FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic blood typing scanner through agglutination

Nuno Miguel Duarte Costa

FOR JURY EVALUATION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Rosaldo José Fernandes Rossetti

Second Supervisor: Joaquim Gabriel Magalhães Mendes

July 11, 2014

Abstract

In this project is presented a study of the applicability of machine learning and computer vision techniques for reading and classifying gel card tests using supervised learning techniques, with the purpose of helping laboratory technicians evaluate ABO typing tests.

The main objective of this project is to build an automated system that performs the ABO typing test returning the result.

The used method consists in three different processes, one where image segmentation is used to segment the blood test in a tube image and extract relevant features using ellipse fitting, and other two who extract interest points from enhanced tube images and use the interest points position for classification.

The test tube images were acquired using a digital camera and 21 different gel cards, with 3 additional images generated digitally, giving a total of 144 gel tube images. Using features extracted from the fitted ellipse, a dataset was built with 144 instances and 6 ellipse attributes: the ellipse centre coordinates, the ellipse semi-major and semi-minor axes, the ellipse area and the ellipse orientation in radians. Another dataset was built using image interest points detected with the FAST algorithm, resulting in a dataset with 829 instances and 2 attributes: the interest point vertical and horizontal position. A final dataset was built using the centre position computed from the detected interest points corresponding to 144 instances, each with two numeric attributes: the vertical and horizontal position of the computed centre.

In the last Section, different classification algorithms were trained with the three built datasets. From the resulting models were chosen three that returned better accuracy, being stored and used in an application built to easily classify new gel cards.

The results show that the applied techniques correctly predicted new card tests for the two majority classes. The three minority classes produced satisfactory results during the training step. These three classes were not tested in the application due to the lack of real samples, only existing digitally created samples.

Resumo

Neste projeto é estudada a aplicabilidade de técnicas de aprendizagem computacional supervisionada e de processamento de imagem na leitura e classificação de cartões de gel usados para determinação do grupo sanguíneo por aglutinação.

O principal objetivo do projeto é construir um sistema autónomo que realize o teste ao grupo sanguíneo retornando o resultado.

O método usado consiste em três diferentes processos, um onde é usada segmentação de imagem para isolar o contorno da amostra de sangue presente em cada imagem de tubo, extraindo características relevantes para classificação através da aproximação do contorno segmentado a uma elipse. Nos dois outros processos, pontos de interesse são extraídos das imagens de tubos e a sua posição é usada para classificação.

As imagens dos tubos de teste foram capturadas usando uma câmara digital e 21 cartões de gel, com 3 imagens adicionais geradas digitalmente, resultando num total de 144 imagens de tubos de gel. Usando as características extraídas dos parâmetros da elipse, foi construido um conjunto de dados com 144 instâncias e 6 atributos: as coordenadas do centro da elipse, o semi-eixo maior e o semi-eixo menor da elipse, a área da elipse e a orientação da elipse em radianos. Um outro conjunto de dados foi construido usando a posição dos pontos de interesse detetados pelo algoritmo FAST, resultando em 829 instâncias e 2 atributos: a posição vertical e horizontal do ponto de interesse na imagem. O último conjunto de dados foi construido usando a posição central do conjunto de pontos de interesse detetados em cada tubo, resultando em 144 instâncias, cada uma tendo dois atributos numéricos: o valor vertical e horizontal da posição central de pontos.

Na ultima parte, diferentes algoritmos de classificação foram treinados com os três conjuntos de dados criados. Dos modelos resultantes, foram escolhidos para analise três modelos com melhor precisão, sendo estes guardados e aplicados numa aplicação criada para a fácil classificação novos cartões.

Os resultados obtidos, mostraram que as técnicas aplicadas conseguiram corretamente avaliar novos cartões de teste para as duas classes maioritárias. Apesar das três classes minoritárias não terem sido testadas na aplicação, devido à inexistência de amostras reais, estas obtiveram resultados satisfatórios no treino dos três diferentes processos.

Acknowledgements

I wish to thank various people for their direct and indirect invaluable contribution to this project.

I would like to begin first by expressing my deep gratitude to Professor Rosaldo José Fernandes Rossetti and Professor Joaquim Gabriel Magalhães Mendes, respectively my research supervisor and co-supervisor, for their guidance, expertise and valuable suggestions during the development of this Master's Dissertation.

I would also like to extend my gratitude to the Clinical Laboratory of Vale do Sousa, specially to Dr. Paulo Jorge Dessa Aguiar for his availability and help in offering the gel cards used in this project.

Finally, I would like to thank my friends and family who, over the past few years, through their love, encouragement and support, have made this learning experience an easier task.

Nuno Miguel Duarte Costa

"There are naive questions, tedious questions, ill-phrased questions, questions put after inadequate self-criticism.
But every question is a cry to understand the world.
There is no such thing as a dumb question."

Carl Sagan, The Demon-Haunted World: Science as a Candle in the Dark

Contents

List of Figures

List of Tables

Abbreviations

2D	2-Dimensional
3D	3-Dimensional
ARFF	Attribute-Relation File Format
DNA	Deoxyribonucleic Acid
DOG	Difference of Gaussians
FAST	Features from Accelerated Segment Test
GLOH	Gradient Location and Orientation Histogram
ID3	Iterative Dichotomiser 3
ISBT	International Society of Blood Transfusion
JUDOCA	Junction Detection Operator Based on Circumferential Anchors
k-NN	k-Nearest Neighbour
LOG	Laplacian of Gaussian
OpenCV	Open Source Computer Vision
PCA	Principal Component Analysis
PCR	Polymerase Chain Reaction
ROC	Receiver operating characteristic
RBC	Red Blood Cell
Rh	Rhesus
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machines
WEKA	Waikato Environment for Knowledge Analysis

Chapter 1

Introduction

This chapter presents the project context, motivation, objectives and structure.

1.1 Context and Motivation

This project is developed in the context of the MSc Dissertation in Electrical and Computers Engineering.

In the Human body, the blood is one of the most important elements to sustain Human life, serving as the delivery process for important substances like nutrients, oxygen, metabolic waste, body regulation functions and system protection. As an essential element for the well being of the Human body, it is important to understand how it behaves, if differences exists between Humans and if they do, how can they be differentiated.

In 1900, the Austrian scientist Karl Landsteiner identified three different blood types, O, A and B, through the existence of two antigens, A and B. Later in 1902, Alfred von Decastello and Adriano Sturli discovered the fourth type, AB, which had both the A and the B antigen. Over the years multiple blood group systems were developed, and as of October 2012, the International Society of Blood Transfusion (ISBT) officially recognizes thirty-three different blood group systems [6]. A blood group system is defined by the ISBT as the system where "one or more antigens controlled at a single gene locus or by two or more very closely linked homologous genes with little or no observable recombination between them" [7].

In clinical laboratories, the gel test is a used method for ABO and Rhesus (Rh) typing. ABO and Rh are blood type systems for blood classification based on the existence or lack of certain Red Blood Cell (RBC) antigen . In the gel card test, after a centrifugation step the test result is determined by a visual inspection of the card to determine how the blood samples reacted to the different antigens. Some systems have been built for automatic determining the gel card test result [8, 9] as well as some alternative systems using the same agglutination concept [10, 11].

This project aims to study the applicability of machine learning and computer vision techniques for reading and classifying gel card tests using supervised learning techniques.

1.2 Objectives

The objective of this project is to build an automated system for ABO typing using machine learning and computer vision techniques. The process should be automated and learn from past misclassification results.

1.3 Report Structure

This report is divided in five chapters. The current chapter contains the project context, motivation and objectives. In the chapter 2 the literature review is presented along some related work. In the chapter 3 the practical problem is presented, framed in the computer vision context and the adopted methodology is detailed. Finally, obtained results are presented in chapter 4 followed by a brief conclusion and future work in chapter 5 .

Chapter 2

Literature Review

A problem of computer vision applied to the analysis of static images is, as described by Trier et al. [12], usually divided in well defined processing steps. Starting with the digital acquisition of the image and its subsequent preprocessing, existing noise is reduced and the image is segmented to isolate its individual characteristics. The next step is to extract and store relevant and descriptive information from the segmented image, and then analyse this information with classification techniques. The process ends with a possible validation of the obtained results, so that wrong outcomes can be corrected. This chapter focuses in the segmentation, feature extraction and classification steps of the problem of computer vision applied to the analysis of static images, providing some used known techniques.

2.1 Image Segmentation

Image segmentation is the process of grouping image pixels with similar characteristics into homogeneous regions. Formally, image segmentation can be defined by [13, 14]:

$$\bigcup_{i=1}^{n} S_i = F \quad with \quad S_i \cap S_j = \emptyset, \quad i \neq j. \tag{2.1}$$

Being F, in the Equation 2.1, the set of all pixels of an image and a function $P()$ the uniformity predicate defined on groups of neighbour pixels, then segmentation is the partitioning of the initial set F into smaller connected subset or regions (S_1, S_2, ..., S_n). The uniformity predicate is $P(S_i) = true$ for all regions (S_i) and $P(S_i \cup S_j) = true$, when S_i is adjacent to S_j.

In computer vision, a good segmentation is essential to ensure good results in the recognition and analysis steps, as such different problems may require different segmentation methods or a combination of multiple methods. The various methods can be broken into measurement-space-based algorithms (e.g. level thresholding of histograms), pixel-similarity-based algorithms (e.g. region growing), pixel-difference-based algorithms (e.g. edge detection), and physics-based schemes [15].

2.1.1 Thresholding

In image segmentation, thresholding is one of the most simple methods to isolate image regions, specially when there is a distinct intensity difference between the object to extract and the image background. In the literature there is a vast number of implementations of thresholding, categorized by Sezgin and Sankur [16] in six groups, according to the kind of information each one uses:

- **Histogram shape-based methods** — Image segmentation is achieved based on histogram properties, like analysing concavities [17], peak [18] and valley analysis [19] and shape-modelling thresholding [20];
- **Clustering-based methods** — The optimal threshold value is obtained using clustering techniques, for example by iteratively determining the cluster mean [21] or minimizing the weighted sum of within-class variance [22]. Fuzzy clustering techniques have also been used in thresholding segmentation [23];
- **Entropy-based methods** — These methods rely on image entropy operations to determine the optimal threshold. These methods are divided by [16] into *entropic thresholding* [24], *cross-entropic thresholding* [25] and *fuzzy entropic thresholding* [26];
- **Object attribute-based methods** — The threshold selection is based in measures of similarity between the original image and its binary version, like image moments [27], edge fields [28] or shape compactness [29];
- **Spatial methods** — These class of algorithms utilizes grey value distribution but also dependency of pixels in a neighbourhood;
- **Local methods** — In these techniques the threshold is estimated for each pixel according to local image characteristics, like contrast [30] or mean and standard deviation [31, 32],

2.1.2 Edge Detection

Edge detection methods can be used as a segmentation tool to highlight image feature boundaries. These methods are usually divided in the literature in sequential and parallel techniques [13]. While in sequential techniques the decision that determines if a pixel represents an edge of a region is dependent in past decisions, in parallel techniques this decision is made based on the pixel in analysis and some of its neighbours. In theory, parallel techniques can be applied to every pixel in the image simultaneously. On the other hand, sequential approaches are dependent on the choice of a good initial seed and how previous decisions will influence the next pixel decision. Parallel approaches use contours detectors for image segmentation with no guarantee that the method result will be a closed region. There are multiple approaches to this problem and it has been shown that regions can be recovered from a contour detector [33, 34]. Early parallel approaches of edge detection respond to local changes in grey level or average grey level of the

image. Roberts, Sobel and Prewitt gradient operators, called first difference operators, and the Laplacian operator, a second difference operator, detect edges by convolving a grey-scale image with local derivative filters. Marr and Hildreth proposed the use of the zero-crossing segments of the Laplacian of Gaussian operator (LOG) for contour detection, or replacing the previous LOG operator by the Difference of Gaussians (DOG) operator for a fast approximation [35].

Canny defined an optimal edge detector method as an algorithm with the three main characteristics [36]:

1. **Good detection** — There should be a low probability of failing to mark real edge points, and low probability of falsely marking non-edge points;

2. **Good localization** — Points marked as edges by the operator should be as close as possible to the centre of the true edge;

3. **Single edge response** — The detector should only produce a single output in response to a single edge.

With these performance criteria Canny proposed a optimal edge detector by convolving the image with an optimal filter, that can be approximated by the first derivative of the Gaussian, and selecting the output maxima as the edges. For the two-dimensional scheme, the image is convoluted with the first order derivatives of the two-dimensional Gaussian. Computing the gradient of the smoothed image, the edges are then located at the maxima of the gradient modulus taken in the direction of the gradient.

Detection of lines, which correspond to local extrema of the grey level image, can also give valuable information in the position of image edges. Haralick [38] proposed an algorithm where lines occur at pixels having zero-crossings of the first direction derivative taken in the direction that maximizes de second directional derivative. Giraudan [39] proposed similar algorithm detecting a line at a negative local maximum of the second derivative of the image. In an analogous way, junctions can identify image corners and the orientations of the edges forming it. Rangarajan et al. [40] derive an optimal detector using Canny's criteria, that locates junctions formed by two linear symmetric edges. The junctions are then located at the local maxima output of the image convolution with twelve different masks that differ in orientation and angle. Elias and Laganière [41] propose an junction detection operator based on circumferential anchors (JUDOCA), that can be split into two main phases, an edge detection phase and a junction detection phase.

2.1.3 Region Operations

Region based approaches attempt to group pixels into homogeneous regions, clustering images features such as brightness, colour or texture. Region growing and splitting are two basic region operators. While the first starts from a initial point as the seed region and then expands to its homogeneous neighbours, the splitting approach uses the whole images as the initial region seed and then keeps splitting the region until it is homogeneous. Region merging is usually used after

these operators to join similar regions. One major drawback is the inherent sequential nature of these techniques.

Arbeláez et al. [34] identify three widely used algorithms of region segmentation: a graph-based region merging algorithm proposed by Felzenszwalb and Huttenlocher [42] which produces segmentations that are not too coarse and not too fine, despite making simple greedy decisions; the mean shift segmentation algorithm by Comaniciu and Meer [43] where regions are formed by grouping pixels whose convergence points are closer than a threshold h_s in the spatial domain and h_r in the range domain; the spectral graph theoretic framework of normalized cuts who partitions the image into regions of coherent texture and brightness [44, 45].

2.1.4 Level Set Segmentation

The level set method, initially introduced for capturing moving fronts [46], is applied to segmentation by trying to represent a contour C in the image plane $\Omega \subset \mathbb{R}^2$ as the zero-level of an embedding function $\phi : \Omega \to \mathbb{R}$: $C = \{x \in \Omega \mid \phi(x) = 0\}$ [47]. There are two different approaches for evolving the zero-level set [48], a global approach that updates the level set function globally (e.g. Simplified Mumford-Shah Model [49]) and an approach that focuses on the local evolution of the curve provided at initiation, only solving the level set equation around a narrow band in the neighbourhood of the zero level set.

Level set methods have been used in the medical field, e.g. Angelini et al. [50] review the use of level set methods in segmentation of medical images, and Jiang and Nie [51] use a level set based on the Mumford-Shah Model in a problem of urine sediment image segmentation.

2.2 Feature Extraction

Feature extraction in image analysis is the process of translating image features to easily processable digital information. Some known techniques are based on shape attributes, contour, texture or local image descriptors. From the past discussed methods for image segmentation some can also be used as image descriptors, such as the image histogram, image edges or image blobs. Local descriptors can be considered a robust method for image representation that allows to identify objects or scenes even without previous image segmentation [52, 53]. These features do not need to translate directly into a object propriety, since the goal is to analyse their statistics.

2.2.1 Region Descriptors

Region descriptors are features extracted from segmented image regions. These features can be simple descriptors as region area and perimeter, shape circularity measurements [54], mean and median values of region pixels, shape eccentricity and elongatedness ratios or region moments.

Gianfranco and Yi [55] proposed the use of region moments as invariant region descriptors. Region moments were presented as descriptors based on an extension of the image moments concept. Region moments represent a statistical description of the region data which is a projection

onto the space of polynomials with two variables. In [55] the authors present three scale and rotation invariant region moments descriptors: the first computes the central moments of the image features of a region; the second one is the set of the Hu's invariants [56] computed over every image feature; and the third computes the radial moments of the image features.

2.2.2 Boundary Descriptors

A region boundary can also be a useful descriptor. Fourier descriptors stand out for being able to approximate a region contour with only a few descriptors by encoding a rough boundary curve in the initials Fourier descriptors, giving each new descriptor a more accurate contour recovered from the inverse Fourier transform [57, 58].

2.2.3 Region Covariance

Tuzel et al. [59] proposed the use of the covariance of multiple region statistics as the region descriptor. The result is a low-dimensional covariance matrix that fuse multiple region features which might be correlated.

$$\mathbf{C}_R = \frac{1}{n-1}\sum_{k=1}^{n}(z_k - \mu)(z_k - \mu)^T \tag{2.2}$$

The covariance matrix of a region R is then given by $\mathbf{C}_R$, in the Equation 2.2, where $\{z_k\}_{k=1\ldots n}$ are the feature points inside R and μ the mean of the points. In the paper, the authors also propose an alternative nearest neighbour search algorithm adapted to the covariance features.

2.2.4 Local Feature Detectors

Lowe presented the scale invariant feature transform algorithm (SIFT) as a procedure of transforming image data into scale-invariant coordinates relative to local features [60]. The process is divided in four major steps:

1. **Scale-space extrema detection** – a DOG function is used to identify potential interest points that are invariant to scale and orientation;

2. **Key point localization** – At each candidate location, a model is fit to determine location and scale. Key points are then selected based on measures of their stability;

3. **Orientation assignment** – One or more orientations are assigned to each key point location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations the features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3-Dimensional (3D) camera viewpoint;

4. **Key point descriptor** – The local image gradients are measured at the selected scale in the region around each key point. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination

Despite generating big descriptor sets, the cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test.

The gradient location and orientation histogram (GLOH), by Mikolajczyk and Schmid [61] is proposed as an extension to the SIFT descriptor. The GLOH was designed to increase robustness and distinctiveness of the SIFT, also applying principal component analysis to reduce the big descriptors size, as suggested by Ke and Sukthankar [62].

Speeded up robust features (SURF) is a fast scale and rotation invariant feature detector and descriptor presented by Bay et al. [63] based on the Hessian-matrix and sums of 2-Dimensional (2D) Haar wavelet being the later obtained with the aid of the integral image.

Features from accelerated segment test (FAST) is a corner detection method developed by Rosten and Drummond [64, 65] that uses the Iterative Dichotomiser 3 (ID3) algorithm for corner pixel selection. A more detailed analysis of this algorithm is present in the fourth chapter of this report.

2.3 Features Classification

One of the last steps in a computer vision problem, is interpreting the extracted representative features. In this project problem, ABO typing, the classification step serves as a prediction process for determining the outcome the ABO test result.

Classification is a supervised learning process which is divided in two main steps, an initial learning step where a classification model is built from a set of previously classified training data, and a test step that uses the obtained model and tests its accuracy. This two step process requires the division of the problem data in two sets of randomly selected tuples, a training set for the initial learning step and a test set for the test step. If the accuracy of the model in the second step is considered acceptable, the obtained model can be used to classify future data tuples.

2.3.1 K-Nearest Neighbour

The k-Nearest Neighbour (k-NN) is simple a non-parametric classification algorithm where unclassified samples are, according to a specified distance measurement, assigned to the class of the nearest K neighbours in the training set. This training set samples exist in a n-dimensional space, where n represents the number of numeric attributes of each sample. To classify a unknown sample, a classifier searches for the K nearest training samples in the n-dimensional space, and assigns it the most common class. This measurement of distance can be, for instance, defined in terms

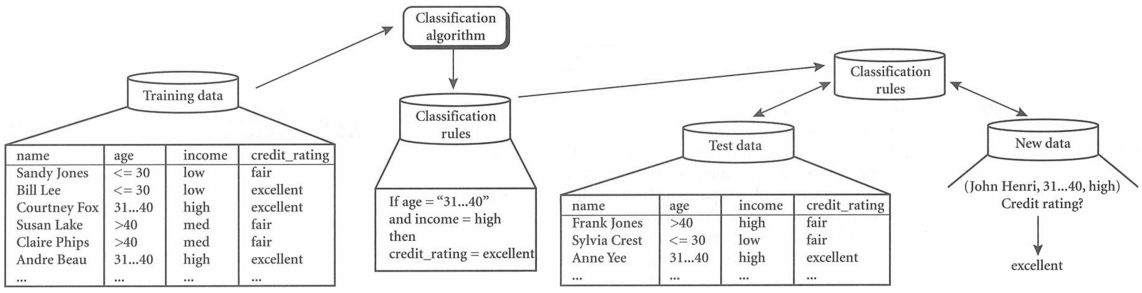


Figure 2.1: The data classification process
In the first step rules are obtained by applying a classification algorithm to the training data, and then the accuracy of those rules are benchmarked in test data. If the accuracy is acceptable, the model can be used in new data tuples [1].

of Euclidean distance, where the Euclidean distance between two points, $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ is

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2.3}$$

Nancy and Asoke [66], use a k-NN classifier to segment retinal blood vessels in digital colour fundus images. The authors use the green channel intensity, the local maxima of the gradient magnitude, and the local maxima of the largest eigenvalue to classify the pixels in colour retinal images into vessel and non-vessel pixels. A post-processing step is used to remove small segments through a iterative thresholding strategy. The authors conclude that the selected three features significantly reduce the processing time when comparing with a competing method that uses thirty-one features, while maintaining comparable sensitivity on pixel classification.

2.3.2 Decision Trees

Decision trees are described by Han et al. [1] as a flow-like tree structure, where each internal node denotes a test on a attribute, each tree branch represents an outcome of the test and each terminal node (leaf node) holds a class label. For a classification problem, having a sample with an unknown class label, the attribute values of the sample are tested against the decision tree, tracing a path from the root to the leaf node, which holds the class prediction for that sample.

In a study to classify a lung cancer dataset, Islam et al.[67] adapted ID3 recursive partitioning as the decision tree algorithm, incorporating entropy into the process. Rajendran and Madheswaran [68] developed a hybrid technique using association rules with decision trees for brain tumour classification.

2.3.3 Naive Bayesian Classification

The naive Bayesian classification technique is a statistical classifier based on the Bayes' theorem, that can predict class membership probability, assuming that all class attributes are independent of

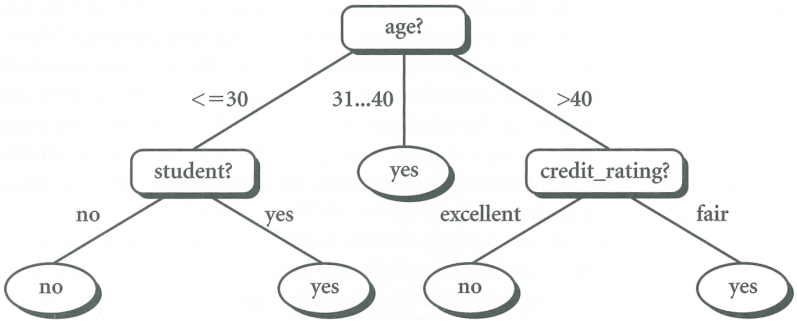


Figure 2.2: Decision tree for classification
Example of a decision tree for classification problem: Does the client buy a computer? [1]

each other. The Bayes' theorem provides a way of calculating the posterior probability, $P(H|X)$, from $P(H)$, $P(X)$ and $P(X|H)$ (Equation 2.4).

$$P(H|X) = \frac{P(XH)P(H)}{P(X)} \tag{2.4}$$

Deepa and Mymoon Zuviriya [69] benchmarked a Naive Bayes classifier, a Support Vector Machine (SVM) classifier and a k-NN classifier with digital retinal images for blood vessel detection. The authors achieved a better result with the SVM classifier than with the Naive Bayes and the k-NN classifiers.

Schmitt and McCoy [70], in object classification and localization using SURF descriptors, use a Naive Bayes and a multiclass SVM algorithms to classify images using Bags of Visual Words composed of histogram vectors. The authors achieved better results with the Naive Bayes classifier, achieving 80% accuracy with a training set of 100 instances and close to 88% accuracy with 100 instances.

2.3.4 Artificial Neural Networks

A neural network is defined by Han et al. [1] as a set of connected layers of input/output units where each connection has a weight associated with it. This weights are adjusted in the learning phase until the network is able to predict the correct class label of the input training set. While having a hight tolerance to noisy data, neural networks also require long training times and usually big training sets.

In feed-forward neural networks, each unit can only output to units in the succeeding layer.

The backpropagation algorithm can be used to train the neural network, by iteratively processing the network's prediction for the training set, adjusting the weights, and trying to minimize the mean squared error between the prediction and the actual class value. This is done from the output layer, through each hidden layer down to the first hidden layer.

Antonie et al. [71] applied neural networks using backpropagation for tumour classification in mammograms. The authors use an existing dataset containing type of tissue and position of the breast, and enhance it with new four statistical parameters extracted from the medical images:

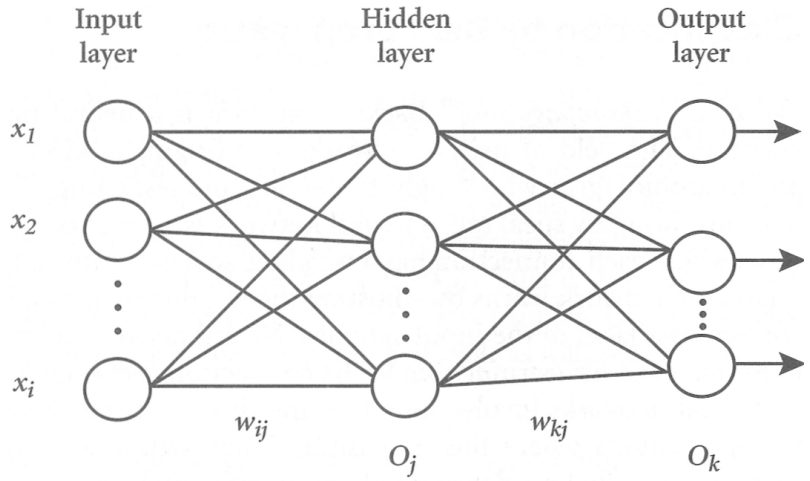


Figure 2.3: A feed-forward neural network with two layers [1].

mean, variance, skewness and kurtosis. This enhanced dataset is then used to build and test two classification models, one using neural networks and other using association rule mining. The first model proved to be less sensitive to database imbalance than the second model built with association rule mining, at the cost of having high training times.

Veluchamy et al. [72] use artificial neural networks to classify blood cells into normal and abnormal. The authors extracted twenty seven features from each cell image from which four were geometrical features, sixteen statistical features and seven moment invariant features. These features were classified with a back propagation neural network with twenty-seven neurons in the input layer, twenty-two in the hidden layer and one in the output layer. The authors achieved a classification efficiency of 80% for normal cells and 66.6% for abnormal cells.

2.3.5 Support Vector Machines

Support Vector Machines (SVM) are supervised learning methods that given a two-class, separable, data set, try to find an optimal hyperplane which separates the two classes.

The optimal hyperplane is the one that maximizes the margin between classes' support vectors dividing the input space, geometrically, into two half spaces. The decision hyperplane can be a linear separator on a two or higher dimensional space, or a non linear separator by relying on a continuous kernel function to divide classes. As noted by Chapelle et al., while SVM are designed for binary classification, they can be adapted to support a multi-class dataset, either by directly modifying its design [73], or by combining multiple binary classifiers in a one-against-one [74] or one-against-all [75] disposition.

Mohapatra et al. [76], use SVM to classify shape and texture features of blood cells for automated leukaemia detection. The authors developed a two step segmentation stage for white blood cells nucleus extraction from microscopic images of blood, first using Gustafson Kessel clustering and followed by nearest neighbour classification in the L*a*b* colour space. From

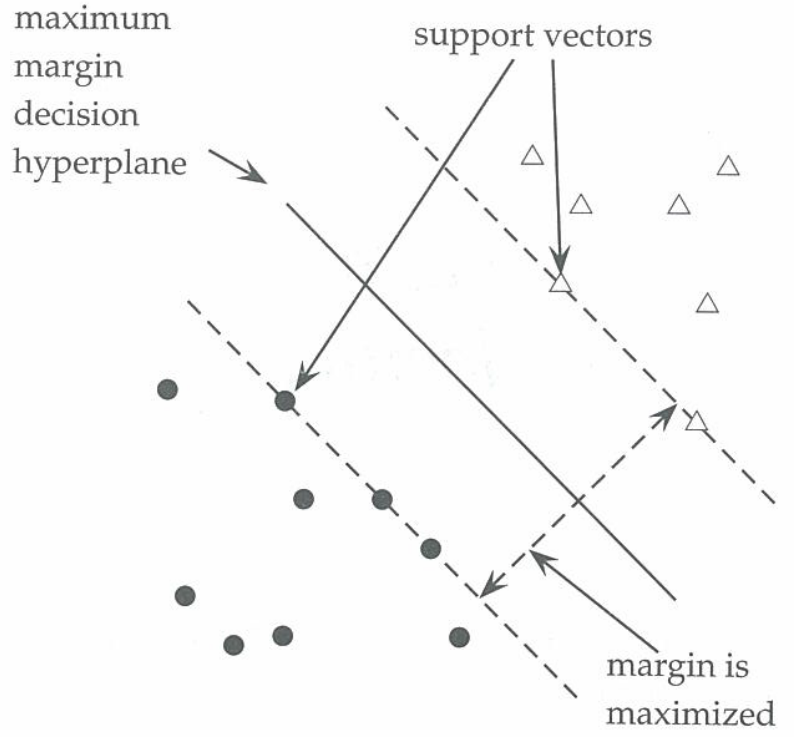


Figure 2.4: Hyperplane separating the two classes of the dataset. [1].

the segmented images, the authors extract shape features using Hausdorff dimension and contour signature, along with colour and texture features. Training the SVM classifier with the selected features, the authors achieve a 93% accuracy in captured images.

Tai et al. [77], propose a hierarchical blood cell image identification and classification method based on a one-against-one multi-class SVM. The authors use Otsu's adaptive threshold algorithm and morphology operations to build a binary image of blood cells, separating touching cells with watershed segmentation. From the segmented image, histogram, area, circularity, cytoplasm ratio and colour were extracted from each cell and applied to the SVM classifier.

2.4 Related Work

Currently, thirty-three different blood group systems are recognized by the ISBT [6]. In 1999, Kane et al. [78] divided the different kinds of blood groups into five types:

1. Blood group antigens including ABO typing, detected by agglutination through an antigen-antibody reaction;

2. Erythrocyte enzyme types;

3. Serum types, differentiated by a polymorphisms of factors that are detected in serum proteins, immunoglobulins and complements;

4. Human leukocyte antigen system, a histocompatibility antigen;

5. Deoxyribonucleic acid (DNA) typing based on length and sequence polymorphisms, that greatly improves individual discrimination.

In the literature sevevral authors have addressed the ABO typing problem. Kobayashi and Akane [79] applied the inverse polymerase chain reaction (PCR) method for simultaneous nucleotides identification. In the inverse PCR method, two distinct regions are inter-ligated for determining three major alleles (A(1), B and O(1)) of the ABO blood group.

Doi et al. [80] developed a ABO genotyping method using a multiplex single-base primer extension reaction, that allows simultaneous detection of six single nucleotide polymorphisms sites in the ABO gene and detection of ABO genotypes from their combinations. The authors achieved good results, correctly identifying all samples of peripheral blood DNA extracted from 103 Japanese individuals.

Srivastava et al. [81], developed a microfluidic platform for quantifying the dependence of RBC responses by ABO-Rh blood type via direct current insulator dielectrophoresis. Video was captured and processed, analysing intensities from five interrogation windows to calculate the fraction of cells in each outlet channel as a function of time. SVM algorithms were used for classification and logit transformation for multivariate hypothesis testing. The authors concluded that a direct current insulator dielectrophoresis device could sort RBC by blood type due to its membrane antigens influence the RBC polarizability in direct current insulator dielectrophoresis field.

Houngkamhang et al. [82] studied readily available antibodies used in standard agglutination tests for their use in ABO blood typing by a surface plasmon resonance imaging technique. The authors used five groups of antibodies to construct detection arrays using a multichannel flow cell, obtaining consistent results with those using standard agglutination technique.

Ferraz et al. [10] presented a blood phenotype system based on the agglutination technique using image processing. The authors developed a portable device for ABO-Rh blood typing and blood phenotype that automates the analysis procedure, including centrifugation, reading and interpretation of results. For interpreting the results, after centrifugation an image of six containers is captured, containing the centrifuged mixture of blood and (specific antigen) reagent. Using image segmentation techniques the authors extract five image features from each container: area, mean value, standard deviation, minimal value, and maximal value, using the standard deviation value for reagent tests classification.

Chapter 3

Methodological Approach

In clinical laboratories the gel test is a used method for ABO and Rh typing. ABO and Rh are blood type systems for blood classification based on the existence or lack of certain Red Blood Cell (RBC) antigens. The proposed gel test by Lapierre et al. [83] uses micro-tubes filled with a mixture of gel, buffer, and reagents that are centrifuged with a suspension of RBCs. After centrifugation in negative reagent reactions the RBCs pass through the gel and collect at the bottom of the micro-tube, whereas in positive reagent reactions they are dispersed throughout the length of the gel. This project aims to develop a prototype capable of automatically reading and determining a gel card test, storing the yielded results. In this system, the captured card's image and serial number should be stored in the system for later evaluation.

This chapter presents an overview of the gel card test with a description of the project tools and the program methodology.

3.1 Gel Card

The gel cards studied in this project are developed by Bio-Rad Laboratories. The gel cards are designed for ABO forward test and the determination of RH1 (D) antigens. Each card has six gel micro-tubes where five of them are impregnated with a reagent specific to the erythrocyte antigen to be determined. The remaining micro-tube is used as control signal for test validity. After centrifugation, non-agglutinated RBCs are collected at the bottom of the micro-tube while agglutinated RBCs are dispersed throughout the length of the gel being their position in the gel the intensity indicator of the reaction (Figure 3.2).

The first three micro-tubes of the card, ABO1(A), ABO2(B) and ABO3(AB) as seen in the Figure 3.1, are specific to the ABO grouping which requires two complementary tests for complete validation. The forward test conducted with anti-ABO1(A), anti-ABO2(B) reagents and if necessary anti-ABO3(AB) reagent need to concord with the reverse test conducted with the A1, B reagent RBC and if necessary A2 and O reagent RBC. If this is not verifiable, and a discrepancy exists between the two tests, complimentary tests should be conducted before any ABO result.

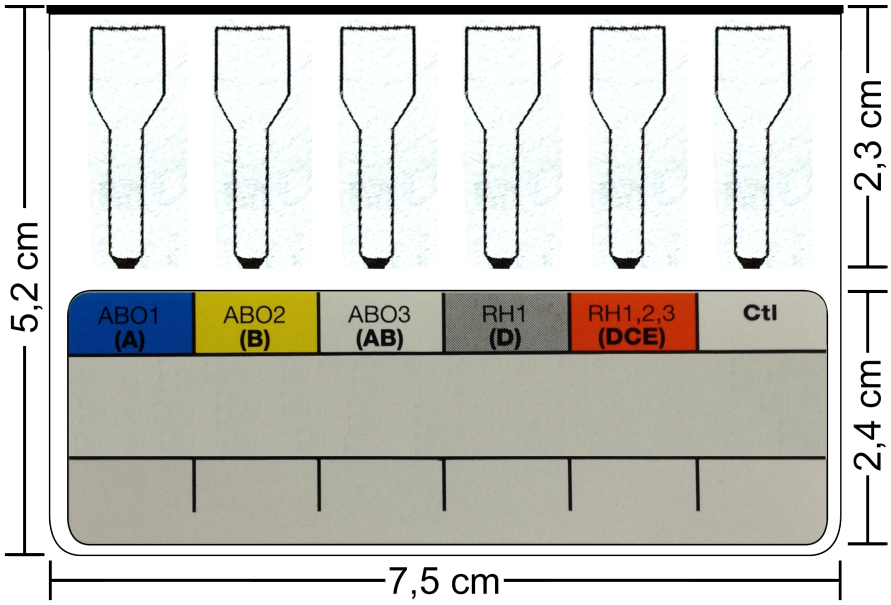


Figure 3.1: Gel card

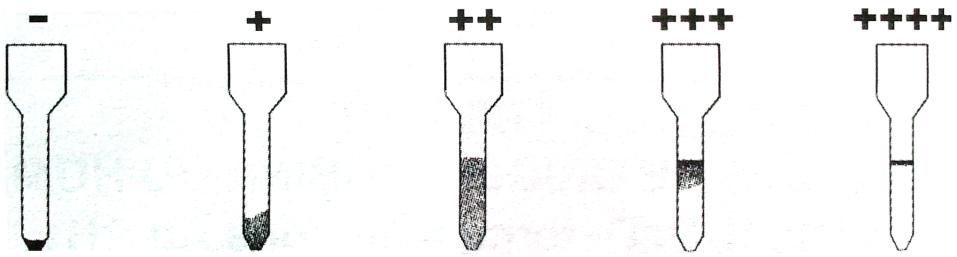


Figure 3.2: Test results for different strengths of reaction [2]

A positive result in the forth card microtube, RH1(D), indicates the presence of antigen RH1(D) on the surface of RBC. This gel card does not allow the detection of all D weak antigens neither the detection of the phenotype RH1 partial category VI (DVI). For these cases, it is recommended an additional complementary test [2]. A positive result in the fifth microtube RH1, 2, 3 (DCE) indicates the presence of antigen RH1 (D) and/or antigen RH2 (C) and/or antigen RH3(E) on the surface of the RBC. The Table 3.1 can be used for easier interpretation of these results.

Groups	ABO Forward Test			ABO Reverse Test			
	Anti-ABO1 (A)	Anti-ABO2 (B)	Anti-ABO3 (AB)	A1	A2	B	O
A	+	-	+	-	-	+	-
B	-	+	+	+	+	-	-
AB	+	+	+	-	-	-	-
O	-	-	-	+	+	+	-

Table 3.1: Reagent result interpretation

3.2 Software Tools

On this project an application was built, capable of reading and classifying test results of gel card micro-tubes. This application was developed in Java using the JavaFX™ library for the user interface, this ensures better code compatibility between platforms. The BoofCV library was used for image segmentation and feature extraction, the Waikato Environment for Knowledge Analysis (WEKA) software workbench for the data classification problem, the H2 Database Engine as local data storage and the ZXing ("Zebra Crossing") library was used for detecting and reading the gel card identification barcode. A preference was given to this three libraries since they are all open source, written entirely in Java code and can be easily embedded in the main application package, do not requiring platform specific configuration. Later in the project development the Open Source Computer Vision (OpenCV) library with desktop Java bindings was included in two different procedures for image feature extraction.

3.3 Methodology

This project is divided in two main steps. First the study and development of the main structure regarding the image segmentation, feature extraction and classification problem in order to identify the six micro-tubes in a gel card photography, extract meaningful and representative information and achieve a test result by means of data classification. In the second step an automated procedure was built from drawn classification results that classifies card images and saves acquired information in a local database. This process should be completely autonomous only requiring human intervention for test validation.

In the first, step three procedures were studied. While the first one extracts predefined region features from segmented image tubes, the other two use the Features from Accelerated Segment Test (FAST) algorithm for corner detection, using detected interest points for extracting features.

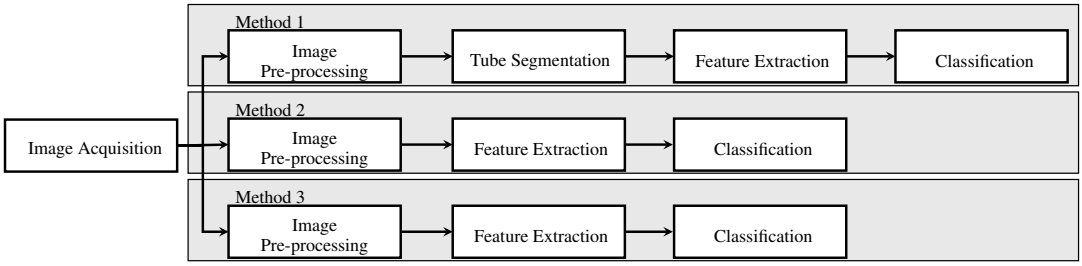


Figure 3.3: Main project structure

3.3.1 Image Acquisition

For the image acquisition process a digital camera (Microsoft® LifeCam Studio™) was used, capable of acquiring images at a resolution of 1920 per 1080 pixels at a focal length starting in 10 centimetres up to infinity. The digital camera has a height of 54 mm, 29 mm in width and 60 mm in length, being small enough for a image acquisition prototype and maintaining a good image resolution.

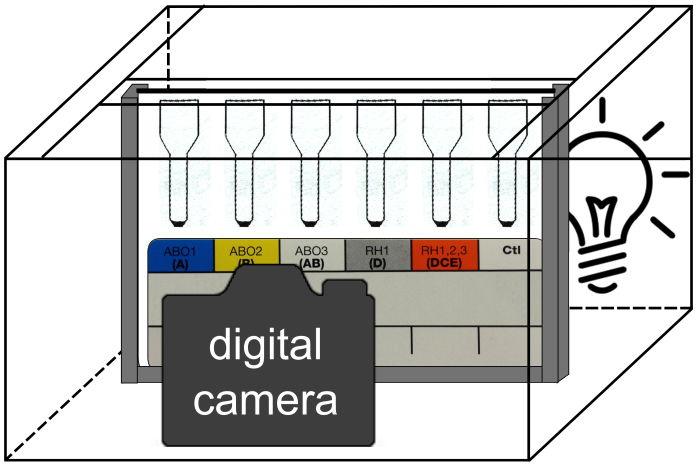


Figure 3.4: Image acquisition system

A physical prototype composed of a base, card stand, white background and a light stand was assembled using steel spring wire, paper and glue. The card stand has a height of 56 mm, 78 mm in width and 5 mm in length, positioned at 105 mm of the digital camera. The white background has a height of 175 mm and 298 mm in width, positioned at 80 mm of the card stand, filling the digital camera's angle of view. As a illumination system, a 23 Watt (168 mA) light source was positioned approximately 190 mm above the card stand using a metal stool as support.

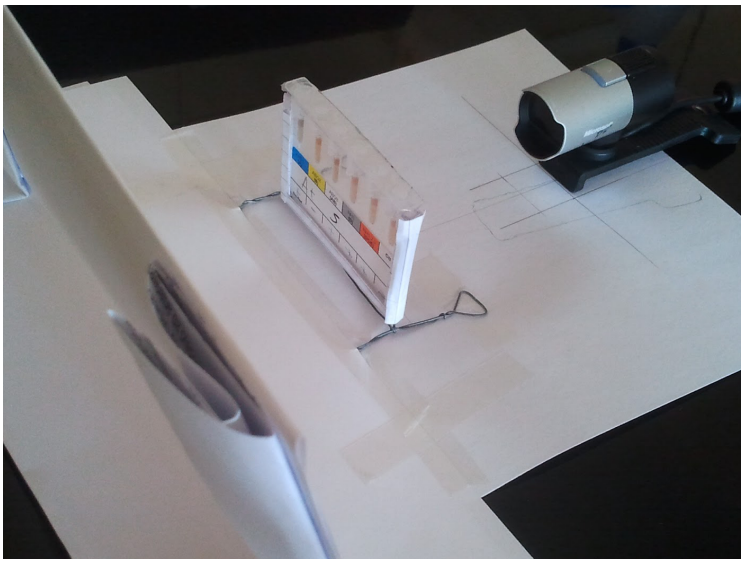


Figure 3.5: Image acquisition prototype

Using this prototype, images of twenty-three different gel cards were captured and saved to the file system of the development platform. Of these images, two were initially kept back for testing later the application. In the supplied gel cards only negative (-) and strong positive (++++) reagent reactions were present, not existing any instances of the other three positive reagent reactions (+, ++ and +++). To try to compensate for this imbalance, three gel card sample images were digitally created from a captured image mimicking the expected results using a sample card template as source.

3.3.2 Image Pre-processing

In this project, all image operations are applied to the green channel of the card image. The green channel was chosen because it exhibited less noise than the other two channels, which could be the result of the Bayer filter mosaic present in the CMOS sensor. The green channel consequently increased the contrast between the blood sample and the gel present in the first two micro-tubes, who have a blue and a yellow coloured gels, respectively.

For extracting the six individual micro-tubes, first the card barcode is read using the ZXing library, storing the card identification number. Next, a template matcher is created using the BoofCV library. The matcher works by moving the template (Figure 3.6) across the card image and for each pixel determining how similar that region is to template using the sum of the difference squared $-\sum_{o,u}\left(I\left(x,y\right)-T\left(x-o,y-u\right)\right)^2$.

Before running the pattern matcher routine, to enhance the pattern present in the gel card, a linear histogram expansion is applied to the card image. Image pixels with a value equal or lower than 50 are set to 0 and pixels with a value equal or higher than 100 are set to 255.

Knowing the gel card dimensions, the barcode width and the top-left coordinate of the pattern in the gel card, the six gel tubes are extracted from the card using the barcode width as a scale factor and the calibration pattern position for card alignment.

Figure 3.6: Alignment pattern

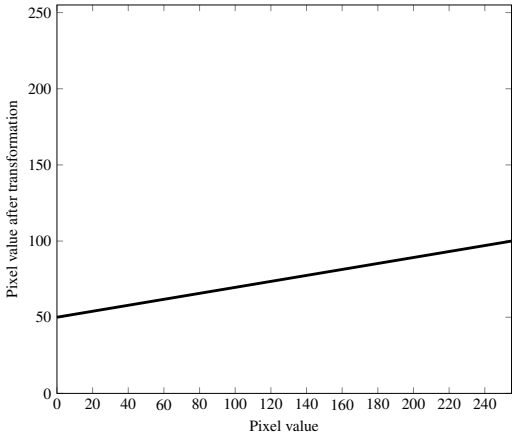


Figure 3.7: Histogram expansion diagram

Figure 3.8: Card image calibration and gel tube extraction

In the two procedures that rely on the FAST algorithm for detecting points of interest, to each individual tube is applied a threshold 5% higher than the tube mean value, removing bright image spots (e.g. light reflections). After, for noise removal, a median filter with a kernel size of 3 is applied to the tube image, followed by an unsharp masking method for increasing the chances of corner detection by the FAST algorithm.

3.3.2.1 Bilateral Filter

The bilateral filter is a non-linear filtering technique which replaces each pixel value with a weighted average of its neighbourhood while preserving sharp edges [84]. The weighted aver-

age is given by the sum of each image pixel existing in a k by l window multiplied by a weighted coefficient $w(i,j,k,l)$, and later divided by the sum of the weighted coefficient $w(i,j,k,l)$ (Equation 3.1). This coefficient depends on its own of two smoothing parameters, the domain or spatial parameter σ_d and the range parameter σ_r (Equation 3.2).

$$g(i,j) = \frac{\sum_{k,l} f(k,l)\, w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)} \tag{3.1}$$

$$w(i,j,k,l) = exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2} \right) \tag{3.2}$$

3.3.2.2 Unsharp Masking

The unsharp masking is a image sharpening technique that enhances image edges by building an edge image and adding it back into the original [3]. This edge image is built by subtracting a smoothed version of the image from the initial image.

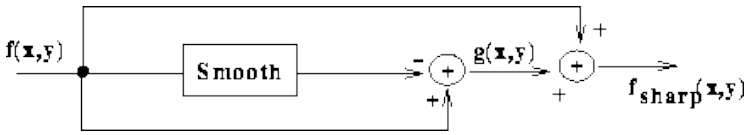


Figure 3.9: Unsharp masking technique [3]

The unsharp masking technique pre-dates digital photography, and was usually used to sharpen images in the darkroom:

> "In fact, before the advent of digital photography, this [unsharp masking] was the standard way to sharpen images in the darkroom: create a blurred ("positive") negative from the original negative by misfocusing, then overlay the two negatives before printing the final image" [84, chap. 3.2 - Linear filterin]

3.3.3 Image Segmentation

In the first tested procedure, image segmentation is applied to each image tube to extract a binary image of the blood test after centrifugation. First, the greyscale tube image is reduced to a binary image using a threshold computed using Otsu's method, followed by a morphological opening. Next, contour information is extracted from each connected cluster of pixels, using the contour to compute the best fit ellipse based on minimizing Euclidean distance. To remove detection errors, from the extracted ellipses is selected the one with the bigger area.

3.3.3.1 Otsu's Method

Otsu thresholding technique is a global thresholding method introduced by Otsu [22] as a non-parametric and unsupervised method of automatic threshold selection for picture segmentation.

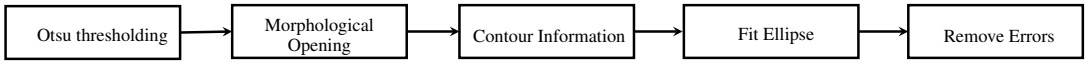


Figure 3.10: Image segmentation steps

This technique iterates through all possible thresholds and evaluates its "goodness", selecting the threshold that better separates image pixels into two classes, foreground and background, i.e. the threshold that minimizes within-class variance.

Using the histogram of a greyscale image with L grey levels, dichotomizing the image pixels into two classes C_0 and C_1 with a threshold k, the probabilities of class occurrence are given by

$$\omega_0 = Pr(C_0) = \sum_{i=1}^{k} p_i = \omega(k) \tag{3.3}$$

$$\omega_1 = Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \tag{3.4}$$

the class mean levels by

$$\mu_0 = \sum_{i=1}^{k} iPr(i|C_0) = \sum_{i=1}^{k} \frac{ip_i}{\omega_0} = \frac{\mu(k)}{\omega(k)} \tag{3.5}$$

$$\mu_1 = \sum_{i=k+1}^{L} iPr(i|C_1) = \sum_{i=k+1}^{L} \frac{ip_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \tag{3.6}$$

and the class variance

$$\sigma_0^2 = \sum_{i=1}^{k} (i - \mu_0)^2 Pr(i|C_0) = \sum_{i=1}^{k} (i - \mu_0)^2 \frac{p_i}{\omega_0} \tag{3.7}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (i - \mu_1)^2 Pr(i|C_1) = \sum_{i=k+1}^{L} (i - \mu_1)^2 \frac{p_i}{\omega_1}. \tag{3.8}$$

Minimizing within-class variance σ_W^2 is the equivalent of maximizing between-class variance σ_B^2, reducing the process to an optimization problem:

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \tag{3.9}$$

being the optimal threshold k^*

$$\sigma_B^2(k^*) = \max_{1 \le k < L} \sigma_B^2(k). \tag{3.10}$$

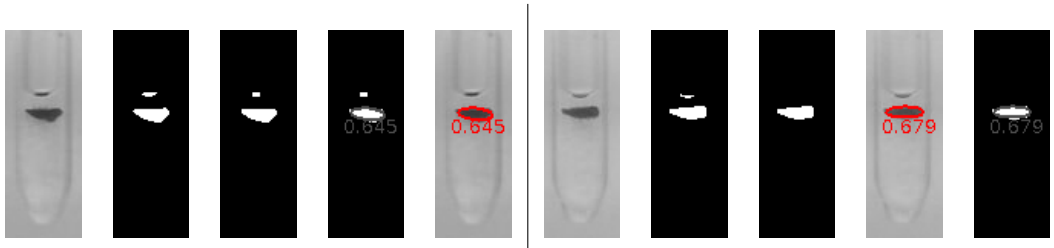


Figure 3.11: Tube segmentation process
(a) Original tube image (sample 1); (b) Binary image using threshold selected with Otsu's Method (sample 1); (c) Binary image after morphological opening (sample 1); (d) Detected ellipse over binary image (sample 1); (e) Detected ellipse over original image (sample 1); (f) Original tube image (sample 2); (g) Binary image using threshold selected with Otsu's Method (sample 2); (h) Binary image after morphological opening (sample 2); (i) Detected ellipse over binary image (sample 2); (j) Detected ellipse over original image (sample 2).

3.3.3.2 Morphological Opening

Opening is a mathematical morphology operator, used to move small pixels from the foreground of an image into the background. The operator works by first running a erosion ($\ominus$) followed by a dilation ($\oplus$), both also morphological operators, using the same structuring element.

$$A \circ B = (A \ominus B) \oplus B \tag{3.11}$$

3.3.3.3 Contour Extraction

In this project a method proposed by Chang et al. [85] is used for labelling connected components and theirs contours in binary images. This method is a linear-time component-labelling algorithm that uses a contour tracing technique to detect the contour of each image component, while identifying and labelling the interior area of each component.

The algorithm scans the image from top to bottom and from left to right per each line. When a contour is encountered, a contour-tracing procedure is used to complete the contour and assign a label to all pixels on the contour (Figure 3.12). After finishing the contour, the initial scanning continues and every time an already labelled pixel is revisited, all foreground pixels surrounding it are assigned the same label.

3.3.3.4 Ellipse Fitting

Ellipse fitting is the process of approximating an ellipse model to image data, reducing and simplifying the available data into fewer variables. Halíř and Flusser [86] propose a non-iterative algorithm, used in this project, for fitting an ellipse to a set of data points. The authors start by analysing a direct least squares based ellipse-specific method by Fitzgibbon et al [87], proposing

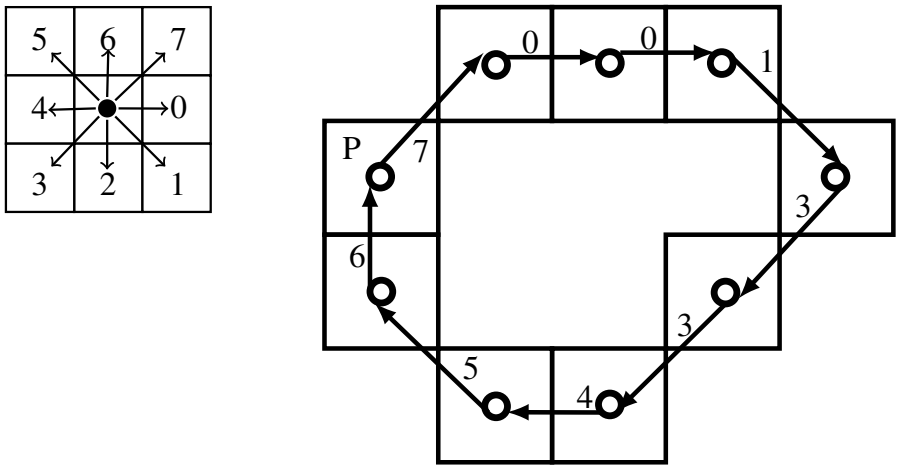


Figure 3.12: Contour-tracing procedure with connectivity 8, starting in P. Contour sequence: 7,0,0,1,3,3,4,5,6

a modified algorithm with the aim of improving numerical stability. The result is an ellipse described using six coefficients (Equation 3.12), where $a\ b\ c\ d\ e\ f$ are coefficients of the ellipse and (x,y) are coordinates of points lying on it.

$$F(x,y) = ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \ \ and \ \ b^2 - 4ac < 0 \tag{3.12}$$

3.3.4 Feature Extraction

In this project three different feature extraction procedures were studied. In the first one (**Method 1**), from the determined ellipse in the segmentation step six features were extracted: the ellipse centre coordinates (x_0, y_0), the semi-major and semi-minor axes (a and b respectively), the ellipse area and its orientation in radians (ϕ).

$$\frac{((x-x_0)\cos(\phi)+(y-y_0)\sin(\phi))^2}{a^2} + \frac{((x-x_0)\sin(\phi)-(y-y_0)\cos(\phi))^2}{b^2} = 1 \tag{3.13}$$

Figure 3.13: Ellipse equation

In this equation, (x_0, y_0) is the ellipse centre, a and b the semi-major and the semi-minor axes and ϕ the counter clockwise angle of rotation from x-axis to the major axis $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$

For the two other procedures (**Method 2 and 3**), the Features from Accelerated Segment Test (FAST) algorithm is used for detecting interest points in the original, non-segmented tube image. In the second method, the position of all detected points is recorded and used for classification, and in the third method the centre of gravity is used to compute the centre position of all points, recording this value and using it for the classification step. This method is better explained in the section 3.3.4.3 of this report. An explanation for use of the FAST algorithm can be found in the following sections with a brief description of the FAST algorithm process for detecting image corners.

The concept behind methods of local image feature detection and description is to first try to identify interesting image points and then for each point extract invariant descriptors, enabling image matching between similar images. This allows to build systems that can efficiently track position, scale or perspective changes between images of the same scene, or find images that are from similar contexts (e.g. pictures of cars).

In this project, the pictures in study are images of gel card tubes. Between them, the only significant change is the position or scale of the tested blood sample, i.e. the blood sample vertical position and its spread across the tube. Using feature descriptors for finding matches between images would not yield good test classification results. Instead, only the initial feature detection is used to find interesting image points and then their position is used for classification.

3.3.4.1 OpenCV Algorithms Benchmarks

The OpenCV software library implements multiple interest point algorithms. A simple routine was implemented that read an existing tube image, applied a bilateral filter followed by an unsharp masking technique, and then detected interest points using the seven different algorithms. The SIFT and SURF algorithms were not studied in this project since both are patented in the US patent office [88, 89].

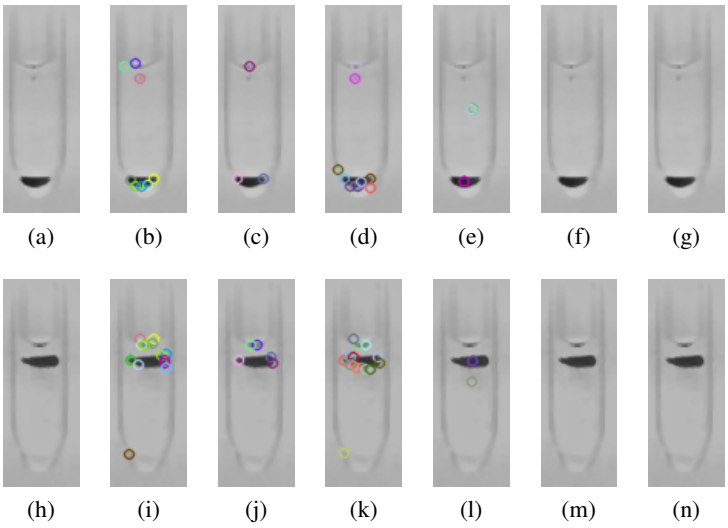


Figure 3.14: Benchmark of feature detection algorithms
(**a** and **h**) BRISK algorithm [90]; (**b** and **i**) FAST algorithm [64, 91]; (**c** and **j**) Harris detector; (**d** and **k**) Modified Harris detector by Shi and Tomasi [92]; (**e** and **l**) Maximally Stable Extremal Regions (MSER) extraction algorithm [93]; (**f** and **m**) Oriented BRIEF (ORB) keypoint detector [94]; (**g** and **n**) Star feature detector. [95]

From the seven tested algorithms, only three consistently detected more than three interest points (Figure 3.14): the FAST algorithm, the Harris detector and the modified Harris detector proposed by Shi and Tomasi. While all three detected relevant interest points, both the FAST and the Shi-Tomasi detectors yielded equivalent results.

When Rosten and Drummond [64] proposed the FAST feature detector, the authors compared its execution time with the SUSAN detector and the Harris detector which was optimized to use Streaming SIMD Extensions vectorising instructions. The FAST detector revealed to be approximately 4.5 times faster than the SUSAN detector and 16.9 times faster than the Harris detector. To confirm this claims a small test was conducted in the development platform, using the three previously identified algorithms (FAST, Harries and Shi-Tomasi). The execution time of each algorithm was recorded over 10000 iterations and the mean execution time was computed (Table 3.2). From the drawn results, the FAST feature detector was chosen for the detection step, since it proved to be most time efficient while detecting relevant image features around the blood test sample.

Detector	FAST	Harris	Shi-Tomasi
Time (microseconds)	18.49 μs	251.58 μs	261.27 μs

Table 3.2: Algorithms execution time

3.3.4.2 FAST Feature Detector

The Features from Accelerated Segment Test (FAST) is a feature detector algorithm proposed by Rosten and Drummond [64, 91] for real-time corner detection. In the FAST algorithm, to determine if a pixel candidate p with an intensity I_p is a corner, a circle of sixteen pixels (a Bresenham circle of radius 3) is considered around the candidate (Figure 3.15).

The pixel p is a corner if there exists a set of $n = 12$ contiguous pixels in the circle which are all brighter that $I_p + t$ or all darker than $I_p - t$, being t a threshold. To speed up the process on false candidates, the algorithm first tests, in pairs, if the four pixels corresponding to the cardinal points (at 1, 5, 9 and 13) are brighter than $I_p + t$ or darker than $I_p - t$, first 1 and 9 and then 5 and 13. For p to be a corner then at least three of these must all be brighter than $I_p + t$ or darker than $I_p - t$. The complete segment test criterion can then be applied to the remaining candidates by examining all pixels in the circle.

Despite its speed, the authors identify four main weaknesses in the detector:

1. The high-speed test does not generalise well for n < 12;
2. The choice and ordering of the fast test pixels contains implicit assumptions about the distribution of feature appearance;
3. Knowledge from the first 4 tests is discarded;
4. Multiple features are detected adjacent to one another.

To address them, the authors propose a machine learning approach for the first three, and the use of non-maximal suppression for the last one.

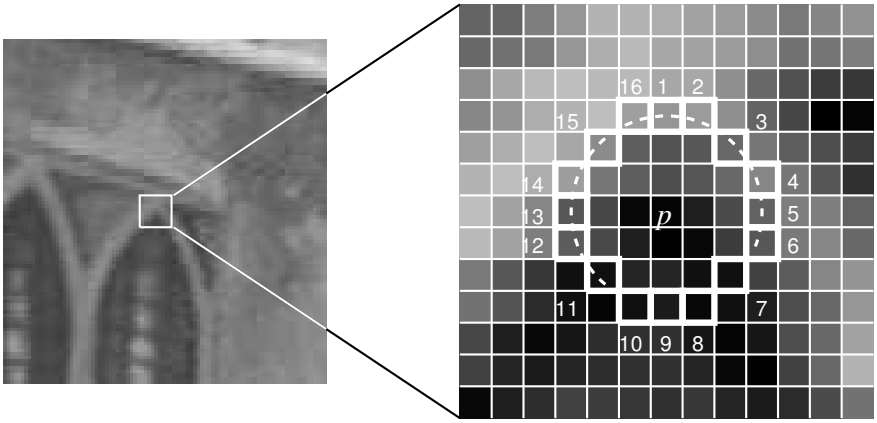


Figure 3.15: FAST Feature detection in an image patch
The highlighted squares are the pixels used in the detection (Bresenham circle of radius 3) [64, 91].

The machine learning process operates in two stages, first by detecting corners from a set of images (preferably from the target application domain) with a slow algorithm which tests all 16 locations around each pixel. Each of the all 16 surrounding pixels $x \in \{1..16\}$ can have one of the three states:

$$S_{p\to x} = \begin{cases} d & I_{p\to x} \leq I_p - t \quad \text{(darker)} \\ s & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b & I_p + t \leq I_{p\to x} \quad \text{(brighter)} \end{cases} \tag{3.14}$$

Each x pixel is then assigned into one of three P subsets, P_d, P_s, P_b, according to its state. In the second stage the same algorithm used in ID3 begins by selecting the x which yields the most information about whether the candidate pixel is a corner measured by the entropy of K_p, being K_p a boolean variable which is true if p is a corner and false otherwise. This process is recursively applied to all the P subsets until its entropy is zero, resulting then in a decision tree which can correctly classify all corners seen in the training set and approximate the rules of the chosen FAST corner detector.

To solve the problem of multiple features detected adjacent to one another, the authors suggest a score function V that is computed for each detected feature point, and non-maximal suppression is applied to remove corners which have an adjacent corner with higher V. The score function V is given by the sum of the absolute difference between the pixels in the contiguous arc and the centre pixel.

$$V = \max\left(\sum_{x\in S_{bright}} |I_{p\to x} - I_p| - t, \sum_{x\in S_{dark}} |I_p - I_{p\to x}| - t\right) \tag{3.15}$$

3.3.4.3 Centre of Gravity Method

The centre of gravity method is an algorithm used to determine the optimal centre location of a group points that minimizes a cost function. This method is usually applied to problems of distribution costs minimization, and is derived from the expression of the centre of mass of a particle system.

Being N the number of detected points, P_x the x coordinate of the detected point and P_y the y coordinate of the detected point, the optimal centre location (C_x, C_y) is given by

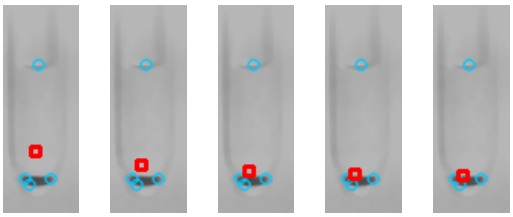
$$C_x = \frac{\sum_{i=1}^{N} P_{i_x} W_i}{\sum_{i=1}^{N} W_i} \tag{3.16}$$

$$C_y = \frac{\sum_{i=1}^{N} P_{i_y} W_i}{\sum_{i=1}^{N} W_i} \tag{3.17}$$

where W_i is the weight associated with the point i, given by

$$W_i = \frac{1}{|P_{i_x} - C_x| + |P_{i_y} - C_y|}. \tag{3.18}$$

In this project, the weight parameter associated with each point is given by its distance to current centre position. This calculus is iterated until the optimal centre coordinates do not change between two consecutive iterations.



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

Figure 3.16: Centre of gravity method over five iterations.

3.3.5 Classification

As mentioned in the section 2.3 of this report, the classification step serves as a prediction process for determining the outcome the ABO test result. Before doing so, first a classification model as to be built so it could be used to classify future gel card tests.

Using the WEKA software workbench, a routine was built to train multiple supervised classification algorithms present in the library. Three different datasets were built, each using one of the previously developed feature extraction processes: ellipse features, position of detected interest points and centre position of detected interest points.

To each dataset was then applied the process detailed in the Figure 3.17:

1. The dataset is split into a training set and a test set using stratified sampling, $\frac{1}{3}$ of the dataset used for testing and the remaining $\frac{2}{3}$ for training. Stratified sampling ensures that original class proportion is maintained in the two new sets;

2. Using the training set, the algorithms are trained with leave-one-out cross validation;

3. After the cross validation process, the average accuracy of each model is computed, determining the three models with better average accuracy;

4. The three resulting models are then tested with the initial test set and the results are recorded for study;

5. The process is repeated using 10-fold cross validation.

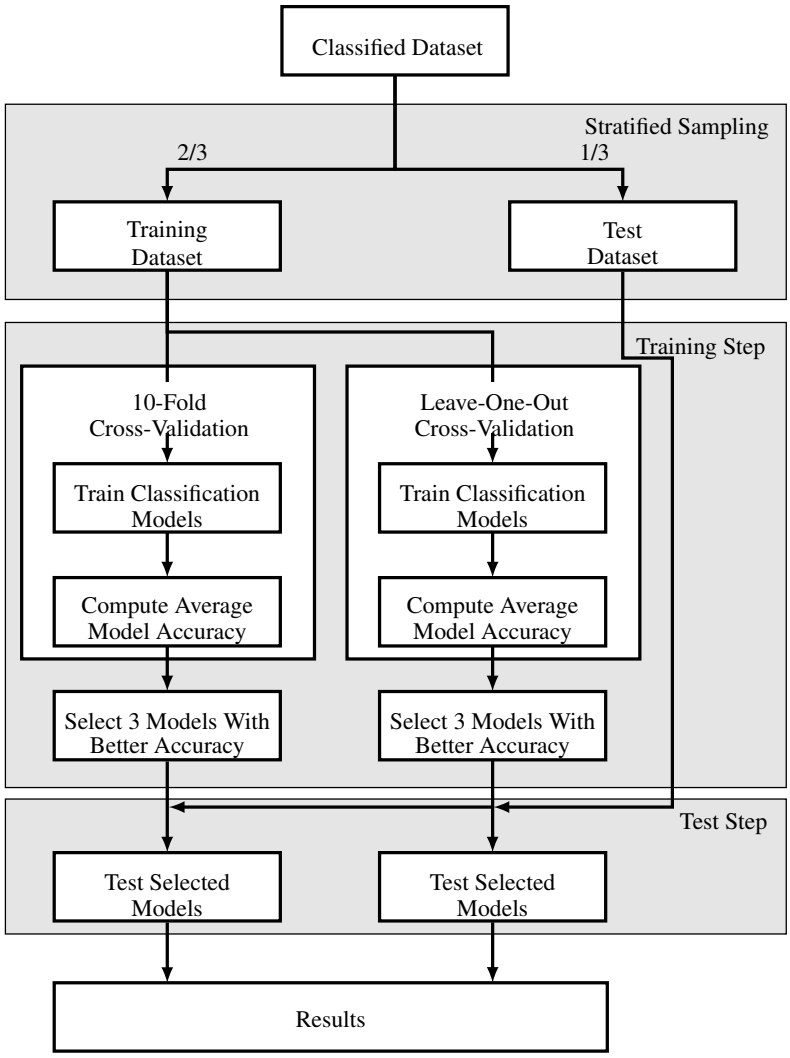


Figure 3.17: Data classification process

The WEKA library divides classification algorithms into eight groups: Bayes, Functions, Trees, Lazy, Rules, Meta, Multi-instance and Miscellaneous. In this project, fifteen of the available algorithms were tested using default parameters:

- **J48** - WEKA implementation of the C4.5 pruned decision tree classifier by Quinlan [96], extended from the ID3 algorithm;
- **Random Forest** - Algorithm introduced by Breiman [97], and defined by the author as a combination of tree predictors in a way that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest;
- **Random Tree** - Builds a unpruned decision tree considering a set of randomly selected K attributes at each tree node;
- **REP Tree** - Builds a pruned decision tree using reduced-error pruning (with backfitting);
- **Logistic** - WEKA process that builds a multinomial logistic regression model with a ridge estimator. The implementation of the algorithm in the WEKA library differs from the proposed method by Le Cessie and Van Houwelingen [98], so the algorithm can handle instance weights;
- **Simple Logistic** - Builds a linear logistic regression model using the boosting algorithm LogitBoost, with simple regression functions as base learners, for fitting the model;
- **Multilayer Perceptron** - Feedforward artificial neural network classifier that uses backpropagation, where all non-numeric nodes in the network are all sigmoid. The network can be monitored and modified during training time;
- **SMO** - Implementation of the Sequential Minimal Optimization (SMO) algorithm by Platt [99] for training a support vector classifier;
- **IBk** - K-NN classifier. The implementation is capable to selecting the appropriate value of k through cross-validation;
- **KStar** - Instance-based learner that uses entropy as a distance measure by Cleary and Trigg [100];
- **Decision Table** - WEKA interface for building a simple decision table majority classifier [101];
- **PART** - Builds a decision list using separate-and-conquer. In each iteration, a partial C4.5 decision tree is built and the best leaf becomes a rule [102];
- **JRip** - Propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), proposed by Cohen [103];
- **OneR** - Implementation of the 1R classifier. First a rule is generated for each dataset attribute, and then the rule with the smallest total error is defined as the classification rule;

3.3.5.1 Cross Validation

Cross validation is a model evaluation method to measure its predictive performance, where the dataset is initially split into k stratified subsets, maintaining class proportion. Then the model is trained using one of the k subsets as a test set and the remaining sets as the training set. This is repeated k times, choosing each time a different subset as the test set. The average error across all k trials is computed, giving a more accurate prediction of the model performance.

While a big number of folds (k) improves the accuracy of the computed average error, the computation time also increases. The number of ten folds ($k = 10$) is commonly used across the literature and it is used also in this project.

Leave-one-out cross validation is a particular case of k-fold cross validation where the number of folds is equal to the number of instances in the dataset. Each dataset instance is then individually used as the test set, using the rest of the dataset for training. This yields the best approximation of the model accuracy, but the computation time is increased to its maximum.

In this project leave-one-out cross validation was also used, since the size of the dataset is relatively small, not having a big impact the computation time.

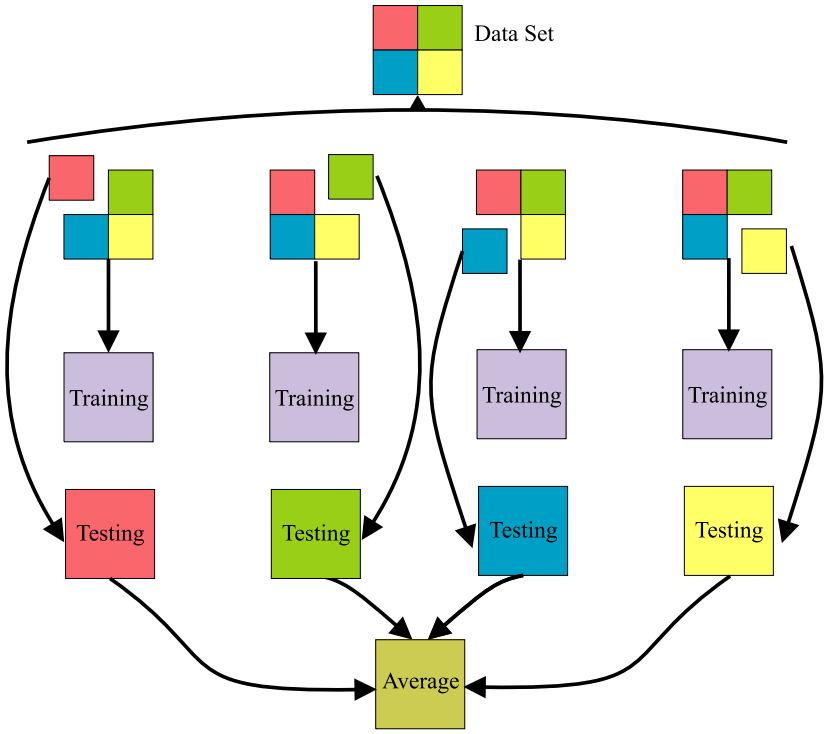


Figure 3.18: Cross validation process [4].

3.3.5.2 Evaluation

For model evaluation a confusion matrix can be used to display prediction accuracy using the number of True Positives (TP), False Negatives (FN), False Positives (FP) and True Negatives (TN) classified instances. The confusion matrix enables a simple visualization of performance metrics of a classification algorithm.

	PREDICTED CLASS			
ACTUAL CLASS		YES	NO	
	YES	TP	FN	TOTAL ACTUAL YES (M1)
	NO	FP	TN	TOTAL ACTUAL NO (M0)
		TOTAL PREDICTED YES (N1)	TOTAL PREDICTED NO (N0)	

Table 3.3: Confusion matrix.

From the confusion matrix, using the equation 3.19 the model accuracy can be determined by dividing the number of correct predictions with the total number of predictions, effectively representing the percentage of correct predictions.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.19}$$

The kappa coefficient is a useful statistic to determine the precision of a model. Vieira and Garrett [5] define the kappa coefficient as an indicative "measure of the magnitude of agreement between observers". The coefficient can also be determined using the confusion matrix, and is given by

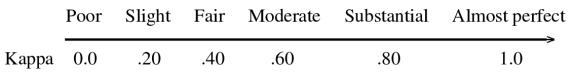
$$\text{Kappa} = \frac{P_o - P_e}{1 - P_e}, \tag{3.20}$$

where P_o is the previously determined accuracy (Equation 3.19), and P_e is

$$P_e = \left(\frac{N_1}{N} * \frac{M_1}{N}\right) + \left(\frac{N_0}{N} * \frac{M_0}{N}\right) \tag{3.21}$$

The authors also provide a interpretation table for the value of kappa, where a kappa of 0.0 indicates that the is a low chance of agreement between observers (in this case the model and the true values), and a kappa of 1.0 a almost perfect agreement.

Precision, recall and F-measure are three cost-sensitive measures that can be computed for each class as a performance measurement of the model. In this context, while precision for a specific class represents percentage of correct predictions for said class, recall is the percentage of correctly predicted class instances. Precision is determined dividing the number of instances



Kappa	Agreement
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21– 0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement

Figure 3.19: Interpretation of Kappa [5].

correctly predicted for a class by the total instances classified as the class (Equation 3.22). Recall is determined dividing the number of instances correctly predicted for a class by the actual number of instances in that class (Equation 3.23).

$$\text{Precision} = \frac{tp}{tp + fp} \tag{3.22}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{3.23}$$

F-measure (also F_1 score) is a score ranging from 0 to 1, given by weighted average between precision and recall of a class, having its worst value at 0 and best at 1.

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.24}$$

In this project, the classification problem in study is not binomial, as seen in the Table 3.3. Instead it is a multinomial classification problem, with five distinct classes (+, ++, +++, ++++ and -). The confusion matrix can still be used to display classification results for each class, having each column representing the number of instances in a predicted class, and each row the actual number of instances of the class.

The model accuracy can still be determined by dividing the number of correct predictions with the total number of predictions, and is given by:

$$\text{Accuracy} = \frac{AA + BB + CC + DD + EE}{TOTAL}. \tag{3.25}$$

		PREDICTED CLASS					
		A +	B ++	C +++	D ++++	E -	
ACTUAL CLASS	A +	AA	AB	AC	AD	AE	TOTAL PREDICTED + (A)
	B ++	BA	BB	BC	BD	BE	TOTAL PREDICTED ++ (B)
	C +++	CA	CB	CC	CD	CE	TOTAL PREDICTED +++ (C)
	D ++++	DA	DB	DC	DD	DE	TOTAL PREDICTED ++++ (D)
	E -	EA	EB	EC	ED	EE	TOTAL PREDICTED - (E)
		TOTAL PREDICTED + (A)	TOTAL PREDICTED ++ (B)	TOTAL PREDICTED +++ (C)	TOTAL PREDICTED ++++ (D)	TOTAL PREDICTED - (E)	TOTAL

Table 3.4: Confusion matrix for multinomial problem.

3.4 Software Application

The software application was developed in Java using the JavaFX^{TM}library for the user interface. For storing program data, the H2 Database Engine was used to save extracted image features and gel card information (Figure 3.20) while tube and gel card images were saved to the file system. In the feature extraction and classification process four additional libraries were used, the ZXing library for detecting and reading the gel card identification barcode, the BoofCV library for image segmentation and feature extraction, the OpenCV library for image feature extraction and the WEKA library for the data classification problem.

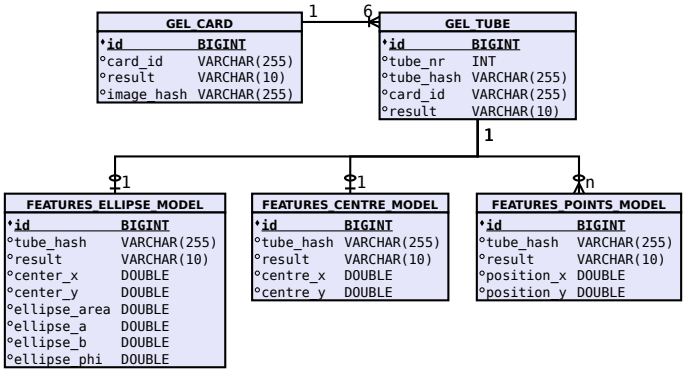


Figure 3.20: Database design.

Two different user interfaces were developed in this project. In the first one the user can easily import gel card images and use them for building the different datasets and train classification models. In the other interface the user can read and classify new gel cards, storing the test result in the database.

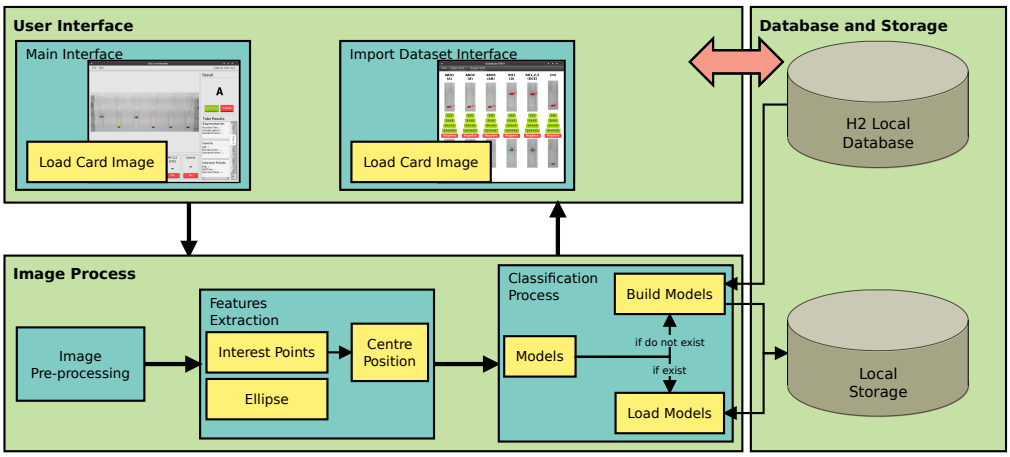


Figure 3.21: Application architecture.

In the first application interface (Figure 3.22), the user is firstly prompted to refer to a system folder that has the gel card images to be interpreted. One by one, the card images are loaded and image features are extracted from each tube, displaying the results in the interface for confirmation. Then, user input is required for classifying each tube test result as one of five possible classes: negative **(-)**, positive **(++++)**, positive **(+++)**, positive **(++)**, positive **(+)**.

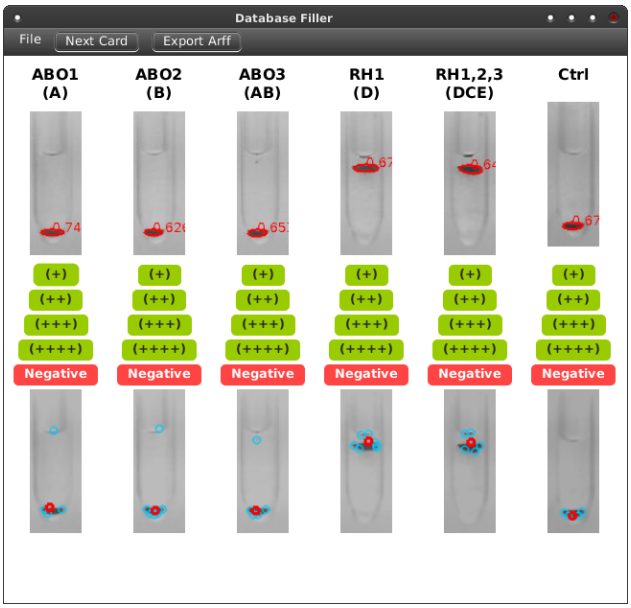


Figure 3.22: Application interface for building training datasets and classification models.

After classifying all tubes, to save the information the user has to request the next card present in the folder. If there is no card images left to process a warning is displayed. In the interface the user also has the possibility to export the current datasets into Attribute-Relation File Format (ARFF), a file format developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato that describes a list of instances sharing a set of attributes.

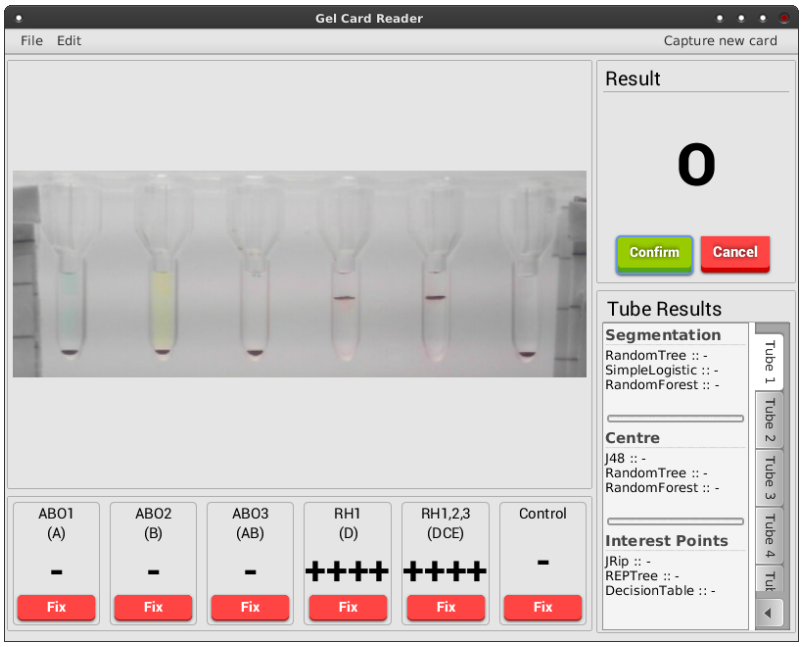


Figure 3.23: Main application interface.

The second interface (Figure 3.23) is the initial screen of the application. From here the user can either analyse a new unclassified card, or open the previous interface for importing data. Opening a new card image, the card is processed and classification results are displayed on the interface. For each tube, the obtained result is displayed at the bottom of the screen with a chance of fixing a classification error. If a test result is corrected by the user, after submitting the new class value, it is added to the training dataset and the classifications models are rebuilt.

In a lateral panel, the card ABO test result is displayed with additional classification information beneath in a tabbed panel. Here, the user can either confirm the test result, recording the values in the database, or dismiss it by pressing the cancel button. In the classification information panel, yielded classification results from the three different models are displayed for each tube.

In the processing phase, each tube outcome is determined using results from the three classification models previously built. Joining the three decisions, the result (out of three) which occurs more times is selected as the tube test result. In a draw condition, the result of the model with better training accuracy is chosen.

Chapter 4

Results

This chapter presents the results obtained in the feature extraction step and in the classification step for the three different feature extraction processes: ellipse features, position of detected interest points and centre position of detected interest points. In each process, the presented results were obtained when evaluating the three best algorithms with the test subset. For all randomization procedures, the chosen random seed was the natural number 1.

Each dataset was built using images of 144 gel tubes, existing 76 **(-)** tubes, 56 **(++++)** tubes, 4 **(+++)** tubes, 4 **(++)** tubes and 4 **(+)** tubes. All gel tubes images of class **(+++)**, **(++)** and **(+)** were digitally created from a sample card.

4.1 Extracted Features Datasets

Three different datasets were built using the extracted image features in the three different methods.

In the first method, six numeric ellipse features were extracted from 144 tubes: the ellipse centre coordinates (centre_x and centre_y), the ellipse semi-major and semi-minor axes (ellipse_a and ellipse_b), the ellipse area (ellipse_area) and its orientation in radians (ellipse_phi). The result was a dataset with 144 instances, with the following class distribution: 76 **(-)** instances, 56 **(++++)** instances, 4 **(+++)** instances, 4 **(++)** instances and 4 **(+)** instances.

Attribute	Minimum	Maximum	Average
centre_x	16.564	33.655	24.301±3.057
centre_y	41.522	118.621	85.999±32.229
ellipse_area	43.022	730.013	148.922±153.818
ellipse_a	6.265	24.531	11.459±3.405
ellipse_b	1.429	11.755	3.745±2.219
ellipse_phi	-1.554	1.563	−0.003±0.455

Table 4.1: Segmentation dataset statistics.

From the extracted ellipse features two interesting patterns arise. In the Figure 4.2, the five different classes are easily separated by the vertical position of ellipse centre. This is an expected

outcome since the blood sample position after the centrifugation step is one of the two main factors for test interpretation, as seen in the Figure 3.2. Still in the first class distribution graph, an outlier exists belonging to the class label **(++)**. Due to improper segmentation in the gel tube, the ellipse fitting process detected an ellipse only on the top of the gel tube (Figure 4.1).

Figure 4.1: Tube image of the detected outlier.

Five different clusters exist in the Figure 4.3, each one corresponding to one of the five classes, having big intervals dividing the classes. The vertical centre position of the ellipse divides the strong positive class and negative class, analogous of what happens in the physical test tube. The other dividing factor is the ellipse semi-major axis length, where strong test results have a smaller value than weak positives. In this figure the dataset unbalance is highly noticeable.

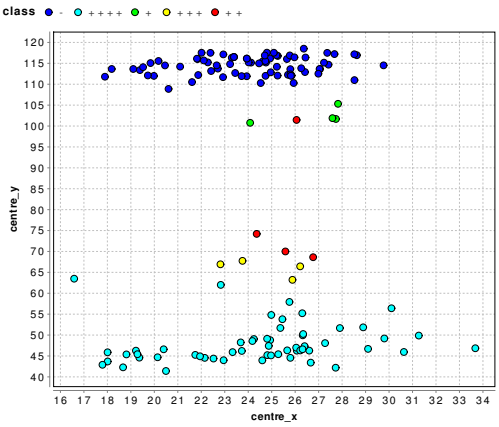


Figure 4.2: Class distribution: horizontal centre position of the extracted ellipse over its vertical centre position;

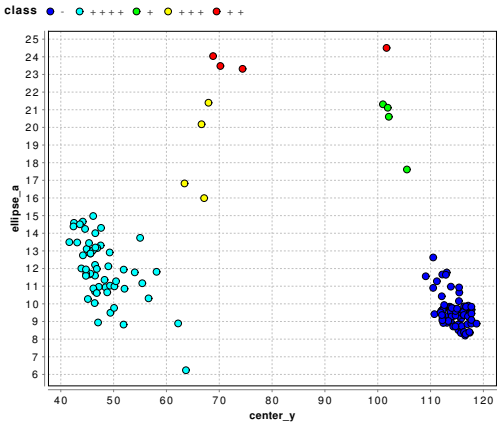


Figure 4.3: Class distribution: vertical centre position of the extracted ellipse over its semi-major axis.

In the second method, the dataset was built using the centre position computed from detected interest points (Section 3.3.4.3) corresponding to 144 instances, each with two numeric attributes: the coordinates of the computed centre (centre_x and centre_y).

In this dataset, the averages of the computed centre positions approximates the averages centre values obtained in the ellipse method, although the range of the vertical centre position was bigger in this method. While the centre averages were similar, the classes are harder to divide in this case.

Attribute	Minimum	Maximum	Average
centre_x	11.680	33.000	24.178±4.077
centre_y	19.990	121.440	85.596±35.298

Table 4.2: Centre of detected interest points dataset statistics.

Still, most of the strong positive class and the negative class instances collect one at the bottom and the other at the top of the graph (Figure 4.4), respectively.

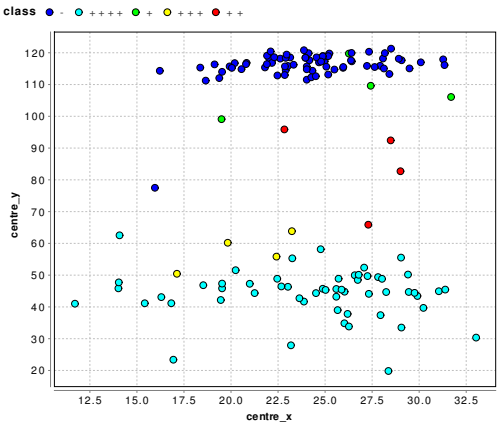


Figure 4.4: Class distribution: horizontal centre position over the vertical centre position.

In the third method, instead of determining the centre position of the detected interest points, the dataset was built using the position of all detected interest points, classifying each point according to the test value present in the image it was detected. From the tube images 829 interest points were detected, with the following class distribution: 400 **(-)** instances, 280 **(++++)** instances, 29 **(+++)** instances, 82 **(++)** instances and 38 **(+)** instances.

Attribute	Minimum	Maximum	Average
position_x	5.000	42.000	24.597±7.692
position_y	3.000	123.000	85.273±34.837

Table 4.3: Position of detected interest points dataset statistics.

Again, the dataset unbalance is noticeable in the Figure 4.5, existing more instances for high and low values of the vertical position. Since the coordinate (0,0) corresponds to the top left pixel of the original tube image, is verified the expected outcome of having most of the negative instances at the bottom of the tube, and most of the strong positive instances at the top of the tube. Some outliers exist in the dataset due to improper detection of residual image reflections and tube margins as image interest points.

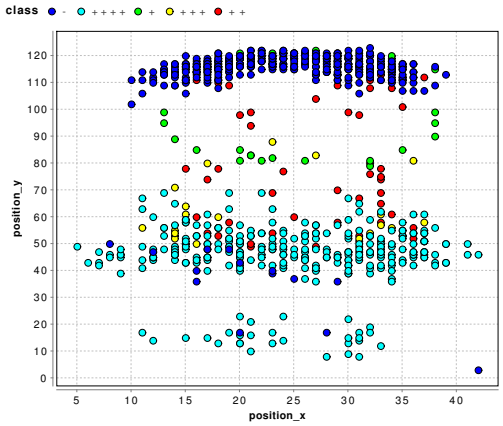


Figure 4.5: Class distribution: horizontal centre position over the vertical centre position.

4.2 Classification Results for Ellipse Features

In this first process, ellipse features extracted from 144 tubes were used to build the classification models. From each tube were extracted six numeric features: the ellipse centre coordinates (x_0, y_0), the ellipse semi-major and semi-minor axes, the ellipse area and its orientation in radians. The result was a dataset with 144 instances, keeping 48 separate for testing. The obtained results are present in the Tables 4.4 and 4.5.

From the classification process, the three algorithms that yielded better accuracy after leave-one-out cross validation were the Simple Logistic algorithm, the Random Forest algorithm and the Random Tree algorithm.

	Simple Logistic					Random Forest					Random Tree				
Correctly Classified Instances	47					48					47				
Incorrectly Classified Instances	1					0					1				
Accuracy	0.9792					1.0					0.9792				
Kappa statistic	0.9634					1.0					0.9634				
Mean absolute error	0.009					0.0225					0.0083				
Class	+	++	+++	++++	-	+	++	+++	++++	-	+	++	+++	++++	-
Precision	1	0.5	1	1	1	1	1	1	1	1	1	1	0.5	1	1
Recall	0.5	1	1	1	1	1	1	1	1	1	0.5	1	1	1	1
F-Measure	0.667	0.667	1	1	1	1	1	1	1	1	0.667	1	0.667	1	1
ROC Area	1	1	1	1	1	1	1	1	1	1	0.75	1	0.989	1	1
TP Rate	0.5	1	1	1	1	1	1	1	1	1	0.5	1	1	1	1
FP Rate	0	0.021	0	0	0	0	0	0	0	0	0	0	0.021	0	0

Table 4.4: Ellipse features classification results for leave-one-out cross validation.

The impact of the small size of the dataset can be seen in the Table 4.4, where the Random Forest algorithm built a perfect model, predicting correctly all instances of the test set. The other two algorithms, Simple Logistic and Random Tree, also approached a perfect model, each only misclassifying one instance.

Analysing the Receiver Operating Characteristic (ROC) area, all three algorithms approach the optimal value of 1 (100%), with the exception of two classes of the Random Tree algorithm. This means that the Random Forest algorithm and the Simple Logistic algorithm both correctly classified 100% of randomly drawn instances for test.

Comparing the Simple Logistic and the Random Tree algorithms, both correctly predicted all strong positive **(++++)** and negative **(-)** instances, the two majority classes, but both struggled with at least two of the minority classes.

To speed up the process, 10-fold cross validation was used to train again the classification models. The yielded result statistics approximated the ones previously determined, having model obtained using the Random Forest algorithm maintained a perfect score of the ROC area for all classes, despite lowering its accuracy. The accuracy of the model built by the Random Tree algorithm decreased, as well as the ROC area for the **(++)** class.

	Simple Logistic				
Correctly Classified Instances	47				
Incorrectly Classified Instances	1				
Accuracy	0.9792				
Kappa statistic	0.9634				
Mean absolute error	0.0128				
Class	+	++	+++	++++	-
Precision	1	1	0.5	1	1
Recall	0.5	1	1	1	1
F-Measure	0.667	1	0.667	1	1
ROC Area	0.967	1	1	1	1
TP Rate	0.5	1	1	1	1
FP Rate	0	0	0.021	0	0

	Random Forest				
Correctly Classified Instances	47				
Incorrectly Classified Instances	1				
Accuracy	0.9792				
Kappa statistic	0.9634				
Mean absolute error	0.015				
Class	+	++	+++	++++	-
Precision	1	1	0.5	1	1
Recall	0.5	1	1	1	1
F-Measure	0.667	1	0.667	1	1
ROC Area	1	1	1	1	1
TP Rate	0.5	1	1	1	1
FP Rate	0	0	0.021	0	0

	Random Tree				
Correctly Classified Instances	45				
Incorrectly Classified Instances	3				
Accuracy	0.9375				
Kappa statistic	0.8908				
Mean absolute error	0.025				
Class	+	++	+++	++++	-
Precision	1	0	0.5	0.95	1
Recall	0.5	0	1	1	0.96
F-Measure	0.667	0	0.667	0.974	0.98
ROC Area	0.75	0.489	0.989	0.983	0.98
TP Rate	0.5	0	1	1	0.96
FP Rate	0	0.021	0.021	0.034	0

Table 4.5: Ellipse features classification results for 10-fold cross validation.

4.3 Classification Results for Centre Position of Interest Points

In the second process, the features dataset was built using the interest points detected in each tube image, computing a global centre position as described in the Section 3.3.4.3 of this report.

The expected outcome would be the detection by the FAST algorithm of interest points around the blood sample present in the gel tube, giving an approximate position of its centre. Due to the detection of image reflections and tube margins as interest points, the centre of gravity method was applied to reduce the influence of these outliers.

As seen previously in the Figure 3.16, iteratively the method approximates the global centre to the neighbourhood of the clustered points. The result was a dataset with 144 instances, one for each tube, with only two attributes: the centre x position and the centre y position. Of these 144 instances, 48 were kept separate for testing, yielding the results present in the Tables 4.6 and 4.7.

	Random Tree					J48					Random Forest				
Correctly Classified Instances	40					45					41				
Incorrectly Classified Instances	8					3					7				
Accuracy	0.8333					0.9375					0.8542				
Kappa statistic	0.7191					0.8868					0.75				
Mean absolute error	0.0667					0.0383					0.055				
Class	+	++	+++	++++	-	+	++	+++	++++	-	+	++	+++	++++	-
Precision	0	0.5	0	0.938	0.96	0	0.5	0	0.95	0.962	0	0.5	0	0.938	0.962
Recall	0	1	0	0.789	0.96	0	1	0	1	1	0	1	0	0.789	1
F-Measure	0	0.667	0	0.857	0.96	0	0.667	0	0.974	0.98	0	0.667	0	0.857	0.98
ROC Area	0.489	0.989	0.457	0.877	0.958	0.848	0.989	0.798	0.983	0.978	0.478	1	0.915	0.993	0.977
TP Rate	0	1	0	0.789	0.96	0	1	0	1	1	0	1	0	0.789	1
FP Rate	0.022	0.021	0.085	0.034	0.043	0	0.021	0	0.034	0.043	0	0.021	0.085	0.034	0.043

Table 4.6: Centre position classification results for leave-one-out cross validation.

In this process the three algorithms that produced a more accurate model after leave-one-out cross validation were the Random Tree algorithm, the J48 algorithm (pruned decision tree classifier) and the Random Forest algorithm. Despite having the same number of instances as the ellipse features dataset, the obtained models yielded worst results. The effects of the small dataset are also visible in Table 4.6, having low precision values for the three minority classes.

Using 10-fold cross validation, the trained model by the Random Tree algorithm maintained its accuracy, correctly classifying 40 instances out of 48. While the Random Tree algorithm kept its accuracy, the J48 algorithm and the Random Forest algorithm both achieved worst results.

	Random Tree					J48					Logistic				
Correctly Classified Instances	40					41					46				
Incorrectly Classified Instances	8					7					2				
Accuracy	0.8333					0.8542					0.9583				
Kappa statistic	0.7141					0.75					0.9242				
Mean absolute error	0.0667					0.062					0.0368				
Class	+	++	+++	++++	-	+	++	+++	++++	-	+	++	+++	++++	-
Precision	0	1	0	0.938	0.923	0	0.5	0	0.938	0.962	0	1	1	1	0.926
Recall	0	1	0	0.789	0.96	0	1	0	0.789	1	0	1	1	1	1
F-Measure	0	1	0	0.857	0.941	0	0.667	0	0.857	0.98	0	1	1	1	0.962
ROC Area	0.489	1	0.457	0.877	0.937	0.848	0.989	0.915	0.891	0.978	0.859	1	1	1	0.977
TP Rate	0	1	0	0.789	0.96	0	1	0	0.789	1	0	1	1	1	1
FP Rate	0.022	0	0.085	0.034	0.087	0	0.021	0.085	0.034	0.043	0	0	0	0	0.087

Table 4.7: Centre position classification results for 10-fold cross validation.

4.4 Classification Results for Position of Interest Points

For the third process, using the same detected interest points, the classification models were trained with each individual point position. Since the number of interest points detected by the FAST algorithm varied with each tube image, the resulting dataset was constituted by a bigger number of instances.

Using the FAST algorithm for corner detection, from the original 144 tube images were extracted 829 interest points, corresponding to a dataset of 829 instances, with 553 instances for

training and for 276 for testing. Of the 829 original instances 38 were extracted from tubes classified as **(+)**, 82 from **(++)** tubes, 29 from **(+++)**, 280 from **(++++)** and 400 from **(-)**.

	Decision Table					JRip					REP Tree				
Correctly Classified Instances	228					229					231				
Incorrectly Classified Instances	48					47					45				
Accuracy	0.8261					0.8297					0.8369				
Kappa statistic	0.7123					0.7258					0.7322				
Mean absolute error	0.134					0.1086					0.0991				
Class	+	++	+++	++++	-	+	++	+++	++++	-	+	++	+++	++++	-
Precision	0.417	1	0.6	0.823	0.869	0.8	0.462	0.5	0.845	0.899	0.571	0.6	0	0.809	0.899
Recall	0.417	0.036	0.333	0.989	0.947	0.333	0.429	0.222	0.926	0.932	0.333	0.321	0	0.989	0.94
F-Measure	0.417	0.069	0.429	0.899	0.906	0.471	0.444	0.308	0.883	0.915	0.421	0.419	0	0.89	0.919
ROC Area	0.865	0.765	0.933	0.97	0.95	0.793	0.695	0.835	0.946	0.927	0.854	0.835	0.942	0.965	0.951
TP Rate	0.417	0.036	0.333	0.989	0.947	0.333	0.429	0.222	0.926	0.932	0.333	0.321	0	0.989	0.94
FP Rate	0.027	0	0.007	0.11	0.133	0.004	0.056	0.007	0.088	0.098	0.011	0.024	0	0.121	0.098

Table 4.8: Results of position of interest points used for classification.

After leave-one-out cross validation, the algorithms that produced a better model accuracy were the Decision Table algorithm, the JRip algorithm (Repeated Incremental Pruning to Produce Error Reduction) and the REP Tree algorithm (decision tree pruned using reduced-error). The increased number of instances produced results statistically more significant. First, all three algorithms produced similar high accuracy (over 80 %) and Kappa statistic values. Despite the increase in the number of instances, the dataset is still highly imbalanced, due to existing more instances for the strong positive **(++++)** and negative **(-)** classes. This leads to worst results on all models for the three minority classes.

	PART					Decision Table					JRip				
Correctly Classified Instances	234					228					233				
Incorrectly Classified Instances	42					48					43				
Accuracy	0.8478					0.8261					0.8442				
Kappa statistic	0.7504					0.7123					0.7478				
Mean absolute error	0.0875					0.1365					0.1012				
Class	+	++	+++	++++	-	+	++	+++	++++	-	+	++	+++	++++	-
Precision	0.75	0.563	1	0.825	0.899	0.417	1	0.6	0.823	0.869	0.625	0.526	0.5	0.858	0.899
Recall	0.25	0.321	0.333	1	0.94	0.417	0.036	0.333	0.989	0.947	0.417	0.357	0.222	0.968	0.94
F-Measure	0.375	0.409	0.5	0.904	0.919	0.417	0.069	0.429	0.899	0.906	0.5	0.426	0.308	0.91	0.919
ROC Area	0.722	0.856	0.94	0.972	0.966	0.823	0.758	0.93	0.967	0.946	0.845	0.7	0.854	0.956	0.93
TP Rate	0.25	0.321	0.333	1	0.94	0.417	0.036	0.333	0.989	0.947	0.417	0.357	0.222	0.968	0.94
FP Rate	0.004	0.028	0	0.11	0.098	0.027	0	0.007	0.11	0.133	0.011	0.036	0.007	0.082	0.098

Table 4.9: Results of position of interest points used for classification.

Similar accuracy results were yielded using 10-fold cross validation to train the models, but now the PART algorithm produced a classification model with better results, replacing the REP Tree algorithm in the top tree. The PART algorithm is described by its authors as a "method for learning decision lists based on the repeated generation of partial decision trees in a separate-and-conquer manner" [102].

4.5 Application Tests

After the classification step, the application described in Section 3.4 of this report was then used to analyse new and unclassified gel cards, employing the three best models trained in each process with leave-one-out cross validation, i. e. ellipse features models, interest points models and centre of interest points models. To determine the test outcome, for each tube, the application joins the three classification decisions yielded by the three models, and chooses the result with the highest occurrence rate (out of three). If all three models disagree, the classification result of the model with better training accuracy is chosen.

The application was tested with the two gel cards initially separated from the group before feature extraction and model training (Figure 4.6), correctly determining the test result for all gel card tubes.

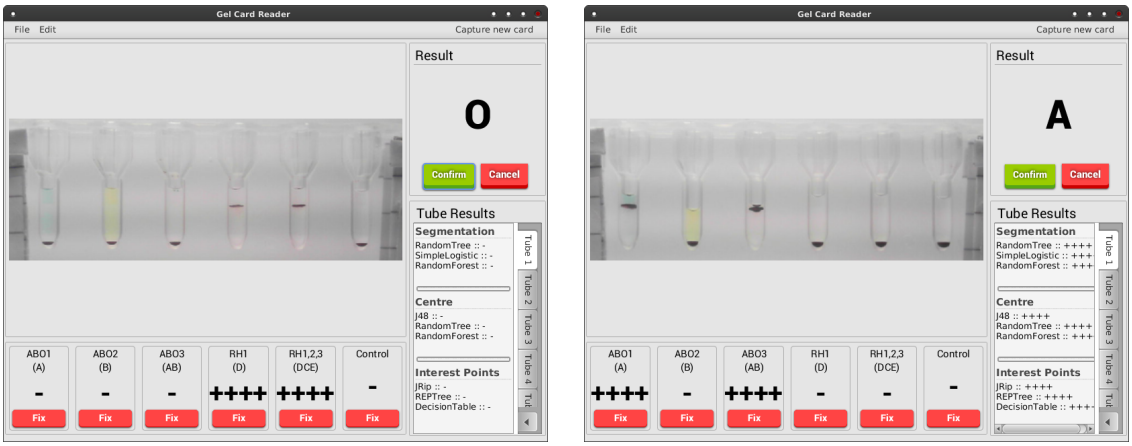


Figure 4.6: Main application interface.

4.6 Conclusions

Comparing the three different processes, it is evident the impact a small and unbalanced training dataset can have in building classification models. While trained models yielded high accuracy results for the two majority classes (**-** and **++++**), the accuracy for the other three minority classes was lower, with ROC areas lower than 0.5. This means that in those instances the models yielded worst results than a pure random decision.

From the three different processes, models built using ellipse features yielded more accurate classifiers. A possible explanation is the similarity between the fitted ellipse characteristics and blood sample for determining the blood test result, since the two main factors for determining the blood test result in the gel tube is the sample vertical position in the tube and its size, i. e. the blood sample spread across the tube. This two attributes can be correlated to the ellipse vertical centre position and the ellipse semi-major axis, who as previously verified in the Figure 4.3, easily divide the five different class labels.

Chapter 5

Conclusions and Future Work

In this dissertation was explored the applicability of machine learning and computer vision techniques for reading and classifying gel card tests used for blood typing. Using open-source libraries, two computer applications were developed, one for reading images of gel cards, extracting image features using three different methods. To each method were applied multiple classification algorithms using leave-one-out cross validation and 10-fold cross validation, and the results were studied for three of the best models.

The second application was developed to use the trained models for classification of new cards, recording each new result and rebuilding the model each time the user fixes a classification mistake. This program also extracts features using the same three different methods, joining yielded results from the three classification models from each method to classify extracted features, and combining again the three tests methods to determine the final tube result.

The software was developed with Java using the JavaFX^{TM} library for the user interface, the BoofCV library for image segmentation and feature extraction, the WEKA software workbench for the data classification problem, the ZXing library for detecting and reading the gel card identification barcode, the H2 database as a data storage for program and classifier data and the OpenCV library with desktop Java bindings for local image feature extraction.

From the three different feature extraction processes, one used the BoofCV library for first segmenting the tube image, isolating the blood sample in the tube, and then fitting an ellipse into the segmented blood shape. From this ellipse were extracted six different features and saved in a dataset: the ellipse centre coordinates, the ellipse semi-major and semi-minor axes, the ellipse area and the ellipse orientation in radians.

The other two feature extraction processes used the OpenCV library for extracting interest points using the FAST corner detection algorithm. For the second dataset each detected interest point position on the tube image was recorded. The third dataset was built with the centre coordinates computed using all the interest points detected in the tube image, as described in the Section 3.3.4.3 of this report.

Due to the small card sample, 24 gel cards, and class imbalance, 76 **(-)** tubes, 56 **(++++)** tubes, 4 **(+++)** tubes, 4 **(++)** tubes and 4 **(+)** tubes, the classification models built in the Chapter 4 yielded

poor statistical results for minority classes. In the third dataset, were verified good results for the two majority classes, indicating that the interest points position detected by the FAST algorithm would successfully represent the information present in the test tube, although features extracted from ellipse fitting would produce better results.

In future work, different dataset sampling process should be studied to try to oppose dataset imbalance, like randomly discard majority class instances, duplicate minority class instances, or both. Also, a bigger and balanced card sample should be used to verify classification accuracy of interest points position as the tube descriptor. Finally, different feature extraction processes should also be tested, to verify if points detected by the FAST algorithm correctly approximate image interest points without compromising detection accuracy.

References

[1] Jiawei Han and Micheline Kamber. *Data Mining: Concepts ans Techniques*. 2001.

[2] Bio-Rad. ScanGel^{TM} Monoclonal ABO/RH - 86496 86426, 2008.

[3] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Unsharp Filter, 2003. URL: http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm.

[4] M Ruiz, L E Mujica, X Berjaga, and J Rodellar. Partial least square/projection to latent structures (PLS) regression to estimate impact localization in structures. *Smart Materials and Structures*, 22(2):025028, February 2013. URL: http://stacks.iop.org/0964-1726/22/i=2/a=025028?key=crossref.0de6e92907954b7faeef3e56759d5368, doi:10.1088/0964-1726/22/2/025028.

[5] Anthony J Viera and Joanne M Garrett. Understanding interobserver agreement: the kappa statistic. *Family medicine*, 37(5):360–3, May 2005. URL: http://www.ncbi.nlm.nih.gov/pubmed/15883903.

[6] ISBT International Society of Blood Transfusion. Table of blood group systems.

[7] ISBT International Society of Blood Transfusion. Blood Group Terminology. URL: http://www.isbtweb.org/working-parties/red-cell-immunogenetics-and-blood-group-terminology/blood-group-terminology/.

[8] P Monod, A Fasel, and P Schwind. CAPILLARY CENTRIFUGATION FOR BLOOD GROUP SEROLOGY DIAGNOSTICS WITH DISTINCT AREAS FOR POSITIVE AND NEGATIVE REACTIONS. *medion-diagnostics.ch*. URL: http://www.medion-diagnostics.ch/images/Capillary_centrifugation_for_blood_group_serology_diagnostics_with_distinct_areas_for_positive_and_negative_reactions.pdf.

[9] Immuno-haematology Reader. Immuno-Haematology Reader.

[10] Ana Ferraz, Filomena Soares, and Vitor Carvalho. A Prototype for Blood Typing Based on Image Processing. *SENSORDEVICES 2013, The Fourth International Conference on Sensor Device Technologies and Applications*, (c):139–144, 2013. URL: http://www.thinkmind.org/index.php?view=article&articleid=sensordevices_2013_10_30_20140.

[11] Mohammad Al-Tamimi, Wei Shen, Rania Zeineddine, Huy Tran, and Gil Garnier. Validation of paper-based assay for rapid blood typing. *Analytical chemistry*, 84(3):1661–8, February 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/22148716, doi:10.1021/ac202948t.

[12] Oivind Due Trier, Anil K. Jain, and Torfinn Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[13] H D Cheng, X H Jiang, Y Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(September):2259–2281, 2001.

[14] Nikhil R. Pal and Sankar K. Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. (cited in Cheng, 2001).

[15] F Kurugollu, B Sankur, and a.E Harmanci. Color image segmentation using histogram multithresholding and fusion. *Image and Vision Computing*, 19(13):915–928, November 2001. URL: http://linkinghub.elsevier.com/retrieve/pii/S026288560100052X, doi:10.1016/S0262-8856(01)00052-X.

[16] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1)(January):146–165, January 2004. URL: http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.1631316, doi:10.1117/1.1631316.

[17] Azriel Rosenfeld and Pilar De La Torre. Histogram concavity analysis as an aid in threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2):231–235, March 1983. (cited in Sezgin, 2004). URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6313118, doi:10.1109/TSMC.1983.6313118.

[18] M. Ibrahim Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Computer Vision, Graphics, and Image Processing*, 49(1):36–51, January 1990. (cited in Sezgin, 2004). URL: http://www.sciencedirect.com/science/article/pii/0734189X9090161N, doi:10.1016/0734-189X(90)90161-N.

[19] Mark J. Carlotto. Histogram Analysis Using a Scale-Space Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):121–129, January 1987. (cited in Sezgin, 2004). URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767877, doi:10.1109/TPAMI.1987.4767877.

[20] N. Ramesh. Thresholding based on histogram approximation. *IEE Proceedings - Vision, Image, and Signal Processing*, 142(5):271, October 1995. (cited in Sezgin, 2004). URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=487786, doi:10.1049/ip-vis:19952007.

[21] T. W. Ridler and S. Calvard. Picture Thresholding Using an Iterative Selection Method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8):630–632, 1978. (cited in Sezgin, 2004). URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4310039, doi:10.1109/TSMC.1978.4310039.

[22] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE TRANSACTIONS ON SYSTREMS, MAN, AND CYBERNETICS*, SMC-9(1):62–66, 1979.

[23] C.V Jawahar, P.K Biswas, and A.K Ray. Investigations on fuzzy thresholding based on fuzzy clustering. *Pattern Recognition*, 30(10):1605–1613, October 1997. URL: http://www.sciencedirect.com/science/article/pii/S0031320397000046, doi:10.1016/S0031-3203(97)00004-6.

[24] J.N. Kapur, P.K. Sahoo, and Andrew K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, March 1985. (cited in Sezgin, 2004). URL: http://www.sciencedirect.com/science/article/pii/0734189X85901252, doi:10.1016/0734-189X(85)90125-2.

[25] C.H. Li and C.K. Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617–625, April 1993. (cited in Sezgin, 2004). URL: http://www.sciencedirect.com/science/article/pii/003132039390115D, doi:10.1016/0031-3203(93)90115-D.

[26] A.G. Shanbhag. Utilization of Information Measure as a Means of Image Thresholding. *CVGIP: Graphical Models and Image Processing*, 56(5):414–419, September 1994. (cited in Sezgin, 2004). URL: http://www.sciencedirect.com/science/article/pii/S1049965284710376, doi:10.1006/cgip.1994.1037.

[27] W H Tsai. Moment-preserving thresholding: a new approach. *Graphical Models and Image Processing*, 19:377 – 393, 1985. (cited in Sezgin, 2004).

[28] Lois Hertz and Ronald W. Schafer. Multilevel thresholding using edge matching. *Computer Vision, Graphics, and Image Processing*, 44(3):279–295, December 1988. (cited in Sezgin, 2004). URL: http://www.sciencedirect.com/science/article/pii/0734189X88901259, doi:10.1016/0734-189X(88)90125-9.

[29] Sankar K Pal and Azriel Rosenfeld. Image enhancement and thresholding by optimization of fuzzy compactness. *Pattern Recognition Letters*, 7(2):77–86, February 1988. (cited in Sezgin, 2004). URL: http://linkinghub.elsevier.com/retrieve/pii/0167865588901225, doi:10.1016/0167-8655(88)90122-5.

[30] J. Bernsen. Dynamic Thresholding of Grey-Level Images. *Proc. of the 8th Int. Conf. on Pattern Recognition*, 1986. (cited in Sezgin, 2004).

[31] W. Niblack. An introduction to digital image processing. *Prentice-Hall, Englewood Cliffs, NJ*, pages 115–116, 1986. (cited in Sezgin, 2004).

[32] J. Sauvola and M. Pietaksinen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000. (cited in Sezgin, 2004).

[33] Aly A. Farag and Edward J. Delp. Edge linking by sequential search. *Pattern Recognition*, 28(5):611–633, 1995.

[34] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, May 2011. URL: http://www.ncbi.nlm.nih.gov/pubmed/20733228, doi:10.1109/TPAMI.2010.161.

[35] D Marr and E Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, 207(1167):187–217, 1980. URL: http://rspb.royalsocietypublishing.org/content/207/1167/187.short.

[36] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767851, doi:10.1109/TPAMI.1986.4767851.

[37] Djemel Ziou and Salvatore Tabbone. Edge Detection Techniques - An Overview. *PATTERN RECOGNITION AND IMAGE ANALYSIS C/C OF RASPOZNAVANIYE OBRAZOV I ANALIZ IZOBRAZHENII*, 8:537–559, 1998. URL: http://nyx-www.informatik.uni-bremen.de/1044/1/ziou_pria_98.pdf.

[38] R. M. Haralick. Ridges and Valley on Digital Images. *Computer Vision, Graphics, and Image Processing1*, (22):28–38, 1983. (cited in Ziou, 1998).

[39] G. Giraudon. Edge Detection from Local Negative Maximum of Second Derivative. *Proceed- ings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 643–645, 1985. (cited in Ziou, 1998).

[40] K. Rangarajan, M. Shah, and D. van Brackle. Optimal Corner Detector. In *[1988 Proceedings] Second International Conference on Computer Vision*, pages 90–94. IEEE. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=589975, doi:10.1109/CCV.1988.589975.

[41] Rimon Elias and Robert Laganière. JUDOCA: junction detection operator based on circumferential anchors. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 21(4):2109–18, April 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/22086500, doi:10.1109/TIP.2011.2175738.

[42] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004. URL: http://link.springer.com/10.1023/B:VISI.0000022288.19776.77, doi:10.1023/B:VISI.0000022288.19776.77.

[43] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1000236, doi:10.1109/34.1000236.

[44] J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=868688, doi:10.1109/34.868688.

[45] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43(1):7–27, June 2001. URL: http://link.springer.com/article/10.1023/A%3A1011174803800, doi:10.1023/A:1011174803800.

[46] James A. Sethian Stanley Osher. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.1266.

[47] Daniel Cremers, Stanley J. Osher, and Stefano Soatto. Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation. *International Journal of Computer Vision*, 69(3):335–351, May 2006. URL: http://link.springer.com/10.1007/s11263-006-7533-5, doi:10.1007/s11263-006-7533-5.

[48] Yonggang Shi and William Clem Karl. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 17(5):645–56, May

2008. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2970527&tool=pmcentrez&rendertype=abstract, doi:10.1109/TIP.2008.920737.

[49] T F Chan and L A Vese. Active contours without edges. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 10(2):266–77, January 2001. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=902291, doi:10.1109/83.902291.

[50] Elsa Angelini, Yinpeng Jin, and Andrew Laine. State of the art of level set methods in segmentation and registration of medical imaging modalities. *Handbook of Biomedical Image Analysis*, 2005. URL: http://link.springer.com/chapter/10.1007/0-306-48608-3_2.

[51] Xin Jiang and Shengdong Nie. Urine Sediment Image Segmentation Based on Level Set and Mumford-Shah Model. *2007 1st International Conference on Bioinformatics and Biomedical Engineering*, pages 1028–1030, 2007. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4272750, doi:10.1109/ICBBE.2007.266.

[52] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, pages 50–54 vol.2. IEEE, 1996. (cited in Tuytelaars, 2007). URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=546722, doi:10.1109/ICPR.1996.546722.

[53] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997. (cited in Tuytelaars, 2007). URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=589215, doi:10.1109/34.589215.

[54] Robert M. Haralick. A Measure for Circularity of Digital Figures. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-4(4):394–396, July 1974. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5408463, doi:10.1109/TSMC.1974.5408463.

[55] Gianfranco Doretto and Yi Yao. Region moments: Fast invariant descriptors for detecting small image structures. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3019–3026, June 2010. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5540052, doi:10.1109/CVPR.2010.5540052.

[56] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962. (cited in Doretto, 2010). doi:10.1109/TIT.1962.1057692.

[57] Charles T. Zahn and Ralph Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, c-21(3):269–281, 1972. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5008949.

[58] E Persoon and K S Fu. Shape discrimination using fourier descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 8(3):388–97, March 1986. URL: http://www.ncbi.nlm.nih.gov/pubmed/21869355.

[59] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. *Computer Vision–ECCV 2006*, 2006. URL: http://link.springer.com/chapter/10.1007/11744047_45.

[60] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. URL: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94, doi:10.1023/B:VISI.0000029664.99615.94.

[61] Krystian Mikolajczyk and Cordelia Schmid. Performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–30, October 2005. URL: http://www.ncbi.nlm.nih.gov/pubmed/16237996, doi:10.1109/TPAMI.2005.188.

[62] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, (20):511–517, 2004.

[63] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer vision and image ...*, (September), 2008. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001555.

[64] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 1508–1515 Vol. 2, 2005. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1544896, doi:10.1109/ICCV.2005.104.

[65] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2008. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4674368, arXiv:arXiv:0810.2434v1.

[66] Nancy M Salem and Asoke K Nandi. Segmentation of retinal blood vessels using scale-space features and K-nearest neighbour classifier. *Acoustics, Speech and Signal ...*, pages 1001–1004, 2006. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1660514.

[67] MR Islam, M Chowdhury, and S Khan. Medical image classification using an efficient data mining technique. *Complexity International*, 12:1–9, 2004. URL: http://www.complexity.org.au/ci/vol12/msid25/file.pdf.

[68] P Rajendran and M Madheswaran. Hybrid medical image classification using association rule mining with decision tree algorithm. *Journal of Computing*, 2(1):127–136, 2010. URL: http://arxiv.org/abs/1001.3503.

[69] M. Deepa and N. Mymoon Zuviriya. Comparative analysis on supervised classification techniques for segmentation and detecting abnormal blood vessels in retinal images. *2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET)*, pages 180–185, December 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6513902, doi:10.1109/INCOSET.2012.6513902.

[70] Drew Schmitt and Nicolas McCoy. Object Classification and Localization Using SURF Descriptors. pages 1–5, 2011. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.375.1456&rep=rep1&type=pdf.

[71] ML Antonie, OR Zaiane, and Alexandru Coman. Application of Data Mining Techniques for Medical Image Classification. *Proceedings of the Seconds International Workshop on Multimedia Data Mining*, pages 94–101, 2001. URL: http://webdocs.cs.ualberta.ca/~zaiane/mdm_kdd2001/papers/mdm01.pdf.gz#page=95.

[72] Magudeeswaran Veluchamy, Karthikeyan Perumal, and Thirumurugan Ponuchamy. Feature extraction and classification of blood cells using artificial neural network. *American Journal of Applied Sciences*, 9(5):615–619, 2012. URL: http://www.thescipub.com/pdf/10.3844/ajassp.2012.615.619.pdf.

[73] Chris Watkins Jason Weston. Multi-class Support Vector Machines. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9594.

[74] M. Pontil and A. Verri. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, June 1998. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=683777, doi:10.1109/34.683777.

[75] H. Bülthoff C. Burges V. Vapnik T. Vetter V. Blanz, B. Schölkopf. Comparison of view-based object recognition algorithms using realistic 3D models. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.9383.

[76] Subrajeet Mohapatra, Sushanta Shekhar Samanta, Dipti Patra, and Sanghamitra Satpathi. Fuzzy Based Blood Image Segmentation for Automated Leukemia Detection. *2011 International Conference on Devices and Communications (ICDeCom)*, pages 1–5, February 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5738491, doi:10.1109/ICDECOM.2011.5738491.

[77] Wei-Liang Tai, Rouh-Mei Hu, Han C.W. Hsiao, Rong-Ming Chen, and Jeffrey J.P. Tsai. Blood Cell Image Classification Based on Hierarchical SVM. *2011 IEEE International Symposium on Multimedia*, pages 129–136, December 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6123336, doi:10.1109/ISM.2011.29.

[78] Masateru Kane, Akiyoshi Nishimura, and Katsuji Nishi. Blood group typing by electrophoresis based on isoelectric focusing. *Analytica Chimica Acta*, 383(1-2):157–168, March 1999. URL: http://linkinghub.elsevier.com/retrieve/pii/S0003267098004966, doi:10.1016/S0003-2670(98)00496-6.

[79] T Kobayashi and a Akane. ABO genotyping by inverse PCR technique. *Legal medicine (Tokyo, Japan)*, 2(1):15–20, March 2000. URL: http://www.ncbi.nlm.nih.gov/pubmed/12935460.

[80] Yusuke Doi, Yuji Yamamoto, Sachiyo Inagaki, Yoshiaki Shigeta, Satoru Miyaishi, and Hideo Ishizu. A new method for ABO genotyping using a multiplex single-base primer extension reaction and its application to forensic casework samples. *Legal medicine (Tokyo, Japan)*, 6(4):213–23, October 2004. URL: http://www.ncbi.nlm.nih.gov/pubmed/15363446, doi:10.1016/j.legalmed.2004.05.005.

[81] Soumya K Srivastava, Andreas Artemiou, and Adrienne R Minerick. Direct current insulator-based dielectrophoretic characterization of erythrocytes: ABO-Rh human blood typing. *Electrophoresis*, 32(18):2530–40, September 2011. URL: http://www.ncbi.nlm.nih.gov/pubmed/21922495, doi:10.1002/elps.201100089.

[82] Nongluck Houngkamhang, Apirom Vongsakulyanon, Patjaree Peungthum, Krisda Sudprasert, Pimpun Kitpoka, Mongkol Kunakorn, Boonsong Sutapun, Ratthasart Amarit, Armote Somboonkaew, and Toemsak Srikhirin. ABO blood-typing using an antibody array technique based on surface plasmon resonance imaging. *Sensors (Basel, Switzerland)*, 13(9):11913–22, January 2013. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3821317&tool=pmcentrez&rendertype=abstract, doi:10.3390/s130911913.

[83] Y Lapierre, D Rigal, J Adam, D Josef, F Meyer, S Greber, and C Drot. The gel test: a new way to detect red cell antigen-antibody reactions. *Transfusion*, 30(2):109–13, February 1990. URL: http://www.ncbi.nlm.nih.gov/pubmed/2305438.

[84] Richard Szeliski. *Computer vision: algorithms and applications*. September edition, 2010. URL: http://books.google.com/books?hl=en&lr=&id=bXzAlkODwa8C&oi=fnd&pg=PR9&dq=Computer+Vision:+Algorithms+and+Applications&ots=gZX5a4pxDC&sig=FJMV4wu-0pGqMlM3twk3vbVO-gA.

[85] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, February 2004. URL: http://linkinghub.elsevier.com/retrieve/pii/S1077314203001401, doi:10.1016/j.cviu.2003.09.002.

[86] Radim Halíř and Jan Flusser. Numerically stable direct least squares fitting of ellipses. *Proc. 6th International Conference in Central Europe ...*, 1998. URL: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Numerically+stable+direct+least+squares+fitting+of+ellipses#0.

[87] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=765658, doi:10.1109/34.765658.

[88] David G. Lowe. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image, 2000.

[89] R Funayama, H Yanagihara, L Van Gool, T Tuytelaars, and H Bay. Robust interest point detector and descriptor, 2012. URL: http://www.google.com/patents/US8165401.

[90] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, pages 2548–2555, November 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6126542, doi:10.1109/ICCV.2011.6126542.

[91] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006. URL: http://edwardrosten.com/work/rosten_2006_machine.pdf, doi:10.1007/11744023_34.

[92] J Shi and C Tomasi. Good features to track. *Proceedings CVPR'94., IEEE*, 1994. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=323794.

[93] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, September 2004. URL: http://linkinghub.elsevier.com/retrieve/pii/S0262885604000435http://www.sciencedirect.com/science/article/pii/S0262885604000435, doi:10.1016/j.imavis.2004.02.006.

[94] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, pages 2564–2571, November 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6126544, doi:10.1109/ICCV.2011.6126544.

[95] Motilal Agrawal, Kurt Konolige, and MR Blas. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. *Computer Vision–ECCV 2008*, pages 102–115, 2008. URL: http://link.springer.com/chapter/10.1007/978-3-540-88693-8_8.

[96] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[97] L Breiman. Random forests. *Machine learning*, pages 1–33, 2001. URL: http://link.springer.com/article/10.1023/A:1010933404324.

[98] Saskia Le Cessie and JC Van Houwelingen. Ridge estimators in logistic regression. *Applied statistics*, pages 191–201, 1992.

[99] JC Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, 1999. URL: http://dl.acm.org/citation.cfm?id=299105.

[100] JG Cleary and LE Trigg. K*: An Instance-based Learner Using an Entropic Distance Measure. *ICML*, 5:1–14, 1995. URL: https://perun.pmf.uns.ac.rs/radovanovic/dmsem/cd/install/Weka/doc/classifiers-papers/lazy/KStar/Cleary95-KStar.pdf.

[101] R Kohavi. The power of decision tables. *Machine Learning: ECML-95*, 1995. URL: http://link.springer.com/chapter/10.1007/3-540-59286-5_57.

[102] Eibe Frank and Ian H Witten. Generating accurate rule sets without global optimization. 1998.

[103] William W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.