Calcium Oxalate (11 images)**	218	35	6	15	0	97.92	7.66
		Triple Phosphate (10 images)	147	2	23	0	0	94.44	1.16
		RBC (5 images)	225	23	15	83	0	55.33	10.98
		WBC (6 images)	190	7	1	11	2	4.76	9.00

Table 4.22 . Counted particles by Naïve Byes classifier

Naïve Byes			pred. Undefined	pred. Crystals - Calcium Oxalate	pred. Crystals- Triple Phosphate	pred. RBC	pred. WBC	Success Rate	Error Rate
	Input Images	**Calcium Oxalate (11 images)**	79	61	94	40	0	97.92	48.91
		Triple Phosphate (10 images)	33	10	117	12	0	94.44	12.79
		RBC (5 images)	33	20	133	146	14	97.33	48.27
		WBC (6 images)	60	4	39	20	88	100	51.66

If we looked to success rates we can deduce that naïve byes achieved the best results, but this is not the only criterion to assess classification model, there is an error rate that means assigning undefined objects to one of the four types (RBC, WBC, Oxalate, and Triple phosphate) by error. We can deduce that

Neural Network model produced the most high success rates and low error rates for all types except for WBC. It’s noticed that WBC was classified successfully with low error rate by k-NN.

4.5. Conclusion

In this chapter, details of experimental procedures were shown. Microscopic urine particles were recognized in two different tracks, first will detect the existence of vague particles, and second will detect the existence of solid particles.

To detect vague particles, a set of preprocessing steps were applied to prepare for texture feature extraction and classification process to assess the methodology and build model. Different classification models were trained and tested with changing parameters’ values to get the best accuracy rate. Best accuracy rate of classifying vague particles was 90.16% with using K-NN and cross validation.

To detect solid particles, image processing operations were added here to get out a binary image in order to extract blobs. Shape and texture features were extracted from blobs. Best accuracy rate of classifying solid particles was 96.41% with using Neural Network and cross validation.

Undefined particles were considered also, and a real experiment with input testing images was applied by entering all extracted blobs to the data mining process and count how many particles are there for each image. Cascade classification was used to filter out as most as possible from undefined particles in order to increase success rates and decrease error rates. Best model that got out best success with error rates was Neural Network.

Chapter 5. Conclusion and Future Works

5.1. Conclusion

Urine analysis is an important test in medical labs and deserves interest as it reveals the presence of many problems and diseases in human body. One of the main parts of that test is handled by human eye observations through microscope. As human eye observation is subjective, time consuming, and causes mistakes, researchers studied methods of automating microscopic analysis with the aid of computer and software systems. Numerous researches were published on that field, but they lack to improvements with respect to methodology and performance.

This research introduced a comprehensive approach for automating procedures for detecting and mining microscopic urine particles. 2193 microscopic images were captured and filtered, 340 of them contain RBC particles, 296 contain WBC, 451 epithelial cells, 220 calcium oxalate, 152 triple phosphate, and others. Experiment was applied in two tracks; first considered vague particles which have very light boundaries like epithelial cells, and second considered the solid particles that have strong boundaries.

In first experiment, histogram and gray level co-occurrence matrix (with different offsets) textural features were extracted from cropped 80x80 image regions. Correlation-based feature selection determined 13 features to be the most discriminative for classification. By trying the five classifiers K-NN, Neural Network, Naïve Bays, Decision Tree, and Rule Induction with 10-fold cross validation, evaluation resulted a performance of accuracy of 90.16% and f-measure of 88.79% by using KNN with k=13 and other adjustments.

In second experiment, Gaussian blur and curvature computing was applied for enhancing images with several experiments for adjusting factors. A modified version of IsoData threshold algorithm was proved to be the best choice, after running a set of histogram-driven threshold algorithms on images. 8-connected blob extraction with manual labeling emerged 268 WBC blobs, 188 RBC blobs, 201 calcium oxalate blobs, and 150 triple phosphate blobs. 16 shape descriptors and 38 textural features were extracted. By trying the five classifiers K-NN, Neural Network, Naïve Bays, Decision Tree, and Rule Induction with 10-fold cross validation, evaluation resulted in a performance of accuracy of 96.41% with minimum f-measure of 93.83% by using neural network. In addition, undefined particles were entered to the second experiment with using cascade classification, by running AODE classifier before the main classifier for filtering undefined particles, but this caused degradation in performance of WBCs recognition.

The chosen approach is considered optimum as it achieved higher performance measurements with strong methodology and evaluation techniques.

5.2. Future Works

In future works, following issues will be considered

1. More particles will be collected from those which were not studied by this research to be studied; this needs more efforts and multiple cameras to run on different medical labs. In addition, a team of experts to label images manually, where manual labeling was very time consuming task in this research.

2. Add the functionality of detecting moving particles such as bacteria which appears like small slow moving sticks inside urine specimen.
3. Overlapped and touching particles produce undefined forms of particles after segmentation and applying threshold. This problem increased error rate, and finding a technique to split touching particles, or extract some particle from overlapping will increase performance of segmentation.

References

[1] Michael L. Astion, Sara Kim, and Amanda Nelson, "*A Two-Year Study of Microscopic Urinalysis Competency Using the Urinalysis-Review Computer Program*". Clinical Chemistry, pp. 757–770, 1999.

[2] MCCULLOUGH B., YING X., MONTICELLO T., and BONNEFOI M., "*Digital Microscopy Imaging and New Approaches in Toxicologic Pathology*". Toxicologic Pathology, 32(Suppl. 2):49–58, ISSN: 0192-6233, 2004.

[3] Jiang X. and Nie S., "*Urine Sediment Image Segmentation based on Level Set and Mumford-Shah Model*". The 1st International Conference on Bioinformatics and Biomedical Engineering, IEEE, 2007.

[4] Masatoshi I., Naoko O., Kogiku S, and Manabu Y, "*How to track spermatozoa using high-speed visual feedback, Engineering in Medicine and Biology Society*". EMBS 2008. 30th Annual International Conference of the IEEE, 2008.

[5] Shapiro L. and Stockman G., "*Computer Vision*". Prentice Hall, ISBN 0-13-030796-3, 2001.

[6] Morris T., "*Computer Vision and Image Processing*". Palgrave Macmillan, ISBN 0-333-99451-5, 2004.

[7] Jähne B. and Haußecker H., "*Computer Vision and Applications, A Guide for Students and Practitioners*". Academic Press, ISBN 0-13-085198-1, 2000.

[8] Davies E., "*Machine Vision: Theory, Algorithms, Practicalities*". Morgan Kaufmann Publishers, ISBN 0-12-206093-8, 2004.

[9] Dhawan P., "*Medical Imaging Analysis. Hoboken*", Publisher: Wiley-IEEE Press, ISBN-10: 0471451312, ISBN-13: 978-0471451310, 2003.

[10] SIMERVILLE J., MAXTED W., and PAHIRA J., "*Urinalysis: A Comprehensive Review*". Am Fam Physician, 71(6):1153-1162, 2005.

[11] Online: "*Urinanalysis*", The Internet Pathology Laboratory for Medical Education, URL: http://library.med.utah.edu/WebPath/TUTORIAL/URINE/URINE.html

[12] "*McGraw-Hill Dictionary of Scientific & Technical Terms, 6E*". The McGraw-Hill Companies, Inc., 2003.

[13] Shapiro G. & Stockman C., "*Computer Vision*". Prentice Hall, 2001.

[14] Nixon M. and Aguado A., "*Feature Extraction and Image Processing*". Academic Press, 2008.

[15] Deng H., Zhang W., Mortensen E., Dietterich T., and Shapiro L., "*Principal Curvature-based Region Detector for Object Recognition*". Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[16] Efford N., "*Digital Image Processing, a practical introduction using Java*". Pearson Education, ISBN 0-201-59623-7, 2000.

[17] Young I, Gerbrands J., and Vliet L., "*Fundamentals of Image Processing*". Delft University of Technology, ISBN 90–75691–01–7, 1998.

[18] Gonzalez R. and Woods R., "*Digital Image Processing, 2nd Ed*". Prentice Hall, ISBN-10: 0201180758 | ISBN-13: 978-0201180756, 2002.

[19] Hu M., "*Visual pattern recognition by moment invariants*"., IRE Transactions on Information Theory, vol. 8, pp. 179-187, 1962.

[20] Huang Z. and Leng J., "*Analysis of Hu's Moment Invariants on Image Scaling and Rotation*". Proceedings of 2010 2nd International Conference on Computer Engineering and Technology (ICCET). pp. 476-480, Chengdu, China. IEEE, 2010.

[21] Heijden F., "*Image Based Measurement Systems: Object Recognition and Parameter Estimation (Design & Measurement in Electronic Engineering)*". Wiley, 1st Ed, ISBN-10: 0471950629 | ISBN-13: 978-0471950622, 1995.

[22] Johnson R. and Kuby P., "*Elementary Statistics*". Duxbury Press; 8th edition, ISBN-10: 0534356761, ISBN-13: 978-0534356767, 1999.

[23] Loeve M. , "*Probability Theory*". Springer-Verlag, 4th edition, ISBN: 0-387-90210-04, 1977.

[24] "*Engineering Statistics Handbook*", High Performance Computing and Communications/Systems Integration for Manufacturing Applications. Online URL: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm.

[25] Dodge Y., "*The Oxford Dictionary of Statistical Terms*". OUP. ISBN 0-19-920613-9, 2003.

[26] HARALICK R., SHANMUGAM K., and DINSTEIN I., “*Textural Features for Image Classification*”. IEEE Transactions on Systems, pp. 610-621, 1973.

[27] Online: “*Create gray-level co-occurrence matrix from image*”. Mathworks Product Documntation, URL: http://www.mathworks.com/help/toolbox/images/ref/graycomatrix.html

[28] Clausi D., “*An analysis of co-occurrence texture statistics as a function of grey level quantization*”. Canadian Journal of Remote Sensing, Vol. 28, No. 1, pp. 45–62, 2002.

[29] Soh L. and Tsatsoulis C., “*Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices*”. CSE Journal Articles. Paper 47, 1999.

[30] Deng K., “*OMEGA: ON-LINE MEMORY-BASED GENERAL PURPOSE SYSTEM CLASSIFIER*”. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Georgia Institute of Technology, 1998.

[31] Hall M., “*Correlation-based Feature Selection for Machine Learning*”. This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at The University of Waikato, 1999.

[32] Rodgers L. and Nicewander W., “*Thirteen ways to look at the correlation coefficient*”. The American Statistician, vol. 42, pp. 59-66, 1988.

[33] Francis DP, Coats AJ, and Gibson D. "*How high can a correlation coefficient be?*". Int J Cardiol, vol. 69, pp. 185–199, 1999.

[34] Han J. and Kamber M., “*Data Mining Concepts and Techniques*”. Diane Cerra, Morgan Kaufmman Publishers. ISBN 13: 978-1-55860-901-3, ISBN 10: 1-55860-901-6, 2006.

[35] LAROSE D., “*DSCOVERING KNOWLEDGE IN DATA An Introduction to Data Mining*”. John Wiley & Sons, Inc., ISBN 0-471-66657-2, 2005.

[36] Online: “*Introduction to Data Mining*”, Liu H., URL: http://www.eas.asu.edu/~mining03/

[37] Ye N., “*THE HANDBOOK OF DATA MINING*”. Lawrence Erlbaum Associates, ISBN 0-8058-4081-8, 2003.

[38] Witten I. and Frank E., “*Data Mining Practical Machine Learning Tools and Techniques*”. Diane Cerra, Morgan Kaufmann Publishers, ISBN: 0-12-088407-0, 2005.

[39] WEBB G., BOUGHTON J., and WANG Z. “*Not So Naive Bayes: Aggregating One-Dependence Estimators*”. Machine Learning, vol: 58, pp. 5–24, 2005.

[40] Williams W. and Lance G., “*Computer programs for hierarchical polythetic classification ("similarity analyses")*”. Computer Journal, pp. 60–64, 1966.

[41] Huang L. and Wang M., "*Image thresholding by minimizing the measure of fuzziness*". Pattern Recognition, vol: 28, pp. 41-51, 1995.

[42] Prewitt J. and Mendelsohn M., "*The analysis of cell images*". Annals of the New York Academy of Sciences, vol:128, pp. 1035-1053, 1966.

[43] Li C. and Lee C., "Minimum Cross Entropy Thresholding", Pattern Recognition, vol:26, pp. 617-625, 1993.

[44] Kapur J., Sahoo P, and Wong A., "*A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram*", Graphical Models and Image Processing. vol:29(3), pp. 273-285, 1985.

[45] Glasbey C., "*An analysis of histogram-based thresholding algorithms*", CVGIP: Graphical Models and Image Processing, vol:55, pp. 532-537, 1993.

[46] Ridler T. and Calvard S., "*Picture thresholding using an iterative selection method*", IEEE Transactions on Systems, Man and Cybernetics, vol:8, pp. 630-632, 1978.

[47] Burges C., “*Dimension Reduction*”. Now Publishers Inc, ISBN-10: 1601983786, ISBN-13: 978-1601983787, 2010.

[48] Fayyad U., Gregory P., and Padhraic S. "*From Data Mining to Knowledge Discovery in Databases*". Association for the Advancement of Artificial Intelligence, vol:17 No:3, 1996.

[49] Zhou X., Xiao X., and Ma C., "*A Study of Automatic Recognition and Counting System of Urine-Sediment Visual Components*", Proceedings of 3rd International Conference on Biomedical Engineering and Informatics, ©2010 IEEE, 2010.

[50] Ranzato M., Taylor P.E., House J. M., Flagan R.C., LeCun Y., and Perona P., "*Automatic Recognition of Biological Particles in Microscopic Images*". Pattern Recognition Letters, Volume 28, Issue 1, Pages 31-39, 2007.

[51] Song X., Sill J., Abu-Mostafa Y., and Kasdan H., "*Image Recognition in Context: Application to Microscopic Urinalysis*". In Advances in Neural Information Processing Systems 12, PAGES 963-969, 2000.

[52] Timna Esther Schneider, "*Automated Classification of Analysis and Reference Cells in Microscopic Images for Cancer Diagnostics*". Proceedings of the 11th International Student Conference on Electrical Engineering, 2007.

[53] Nugent C., and Cunningham P., "*Object Recognition and Active Learning in Microscope Images*". 2006.

[54] Li Y., Su G., and Li Z, "*Automatic Detecting Particle Objects in Image*". International Journal of Information Technology, Vol. 11, No. 7, 2005.

[55] Luo H., Ma S., Wu D., and Xu Z., "*Mumford-Shah Segmentation for Microscopic Image of the Urinary Sediment*". Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering, 2007.

[56] Li C., Fang B., Wang Y., Lu G., Qian J., and Chen L., "*Automatic detecting and recognition of casts in urine sediment images*". Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, ©2009 IEEE, 2009.

[57] Cao G., Zhong C., Li L., and Dong J., "*Detection of Red Blood Cell in Urine Micrograph*". Proceedings of 3rd International Conference on Bioinformatics and Biomedical Engineering, ©2009 IEEE, 2009.

[58] Santosa A., Ramiroa C., Descob M., Malpicaa N., Tejedorb A., Torresb A., Ledesma-Carbayoa M. J., Castillab M., and García-Barrenob P., "*Automatic detection of cellular necrosis in epithelial cell cultures*". Medical Imaging 2001, Proceedings of SPIE Vol. 4322, 2001.

[59] MENDEZ D., QUEZADA A., and LEHMAN M, "*Application of Neural Networks and Fractals for Urinalysis*". 2004.

[60] Hans C., Merchant F., and Shah S, "*Decision Fusion for Urine Particle Classification in Multispectral Images*". Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, ACM New York, NY, USA ©2010 Pages 419-426, 2010.

[61] CHEN L., FANG B., WANG Y., LU G., QIAN J., and LI C, "*AUTOMATED CLASSFICATION OF PARTICLES IN URINARY SEDIMENT*". Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, ©2009 IEEE, 2009.

[62] Mei-li S., and Rui Z., "*Urine Sediment Recognition Method Based on SVM and AdaBoost*". International Conference on Computational Intelligence and Software Engineering, 2009. ©2009 IEEE, 2009

[63] Calva D., García M., Martínez C., Salgado G., and Lehman M., "*Urine and Copro Recognition with Generalized Entropy and Neural Networks*". IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.

[64] Li Y., and Zeng X., "*A new strategy for urinary sediment segmentation based on wavelet, morphology and combination method*". Computer methods and programs in biomedicine, © 2006 Elsevier, 2006

[65] Mumford D. and Shah J., "*Optimal approximations by piecewise smooth functions and associated variational problems*". Comm. Pure Appl. Math., 42(5):577–685, 1989.

[66] Olson D. and Delen D., "*Advance Data Mining Techniques*". Springer, ISBN: 978-3-540-76916-3, 2008.