

Nuno Filipe Girão Santos Almeida

# Implementation and Optimization of Photocoagulation Retinal Treatment Detection Algorithms

Dissertation presented to the University of Coimbra to obtain the Master's Degree in Electrical and Computer Engineering.  
Area of Telecommunications

May/2016



UNIVERSIDADE DE COIMBRA





**Faculty of Science and Technology**  
**University of Coimbra**

# **Implementation and Optimization of Photocoagulation Retinal Treatment Detection Algorithms**

Dissertation supervised by Professor Luís Alberto da Silva Cruz, of the Department of Electrical and Computer Engineering of the Faculty of Sciences and Technology of the University of Coimbra and Engineer Carlos Manta Oliveira, of Retmarker S.A., presented to the University of Coimbra to obtain the Master's Degree in Electrical and Computer Engineering.

Nuno Filipe Girão Santos Almeida

May 2016



Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são pertença do autor da mesma, e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This copy of the thesis is provided on the condition that anyone who consults it is understood to recognize that its copyright rests with its author and that no quotation from the document and no information derived from it may be published without proper acknowledgement.



*“I’m a great believer in luck and I find the harder I work, the more I have of it.”*

*Thomas Jefferson*





# Acknowledgements

University of Coimbra is a great institution to which I must express my gratitude. The knowledge acquired and friends, professors and all the people I have encountered and that made me the person I am today. I would like to express sincere appreciation to Professor Luís Alberto da Silva Cruz and Carlos Manta Oliveira, my thesis supervisors, for their full support, guidance and understanding throughout my research and development of this project. Through them, I had the opportunity to work with a marvelous team at *Retmarker S.A.*, namely João Diogo Ramos, Frederico Teles, Adriano Galante, João Pinto, Guilherme Nunes and other colleagues I met in *Critical Software S.A.*. This first contact with the industrial reality has given me a priceless insight about working in a real-world environment.

I am also very thankful for having the opportunity to work with such great colleagues from the Laboratory of Image Processing. Thank you Pedro Rocha, Nuno Carvalho, Ricardo Rocha, Rui Peliteiro and especially João Sousa, for the support, help, suggestions and the pleasure of working with you.

My academic journey was also made easier by the company and unconditional support or my friends, Cláudia Gaspar, Tiago Jesus, João Brito Dias, João Antunes, Tiago Cavaleiro, Marília Meco, Magda Santos, Maria João Gouveia, Carlos Ramos, Cátia da Silva, Pedro Duarte, André Leitão, Tiago Dias and lastly, my most important friend of all and better half, Inês Trindade, who stood by me in the most difficult part of this journey and of my life, believing in my work and supporting me each day.

Finally, I would like to acknowledge my family as the most important supporter of all. To my father José Augusto Almeida for his guidance and knowledge, to my mother Helena Girão for her kindness and unlimited love, to my brother João Pedro for his comprehension, my grandparents Amadeu and Dulcelina, José and Gilberta, all my cousins, my uncle Cavaco and aunt Manuela and a special thanks to my aunt Helena Almeida and uncle João Pessoa for the tireless devotion, despite the sleepless nights.



# Resumo

A Retinopatia Diabética é uma complicação da Diabetes Mellitus que pode causar perda de visão, podendo ser prevenida e tratada. O tratamento mais comum desta patologia é feito através de foto-coagulação laser destruindo na retina vasos sanguíneos frágeis, prevenindo também que outros se desenvolvam, retardando o avanço da doença.

A maneira mais eficaz de detetar esta doença é a realização de rastreios periódicos. Nestes, são frequentemente empregues métodos de deteção automática de lesões associadas à Retinopatia Diabética, como é o Retmarker. No entanto, estes sistemas podem apresentar baixo desempenho devido à presença de marcas laser, como as causadas por tratamentos da Retinopatia Diabética com recurso a foto-coagulação. Esta dissertação, proposta pela Retmarker S.A., apresenta melhoramentos de um método de deteção automática de marcas laser para que seja evitado o processamento desnecessário de imagens de pacientes já tratados e para prevenir a classificação incorreta de imagens contendo marcas laser.

Três algoritmos de segmentação do estado da arte foram otimizados para detetar regiões candidatas a marcas laser na imagem fonte. A partir destas e da restante imagem, várias características são calculadas de forma a treinar classificadores binários capazes de distinguir corretamente imagens contendo marcas laser.

Máquinas de Suporte Vetorial e dois classificadores baseados em árvores de decisão foram testados, uma vez que o melhor resultado obtido pelo algoritmo que se pretende otimizar é baseado num classificador em Árvore de Decisão C4.5. Seis variantes de treinos de classificação foram utilizadas e o seu desempenho foi avaliado em conjuntos de dados de teste. O classificador baseado em Máquinas de Suporte Vetorial foi aquele que obteve a melhor eficiência com 99,19% de Especificidade e 89,63% de Sensibilidade. Foi conseguida uma otimização do tempo de execução de aproximadamente 80%.

Os resultados são apresentados e discutidos e finalmente são feitas sugestões de continuação do presente trabalho.

**Palavras-Chave:** Retinopatia Diabética, Retinografia digital da retina, Processamento de Imagem, Deteção Automática, Classificação, Extração de características, Marcas Laser



# Abstract

Diabetic Retinopathy is acknowledged as a complication resulting from Diabetes Mellitus and a sight-threatening disease, that can be prevented and treated. The most common treatment for this condition is made by laser photocoagulation to destroy the fragile new vessels and prevent the growth of new abnormal ones.

The most effective way of detecting this disease is by regularly surveying populations through screening initiatives. In screening initiatives, automatic lesion detection methods are often employed to detect Diabetic Retinopathy-related lesions, as is the case of Retmarker. These systems can, however, show low performance when processing images presenting laser treatment scars such as the ones left by Diabetic Retinopathy treatments.

This thesis, which was proposed by Retmarker S.A, presents improvements to an existing algorithm used in the automatic detection of lasermarks to avoid the further processing of images belonging to already treated patients and also to prevent the misclassification of images presenting laser treatment scars.

Three state-of-the art segmentation algorithms were optimized in order to identify lasermark candidate regions in an input image. From these candidates and from the image itself, various features are computed in order to train binary classifiers capable of correctly distinguish images containing lasermarks.

Support Vector Machines and two Decision Tree based classifiers were tested, as the best result obtained by the previous algorithm was achieved using a C4.5 Decision Tree classifier. Six classification training variants were used and their performance was evaluated on test datasets. Support Vector Machines was the algorithm that achieved the best classification efficiency with 99.19% Specificity and 89.63% Sensibility. The time-efficiency improvements reached about 80%.

All the presented results are discussed and finally improvements to be made and future work are suggested.

**Key-words:** Diabetic Retinopathy, Digital Fundus Image, Image processing, Classification, Feature Extraction, Photocoagulation, Laser Marks



# Abbreviations

<b>50HP</b>	50 Healthy Patients Dataset
<b>ACC</b>	Accuracy
<b>ADT</b>	Alternating Decision Tree
<b>AIBILI</b>	Association for Innovation and Biomedical Research on Light
<b>ARS</b>	Administração Regional de Saúde do Centro
<b>ASIC</b>	Application Specific Integrated Circuits
<b>BAT</b>	Before and After Treatment Dataset
<b>BF</b>	Bilateral Filter
<b>CHT</b>	Circular Hough Transform
<b>CM1</b>	Color Map 1
<b>CM2</b>	Color Map 2
<b>CM3</b>	Color Map 3
<b>DT</b>	Decision Tree
<b>DFI</b>	Digital Fundus Image
<b>DME</b>	Diabetic Macular Edema
<b>DR</b>	Diabetic Retinopathy
<b>MA</b>	Micro-Aneurysm
<b>EX</b>	E-ophta EX Dataset
<b>FAZD</b>	Foveal Avascular Zone Detection Dataset

<b>F</b>	Incorrect Classification
<b>FN</b>	False Negatives
<b>FP</b>	False Positives
<b>FVF</b>	Frangi Vesselness Filter
<b>GPC</b>	Generic Parameter Classification
<b>GR</b>	Gain Ratio
<b>HRF</b>	High-Resolution Fundus
<b>IDF</b>	International Diabetes Federation
<b>IG</b>	Information Gain
<b>LMS</b>	Laser Mark Segmentation
<b>M</b>	Messidor Dataset
<b>MA</b>	E-ophtha MA Dataset
<b>MEP</b>	Macular Edema Photocoagulation
<b>NOEX</b>	E-ophtha NOEX Dataset
<b>NOMA</b>	E-ophtha NOMA Dataset
<b>NOMASCR</b>	The Training Dataset
<b>OD</b>	Optic Disc
<b>PDJD</b>	Proprietary Dataset João Dias
<b>PR</b>	Proliferative Retinopathy
<b>PRP</b>	Pan-Retinal Photocoagulation
<b>ROI</b>	Region of Interest
<b>SBF</b>	Shiftable Bilateral Filter
<b>SCR</b>	Proprietary Dataset Screening
<b>SEN</b>	Sensitivity



<b>SCR15</b>	Proprietary Dataset SCR15
<b>SM</b>	Single Measure
<b>SPC</b>	Structural Parameter Classification
<b>SPE</b>	Specificity
<b>SVM</b>	Support Vector Machine
<b>T</b>	Correct Classification
<b>TN</b>	True Negatives
<b>TP</b>	True Positives
<b>VBR</b>	Vessel-Based Registration Dataset
<b>WEKA</b>	Waikato Environment for Knowledge Analysis



# Contents

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Contextualization . . . . .	1
1.2 Document Overview . . . . .	4
<b>2 Considerations and State-of-the-art</b>	<b>5</b>
2.1 Diabetic Retinopathy Considerations and Treatment . . . . .	6
2.1.1 Photocoagulation Treatment . . . . .	7
2.2 Lasermarks detection state-of-the-art . . . . .	9
2.2.1 Detection of Laser Marks in Retinal Images . . . . .	9
2.2.2 Laser Marks Detection From Fundus Images . . . . .	10
2.2.3 Contributions to the Automatic Detection of Laser Marks in Retinal Digital Fundus Images . . . . .	11
2.3 Lasermark Segmentation Algorithms in Retinal Images . . . . .	12
2.3.1 Laser Mark Segmentation . . . . .	13
2.3.2 Frangi Vesselness Filter . . . . .	14
2.3.3 Circular Hough Transform . . . . .	15
2.4 Classification Algorithms . . . . .	17
2.4.1 C4.5 Decision Tree . . . . .	17
2.4.2 Alternating Decision Tree . . . . .	18
2.4.3 Support Vector Machines . . . . .	19
2.5 Test Platform . . . . .	20
2.6 Materials . . . . .	20
2.6.1 Public Datasets . . . . .	21
2.6.2 Proprietary Datasets . . . . .	22
2.6.3 Validation Datasets . . . . .	23

<b>3</b>	<b>Algorithm Optimizations</b>	<b>25</b>
3.1	Pre-processing Optimization . . . . .	27
3.2	Circular Hough Transform Optimization . . . . .	31
3.2.1	Guided Filter . . . . .	32
3.2.2	Shiftable Bilateral Filter . . . . .	33
3.2.3	LMS and FVF Optimization . . . . .	35
3.3	Code-based Optimizations . . . . .	37
3.4	Feature Selection and Binary Classification . . . . .	44
<b>4</b>	<b>Experimental Results</b>	<b>47</b>
4.1	Feature Selection Results . . . . .	47
4.2	Classification Results . . . . .	48
4.3	Optimization Results . . . . .	49
4.4	Discussion . . . . .	51
<b>5</b>	<b>Conclusion and Future Work</b>	<b>57</b>
	<b>Appendix A Datasets Examples</b>	<b>59</b>
	<b>Appendix B Initial Algorithm Considerations</b>	<b>73</b>
	<b>Appendix C Code Sections Efficiency</b>	<b>76</b>
	<b>Appendix D Feature Selection</b>	<b>80</b>
	<b>Appendix E Other Classification Results</b>	<b>83</b>
	<b>Appendix F Classification Models</b>	<b>87</b>

# List of Tables

2.1	Dias <i>et al.</i> laser mark detection algorithm performance. . . . .	10
2.2	Faraz Tahir <i>et al.</i> laser mark detection algorithm performance. . . . .	11
2.4	Test datasets used. . . . .	21
2.5	Validation datasets. . . . .	21
3.1	Algorithm’s functions and their operations, by computing order. . . . .	41
4.1	<b>GR_SVM</b> classifier tested on test datasets. . . . .	49
4.2	<b>GR_SVM</b> classifier tested on validation datasets. . . . .	49
4.3	Initial algorithm processing time for each test dataset . . . . .	50
4.4	Final algorithm processing time for each test dataset . . . . .	50
4.5	Efficiency improvements quantified. . . . .	50
A.1	DFIs not considered from the Messidor Dataset. . . . .	59
A.2	BAT Dataset discrimination. . . . .	68
B.1	Kernels and Constants used. . . . .	73
C.1	Initial algorithm classification with Sousa <i>et al.</i> DT classifier. . . . .	76
C.2	Algorithm classification with Sousa <i>et al.</i> DT classifier, after preprocessing modifications. . . . .	77
C.3	Algorithm classification with Sousa <i>et al.</i> DT classifier, after CHT modi- fications. . . . .	78
C.4	Algorithm classification with Sousa <i>et al.</i> DT classifier, after FVF and LMS modifications. . . . .	79
D.1	IG-filtered and GR-filtered feature subsets and their respective individual value. . . . .	82
E.1	<b>IG_DT</b> classifier tested on test datasets. . . . .	83
E.2	<b>GR_DT</b> classifier tested on test datasets. . . . .	84
E.3	<b>IG/GR_ADT</b> classifier tested on test datasets. . . . .	85

E.4	<b>IG_SVM</b> classifier tested on test datasets. . . . .	86
-----	---	----

# List of Figures

1.1	Worldwide prediction of Diabetes' prevalence and progression by IDF region. . . . .	1
2.1	Anatomy of the human eye's retina [8]. . . . .	6
2.2	Examples of Photocoagulation patterns. . . . .	7
2.3	LMS processing steps . . . . .	14
2.4	Patterns in 2D, depending on the value of the eigenvalues. . . . .	15
2.5	Example image of the detected structures using FVF. . . . .	15
2.6	CHT voting pattern: a) Detected edges, b) Circle contributions of each edge [37] and c) Accumulator array correspondent to a circle detection. The obtained peak is the center of the circle [34]. . . . .	16
2.7	Decision Tree classifier example. . . . .	17
2.8	Alternating Decision Tree classifier example (Fig. 1 of [43]). . . . .	19
2.9	Non Linear SVM Example. . . . .	20
2.10	RFI of a healthy patient (a) and of a patient with laser treatment marks(b). . . . .	21
3.1	Initial algorithm mean processing time, per section. . . . .	27
3.2	Cropping Example. . . . .	28
3.3	L* color component before(a) and after(b) uneven illumination correction and contrast enhancement. . . . .	30
3.4	Image after pre-processing step - RGB (a) and Green Channel (b). . . . .	31
3.5	Circles detected by CHT with Guided Filter. . . . .	32
3.6	Circles detected by CHT with (a) BF and (b) SBF. . . . .	33
3.7	Ratio(D/di) for all the circles detected in the dataset NOMASCR. . . . .	34
3.8	FVF and Jerman vesselness filters' obtained segmentation. . . . .	36
3.9	Matlab profiler's example report. . . . .	38
3.10	A Matlab profiler's report. . . . .	39
3.11	Variable growing inside a loop. . . . .	40
3.12	Variable pre-allocated and later truncated. . . . .	40

3.13	Vectorization example. . . . .	42
4.1	Best subsets obtained with the IG-filtered features. . . . .	47
4.2	Best subsets obtained with the GR-filtered features. . . . .	48
4.3	Misclassified images of HRF dataset . . . . .	53
4.4	The only False Positive obtained in the BAT dataset. . . . .	56
A.1	Messidor example images. . . . .	59
A.2	MA example images. . . . .	60
A.3	NOMA example images. . . . .	61
A.4	EX example images. . . . .	62
A.5	NOEX example images. . . . .	63
A.6	VBR example images. . . . .	64
A.7	50HP example images. . . . .	65
A.8	FAZD example images. . . . .	66
A.9	SCR example images. . . . .	67
A.10	BAT example images. . . . .	68
A.11	PDJD example images. . . . .	69
A.12	SCR15 example images. . . . .	70
A.13	CDB example images. . . . .	71
A.14	HRF example images. . . . .	72
B.1	Algorithm to be optimized. . . . .	74
B.2	Computed Features. . . . .	75
C.1	Initial algorithm mean processing time, per section. . . . .	76
C.2	Algorithm mean processing time, per section, after preprocessing improvements. . . . .	77
C.3	Algorithm mean processing time, per section, after CHT improvements. . . . .	78
C.4	Algorithm mean processing time, per section, after FVF and LMS improvements. . . . .	79
C.5	Algorithm mean processing time, per section, after code-based improvements. . . . .	79
F.1	Decision Tree Classifier trained using the features <b>IG_DT</b> . . . . .	87
F.2	Decision Tree Classifier trained using the features <b>GR_DT</b> . . . . .	88
F.3	Alternating Decision Tree Classifier trained using the features <b>IG_ADT/GR_ADT</b> . . . . .	89



# Chapter 1

## Introduction

### 1.1 Problem Contextualization

Diabetes Mellitus, or simply Diabetes, is a metabolic chronic disease that compromises the human body's absorption of sugar. This problem arises from the inability of pancreas to produce enough insulin or when the body does not respond properly to the produced insulin. Diabetes has a preoccupying incidence, with the International Diabetes Foundation/IDF, in 2013, estimating 382 million people living with diabetes (8.3% of the global scale adult population), and predicting an increase of 55% until the year 2035 [1]. In addition, IDF estimates that 46% of these cases are still to be diagnosed.

These numbers introduce diabetes as a severe global-scale disease that presents a subject's age as the main associated risk factor.

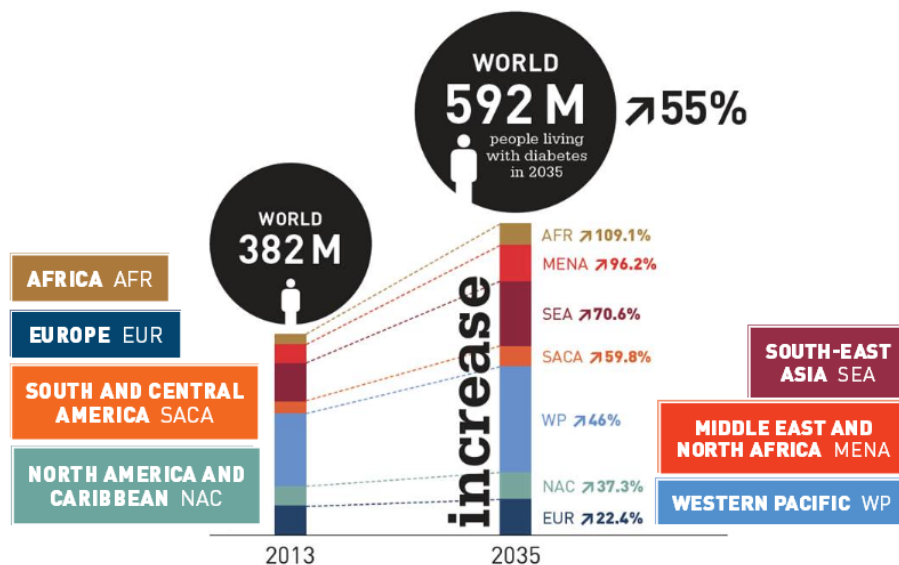


Figure 1.1: Worldwide prediction of Diabetes' prevalence and progression by IDF region. [1]

Diabetes presents a various number of complications, most of which relate to blood vessels deterioration. One of these complications is *Diabetic Retinopathy*, which affects the retina potentially causing vision loss. The high prevalence associated with Diabetic Retinopathy [2] indicates that a substantial part of the estimated worldwide diabetics suffer from this condition. In 2012, about 160 million people were estimated as having developed Diabetic Retinopathy(DR) [3].

In addition, DR is identified as one of the four leading causes of avoidable blindness by VISION 2020 [4]. DR is in fact a treatable and preventable condition, nevertheless, since the evolution of the disease greatly compromises visual acuity, its diagnosis must be made in its earlier progression stages, unveiling related symptoms as soon as possible and for the prescription of adequate treatment as a way of preventing sight-loss complications.

Laser photocoagulation is an intervention that is commonly used to treat DR, in which a laser beam is applied to the retina with the purpose of preserving vision by destroying, sealing and stopping the growth of the leaking blood vessels. This treatment unavoidably damages the light sensitive cells of the eye but is very effective in preventing the disease progression. As a consequence, diabetic subjects need periodic professional observation to prevent and monitor the progression of DR.

Promoting regular screenings in conjunction with a thorough management of glycemia and hypertension can work in reducing the number of people who develop this vision-threatening retinopathy [5]. Screening programs survey populations recurrently, and are acknowledged as one the most effective ways of detecting DR but, due to the quantity of subjects being observed in a short period of time, some of these screenings' restrictions must be taken into account. These programs involve complex logistics, involves numerous human resources, high amounts of digital storage, advanced equipment, among others. Therefore, the management of such screening initiatives becomes more difficult as the number of patients involved increases.

One way of reducing these requirements, could be to assess whether a patient needs treatment. If not, the patient is removed from the screening program, immediately decreasing the number of patients observed and consequently all the associated costs. The larger the number of patients correctly identified as not needing treatment, the more impactful this evaluation becomes.

Digital fundus photography is one of the most used diagnostic techniques employed in screening programs and produces images of the patient's retina for professional grading by ophthalmologists or optometrists. As a matter of fact, an alternative to the grading

procedure is the employment of automatic methods for the detection of lesions, such as the ones related to DR, through the processing of retinal fundus images [6].

Retmarker, a product developed by Retmarker S.A., contains examples of these algorithms [6], and contributes in alleviating the human grading burden and consequently the management of screening events, detecting DR-related lesions such as Microaneurysms (MAs) and Exudates (EXs).

Photocoagulation scars, as the ones resultant from DR treatment, may be misclassified as MAs and particularly as EXs (which appear bright in retinal fundus image, similarly to recent laser photocoagulation treatment scars). In other words, the presence of laser marks may cause misbehaviors of these automatic pathology detection algorithms.

Laser marks detection is a rather new research problem, with only a few scientific works addressing this specific subject.

For all the stated reason, an important and relevant task would be to implement a highly efficient and automatic method, that detects photocoagulation treatment laser marks, the primary objective of this thesis. Further motivation exists on the essential human burden reduction and processing time decrease in conjunction with the enhancement of existing methods.

In collaboration with Retmarker S.A., a recent work [7] presented an automatic algorithm for the detection lasermarks in digital fundus images, which achieved promising results. This algorithm was the starting point of this work.

In this work, and with the objective of improving the efficiency of the processing steps of the mentioned algorithm, a thorough optimization and search for alternative methods were made, aiming for improved time efficiency and performance in the classification of retinal images. Three binary classification algorithms were proposed and evaluated, using a total of fourteen datasets, eleven of which were used in [7], allowing classification performance comparisons and measurements of the optimization gains. In addition, three validation datasets were used.

The contribution of the present thesis consists in the optimization of algorithm in [7], supported by the introduction of new classification features, different binary classifiers, and the substitution or enhancement of that approach methods by some which proved more efficient in either efficiency or final classification. Examples of these improvements could be the introduction of further adaptiveness in pre-processing by using smaller and more-informative images, the introduction of Shiftable Bilateral Filter (a highly efficient approximation of the traditional bilateral filter), Matlab code-based optimizations, the im-

plementation of classification models, and the inclusion of a less noise-sensitive binary classifier, Support Vector Machine. As a result, the former algorithm was improved by about 80% in what time efficiency is concerned, while the accuracy of the method was also improved. Based on the previous results of the previous algorithm, which were improved in the present work, a scientific paper was proposed.

The contents of this thesis were produced in collaboration with Retmarker S.A., and the enhancements and optimizations performed to the existent algorithm were made with the objective of its implementation in the Retmarker family of products of the same company.

## 1.2 Document Overview

This thesis is structured in 6 chapters and 6 Appendixes.

**Chapter 2: Considerations and State-of-the-art** - Introduces Diabetic Retinopathy and its treatment, the state-of-the-art of this particular subject, three classification algorithms and finally the Materials used in this work.

**Chapter 3: Optimization methods** - Thoroughly describes the algorithm that was enhanced and the main optimization steps taken for its optimization.

**Chapter 4: Experimental Results** - Exposes the final algorithm's classification and time efficiency results, quantifying the optimization steps that were made. The results obtained are considered, explained and discussed.

**Chapter 5: Conclusion and Future Work** - The obtained results are compared with the proposed objectives and further improvements are suggested for the future.

**Appendix A** - Samples from the Retinal Fundus Images Datasets used in this work, and some related information.

**Appendix B** - Presents relevant information regarding the previous algorithm to be optimized.

**Appendix C** - Displays the mean time percentages of each processing step of the initial algorithm and their optimizations. Classification results based on the Decision Tree classifier obtained in [7] are presented, to infer about the impact of each optimization.

**Appendix D** - Introduces the theory of information measures by which the relevance of each computed image feature was rated, and two feature subsets used in the training of binary classifiers.

**Appendix E** - Comparative analysis of the classification performances.

**Appendix F** - Presents the classification models of the tested Decision Tree and Alternating Decision Tree classifiers .

## Chapter 2

# Considerations and State-of-the-art

Diabetes Mellitus' associated complications normally manifest themselves by the deterioration of the blood vessels. Diabetic Retinopathy (DR), which particularly affects the eye's retina, is one of these complications. The condition itself and its most common treatment, laser photocoagulation, will be introduced in this section. Retina, the most DR-affected eye structure, is one of the coatings of the human eye and grants it light-sensitivity by incorporating *rods* and *cones* cells that enable vision by color differentiation and depth perception of the surrounding environment.

Digital Fundus Images (DFI) results from a non-invasive imaging procedure for retinal observation. It is simply a photography of the fundus (Latin for *bottom*) of a subject's eye, acquired with a specialized camera, composed of an intricate microscope attached to a flash-enabled camera. DFIs are used to detect anomalies associated to retinal pathologies that affect the eye and to monitor the progression of diseases. It is essential in the detection of conditions such as macular degeneration, retinal neoplasms, glaucoma, diabetic retinopathy and many others.

Figure 2.1 presents an example of a DFI, where some anatomical structures can be identified: the *Optic Disc* (**OD**), an ellipsoidal structure where the optic nerve leaves the retina and where light-sensitive cells do not exist; the *macula*, a highly pigmented yellow area near the centre of the retina which is responsible for central high resolution vision; the *fovea*, a small pit near the macula centre that contains the largest concentration of cone cells in the eye, and the *vascular network* of veins and arteries irrigating the eye. These are the four main and most relevant anatomical landmarks in the context of this work.

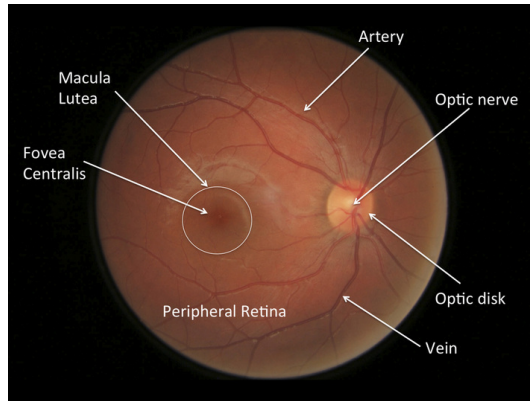


Figure 2.1: Anatomy of the human eye's retina [8].

Nowadays, medical image processing is an important research area, which developed techniques and systems increasingly contribute to the automatic diagnosis of pathologies, by the means of visual representation and imaging. Image processing has a fundamental role in providing the necessary adaptiveness, robustness and automatical analysis for the efficient and accurate pathology detection through visual data processing.

## 2.1 Diabetic Retinopathy Considerations and Treatment

DR can develop into two complications: Nonproliferative DR and Proliferative Retinopathy.

*Nonproliferative DR*, the earlier stage of DR, occurs when the walls of blood vessels weaken and become abnormal, leaking blood and fluids into the surrounding tissue, originating exudates (protein and lipid deposits). If the fluid or blood leakage occurs near the macula (responsible for central high definition sight) vision will be impaired due to fluid leakage causing macular swelling, which blurs central vision. This condition is called Diabetic Macular Edema and can occur in any of the stages of DR. Microaneurysms and small hemorrhages are also signs of this stage of DR.

*Proliferative Retinopathy* is the advanced stage of DR which takes place when the severely diminished blood flow causes the damaged retina to initiate a regeneration process through generation of new blood vessels. However, these new vessels are very fragile and tend to break and leak blood. When they bleed into the vitreous humor, they can block the passage of light and cause a sudden loss of vision. The blood is usually reabsorbed, but scar tissue may form, which can lead to severe impairment and even permanent vision loss. About half of the subjects with Proliferative Retinopathy also develop Diabetic Macular Edema.

### 2.1.1 Photocoagulation Treatment

Laser photocoagulation is a commonly used intervention to treat DR, in which a laser beam is applied to the retina with the purpose of preserving vision by sealing or preventing the growth of abnormal blood vessels. While preventing the rapid progression of the disease, this treatment unavoidably causes damage to the light sensitive cells of the eye, one more reason why patients with this condition must be identified before visual acuity loss occurs [9, 10].

Two main types of laser treatment are prescribed for the treatment of DR, Pan-Retinal Photocoagulation and Macular Edema Photocoagulation.

*Pan-Retinal Photocoagulation* (PRP), also know as Scatter Photocoagulation, is the most proven and accepted treatment for PR and is used to slow the growth of new abnormal blood vessels, which have developed over a wider area of the retina. PRP involves applying a substantial number of large and intense laser burns. The main purpose of PRP is forcing the regression of the new vessels' growth in order to delay the progression and limiting the damage to the OD and macula to preserve central, high resolution vision. As a result, there is an improved tissue oxygenation of the peripheral retina.

*Macular Edema Photocoagulation* (MEP) is the most indicated treatment for Diabetic Macular Edema and extremely difficult to perform. Extreme precision is required to apply MEP which consists in applying low intensity small burns in a "C" shaped area around the macula, helping to clear the edema. If by mistake the laser operator aims the photocoagulation laser beam at the fovea, visual loss may result.

The scars produced by each of these treatments differ in size, location, intensity and number. Consequently patterns generated by these two different types of treatment are also different with MEP scars being smaller, fewer, and much more difficult to detect when compared to PRP's [9, 11].

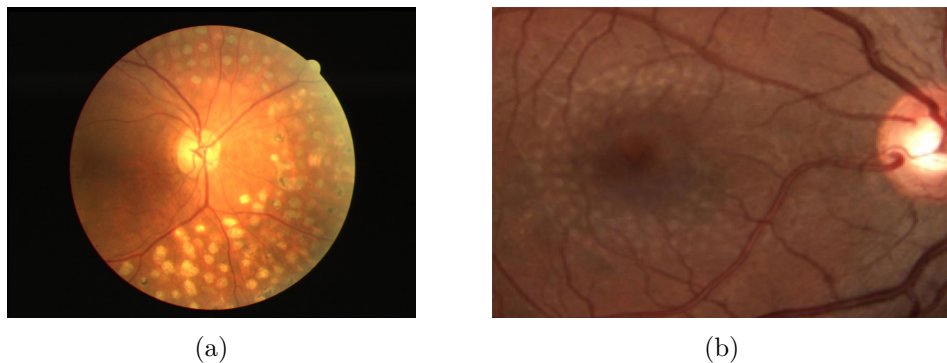


Figure 2.2: Example patterns: (a) PRP (b) MEP

In either case, the scars left on the retinal surface area have specific geometric, distribution and intensity characteristics. From the inspection of DFIs of a DR screening coordinated by ARS Centro which resulted in some of the lasermarks-containing datasets of this work, it can be concluded that laser mark patterns present the following properties:

- They appear in clusters and therefore are not randomly distributed over the retina;
- They occur in periphery regions, far from the fovea, the OD and the main veins and arteries of the vascular network (Figure 2.1);
- They present a dark or bright appearance and their color is either yellow or green. This fact relates to how recently the treatment was performed;
- They usually have a circular form but sometimes present an asymmetric shape, especially when closer to the vascular network;

The observed characteristics result from professional medical methodologies, healing patterns and laser specifications that do not present great variations between patients. It can thus be concluded that these photocoagulation patterns are generally the ones found in most lasermark-containing DFIs.

A final consideration regarding photocoagulation treatment must be made. As stated, four main anatomical structures can be identified in the human retina, and consequently in Retinal Fundus Images: Macula Lutea, the OD, the Vascular Network and Fovea. The three latter structures are important in this context for a treatment-related reason, which is the location of lasermarks. When patients are submitted to laser treatment none of these structures will be photocoagulated. For medical reasons, the OD is never subjected to the laser beam action. The Vascular Network, which sometimes receives this treatment to prevent the growing of new abnormal blood vessels, has the singularity of not developing scars in its vicinity, moreover, main arteries and veins are not photocoagulated as the eye's nourishing would be compromised. If the fovea (from where approximately half of the optic nerve fibers carry information being responsible for the central/sharp vision) would be directly targeted with the laser beam, central vision would be harmed.

Therefore, the anatomical landmarks of the retina, that are present in a DFI, can be used as rejection criteria for laser mark candidates.



## 2.2 Lasermarks detection state-of-the-art

As introduced, the task of classifying a patient as treated with laser photocoagulation can be reduced to the observation of whether the retina presents or not laser treatment scars.

Considering this task, statistical performance measurements will be introduced, after what the three major state-of-the-art proposed solutions will be presented.

**Sensitivity** and **Specificity** are two statistical measures that can be used to quantify the performance of a binary classifier. Their values are calculated as exemplified in Equations (2.1) and (2.2).

$$Specificity = \frac{TN}{TN + FP} \quad (2.1) \quad Sensitivity = \frac{TP}{TP + FN} \quad (2.2)$$

where,

- TP are true positives, retinal images with lasermarks that were correctly classified;
- TN are true negatives, retinal images without lasermarks that were correctly classified;
- FP are false positives, retinal images without lasermarks which were classified as "Laser";
- FN are false negatives, retinal images with lasermarks which were classified as "No Laser".

### 2.2.1 Detection of Laser Marks in Retinal Images

Dias *et al.* suggested one and the very first approach for the automatic detection of laser marks in DFIs [12]. This method uses two binary classifiers: GPC (Generic Parameter Classification) and SPC (Structural Parameter Classification), that identifies an image as "Laser", i.e. showing the presence of lasermarks, should both the outputs of GPC and SPC be positive and as "No Laser" if either one of them has a negative output.

GPC classifier makes use of various values, obtained through color, focus, contrast and illumination analysis of the input DFI. A total of fourteen features that are indicators of general quality parameters are the inputs to this classifier. This classifier takes advantage of the fact that general quality parameters are divergent in images that contain and images that do not contain lasermarks.

A first dataset of 40 DFIs with lasermarks and 176 without lasermarks was used to train the GPC binary classifier, using 4-fold validation, with 75% of the total images used

for training and 25% for testing. The ratio between images with and without lasermarks was the same for the training and test subsets.

SPC has four features as inputs: Simple Measure (SM), Colormap 1 (CM1), Colormap 2 (CM2), and Colormap 3 (CM3). SM, or *single measure*, is computed from the image structural information. First, a Sobel operator is applied to obtain a gradient image, where several morphological erosion operations with different structuring elements are performed. The resulting images are summed, obtaining a final binary image from which irrelevant, small objects are removed. From this image, SM counts the total foreground pixels with candidate lasermarks.

CM1, CM2 and CM3 are obtained by histogram projection with three different colormaps, built with statistical measurements made on a second dataset with 101 DFIs presenting lasermarks and 141 without lasermarks. CM1, CM2 and CM3 are computed from the obtained colormaps called *eye-fundus*, *bright marks* and *dark marks*. These 4 features are inputs to a Feed Forward Neural Network, trained and validated with a 4-fold cross validation.

A third and final dataset (constituted by the Messidor dataset [13] of 1200 retinal images without laser marks, plus 996 other images of which 101 present laser marks) was used to perform classification evaluation. The performance of the algorithm is presented in Table 2.1,

Dataset	Statistical Performance
Sensitivity (101 images)	63.37%
Specificity (2095 images)	99.90%

Table 2.1: João Dias *et al.* laser mark detection algorithm performance. Adapted from Table 1 of [12].

The importance of detecting lasermarks was acknowledged by the authors, who conclude that these results are remarkable, since the obtained specificity is nearly 100%, a desirable goal since classifying subjects as treated when they were not, is not medically acceptable.

### 2.2.2 Laser Marks Detection From Fundus Images

Tahir *et al.* presented later an alternative solution for the automatic detection of laser scars [14]. This method has four main steps: pre-processing, candidate region identification, feature extraction and classification. Pre-processing step concerns the fact that retinal

images are often noisy and poorly illuminated. The green channel of the RGB fundus images is used for being the one with the better contrast. A circular kernel averaging filter reduces the noise in the green channel after what adaptive histogram equalization [15] is performed, to enhance contrast. Illumination is then corrected using a top-hat filter.

To assess candidate regions, all possible lasermarks are identified. A binary image containing the masked candidate regions is computed by thresholding this image, with a small-valued threshold which still obtains undesirable false detections.

In order to remove false detections, 10 features are computed for each candidate region [14]: Compactness, Max hue, Max saturation, Standard deviation of saturation, Intensity mean, Intensity max, Intensity standard deviation, Mean Red channel, Max Red channel and Max Green channel. Using a minimum distance classifier, each candidate region is evaluated as being or not a real lasermark, based in the computed features. The image is classified as presenting lasermarks if any candidate is considered to be a lasermark.

The proposed algorithm achieved the performance values presented in Table 2.2, evaluated on a locally gathered dataset consisting of 51 images containing lasermarks and 329 other images without lasermarks. All images have the same resolution and were acquired with the same camera, and were labeled by ophthalmologists as having or not lasermarks.

Dataset	Statistical Performance
Sensitivity (51 images)	94%
Specificity (329 images)	97%

Table 2.2: Faraz Tahir *et al.* laser mark detection algorithm performance. Adapted from Table 3 of [14].

These results indicate that the proposed algorithm has a very good performance with 94% sensitivity, even if specificity is 3 percentual points away from 100%. However, it must be pointed out that the images containing lasermarks were taken with the same camera, and all have the same resolution.

### 2.2.3 Contributions to the Automatic Detection of Laser Marks in Retinal Digital Fundus Images

Sousa *et al.* [7] made a remarkable contribution to this subject, by developing an automatic algorithm for classification of images containing lasermarks. Combining a total of 65 features, descriptive of the image, which were input to five different tree-based classifiers. For the training these classifiers, a total of 855 images were used, with 203 of these being lasermark-containing DFIs, and a second dataset, used for testing, was constituted by

with 1749 images, of which 135 present lasermarks.

Promising results were reached using a Decision Tree classifier, as shown in Table 2.3.

Measure	Statistical Performance
Specificity	98,9%
Sensitivity	88.1%

Table 2.3: Sousa *et al.* results on Decision Tree classifier. Adapted from Table 5.12 of [7].

While based on state-of-the-art approaches, this work introduced new concepts in the detection process, including the use of a scaling factor able compensate for resolution variations, and the introduction of the segmented vascular network as lasermark-free region (Figure 2.1). This work used a total of 11 datasets, 3 of which contain lasermarks. This image variety, with and without lasermarks, ensured for a reliable validation of the algorithm.

The primary objective of this method was in reducing the workload adressed to optometrists and ophthalmologists, and also to bypass time-consuming processing steps that are unnecessary if a patient has already been treated. The capacity of images presenting lasermarks to deteriorate the performance of further detection algorithms was also a concern, as stated in previous contributions to this particular subject.

The present thesis, whose main objective was implementing a time and quality-efficient algorithm for detecting laser marks in DFIs, had the method just described as the primary reference.

## 2.3 Lasermark Segmentation Algorithms in Retinal Images

Image processing is a form of signal processing in which the input is an image, a series of images or a video. In most image processing methods image is treated as a two or three dimensional signal, to which standard signal processing techniques are applied in order to obtain modified output image(s) with the expected form. Image Segmentation, as an image processing technique, is a procedure where a digital image is automatically or semi-automatically partitioned into multiple segments. Segmentation is used to simplify or change the representation of images in order to ease their visual or computational analysis.

Several segmentation algorithms are used in the digital processing of retinal images with various objectives. Examples are the Sekhar *et al.* [16] algorithm with the application of the Circular Hough Transform to locate the OD which is circular bright object similar to some laser marks or Salem *et al.* [17] who suggested a segmentation of retinal blood

vessels based on the analysis of the Hessian matrix's eigenvalues, with authors predicting the same approach could be used to detect blob structures, one of the characteristics of lasermarks.

The already referred anatomical structures that are present in human retina, are of interest for this work. Different approaches exist for their detection and example methods for segmenting the vascular network are presented in [18–20]. The OD can be detected, for example, using Circular Hough Transform as in [16] or using a morphological adaptive approach using the segmented vascular network as in Welfer *et al.* [21]. Niemeijer *et al.* [22] also present an automatic method for the segmentation of all these major anatomical structures.

In this thesis, the interest of segmentation is specifically related with segmenting lasermarks, and particularly in optimizing the three image segmentation methods that were used in [7]: Circular Hough Transform (**CHT**), Laser Mark Segmentation (**LMS**) and Frangi Vesselness Filter(**FVF**), which will be presented in the next subsections.

### 2.3.1 Laser Mark Segmentation

A Laser Mark Segmentation (LMS) algorithm was proposed by Sohini Roychowdhury [23]. This algorithm was suggested in the context of segmenting scars resultant from photocoagulation treatments.

The proposed algorithm is preceded by pre-processing steps (Figure 2.3 a),b) and c)) in which the image's green channel is extracted. Using only this channel, pixel intensities are scaled to a [0,1] range, to which contrast enhancement is applied, obtaining image I. I is eroded using a morphological structuring element disk of radius  $r = 15$ , followed by image reconstruction as a way of enhancing the compactness of bright regions and resulting in an image J. J is then subtracted from I, scaled to [0,1] and contrast enhanced, resulting in a morphologically transformed image, K. In image K the pixels of the Vascular Network and of the OD's neighborhood are ignored, through masking operations, and finally a threshold value obtained with Otsu's method [24] is applied adaptively resulting in the final image  $I_M$  (Figure 2.3 d)).

Four discriminating features are computed from the image: Roundedness, Solidity, Filled Rate and Compactness[23], which will be reviewed further ahead. These features evaluate if a candidate region is indeed a lasermark(Figure 2.3 e))

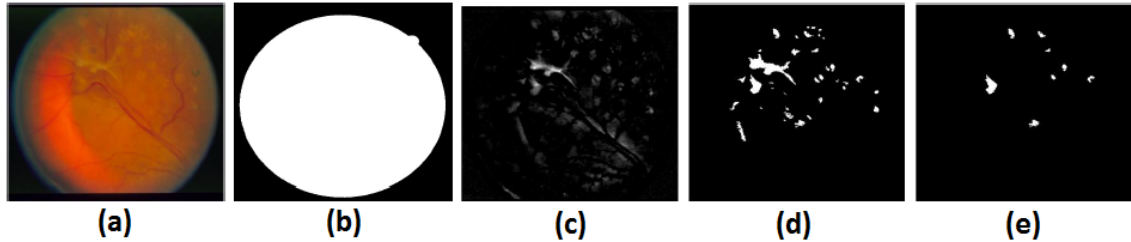


Figure 2.3: LMS pre-processing steps, candidate regions and final detected lasermarks (adapted from Figure 6.2 of [23]).

### 2.3.2 Frangi Vesselness Filter

In 1998, Alejandro Frangi *et al.* [25] developed a vessel enhancement filter using the eigenvalues of the Hessian matrix of an image<sup>2.3</sup>. This enhancement was performed in Digital Subtraction Angiography 2D images and 3D aortoiliac and cerebral magnetic resonance angiographies showing the filter simultaneous properties of background and noise suppression while segmenting vessels in maximum intensity projections and volumetric displays.

This method was used in different medical image processing problems since then. Examples are Pulmonary Vessel segmentation [26], 3D lung vessel segmentation [27], in coronary artery segmentation in Computed Tomography (also known as CT) images [28–30] and even in vessels enhancement in DFIs [17, 31, 32].

As cited methods have proven, Frangi Vesselness Filter (FVF) detects tubular, ridge and blob-like objects in medical diagnosis images with good performance. FVF searches for this kind of geometrical forms but, based on the fact that the size of these forms may vary, a multiscale approach is used. Therefore FVF stands as an iterative algorithm that merges the vesselness measure computed for each scale size  $\sigma$ . For each iteration, the retinal image is convoluted with a Gaussian filter with a kernel of size  $\sigma$ . Then, the Hessian matrix is calculated, according to Equation (2.3), where  $I$  is the processing image,

$$H(I) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (2.3)$$

$\lambda_1$  and  $\lambda_2$ , eigenvalues of the Hessian matrix, are calculated with  $|\lambda_1| < |\lambda_2|$ . Figure 2.4 is a summary of the type of objects detected by the 2D Frangi Filter as a function of the eigenvalues.

Analysing eigenvalues of the Hessian is the equivalent of obtaining the main directions in which the second order local structure of the image can be decomposed. Therefore, low  $\lambda_1$  values occur when the structures are tubular while high  $\lambda_1$  values occur when the structures are blob-like. Also, if  $\lambda_2 < 0$ , the structures are bright, otherwise the objects

$\lambda_1$	$\lambda_2$	orientation pattern
N	N	noisy, no preferred direction
L	H-	tubular structure (bright)
L	H+	tubular structure (dark)
H-	H-	blob-like structure (bright)
H+	H+	blob-like structure (dark)

Figure 2.4: Patterns in 2D, depending on the value of the eigenvalues. N - Null, L - Low, H - High (adapted from Table 1 of [25]).

have a dark appearance.

Frangi *et al.* [25] originally proposed a 2D vesselness measure for each scale  $V_\sigma$ , according to Equation (2.4),

$$V_\sigma = \begin{cases} 0, & \lambda_2 > 0 \\ \exp(-\frac{R_b^2}{2\beta^2})(1 - \exp(-\frac{S^2}{2c^2})), & \lambda_2 < 0 \end{cases} \quad (2.4)$$

where  $R_b = \frac{\lambda_1}{\lambda_2}$  is the blobness measure and  $S = \sqrt{\lambda_1^2 + \lambda_2^2}$  is the Frobenius norm. If the value of this norm is low, it means that  $V_\sigma$  is low in the background pixels, where no structure is present and the eigenvalues are small for the lack of contrast.

For every pixel, the value with the maximum vesselness measure  $V_\sigma$  of computed scales (all the  $\sigma$  values) is returned. An example of the filter's performance with  $\sigma = \{1,3,5,7,9\}$  is shown by Figure 2.5.

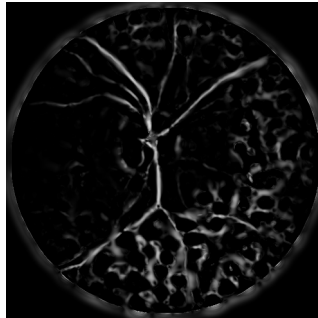


Figure 2.5: Example image of the detected structures using FVF with Figure 2.2a as input.

### 2.3.3 Circular Hough Transform

The Hough Transform is an important method patented by Paul Hough in 1962 for the automatic detection of lines in an image. The Circular Hough Transform (CHT) is one of the many variations of this first transform [33], which identifies circles and almost-circular shapes. Several medical image processing problems are solved by this method,

as is the case of MAs detection [34] or, as mentioned before, the detection of the OD [16, 35]. As explained, laser marks may present a circular or asymmetric shape making this algorithm one of the most effective in detecting lasermark candidates, mainly because of its robustness to image variations like illumination and noise, often present in digital fundus images.

To present a brief explanation of the method, in a two-dimensional space, a circle can be described by:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.5)$$

where  $(a,b)$  is the circle's center and  $r$  is its radius. CHT starts by applying an edge detection method like Canny or Sobel [36] after what circular patterns are searched by using a technique equivalent to a convolution of the edge image with a circle operator of a given radius  $R$ . In other words, all the edge candidates cast "votes" of all the points of a circle of radius  $R$  and centered in each of these edges. This process can be interpreted as a so-called voting scheme in an accumulator matrix, and where a peak is resultant from the overlap of each circle's contribution at the center of the original circle. This method equivalent to a voting scheme can be exemplified by Figure 2.6, where Figure 2.6b) exemplifies the previously detected candidate edges (Figure 2.6a)) lying on an actual circle (solid circle), and the voting patterns generated by them (dashed circles) which coincide at the center of the actual circle. Finally Figure 2.6c) is an example peak of the computed accumulator space.

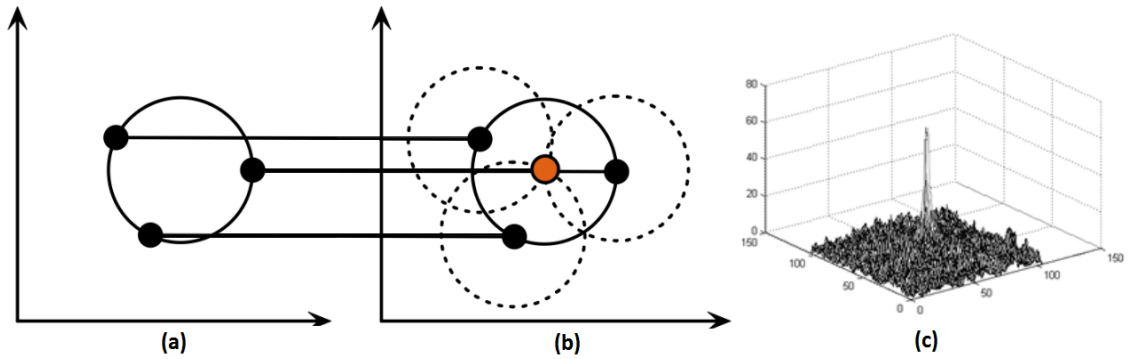


Figure 2.6: CHT voting pattern: a) Detected edges, b) Circle contributions of each edge [37] and c) Accumulator array correspondent to a circle detection. The obtained peak is the center of the circle [34].

After the accumulator matrix is calculated and normalized, circles are easily found by finding peaks that exceed a certain sensitivity threshold.



## 2.4 Classification Algorithms

A classifier is an algorithm that given set of occurrences classes identifies new observations as belonging to one of these classes, based on the information of training data that contains past observations whose class is known. The classification algorithms used in this work are considered to be examples of supervised learning, since there is the presence of a training set of correctly identified situations. This training information is presented to a classifier in the form of nominal or numerical descriptors that describe each instance.

Examples of classification problems would be classifying a patient as "Sick" or "Not sick", identifying a circular shape as "Circle" or "Not a Circle" or even classifying retinal fundus images as "Contains Lasermarks" or "Does not contain Lasermarks". These examples are similar to the problem at hand: returning a final binary decision, classifying an image as containing lasermarks (or "Laser", the positive case), or as not containing lasermarks (or "No Laser", the negative case).

In this work three binary classifiers were used - Decision Tree (**DT**) [38], Alternating Decision Tree(**ADT**) [39], and Support Vector Machines (**SVM**) [40].

### 2.4.1 C4.5 Decision Tree

Ross Quinlan introduced Iterative Dichotomiser 3 (ID3) [38], an algorithm that builds a classification model which can be expressed by a decision tree. Figure 2.7 exemplifies a DT that starts as a single node, "Outlook?", with three possible outcomes, and requires a decision to proceed. One of these outcomes is traversed and either another node is reached (and a new decision must be made), or a final decision is obtained. Since all the five leaf nodes of this tree contain a set of two possible outputs, this classifier returns a binary decision.

For example, an instance of [*Outlook=sunny, humid=normal*] would be classified as "True" by the decision tree classifier of figure 2.7.

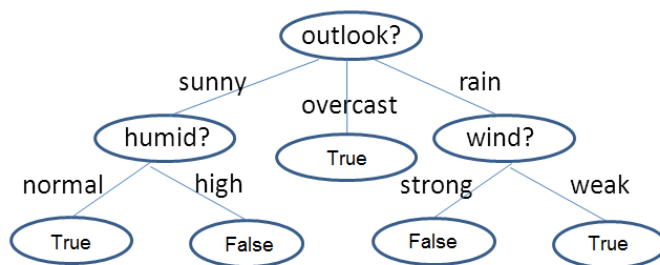


Figure 2.7: Decision Tree classifier example (adapted from [41]).

During training phase, in which ID3 has the objective of creating the smallest tree pos-

sible, each node's attribute is obtained by computing the information gain of all attributes and selecting the highest information gain of them as the attribute for the corresponding decision node. The ID3 algorithm stops after all attributes are used or after every instance of the training set was correctly classified by the model.

However, the ID3 algorithm has some limitations. While searching for the optimal solution, it solely maintains a single current hypothesis, while earlier (and maybe better) versions are eliminated; it only operates with nominal attributes; it performs no backtracking in search, meaning it may converge to locally optimal solutions that are not globally optimal, which can in turn lead to overfitting.

To overcome these issues, Quinlan *et al.* [42] developed an extension of ID3 - the C4.5. The main improvements of C4.5 were the compatibility with continuous variables, the use Gain Ratio D instead of Information Gain D for attribute selection at each node, and *pruning*, a process where the final tree has some of its branches removed as a way of reducing the overfitting probability. This new algorithm has the same principles of ID3, despite the enhancements introduced.

## 2.4.2 Alternating Decision Tree

ADTs were introduced by Freund and Mason [39]. The algorithm was later reviewed and optimized by Pfahringer *et al.* [43], who called it the Alternating Decision Tree Induction algorithm. This latter version of the algorithm is included in WEKA, as referred in [43].

A standard DT has interior nodes where decisions are made and leaf nodes where "arriving" instances are classified. On the other hand, ADT consists in an alternation of decision nodes that specify a predicate condition, and prediction nodes, which contain a single number that is added or subtracted to the instance's "score". Prediction nodes can be interior or leaf nodes and the root node is always a prediction node.

The way an instance is classified differs from a typical DT in the fact that all the paths for which decision nodes are true are followed, summing any traversed prediction nodes and obtaining a final score. If the final score is positive, the instance is classified with one of the classes, and if the score is negative it is classified with the remaining one.

The concept can be explained by Figure 2.8. An instance of  $[A1=true, A2=false]$  would enter the root node adding +0.5 to its initial null score. Two decision nodes are reached after the root where boolean decisions lead to one of two new paths. As indicated by the leftmost horizontal arrow, since the prediction node condition of  $[A1=true]$  is verified, the "y" path is traversed adding -1.2 to the present score of 0.5. Similar operations

are performed in the remaining prediction nodes. The final score obtained by the instance can be expressed by the sum  $0 + 0.5 + (-1.2) + (-3.4) + 0.2 = -3.9$ , where all traversed leaf nodes are indicated by horizontal arrows in the figure. This instance would be classified with the "negative class", since  $-3.9$  is a negative real number.

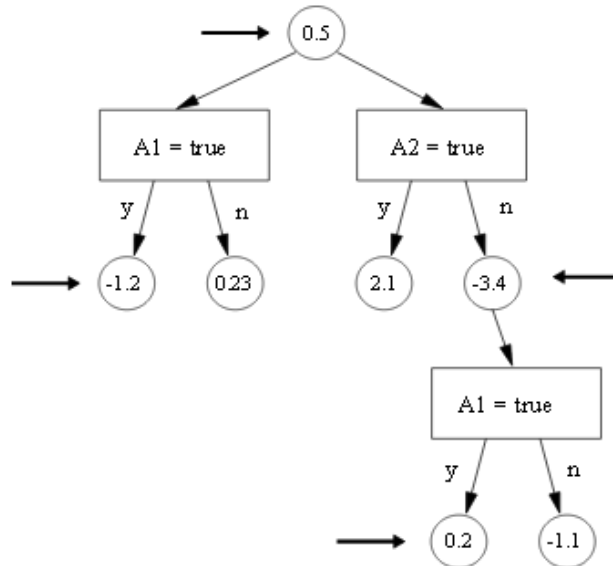


Figure 2.8: Alternating Decision Tree example (Fig. 1 of [43]).

In addition, ADT uses the boosting procedure, which reduces bias and variance being a supervised learning algorithm. Boosting aims to produce strong learners, well-correlated with the with true-classification, even if only a set of weak learners is provided; each boosting iteration adds a test (called *weak hypothesis* in both [39] and [43]) and two predictor nodes to the tree. In each iteration the test chosen is the one that minimizes a function that measures the impurity of the test. For each boosting iteration, the minimization function has to be computed for each possible test, which results in an algorithm quadratic in the number of boosting iterations.

### 2.4.3 Support Vector Machines

Support Vector Machines (SVM) are mathematical models that work by finding a hyperplane or a set of hyperplanes in a high-dimensional space to separate members of two classes as widely as possible. SVM were introduced by Cortes and Vapnik [40] but can be traced back as early as 1964 by Vapnik and Lerner. SVM are well known for their robustness, resilience to overfitting and overall performance being acknowledged as one of the most successful binary classification algorithms. SVM can be applied to two different types of data.

When applied to *Linearly Separable Data*, which can simply be defined as points in a

plane/space whose classes can be separated by a line/hyperplane, the parameter to be maximized is called *margin* and represents the distance of the separation line or hyperplane to the closest members of each class. When applied to *Linearly Unseparable Data*, or instances of data that cannot be separated by a line/hyperplane in the plane/space, a positive slack variable, which allows for the misclassification of points, must be minimized aiming to reduce the number of misclassifications. Another subtlety of SVM when applied to *Linearly Unseparable Data* is the use of kernel-based transformations into higher dimensional spaces in search of other possible hyperplanes for the separation of data.

This process can be explained by Figure 2.9. The linearly nonseparable data in (A) can be imagined as a in  $\mathbb{R}^2$  representation of the dataset present in (B), in  $\mathbb{R}^3$ , which can be linearly separated by a hyperplane((C)). Therefore, it can be assumed that if we work in  $\mathbb{R}^3$  a good linear SVM decision boundary can be found.

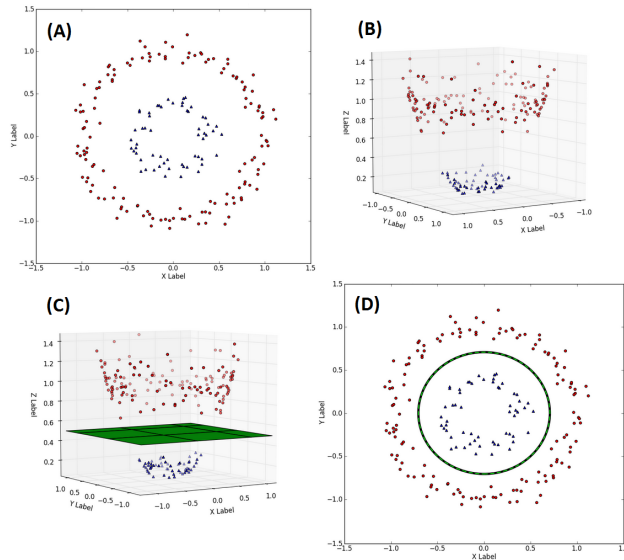


Figure 2.9: Non Linear SVM Example.

## 2.5 Test Platform

The test platform used in this work has the following specifications: **Processor:** Intel Core i5-2410M CPU @ 2.30Ghz, **RAM:** 4GB, **Operative System:** Windows 7 Professional.

## 2.6 Materials

This work used Matlab R2008b and WEKA 3.6, a machine learning/data mining software written in Java. For all the tests performed in this work, several digital fundus images datasets were used(Tables 2.4 and 2.5), consisting of 10 public datasets(1 through 8, 13

and 14) and 4 proprietary datasets (9 through 12). Test datasets (Table 2.4) were used to evaluate the performance of the proposed method while validation datasets (Table 2.5) were classified to validate the final method with different datasets than those used in the testing of the previous method in [7]. Each image of these datasets is to be classified as correspondent to one of the considered binary classes "Laser" and "No Laser". Examples of these classes are presented in Figure 2.10.

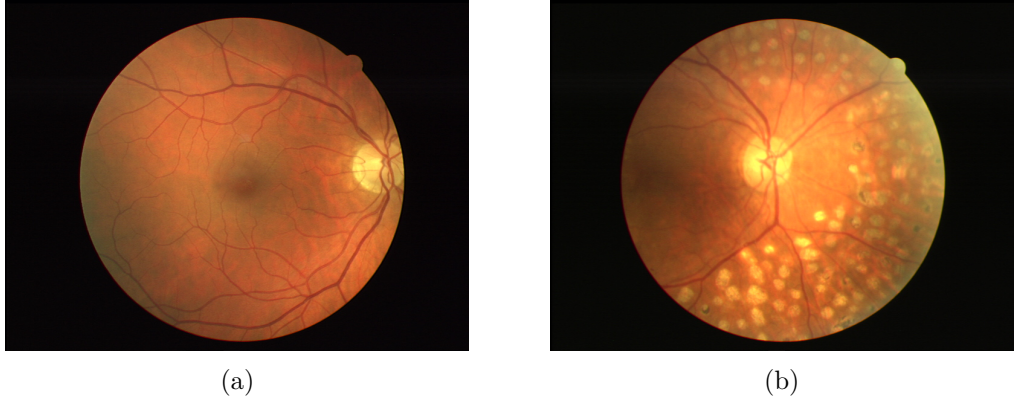


Figure 2.10: RFI of a healthy patient (a) and of a patient with laser treatment marks(b).

Table 2.4: Test datasets used.

N <sup>o</sup>	Dataset	Reference	N <sup>o</sup> of Images	Laser Marks
1	Messidor ( <b>MESSIDOR</b> )	[13]	1187	No
2	e-ophtha MA ( <b>MA</b> )	[44]	148	No
3	e-ophtha NOMA ( <b>NOMA</b> )	[44]	233	No
4	e-ophtha EX ( <b>EX</b> )	[44, 45]	47	No
5	e-ophtha NOEX ( <b>NOEX</b> )	[44, 45]	35	No
6	Vessel-Based Registration ( <b>VBR</b> )	[46]	22	No
7	50 Healthy Patients ( <b>50HP</b> )	[47]	100	No
8	Foveal Avascular Zone Detection ( <b>FAZD</b> )	[48]	60	No
9	Screening ( <b>SCR</b> )	-	622	Yes
10	Before and After Treatment ( <b>BAT</b> )	-	49	Yes
11	Proprietary Dataset João Dias ( <b>PDJD</b> )	[12]	101	Yes

Table 2.5: Validation datasets.

N <sup>o</sup>	Dataset	Reference	N <sup>o</sup> of Images	Laser Marks
12	Screening Laser 2015 ( <b>SCR15</b> )	-	99	Yes
13	High-Resolution Fundus ( <b>HRF</b> )	[49]	45	No
14	Chasedb ( <b>CDB</b> )	[50]	28	No

### 2.6.1 Public Datasets

The datasets 1 to 8 in Table 2.4 result from previous studies and are publicly available online. All these represent images without lasermarks. These datasets have different

origins and characteristics:

- **MESSIDOR** is one of the most used Datasets for teaching purposes, and is also a resource for many scientific studies. Its images were obtained using a color video 3CCD camera, from a *Topcon TRC NW6 non-mydratic retinograph*. Messidor is composed of 1200 images of resolutions 1444x960, 2240x1488 and 2304x1536. According to image processing experts at *Retmarker S.A.*, 13 images presenting lasermarks were excluded from this dataset (See Appendix A).
- **MA**, **NOMA**, **EX** and **NOEX** are part of a project called ANR-TECSAN - TELE-OPHTHA funded by the French Research Agency. All images of MA and EX contain DR-related lesions, Microaneurysms and Exudates, respectively. NOMA and NOEX do not present signs of DR or treatment, however, EX contains many optical artifacts capable of misleading automatic detection algorithms. All 363 images have resolutions of 1440x960, 1504x1000, 2048x1360 or 2544x1696.
- **VBR** images were acquired with a *Topcon 3D OCT-1000* camera and have a resolution of 1200x1143. The dataset consists of 22 pairs of images, 22 are fundus images and 22 are Optical Coherence Tomography images. The former are the ones we are interested in for this work.
- **50HP** images are from 50 healthy subjects' left and right eyes. Their resolution is 1612x1536.
- **FAZD** consists of 60 healthy patients retinal images of resolution 720x576.

## 2.6.2 Proprietary Datasets

Datasets 9 to 11 (table 2.4) are proprietary datasets made up of images with lasermarks. Some observations on these datasets are presented below.

- **SCR** is composed of 419 retinal images without laser marks and 203 retinal images having lasermarks. These images result from a DR screening program managed by *ARS Centro*, and were classified by professional optometrists as having or not laser marks. All of the images are non-mydratic and have a 45° FOV and they were acquired between February, 2014 and January, 2015. There are 94 images captured using *Nidek AFC-330 Retinal Camera* 76 of these with a resolution of 1920 by 1920 pixels, and the remaining 18 with 2448x2448. There is one image captured using *CSO Cobra Retinal Camera* with 60°x45° FOV and a resolution of 1624 by 1232 pixels. The remaining 527 images were captured using *Canon CR6-45NM Retinal*

*Camera* and have a resolution of 768 by 584 pixels. The totality of images, 622, corresponds to the number of patients; only one image per patient was selected.

- **BAT** is composed of 34 retinal images with and 15 without laser treatment scars. These images were provided by Centro Cirúrgico de Coimbra (CCC). This dataset is composed by images from 9 patients (See Appendix A)
- **PDJD** is composed of 101 retinal images with laser marks from a screening program managed by ARS Centro previous to 2013. All of the images are non-mydratic and have a 45° FOV and were captured using *Canon CR6-45NM Retinal Camera* and have a resolution of 768 by 584 pixels. This dataset contains images selected by image processing experts and they were not classified by any optometrist or ophthalmologist.

### 2.6.3 Validation Datasets

Validation datasets (Table 2.5) were obtained and used for the purpose of validating the performance of the final optimized algorithm proposed in this thesis.

- **SCR15** is composed of images from the same and ongoing screening program from **SCR** but were obtained from February, 2015 to November 2015. As a result this is also a proprietary Dataset. All images were acquired with a *Nidek AFC-330 Retinal Camera* and have resolutions of 1920x1920 or 2448x2448. Only one image per patient was selected.
- **HRF** is a public dataset aggregating 15 images of healthy patients, 15 images of patients with diabetic retinopathy and 15 images of patients suffering from glaucoma. All images have a resolution of 3504x2336.
- **CDB** is a dataset of 28 images of health patients publicly available online. Its images have a resolution of 999x960.

Samples of all these datasets are presented in Appendix A.





## Chapter 3

# Algorithm Optimizations

In this chapter, the optimization steps performed on the aforementioned algorithm will be reviewed and explained. In order to do this, the algorithm itself must be further explained.

### Algorithm Review

The algorithm involves four main processing steps: *Pre-processing*, *Segmentation*, *Features Computation* and finally *Classification*. Appendix B.1 describes these processing steps in detail, which involve the following operations, for any input image and in this specific order: **Pre-processing**: *resizing the image*, performed using a bicubic interpolation if the image does not fit a specific resolution range; *segmentation of the anatomical structures*, using a contourlet transform for the Vascular Network and using the method described in [35] for the OD; computation of the *Region of Interest (ROI) mask*; *uneven illumination correction* to correct luminance; *adaptive histogram equalization* to enhance contrast, and finally *channel extraction* where the most relevant channels for detection are extracted: Green, Red, Hue and Saturation.

After pre-processing, the segmentation algorithms described in subsection 2.3 are applied to the green channel of the resultant image. For each algorithm, regions that are considered to be false detections are discarded resulting in final set of candidate regions, obtained by merging the regions identified by each segmentation algorithm. The binary image containing these candidates is used to compute several candidate-related features. A summarized description of the 65 features computed by the main algorithm is presented in Appendix B.2, where the first 12 features represent geometrical features, features 13 to 22 represent spatial distribution features, 23 to 38 are intensity based features in the four image channels extracted, and the last 27 features are texture features based the majority of which are obtained through the Gray-Level Co-occurrence Matrix of the image [51].

The classification step, which was not part of this method, was developed using WEKA in order to test and obtain the final classification using five different tree-based classifiers. C4.5 Decision Tree was the classifier used to obtain the best classification in both accuracy and specificity. For this reason, in this work the DT classifier obtained by Sousa *et al.* was implemented in Matlab and the time efficiency and the classification improvements or deteriorations of each optimization were evaluated.

### Efficiency Analysis

As previously mentioned, one of the identified problems regarding images containing laser marks is their capacity to mislead further processing steps, namely the identification of DR-related retinal lesions. With this problem in mind, it is clear that the algorithm proposed by [7] may increase the efficiency of other automatic detection methods. Efficiency is then crucial, since in order to increase the efficiency of such algorithms, as is the case of the implemented algorithms in Retmarker, automatic detection of lasermarks has to be performed before these other methods.

As an increased processing time of the major Retmarker algorithm is not desired, an extensive optimization needed to be made.

In this work, several optimization procedures were performed. This chapter aims to explain these improvements, why they were made, and the overall impact in the final processing time, and constitutes the primary contribution of the present thesis to this field of knowledge, both in the optimization approach and on the strategies which were proved successful, both in time efficiency and classification performance.

In order to properly evaluate the algorithm most time-expensive steps, the code was divided in parts where some of the features are computed, and mean times for each code section were obtained. The reason for this approach is that many of the features are obtained by the same calculations. For example, Frangi Vesselness Filter returns 4 different features; *likelihood\_FVF* and *number\_FVF* are both obtained precisely after the removal of candidate regions considered to be false detections, meaning that considering each individual feature processing time was not the best strategy.

The eight sectioned parts of code are the following: *Pre-processing*, *Texture*, *CHT*, *LMS*, *FVF*, *Intensity Features*, *Spatial Distribution* and *Classification*. Figure 3.1 represents how the mean processing time is distributed in these code sections.

Figure 3.1 represents the time percentage associated with each of these code sections. It can be concluded that *CHT* is the most time expensive part of the algorithm with

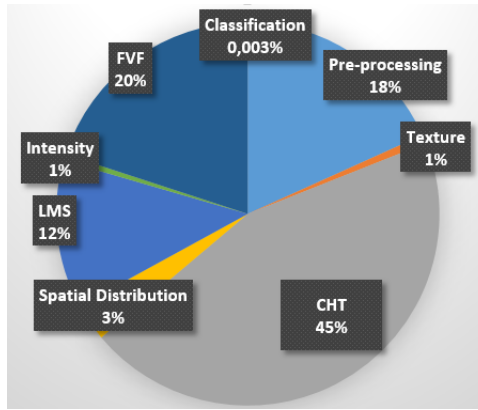


Figure 3.1: Initial algorithm mean processing time, per section.

45% of the mean processing time, followed by *FVF*, *Pre-processing*, and *LMS* with 20%, 18%, and 12% respectively. It is worth noting that these four sections represent 95% of the processing time and that the remaining four sections are neglectable at this point. *Pre-processing* was the first code section to be optimized, and for a particular reason: all the steps of *pre-processing* affect the performance of the segmentation algorithms and of feature extraction. However, *CHT* is acknowledged as the primary part of the algorithm to be optimized.

Throughout this chapter, after each modification introduced in each section, if a modification used different concepts from the ones used in the previous version, a classification performance evaluation of the algorithm was obtained on the testing images used in [7]. This was accomplished using the classification algorithm to obtain the best performance in [7], a DT classifier using eight features. In other words, everytime a major modification was introduced, the classification resultant from the respective optimization was compared with the reference classification in order to understand wether the applied modification affected the results or not.

### 3.1 Pre-processing Optimization

#### Cropping

The algorithm that is the foundation of this thesis had an extensive share of *pre-processing*, as stated. The first part of it was a resizing.

While this step was maintained, a further cropping of the image was considered and implemented. The reason behind this choice, in addition to the computational efficiency of a smaller image, was the unnecessary time-consuming processing of the dark peripheral regions surrounding the ROI. Furthermore, a great variance of the size of this "black

border" is noticeable in the testing datasets (See Appendix A.14f) and due to this, aiming for adaptiveness, the unnecessary borders were cropped.

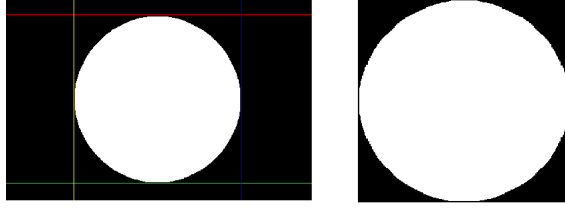


Figure 3.2: Cropping Example.

The cropping boundaries are achieved searching through the middle vertical and middle horizontal vectors of the ROI image for the first pixel with value "1" (or belonging to the ROI), in ascendant and descendant directions. This way, four quantities that represent the unnecessary to process dark regions are obtained. Margins of 1% of the total number of rows are subtracted to the vertical quantities and of 1% of the total number of columns is subtracted to the horizontal quantities, each of them rounded to the next integer.

This margin is considered since many kernel based operations are performed in the image and due to a number of 2D convolutions being performed using by the *conv2* Matlab function, which applies the Zero Padding principle [52], visual artifacts appeared in periphery of the images, interfering with the segmentation algorithms and, for example, Texture-based Features.

This was the first major enhancement to the algorithm and due to it the algorithm had its time performance increased substantially, particularly in the datasets with small ROIs and consequently fewer quantities of useful information.

### Scaling Constant

Adding robustness and versatility is desired for such an algorithm, aiming for the most automated processing possible. For this reason, many of the previous algorithm operations and kernel sizes related to pre-processing and feature extraction were scaled through the introduction of a *scaling constant*. This *constant* is given by,

$$scaling\_constant = \frac{ROI\_d - OD\_d}{480 - 110} = \frac{ROI\_d - OD\_d}{370} \quad (3.1)$$

where  $ROI\_d$  and  $OD\_d$  are the ROI diameter and the OD diameter, respectively.

This constant resulted from empirical observations on the training dataset used in both [7] and this work, in which the mean value of the ROI diameter was found to be close to

480 pixels and the mean value for the OD diameter was about 110 pixels.

After experimenting modifications to this constant it was verified to be affecting the overall classification results negatively. The inclusion of the  $ROI\_d$  and the factor of 480 could explain this effect. Including the ROI size of the processed image in this constant removes adaptiveness to the Field of View (FOV) factor since for example two images with the same  $ROI\_d$  can be achieved with cameras presenting different FOVs, resulting in a considerable difference in the amount of information presented to the algorithm.

Moreover, as it can be exemplified by images in Appendix A, many images do not present a circular shape. For that reason variants of the constant were experimented. These included factors like the ROI's area, a ratio between the ROI's area and the OD's area, the OD's area, and all the mean value of these factors measured on the training dataset.

The chosen alternative to the first constant was the diameter of the OD divided by the mean OD of the dataset later used for training(110 pixels):

$$scaling\_constant\_2 = \frac{OD\_d}{110} \quad (3.2)$$

In addition, all measures obtained from the image during feature extraction (as laser-marks sizes and distances between them) are normalized to the size of the OD. The new constant and kernel scaling are expressed in table B.1 of Appendix A.

This modification was made since it did not affect the overall processing time of the algorithm while resulting in an improvement of the final classification accuracy.

## Image Enhancement

The 18% mean time of pre-processing is mainly related to two different image enhancements: *uneven illumination correction* and *contrast enhancement*. Although other improvements were made, these corrections are the only part of pre-processing (in conjunction with the applied cropping) in which any modification directly affects the final pre-processed image, and for this reason they were addressed at this point.

Uneven illumination and poor contrast are normal problems in digital fundus images, and reasons for these occurrences include several technical and instrumentation specific characteristics. Some of these include: visual artifacts caused by inefficient aiming of illumination, the presence of impurities on the camera lens, focus-associated blurring, or

even the presence of lasermarks or bright lesions.

This lack of proper illumination and/or contrast is often enough to introduce misbehaviours in some algorithms or, in this case, impair the proper segmentation of lasermarks and of anatomical structures.

Aiming to reduce the complexity of the former method, several state-of-the-art illumination enhancement techniques [53–55] were considered, namely Gamma Correction of the Value/Brightness channel in HSV colorspace [56], Top-Hat filtering [57], Mahalanobis Distance [54] and Homomorphic Filtering [55].

In the initial algorithm, the image was converted to  $L^*a^*b$  colorspace [58] where its Luminance Channel( $L^*$ ) color component was adapted to normalize the illumination of the final image. Since Luminance values represent the brightness of the corresponding pixel, a mean luminance value map, normalized to  $[0,1]$  was obtained, after what the final  $L^*$  color component had its brightest areas luminance reduced and its darkest areas luminance increased. After this process, Adaptive Histogram equalization was applied to this  $L^*$  corrected channel enhancing contrast. Finally  $L^*$  replaces the initial  $L$  channel in the  $L^*a^*b$  colorspace image, which is then converted back to the RGB colorspace, resulting in a contrast and illumination enhanced image 3.3.

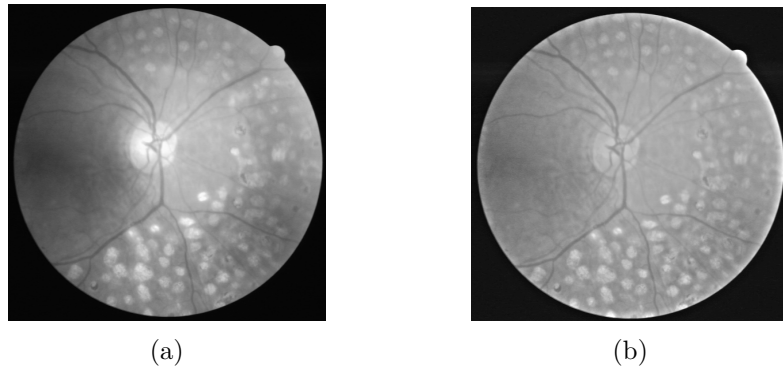


Figure 3.3:  $L^*$  color component before (a) and after (b) uneven illumination correction and contrast enhancement.

This image enhancement is exemplified in Figure 3.4a. This method presented better results than any of the mentioned alternatives in conjunction with the segmentation algorithms, being confirmed as the best approach for this specific method. Even though it must be pointed out that Gamma Correction of the Value channel in HSV colorspace was the method to better approximate this previous approach.

It is worth noting that, in the majority of Digital Fundus Image processing problems, Green Channel (Figure 3.4b) is state-of-the-art acknowledged as the channel to generally

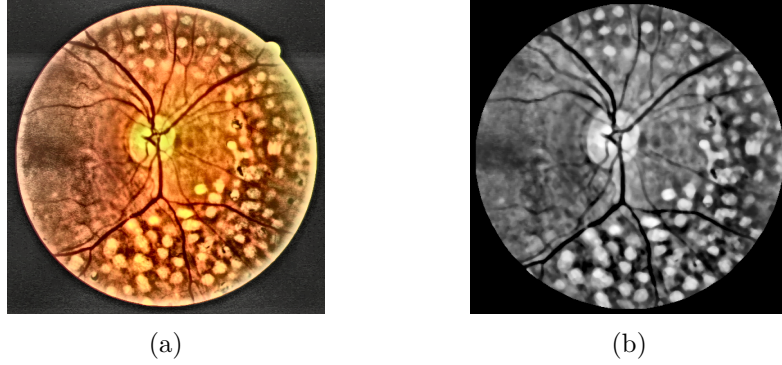


Figure 3.4: Image after pre-processing step - RGB (a) and Green Channel (b).

present a better contrast [14, 53, 59, 60]. For this reason, additional processing is only performed in this channel. CHT was the next code section to be optimized, for the general impact it has on the algorithm’s mean time performance.

### 3.2 Circular Hough Transform Optimization

CHT detects circles in the preprocessed green channel of the image. While the edge detector present in CHT, usually Canny detector, removes the noise by using a Gaussian filter, this type of filtering weakens important edges, decreasing true detections and possibly increasing false detections. An alternative approach was achieved by Jie and Ning’s in [61] and reproduced by [7] that consisted in applying a bilateral filter (**BF**) to the image and then using Otsu’s method for thresholding [24]. BF does perform a suitable smoothing for this problem, however, it totals a computation time as high as 14 seconds for some images. This is because Bilateral Filter computes an additional range kernel for restricting the neighborhood pixels’ averaging, resulting in a nonlinear and computationally intensive averaging process.

For this reason an alternative filter was the first idea to be tested. BF proved very effective for this application in [7] because of its smoothing while preserving edges property. Searching for methods with similar properties [62], three options were considered: Anisotropic Diffusion [63], Guided Filter [64] and Shiftable Bilateral Filter (SBF)[65]. Anisotropic Diffusion is an iterative method that was proved slower than BF without the introduction of improvements. Both Guided Filter and SBF are much faster than Bilateral Filter (both about 30 times faster for an example resolution of 584x768) and were chosen to be tested, replacing the former BF. These steps required extensive testing since CHT is the best suited segmentation algorithm for the detection of lasermarks in this method (as pointed by Sousa *et al.*), which meant that a poorly implemented CHT could compromise

the whole algorithm. Adapting these two filters to this problem and choosing one of them was one the primary concerns of this thesis.

### 3.2.1 Guided Filter

Guided Filter [64] is an edge-preserving smoothing filter that computes its filtering output using a guidance image. The guidance image can be the image itself, a further processed version of the image, or a completely different image (of the same size as the primary input image). This filter has been used in applications like ASIC real-time video processing [66], haze removal [67] or Visual tracking [68]. Being a neighborhood operation in the processed image, this filter also takes into account statistics of the corresponding spatial window in the guidance image.

When the guidance image provided is the input image itself, the filter takes into account its edges and promotes *structure transference* while smoothing background. The authors state that Guided Filter produces outputs similar to Bilateral Filter but that it has a better performance near edges and, apparently, this was the primary problem associated with Guided Filter for the implemented method. What generally happens is that Guided Filter results in a much larger number of CHT circle detections (compared to BF), many of which are false detections, when applied with an adaptive regularization parameter  $\epsilon$  (based on the difference between maximum and minimum gray levels of the image), a kernel similar to BF, and the CHT threshold of 0.225 used in the previous method (for the consideration of a circle in its normalized to [0,1] range accumulator array).

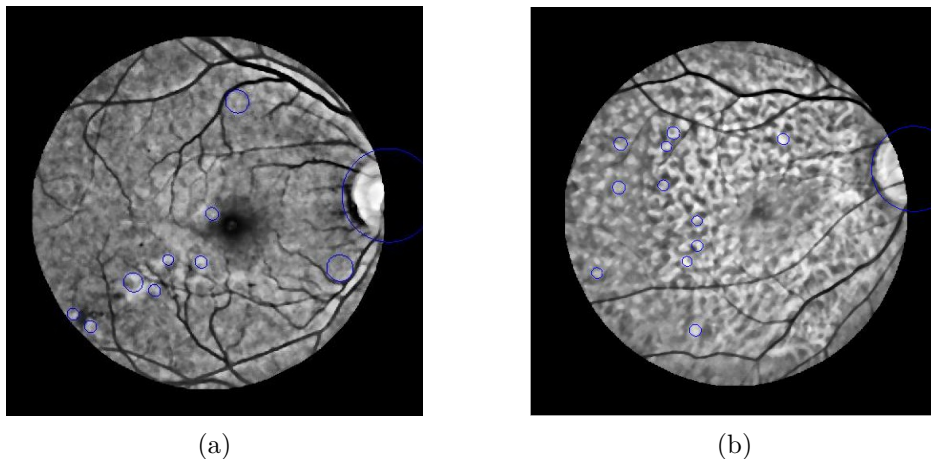


Figure 3.5: Circles detected by CHT with Guided Filter.

The threshold parameter of CHT was modified aiming to prevent this false detections problem, unsuccessfully. When this threshold was increased, a lower rate of false detections was verified but the true detection rate deteriorated in the same manner, aggravating the



final classification. On the other hand, when this threshold was increased, many circles were detected both in images that have them and in images that do not (Figure 3.5).

It was therefore concluded that, for this specific application, Guided Filter does not equals BF's ability of smoothing the image without compromising edges.

### 3.2.2 Shiftable Bilateral Filter

Shiftable Bilateral Filter (SBF) is a method proposed by Chaudhury *et al.* [65] that approximates BF using trigonometric kernels (raised cosines) as a substitute of the gaussian range kernel that restricts the performed averaging. Time efficient approaches aiming to approximate Bilateral Filter's output had already been proposed with polynomial kernels [69], but in [65] it has been observed that, for a fixed number of terms, the approximation quality with trigonometric kernels surpasses the one using polynomial kernels. Later, the same author proposed an improved and revised version of the Shiftable Bilateral Filter (SBF) in [70].

SBF presents a good approximation of BF combining the background smoothing while preserving edges property and a high computational efficiency. In the context of this work, and much like Guided Filter, a larger number of CHT circle detections was verified. However, for SBF, an adjustment of the CHT threshold was enough to obtain an efficient circle detection. Figure 3.6 presents a comparison for the same image between the CHT method output with the former BF and with SBF, respectively, aiming to compare the detections obtained with the initial CHT and CHT using SBF.

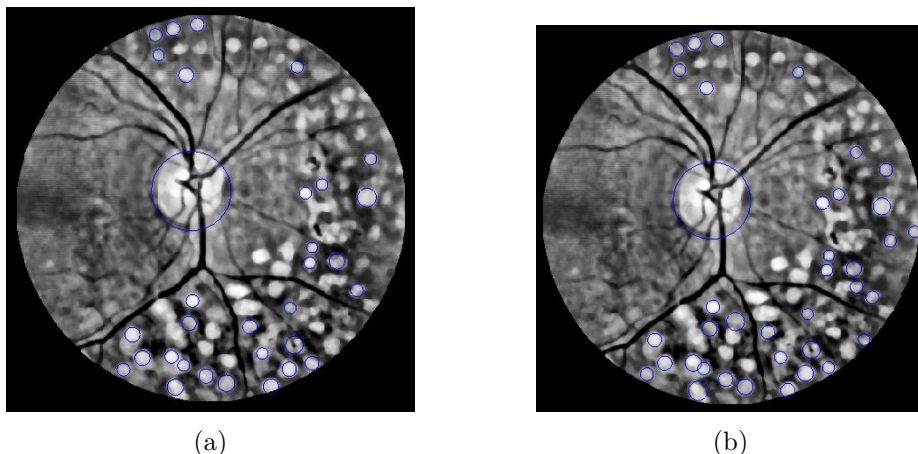


Figure 3.6: Circles detected by CHT with (a) BF and (b) SBF.

Detections obtained with the new method resemble the ones with the former one implemented in the original algorithm, a fact verified for the remaining test datasets. This resemblance was reached employing an adequate threshold for the consideration of

circles of 0.275, in opposition to the former 0.225 (in the normalized accumulator array).

This threshold increase contributed to reduce the rate of false detections of the previous method since, from this moment on, a greater number of votes is needed for a circle to be detected (subsection 2.3.3).

A range of radii had to be chosen in order to minimize false detection rate, promote versatility, and minimize the circle radii range to be searched by the method. To do this, the new optimized CHT was applied to the training dataset with a large range of radii. For each image, the diameters of the detected circles were stored in a vector, and for each diameter value, a ratio between it and the diameter of the OD was calculated. The results are presented in the histogram of Figure 3.7.

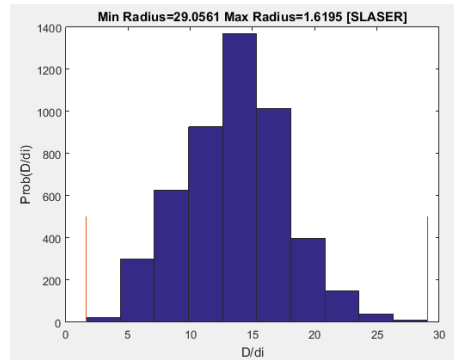


Figure 3.7: Ratio( $D/d_i$ ) for all the circles detected in the dataset NOMASCR.

In this histogram, the vertical red and blue lines indicate the maximum and minimum radius value detected, respectively. It can be assessed from the graphic that if a range in  $D/d_i$  of  $[4,23]$  is chosen, a minor error is committed. This range defined a CHT radii search range of,  $\left[\frac{D}{23}, \frac{D}{4}\right]$ , where  $D$  is the Diameter of the of the optic disc.

For CHT and the remaining segmentation algorithms one of the rejection criteria for the invalidation of lasermarks was their location. In the work developed by Sousa *et al.* [7], if a detected circle lies in the vicinity of the Vascular Network or of the OD, it is ignored.

In this work it was verified that computing the location of the Fovea did not represent a significant aggravation of the computational efficiency, and for this reason, Fovea's location was included as a third rejection criteria for the three segmentation algorithms, contributing for the rejection of more false detections. If the distance between the center of a detected circle to the center of the Fovea is less than half the diameter of the OD, it is discarded.

### 3.2.3 LMS and FVF Optimization

As explained in subsection 2.3, FVF detects tubular and blob-like structures and LMS detects regions based on a morphologically transformed image  $I_M$ . After experimenting the methods, the concepts behind each of them were not modified although their computation times were optimized significantly.

Both these methods, in addition to morphological operations, image reconstruction, and the application of multi-scale Frangi Filter 2D (in the case of FVF) [25], make use of a Matlab function from the image processing toolbox called *regionprops*, a useful measurement tool for binary images. However, some of the measures performed by *regionprops* require a significantly computational complexity, the majority of which are used by FVF and LMS.

FVF consists on the application of a two-dimensional Frangi filtering to the image for the detection of blob structures in the image. This filter is explained in section 2.3. One of the options for its optimization could be to reduce the number of individual values of  $\sigma$  of the original algorithm  $\sigma = \{1, 3, 5, 7, 9\}$ , which contribute similarly for the total mean time. For each  $\sigma$ , the algorithm performs an iteration and computes a vesselness measure  $V_\sigma$  for each pixel. The final filter output per pixel is based on the maximum vesselness measure obtained for each of the computed scales  $\sigma$  (subsection 2.3.2). For this reason, the mean contribution of each individual  $\sigma$  was measured, in the training dataset. The minimum mean contribution was presented by  $\sigma = 1$  with 5.96%. The algorithm was tested using only the most relevant sigma scales,  $\sigma = \{3, 5, 7, 9\}$ . While the visual difference of the generated filtered images is minimal, the method classification's performance was affected negatively.

As the method was not able to be optimized by a fewer-calculations tradeoff, an alternative to FVF was considered: a similar multi-scale vesselness filter algorithm proposed by Jerman *et al.* [71]. The author proposes a new vessel enhancement filter which presents notable results and when compared to other four state-of-the-art enhancement filters.

Images (b) and (c) of Figure 3.8 show the best set of parameters which led to the most efficient detection for each of the two methods when performed on image (a).

FVF, for this specific task, revealed itself more efficient in properly segmenting regions that truly are lasermarks in a time-efficient fashion, and for this reason it is confirmed as the state-of-the-art best segmentation algorithm for this optimization purpose. Jerman's method seems more accurate presenting better definition and detail of marks, however, it presents false detections in some "No Laser" datasets' images and was not considered due

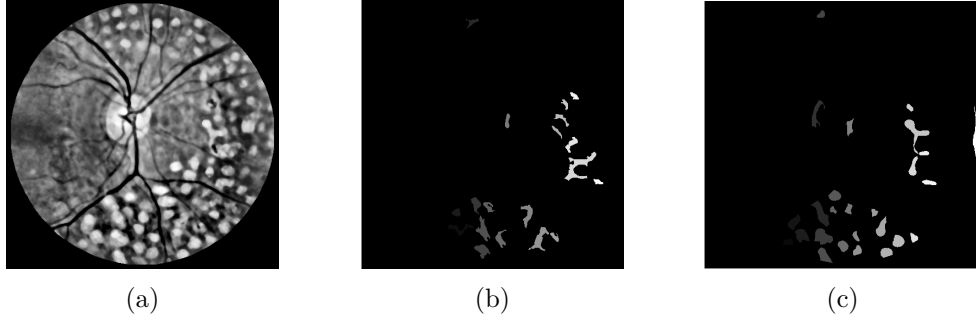


Figure 3.8: (a) Test Image (b) Frangi and (c) Jerman vesselness filters obtained segmentation.

its lack of accuracy, for this specific method. It needs, unlike FVF, a smaller amount of sigma scales ( $\sigma = \{1, 3, 5\}$ ) and provides very detailed segmentations. Comparing only the filters computations FVF is faster since, in order to obtain final regions exemplified in image (c) of Figure 3.8, additional processing steps of the filter's output were required and must be summed up; these include the removal of a circular shape artifact surrounding the image (through the computation of a secondary ROI) or median filtering of the output. Since the interest lies in optimizing the overall time efficiency of the algorithm, FVF was maintained.

The selection and evaluation of blobs structures of FVF is made by using *regionprops*, specifically two different calls of this Matlab function. The first call measures the segmented candidate regions (after Otsu's method for thresholding is applied to the filtered image) in calculating four metrics: its *Centroid*, *Major Axis Length*, *Minor Axis Length*, and *Eccentricity*. After this, the *Eccentricity* (ECC) parameter is used to validate candidates removing each region that verifies  $ECC > 0.92$ . In a next step the Matlab function *bwselect*, which in a binary image keeps only the selected regions, was used to only maintain in the thresholded binary image the valid candidates. A second call of the computationally complex *regionprops* is made to compute once again the measures of the valid marks. ECC is later used to compute the likelihood of each mark to be a lasermark, and to weight the area of each mark resulting in two specific features (*likelihood\_FVF* and *weighted\_area\_FVF*).

These steps can be reduced to one call of the *regionprops* function saving all the information of each mark in a matrix and removing all considered-to-be false detections and selecting them with *bwselect* for the obtention of the final candidates binary mask. Later the candidate lasermarks are only considered to be valid if not located in the neighborhoods of the vascular network, of the OD or of the fovea.

The LMS algorithm was reproduced from [23] in [7] and presented seven different calls of the already mentioned *regionprops* function. Manipulating vectors, an approach was obtained using one call of the function, which resulted in significant decrease of the computational time. Furthermore, Roundedness and Solidity, described below, were calculated for all the marks, when several of these values were later discarded.

After the call of *regionprops*, candidates with an area lower than 100 pixels are considered too small for representing lasermarks and are removed. In addition, and much like the remaining segmentation methods, candidate lasermarks are considered invalid if located in the neighborhoods of the vascular network, of the OD or of the fovea.

Finally, and only for each validated segmented region, *Roundedness* ( $f_1$ ) and *Solidity* ( $f_2$ ), two of the features established in [23], are calculated. These are defined as:

*Roundedness* is defined as the ratio of the major by minor axis length of the region - since most lasermarks have a circular form this measure defines how circular the segmented region is.

*Solidity* is the ratio of the candidate's region area by its convex hull area (the area enclosing the region) - laser marks are convex structures having high solidity, close to unit.

If these parameters verify:  $1 < f_1 < 2.5$  and  $f_2 > 0.8$  the candidate is kept. If one of these conditions is not verified, the candidate is discarded.

After only the relevant candidates are retained, two new classification features that were introduced are calculated: **roundedness\_LMS** and **solidity\_LMS**, which respectively represent the mean Roundedness and the mean Solidity of the valid lasermark candidates detected by LMS. This additional processing does not represent a significant increase of the computational time.

Comparatively, LMS is the method that presents more false detections, which was the main reason for the inclusion of these two new features as it seemed imperative to include measures which evaluate the detected marks likelihood of being in fact lasermarks and LMS's performance in the data mining process. CHT and FVF already computed two geometry-related features and one likelihood-related features regarding detected marks.

### 3.3 Code-based Optimizations

The algorithm to be optimized was written in Matlab, which means several notions must be taken into account in order to perform a conscient optimization regarding all possible resources this language offers.

The optimizations referred in this section did not solve code bottlenecks but rather worked as an overall optimization of the entire algorithm, yet they represent a considerable optimization contribution. While their main objective is to reduce computational time, they do not interfere with the algorithms final result; they only introduce differences in the approach of solving mathematical and programming operations.

Matlab is an interpreted language. In other words, Matlab code does not need to be explicitly compiled. As a consequence, Matlab code is (to some extent) interpreted line by line, executing the commands it is instructed to, which means spending time reading each line. This is the reason behind Matlab's interpreter inferior speed when compared to pre-compiled languages such as C/C++ or Fortran.

However, Matlab has precompiled functions that compute basic matrix/vector operations, meaning Matlab is able to equal and even surpass other compiled languages in what concerns matricial operations. This section explains the majority of the code-based optimization steps that were made.









Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">extract_features_laser_final</a>	1	5.469 s	0.044 s	
<a href="#">LMS</a>	1	2.885 s	0.017 s	
<a href="#">regionprops</a>	8	2.699 s	0.016 s	
<a href="#">regionprops&gt;ComputeConvexArea</a>	1	2.353 s	0.019 s	
<a href="#">regionprops&gt;ComputeConvexImage</a>	1	2.334 s	0.048 s	
<a href="#">roipoly</a>	1643	1.294 s	0.076 s	
<a href="#">colorspaces\private\applyformsequence</a>	4	1.136 s	0.000 s	
<a href="#">regionprops&gt;ComputeConvexHull</a>	1	0.992 s	0.065 s	
<a href="#">poly2mask</a>	1643	0.952 s	0.146 s	

Figure 3.9: Matlab profiler's example report (before LMS optimization).

In order to quantify the processing time of each considered code section, the performance measure functions *tic/toc* were used to accomplish precise measurements. But to discover which functions were contributing most to the high computational mean time of the algorithm, Matlab's Profiler tool was used. This tool records functions' execution time and was of great utility in uncovering the most time consuming tasks and where to apply a greater optimization effort.

Figure 3.9 is the report of Matlab's profiler before the last optimizations of section 3.2.3 were implemented. Note that running the profiler increases computational time significantly. From this report it can be noted that inside the main extraction function, *extract\_features\_laser\_final*, almost 5,5 seconds are spent. LMS is one of the most time-

expensive functions inside this main function. However, almost all the function's time is spent by *regionprops*, a native matlab function. *Regionprops* was being called as follows:

```
stats = regionprops(bw_LMS,'Centroid','MajorAxisLength',
'MinorAxisLength','Area','ConvexArea');
```

after what detected regions were evaluated. Then *bwselect* was used to choose only the valid candidate marks, based in the detected centroids, and then another *regionprops* call was made to calculate Major Axis Length, Minor Axis Length and Centroids, when these regions had already been segmented. The approach used instead was to successively delete the matrix storing the data segmented by *regionprops*. This procedure saved more than one second for each image as the profile call in Figure 3.10 shows.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">extract_features_laser_final</a>	1	4.721 s	0.067 s	
<a href="#">LMS</a>	1	1.633 s	0.017 s	
<a href="#">regionprops</a>	7	1.418 s	0.011 s	
<a href="#">colorspaces\private\applyformsequence</a>	4	1.233 s	0.001 s	
<a href="#">regionprops&gt;ComputeConvexArea</a>	1	1.197 s	0.009 s	
<a href="#">regionprops&gt;ComputeConvexImage</a>	1	1.188 s	0.020 s	

Figure 3.10: Matlab profiler's report after LMS optimization.

The first step taken was to analyse the presence of repeated operations, as removing them contributes to accelerate the algorithm and decreases the ammount of code to be optimized. As a result a great number of repeated operations were eliminated. The most relevant examples are the stated repetitive calls of *regionprops* in the calculations of both FVF and LMS and in the calculations of intensity features on the four extracted channels of the image. Cropping the image also contributed to eliminate repeated or unproductive operations, for example computing CHT, LMS and FVF in the outer regions of the image.

All code sections lacked pre-allocation, to some extent. **Pre-allocation**, as in any other programming language is defined as finding a place in memory large enough (or a *range of adresses*) to store the information of a variable (e.g. a matrix) in a continuous way.

Differently to other programming languages, Matlab performs its own pre-allocation. However, memory-related problems emerge from this mechanism as Matlab requires variables to be saved continuously in memory.

A practical example: when a vector with 100 elements is declared and stored in memory, but later 50 more elements are concatenated to it, if any of the 50 addresses imediately

next to the 100 initial ones is occupied, other contiguous and empty 150 addresses have to be found where the new vector of 150 elements will be stored. Therefore, if Pre-allocation is not performed by the programmer, the processor will likely search the system memory for new space to store variable everytime a variable is grown. For this reason, pre-allocating promotes memory and time efficiency. In this example, if the final vector size was previously known, 150 elements could have been allocated in the first declaration of the variable, guaranteeing that the processor can quickly access the whole variable continuously, hence efficiently.

```

if isempty(centers_phd)==0
    for c=1:size(centers_phd,1)
        if (roundedness(c,1)>1 && roundedness(c,1)<2.5...
            && solidity(c,1)>0.8 && A(c,1)>constant_area_LMS)
            centers_auxiliar=[centers_auxiliar;centers_phd(c,:)];
        end;
    end;
end;

```

Figure 3.11: Matrix *centers\_auxiliar* is growing inside a loop.

A bad practice in Matlab, related to the pre-allocation topic, is allowing a variable to grow inside a loop (Figure 3.11). This is perhaps one of the worst pre-allocation practices in Matlab, since it implies the search in every iteration for a new space in memory for new memory contiguous addresses. This is increasingly inefficient the greater the matrix to be stored. Whenever a loop cannot be avoided, a considerable (predicted and protected) amount of space can be allocated to a matrix variable, and the number of iterations inside the loop is stored. When the last iteration is made, the matrix or vector can simply be truncated in a number of elements equal to the number of iterations, avoiding the memory-expensive growth inside a loop as Figure 3.12 exemplifies.

```

aux_count=0;
if isempty(centers_phd)==0
    for c=1:size(centers_phd,1)
        if (roundedness(c,1)>1 && roundedness(c,1)<2.5...
            && solidity(c,1)>0.8 && A(c,1)>constant_area_LMS)
            aux_count=aux_count+1;
            centers_auxiliar(aux_count,1:2)=centers_phd(c,1:2);
        end;
    end;
end;
centers_auxiliar=centers_auxiliar(1:aux_count,:);

```

Figure 3.12: Variable pre-allocated and later truncated.



Understanding this necessity, the variables used in each code section were pre-allocated, either with a size equal to the size they would present in final calculations or with a considerable and protected size and later truncated. A total of 51 growing inside loops matrices were pre-allocated throughout the initial algorithm.

Functions are an essential part of programming, allowing code to be organized and structured, combining a grow in complexity and readability of code. Therefore, sectioning code into functions may help increase the overall execution of code, as they handle variables internally returning and keeping in memory only those that will be further manipulated or presented in the final return of a main function. This is of great relevance specifically in Matlab, as there is the necessity of using multiple variables since Matlab is more efficient in creating new variables in opposition to typecasting. As a consequence, creating more variables is normally more efficient, but keeps them in memory and functions allow unused variables to be discarded, promoting cache and general programming efficiency.

In order to take advantage of function handling, the code was sectioned into eight functions, resembling the code sections mentioned in the beginning of this chapter (Figure 3.1 presents these functions). From this point, the main algorithm's function receives an image which inputs the pre-processing function, after six additional functions compute the returning features, which in turn are input to the eight and final function that classifies the image as "Laser" or "No Laser".

This way, fewer variables are stored in memory while the extraction is being made, resulting a cleaner, faster and easier to understand and optimize code. Table 3.1 summarizes the operations performed by each function.

Table 3.1: Algorithm's functions and their operations, by computing order.

Function Name	Performed Operations
Pre-Processing	Obtains ROI, Vascular Network, OD, Fovea, Constants. Crops the image. Corrects brightness and contrast. Extracts channels Green, Red, Hue and Saturation.
CHT	Finds circles and circular structures in the image; Applies rejection criteria to invalidate false detections.
FVF	Finds blobs structures in the image; Applies rejection criteria to invalidate false detections.
LMS	Finds structures resembling lasermarks; Applies rejection criteria to invalidate false detections.
Intensity	Computes Intensity-based features considering the candidate lasermarks detected in the segmentation algorithms for the four considered channels.
Spatial Distribution	Computes Spatial Distribution features considering the candidate lasermarks detected in the segmentation algorithms.
Texture	Computes Texture features in the image's green channel pixels inside the ROI.
Classification	Receives the final features and classifies the image as "Laser" or "No Laser".

In Matlab it is also possible to declare secondary functions inside primary functions. If a function is only used by any other particular function, it can be declared inside this last one, granting that when the first function is called, Matlab already knows where to find the inside-declared function. This work took advantage of this fact. Examples, of this application are: in CHT *chaccum* function, which computes the accumulator array of the transform; *MaxFilter*, that computes the T parameter of SBF; *Hessian2D*, that obtains the Hessian matrix of the image for further processing by FVF, and all the feature-calculating functions, which are secondary functions of the main extraction function.

As mentioned, the *Matrix Laboratory* code is mostly interpreted line-by-line resulting in control flow loops, as *for* and *while*, being inefficient practices in this programming language. One way of avoiding loops is by using *Vectorization*.

```

for iii=1:size(centroids,1)
    if ec(iii,1)<0.92
        c_auxiliar_blob=[c_auxiliar_blob;centroids(iii,:)];
    end;
end;

```

---

```

if ~isempty(centroids)
    ec_aux=ec<0.92;
    c_auxiliar_blob_1=nonzeros(centroids(:,1).*ec_aux);
    c_auxiliar_blob_2=nonzeros(centroids(:,2).*ec_aux);
    c_auxiliar_blob=[c_auxiliar_blob_1 c_auxiliar_blob_2];

else
    c_auxiliar_blob=[];
end

```

Figure 3.13: Vectorization example.

A vectorized computation is one that takes advantage of vector operations. Vectorization is one of the most general and effective techniques to write efficient Matlab code, however, it is not always an easy task and many approaches to vectorization and ways of programming it exist, hence vectorization is not universal, being highly dependent of each particular situation. An example is shown in Figure 3.13 where the *for* cycle can be easily avoided. Only one condition is verified to enter the cycle which makes vectorizing it a simple task by using general vector logic. The vectorized code is faster than the one containing a for loop even when *centroids* variable has a single pair of elements, the x and y coordinates of a centroid. In this particular case, the efficiency improvement grows linearly the more values *centroids* has. As a comparison example, if *centroids* is a 200x2

matrix,

**Loop:** *Elapsed time is 0.001586 seconds.*

**Vectorized:** *Elapsed time is 0.000058 seconds.*

or approximately 27 times faster. If centroids is a 2000x2 matrix,

**Loop:** *Elapsed time is 0.030793 seconds.*

**Vectorized:** *Elapsed time is 0.000113 seconds.*

or approximately 273 times faster.

A good understanding of referencing also strengthens good practices for Matlab programmers, allowing a broader range of vectorizing options. For example, a 5x5 matrix A is stored in memory linearly, as a one dimension 25-element array, in column-major order. An index is the reference to to an element's position in that one-dimensional array, such as A(13). A(13) is the exact same value as A(3,3), but it is faster to scan down columns than down rows. This is because column-major order means that elements along a column are sequential in memory, while elements along a row are further apart. This referencing awareness is even more important the larger the matrices to be computed are.

Helpful Vectorization Matlab functions in this work were *min*, *max*, *mat2gray*, *logical*, *reshape*, *repmat*, *meshgrid*, *cumprod*, *accumarray*, *norm*, *ones*, *zeros*, *nonzeros*, *sort*, *find*, among others. Different functions were also tried in order to improve code execution time. *logical* is a more efficient function in the use of memory than *bwlabel* for dealing with binary images. *strtok* is useful for dealing with input names but required variables growing inside loops, so *fileparts* was used instead.

When optimization and the use of another functions is concerned, *mex* files have to be referred. Binary *mex* files, or Matlab Executable files, are dynamically linked subroutines that the Matlab interpreter loads and executes like a native Matlab function. In other words, it is a way of calling routines of other pre-compiled languages like C/C++ or Fortran. In latest Matlab versions mex files are not recommended by Mathworks. However, in version Matlab 2008b *mex* files were considered. But since the majority of the code was vectorized or already took advantage of Matlab most powerful tools, mex files ended up not being used. At this point, the functions that presented a larger execution time and that could be called as mex files were *regionprops*, *medfilt2* and *applycform*, all of which are Matlab native functions. Nevertheless, *medfilt2* already calls a *mex* file in its algorithm form, for efficiency reasons.

### 3.4 Feature Selection and Binary Classification

The last optimization to be made relies on a feature selection step for the training of binary classifiers, since the classification models that will be chosen will receive as input only some of the 67 features computed.

The chosen training dataset NOMASCR, has a total of 855 images, of which 203 are "Laser" images and 652 are "No Laser" images. These 855 images result of merging datasets NOMA (233 images) and S (203 "Laser" images and 419 "No Laser" images). This was the chosen training dataset in order to allow classification and time-efficiency comparisons with the previous method.

WEKA, a machine learning algorithm package written in Java, was the primary tool for data exploration, providing an overview of all the data per instance, allowing an efficient analysis of features and of their utility in separating classes. In this work three binary classifiers were considered: Decision Tree (**DT**) [38], Alternating Decision Tree(**ADT**) [39], and Support Vector Machines(**SVM**) [40].

To obtain subsets of features for each of the chosen classifiers, two types of feature selection methods were used - Filter methods and Wrapper methods. Filter methods feature selection is important in several classification settings. Filter methods analyze intrinsic data properties, independently of the used classifier. By ordering these features by their measure of importance with a filter method, a threshold can be applied removing less important features. In these methods usually the potential interactions among elements of the joint set of features are neglected. For this reason, sometimes filter methods result in redundant features.

Wrapper methods consider a specific classifier and evaluate subsets of variables while taking into account, unlike Filter Methods, the possible interactions between features, which is translated in choosing from all possible subsets of features, therefore, higher computational power is required. Wrapper Methods also present limitations, like the increasing overfitting risk when the number of instances is insufficient or the significant computation time given a large feature set.

Embedded Methods are a coupling of both the above described methods. They aim to combine the advantages of both Filters and Wrappers and minimize their limitations [72].

The feature selection procedure was to filter data with two Filter Methods - Information Gain(IG) and Gain Ratio(GR) - obtaining two initial subsets (Appendix D) after what Wrapper methods for each classifier were applied to the IG and GR filtered subsets, resulting in six final subsets of features later used to train the classifiers.

Two initial subsets were obtained by applying filter methods (IG and GR) to the training dataset, thresholded by  $score \geq 0.1$ , resulting in two different feature subsets, one for each filter.

Wrapper methods were applied to these two subsets, one for each of the used classification algorithms, resulting in 6 more feature subsets which were used to train classifiers. WEKA was set to perform inner stratified 5-fold cross-validation on each of these generated training datasets.

The following Chapter describes results of the feature selection approach, the obtained results of the test datasets' classification with each trained classifier followed by a validation in three other datasets, and finally the efficiency of the performed optimization is compared to the former method.



## Chapter 4

# Experimental Results

In this chapter, the obtained experimental results will be presented and discussed. These results are subdivided into Feature Selection, Classification, and Optimization reflecting all subjects this work addressed. In addition, Appendix C refers general mean time processing divisions between algorithms' functions and the classification obtained for the best classifier obtained in [7], after each major optimization was performed. A brief summary of the image classification features used in this work is presented in Figure B.2 of Appendix B.

### 4.1 Feature Selection Results

The IG-filtered subset contains 43 features and the GR-filtered subset contains 44 features (Appendix D). To these, a subset evaluator wrapper method was applied for each classifier.

The features of the best subsets found for each classifier on the IG-filtered features are the following:

<b>Best Subset: IG-DT</b>	<b>Best Subset: IG-ADT</b>	<b>Best Subset: IG-SVM</b>
number_CHT uniformity entropy trace max_eigen autocorrelation mean_laser_red mean_laser_sat	number_CHT area_CHT likelihood_CHT autocorrelation energy sum_var radius_mean mean_laser_sat var_laser_red variance_var_green variance_var_hue	number_CHT weighted_area_CHT skewness entropy number_LMS hull moran_i moran_null_hipot autocorrelation variance_var_green variance_var_hue mean_laser_hue
<b>Accuracy: 93.0%</b>	<b>Accuracy: 93.7%</b>	<b>Accuracy: 93.6%</b>

Figure 4.1: Best subsets obtained with the IG-filtered features.

The features of the best subsets found for each classifier on the GR-filtered features are the following:

<b>Best Subset: GR-DT</b>	<b>Best Subset: GR-ADT</b>	<b>Best Subset: GR-SVM</b>
number_CHT area_FVF cluster_prominence sum_avg mean_laser_red var_laser_sat variance_var_green	number_CHT area_CHT likelihood_CHT autocorrelation energy sum_var radius_mean mean_laser_sat var_laser_red variance_var_green variance_var_hue	number_CHT skewness solidity_LMS hull_area moran_i moran_null_hipot roundedness_LMS sum_avg mean_laser_green max_laser_green mean_laser_sat
<b>Accuracy: 94.0%</b>	<b>Accuracy: 93.7%</b>	<b>Accuracy: 93.7%</b>

Figure 4.2: Best subsets obtained with the GR-filtered features.

## 4.2 Classification Results

In this section, the final classification results obtained are presented. The six feature subsets presented in the previous section were used to train the respective binary classifier.

Since both ADT-based wrappers returned the same features, five different classifications were obtained. Stratified 5-fold cross validation is performed on the training dataset (NOMAS) after what the classifier is tested with the remaining datasets. Some statistical measures accompany each classification.

In addition to what has been introduced in subsection 2.2,

- $T = TP + TN$ , or the correct classifications;
- $F = FN + FP$ , or the incorrect classifications;

To avoid redundancy only GR\_SVM, the classifier with the best performance, will be presented in this chapter. The remaining classifiers' results can be consulted in Appendix E.

After a stratified 5-fold cross validation using an SVM classifier with the features GR\_SVM on the images of the training dataset, the following results were obtained

<b>GR_SVM</b>	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
NOMASCR	160	641	801	11	43	54	98,31	78,82	93,68

The trained classifier resulted in a SVM model which WEKA needed 0.28 seconds to build. By testing this classifier in all the test datasets, that total 1749 images (135 "Laser"



and 1614 "No Laser"), the following results were obtained:

Table 4.1: **GR\_SVM** classifier tested on test datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1181	1181	6	0	6	99,49	-	99,49
MA	0	148	148	0	0	0	100,00	-	100,00
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	35	35	0	0	0	100,00	-	100,00
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	100	100	0	0	0	100,00	-	100,00
FAZD	0	54	54	6	0	6	90,00	-	90,00
BAT	22	14	36	1	12	13	93,33	64,71	73,47
PDJD	99	0	99	0	2	2	-	98,02	98,02
Total	121	1601	1722	13	14	27	99,19	89,63	98,46
<b>Specificity: 99.19%, Sensitivity: 89.63%, Accuracy: 98.46%</b>									

Additionally, datasets **SCR15**, **CDB**, and **HRF** were classified as a way of validating the optimized method.

Table 4.2: **GR\_SVM** classifier tested on validation datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
SCR15	86	0	86	0	13	13	-	86,87	86,87
HRF	0	43	43	2	0	2	95,56	-	95,56
Chasedb	0	28	28	0	0	0	100,00	-	100,00

### 4.3 Optimization Results

The results in the previous section and Appendix E present **GR\_SVM** as the training subset with the best classification, with the respective SVM classifier. For this reason, the optimized algorithm includes classification based on this model. As stated, the described algorithm has been optimized in order to obtain a satisfying time efficiency. Initially, the algorithm in [7] presented the time-efficiency results in Table 4.3:

With the optimizations performed, time efficiency results are presented in Table 4.4.

In order to quantify the contributions of the performed optimizations, the mean time for each code section (or each function) was obtained for both the initial and the optimized algorithm using the same datasets (Test Datasets - Table 2.4). These mean times are presented in table 4.5, where mean processing times are presented, namely the initial, final and resultant from each optimization step. The final column represents the final time-efficiency improvements of each code section.

These efficiency measurements translate into a mean decrease of time of 79,07% in which

Table 4.3: Initial algorithm processing time for each test dataset.

Dataset			Processing Time		
Name	N° Images	Mean Pixels	Seconds	Minutes	Mean/image(s)
MESSIDOR	1187	696320	24797,77	413,296	20,89
MA	148	699392	3755,84	62,597	25,38
EX	47	699392	1346,53	22,442	28,65
NOEX	35	699392	937,52	15,625	26,79
VBR	22	619008	458,74	7,646	20,85
50HP	100	619008	2309,62	38,494	23,10
FAZD	60	414720	1413,69	23,561	23,56
BAT	49	715037	1319,72	21,995	26,93
PDJD	101	680767	1687,80	28,130	16,71

Table 4.4: Final algorithm processing time for each test dataset.

Dataset			Processing Time		
Name	N° Images	Mean Pixels	Seconds	Minutes	Mean/image(s)
MESSIDOR	1187	696320	5245,81	87,430	4,42
MA	148	699392	721,16	12,019	4,87
EX	47	699392	234,08	3,901	4,98
NOEX	35	699392	168,86	2,814	4,82
VBR	22	619008	101,95	1,699	4,63
50HP	100	619008	561,90	9,365	5,62
FAZD	60	414720	263,73	4,396	4,40
BAT	49	715037,0079	325,18	5,420	6,64
PDJD	101	680767,3663	337,60	5,627	3,34

Table 4.5: Efficiency improvements quantified.

Function	Mean time (seconds)					Reduction(%)
	Initial	Cropping	CHT	FVF&LMS	Code-based	
Pre-processing	5,967	4,424	4,550	4,512	1,642	72,48
CHT	9,753	9,364	1,688	1,631	1,030	89,44
FVF	4,341	3,647	3,641	2,641	0,759	82,52
LMS	2,709	2,338	2,475	1,575	0,748	72,39
S. Distribution	0,694	0,415	0,519	0,446	0,294	57,64
Intensity	0,117	0,041	0,072	0,054	0,022	81,20
Texture	0,160	0,175	0,199	0,194	0,055	65,63
Classification	0,001	0,001	0,001	0,001	0,001	0,0
Total	21,741	19,405	12,144	10,052	4,550	79,07

several different resolutions, present in the test datasets, were considered (See Chapter 2.6).

## 4.4 Discussion

On a first and introductory analysis, it seems clear that the former method in [7] was improved both in classification and time efficiency performance by the introduction of alternatives in the majority of the algorithm processing steps.

Considering the performed feature selection, it is interesting to note that Alternating Decision Tree-oriented wrapper methods chose exactly the same features from two different sets of pre-filtered features. This could be explained by the training process of this classifier, in which boosting is taken into account. Boosting is based on an impurity measure that is computed in order to add new features to the classifier.

Another consideration is the fact that in [7] it was pointed that texture descriptors (which are a total of 27 different features) were not present in any of the wrapper subsets used for training classifiers. In the subsets presented in this work this is not the case. All the wrapper-resultant subsets present one or more Texture Features, and in fact the features **energy**, **entropy**, and **skewness** are present in more than one of these subsets.

Texture features, which were considered not so relevant for the detection of lasermarks in the previous algorithm, present an important contribution in this thesis' final results. These features were calculated using only the pixels contained in the ROI (in both works), which means that except for pre-processing changes, the image information remains the same in these two approaches.

It must be stated that Circular Hough Transform is the best segmentation algorithm of this method. The author of the previous method concluded this as well and after major changes in its processing, with a new filter, new restrictions and new parameters, this continues to be the case. The motivation for this statement relies on empirical observations and the fact that **number\_CHT** (the number of circles detected by the CHT method) is always choosed by wrappers. This is the case in the presented subsets and was also verified throughout the whole testing phase. The fact that the ADT classifier (both IG and GR filtered) kept 3 of the 4 CHT-related features also points in this direction as the impurity measure of ADT ranked these features high enough to be present in the final subset.

A curious occurrence in these results is the fact that the features that were introduced by this work (subsection 3.2.3), **roundedness\_LMS** and **solidity\_LMS**, were both

selected by the wrapper which originated the classifier that obtained the best performance; in fact it was the only wrapper method to chose them. A closer inspection of the GR and IG filtered subsets presented in Appendix D it can be noted that the IG filtered subset only included **solidity\_LMS** which was the third worst ranked feature by this measure. In the GR filtered subset both the features were kept and they rank last and third worst in this measure.

Considering the results presented in Appendix C, it can be first noted that classification results were affected as soon as the first pre-processing modifications were introduced. This is not surprising at all, since the trained classifier which presented the initial results would need to be retrained in order to achieve the optimal result. However, this classifier was only used as a reference to understand if the classification was being affected. It can be observed that classification accuracy did not vary in a great manner with all the optimizations introduced, which means information was not lost. This was also the case after the FVF and LMS optimizations and the following code-based optimization do not introduce any differences in classification results, as they were purely computational approach changes. After this last optimization step (See Figure C.5) pre-processing is now the most time expensive step with 36% of the mean processing time and CHT, which initially represented almost half of the processing time per image, occupies now only 23% of this total mean time per image. Classification was and still is the quickest function to be computed. The performed optimizations are concluded successful has the mean processing time was reduced in about 80%.

Regarding classification results, and as referred in 2.2.3, the previous method had the best classification performance with a C4.5 Decision Tree classifier that presented an Accuracy of 98,1%, a Specificity of 98,9% and a Sensibility 88,1%. Given this fact, Decision Tree was again considered in this work, in conjunction with another very similar tree-based binary classifier, Alternating Decision Tree, and Support Vector Machines.

This work reassured that Tree-based classifiers are a good approach for this specific problem and introduced Support Vector Machines to this specific classification problem improving the efficiency of the method in which the obtained performance of 98,46% Accuracy, 99,19% Specificity and 89,63% represent an improvement of the former method of 0.29 p.p. in Specificity, of 1.53 p.p. in Sensitivity and of 0.36 p.p. in total Accuracy.

Additional explanations should be made regarding the validation datasets classification (Validation Datasets - Table 2.5). The results presented in Table 4.2 show a total of 2 False positives obtained in the HRF dataset and in addition 13 images with lasermarks

failed to be detected. Considering the most concerning of these false classifications, the false positives, the HRF misclassified images were further inspected and are presented in Figure 4.3.

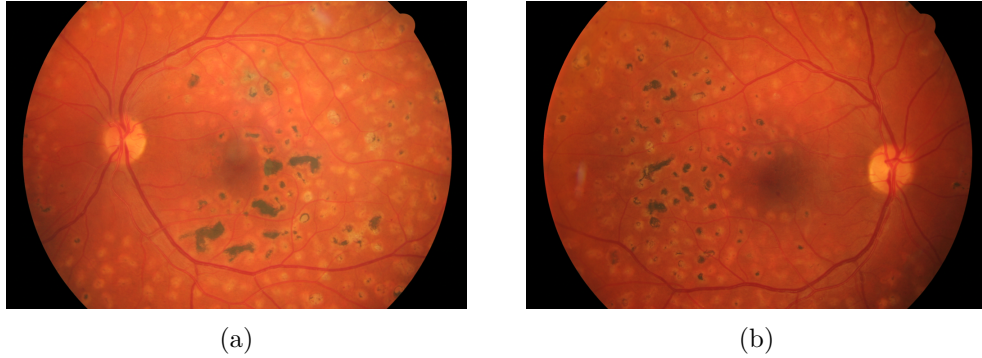


Figure 4.3: Misclassified images of HRF dataset.

In its official site, the dataset is referred as containing 15 images of healthy patients, 15 images of glaucomatous patients and 15 images of patients with Diabetic Retinopathy images, but no reference to the presence of lasermarks is made. By consulting image processing professionals at Retmarker, it was concluded these two images actually present photocoagulation lasermarks. It is interesting to verify a useful detection by the final algorithm, which detected lasermarks in images of a public dataset whose presence was not referred. This consideration means the classification results of the validation datasets must be re-examined, for a total of 73 images without lasermarks and 99 images with lasermarks. Of these 172 images, 13 images were misclassified, with zero false positives and 13 false negatives resulting in an accuracy of 92,44%, a specificity of 100% and a sensitivity of 86,87%. These results are similar to the ones obtained in the previous test datasets, except in total accuracy. However, comparatively, it must be noted that the validation datasets have almost as many lasermarks images as the test datasets (99/135, respectively) but that the number of no laser images is clearly unbalanced (73/1614, respectively). The test datasets present, in fact, a better reality approximation of DR incidence. In the conjunction of all tested images, with this correction, the following results were obtained:

Images	Accuracy
No Laser (1687 images)	99,23%
Laser (234 images)	88,46%
Total (1921 images)	97,92%

Observing time-efficiency results, a decrease of almost 80% of the mean computational time was obtained, where a wide variety of image resolutions is included. It can further

be noticed that the datasets which presented the greatest computational mean times were 50HP and BAT with 5,62 and 6,64 seconds per image, while PDJD presents the lowest mean time of 3,34 seconds per image. Both 50HP and BAT present the images with the greatest resolution of the datasets used in this work when compared to the ROI size of the images, or in other words when compared to the quantity of useful information. PDJD presents an the exact opposite example; its images present a small ROI which decreases its mean computational as less information is processed.

The improvements that were introduced in this work to the previous method, result from various modifications that were implemented or tried. These include:

- State-of-the-art segmentation algorithms being tested against other valid or proven methods as Guided filter or Adaptive Bilateral Filter, and other new methods like the vessel enhancement filter in [71];
- Pre-processing techniques were also a focus with Gamma Correction or Brightness Value Enhancement in HSV colorspace being good alternatives but that were unable to outperform the existent ones, which were kept. On the other hand, further cropping the images presented relevant time efficiency improvements;
- A modified constant for scaling calculations and measurements and to introduce further adaptiveness of kernel-based operations;
- The introduction of two new features (**roundedness\_LMS** and **solidity\_LMS**) related to one of the segmentation algorithms;
- The training of binary classifiers, introducing two different classifiers;
- The use of a binary Support Vector Machines classifier that presented the best classification results;
- Extensive data mining and a great number of tests were performed on the algorithm.

Guided Filter and its application requires further explanation. It is referred by its authors as having a better performance near edges. It was verified in this work that for this specific application it performs so well near edges that even weak edges are maintained extremely augmenting the false detection rate, affecting classification negatively.

Observing the Feature subsets obtained in the previous method, it was observed that while spacial and intensity descriptors were always present, the majority of chosen features were ones related to CHT, FVF and LMS. Since the results presented in this work are

on par with the best results presented in [7] and that only at most 3 out of 11 features in the subsets are related to the segmentation algorithm, it is safe to assume the current method extracts more information from images than the previous one, particularly in what concerns intensity-based descriptors (features 50 to 65). Even if CHT has been modified considerably, the principles of LMS and FVF did not suffer major changes. Since the same datasets and similar segmentation algorithms were used, the methods should perform similarly, but the remaining features appear more frequently. This is due to an important pre-processing that was made: a further cropping of the image. Further inspection of the datasets show several different sized ROIs, thus variable unnecessary information. As a consequence, further adaptiveness is obtained with cropping since pre-processed images presented to the segmentation algorithms have a greater uniformity in size than before.

Retraining the classifier when more images are acquired is a possibility since their training time is really small compared to the processing time of an image, a difference of one order of magnitude.

Also of note, public datasets of DFIs containing lasermarks were not found throughout the elaboration of this work or do not exist. All the datasets with lasermarks used in this work were provided by Retmarker S.A..

The 5-fold cross validation performed on NOMASCR, for the subset of features **GR-SVM**, classified correctly 93,68% of the images, with a Specificity of 98,31% and Sensibility of 78,82%. This sensibility differs considerably from the global one of classification results. As referred in subsection 2.2.1, the PDJD dataset (which resulted from that same study), lacks variation in resolution, quality, models of cameras and even types of photocoagulation laser marks. This was the first study of this specific topic, and PDJD was used to test the implemented method in conjunction with the Messidor dataset and 996 more "No Laser" images, justifying the use of the dataset. Furthermore, if PDJD lacks these variations it is less representative of the reality and for that reason a sensibility of 98.02% was obtained in this particular Dataset.

Concerning the remaining datasets, the only dataset in which results are less appealing is the BAT dataset with 93,33% Specificity and 64,71% Sensibility. Yet, BAT is a really unique dataset in the thesis and it is composed of images of patients before and after photocoagulation treatment of a total of nine patients (kindly provided by Centro Cirúrgico de Coimbra).

For all the 9 patients, at least one true positive was detected. In other words, considering only the lasermarks images of each patient one or more was detected as containing

lasermarks (of the images that in fact contain, lasermarks) which is equivalent to a sensibility of 100% for these specific patients since, in a screening initiative, this is the necessary condition for admitting a treated patient. In a clinical context the evaluation that is made is always a per patient one. The false positive that was obtained represents the misclassification of that patient, which translates in an accuracy of 88.89%. The obtained False Negative is presented in figure 4.4. The patient to which this image belongs, suffers from a condition called Central Retinal Vein Occlusion, a vascular disorder commonly characterized by superficial hemorrhages that originate the lesions that can be observed in the figure, which were misinterpreted by the algorithm.

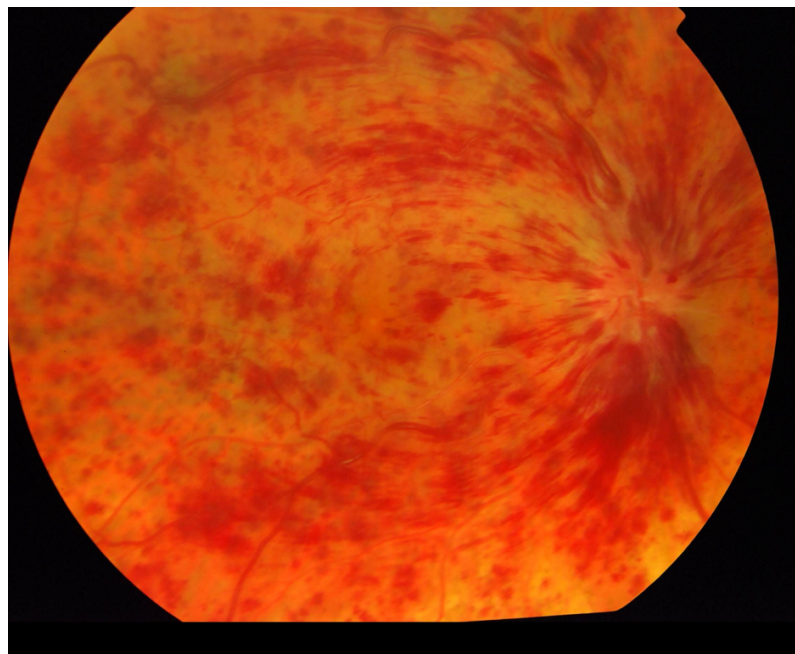


Figure 4.4: The only False Positive obtained in the BAT dataset.



## Chapter 5

# Conclusion and Future Work

This work was a practical implementation of Sousa *et al.* [7] algorithm and had the primary objective of optimizing it. Several aspects of pre-processing, detection and classification were enhanced.

This work contributed to validate state-of-the-art methods and improve some of them. Moreover, due to the optimization performed and the obtained classification efficiency the algorithm is ready to be employed in screening initiatives which is also relevant as a way of enhancing other detection methods, without a dramatic time increase. Thus, this thesis had specific guidelines.

Firstly, the previous method was carefully measured in both time and classification performances, after which a bibliographic research of existent methods was performed, searching for optimization opportunities.

Considering the previously studied execution time of the algorithm, some parts were considered first as were percentually considered to be the most time consuming tasks.

Several methods were then tested for comparison with the already available methods, in search of the best capable and time-optimal sequences of possible pre-processing and feature extraction providing adaptiveness to provided images.

Data inspection and mining was always present using tools like WEKA and Matlab, in the search for the best suitable data for classification.

An extensive optimization taking into account Matlab tools and subtleties was applied with the objective of presenting the most efficient algorithm possible for real-time applications, as Retmarker.

The final evaluation of the applied methods was made by choosing adequate classifiers, training them and testing their respective classification performance. Three additional datasets were obtained for the purpose of validating the algorithm after a first test with

the remaining test datasets.

In the light of these results, it can be concluded that the main goal of this work was accomplished: to optimize, implement and evaluate the previous algorithm for the detection of laser marks with the purpose of its integration on the main Retmarker algorithm, without compromising the classification performance obtained by the previous method.

The grading of Retinal Fundus Images is a complex task, even for human experts and the optimized method provides a contribution in this subject by time-efficiently identifying a noticeable quantity of specific lasermark-containing Digital Retinal Fundus Images in an timely fashion, since a reduction rounding 80% of the processing time was achieved without degradation of the previously obtained results and even improving them, even if marginally.

The results obtained are concluded similar to the previous state-of-the-art methods of João Dias *et al.* [12], Faraz *et al.* [14] and Sousa *et al.* [7]. A constant comparison with the last method is provided throughout this thesis, as it was the starting point of this work.

Regarding the two previous approaches of [12] and [14], the presented method obtained a marginally lower specificity but a much higher sensibility than [12] even if the test datasets are not exactly the same. Relatively to the method proposed in [14], the presented method obtained a higher specificity, but a lower sensibility, yet the images used in this work are not publicly available which complicates comparisons with this method.

Relevant conclusions can be taken from this work, as the mentioned contribution of texture descriptors which presented a pertinent contribution in this classification approach, or as the SVM supremacy over all the other tested classifiers.

A suggestion for a future work is a partnership with a specialized treatment institution in order to gather more images with and without laser treatment scars with the objective of improving the existent classifier by an online training approach, in other words, re-training the classifier in real time when new images enter the system. This approach would provide additional robustness to the method, since new images with resolution and camera type equal to the very next ones to be classified would enter the system.

Another suggestion could be the addition of histogram projection in different colormaps as suggested by [12]. Since lasermarks are either dark or bright it seems this technique could be explored with greater depth.

# Appendix A

## Datasets Examples

Dataset Messidor (1187 retinal fundus images)

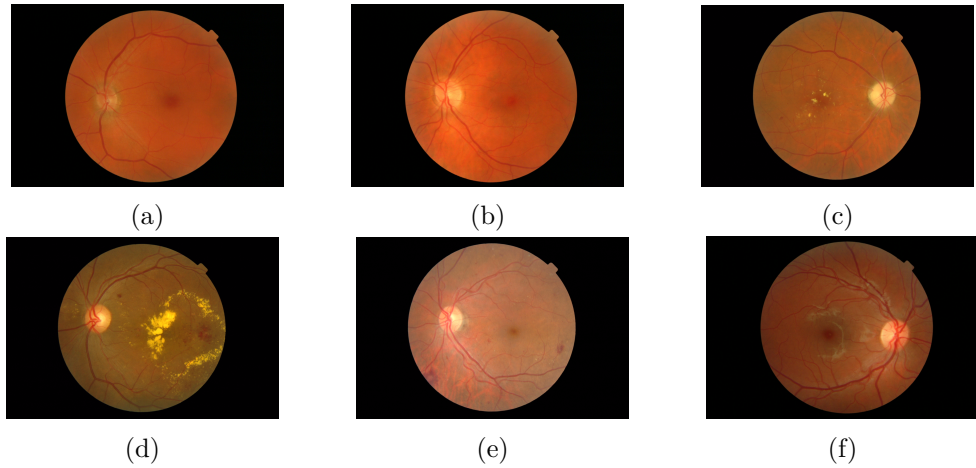


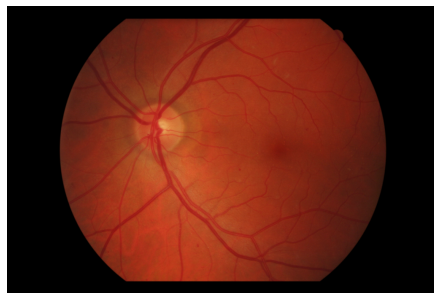
Figure A.1: Messidor example images.

As stated in section 2.6 of Chapter 2, 13 images of the original Messidor dataset (which has a total of 1200 images) were not considered. These images were identified as containing lasermarks by Retmarker S.A. image processing professionals and, since the Messidor dataset was initially chosen as one of the best measures of specificity, they were not considered.

Messidor	
20051020_44843_0100_PP.png	20051020_54209_0100_PP.png
20051021_40377_0100_PP.png	20051021_40450_0100_PP.png
20051021_59459_0100_PP.png	20051021_59504_0100_PP.png
20051205_31396_0400_PP.png	20051212_41432_0400_PP.png
20051213_62648_0100_PP.png	20051214_40912_0100_PP.png
20051214_41582_0100_PP.png	20051214_56821_0100_PP.png
20051214_57940_0100_PP.png	-

Table A.1: DFIs not considered from the Messidor Dataset.

Dataset MA (148 retinal fundus images)



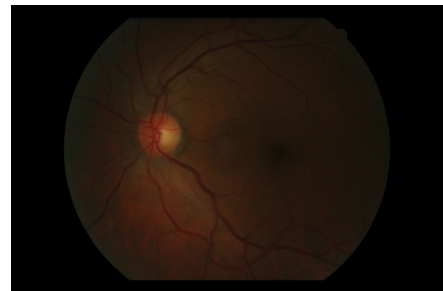
(a)



(b)



(c)



(d)



(e)



(f)

Figure A.2: MA example images.

Dataset NOMA (233 retinal fundus images)

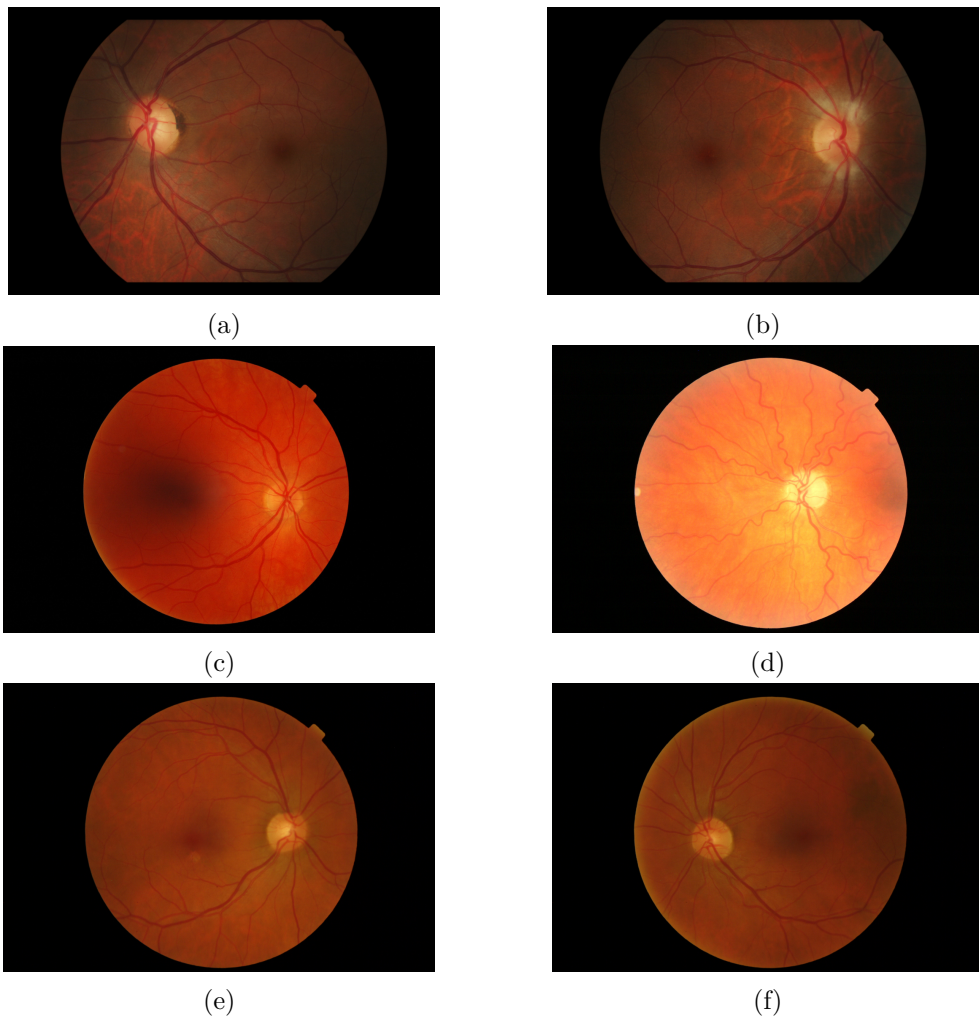


Figure A.3: NOMA example images.

Dataset EX (47 retinal fundus images)



(a)



(b)



(c)



(d)



(e)



(f)

Figure A.4: EX example images.

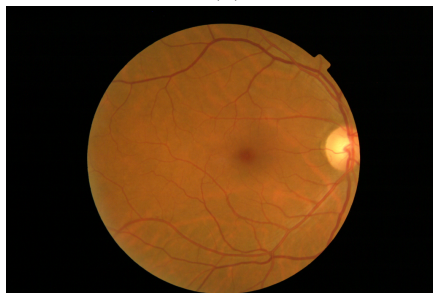
Dataset NOEX (35 retinal fundus images)



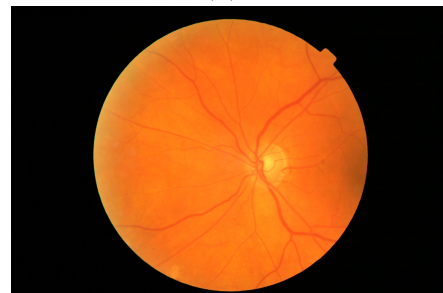
(a)



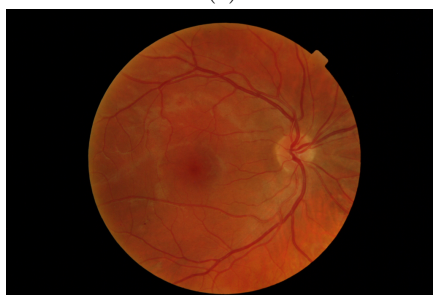
(b)



(c)



(d)



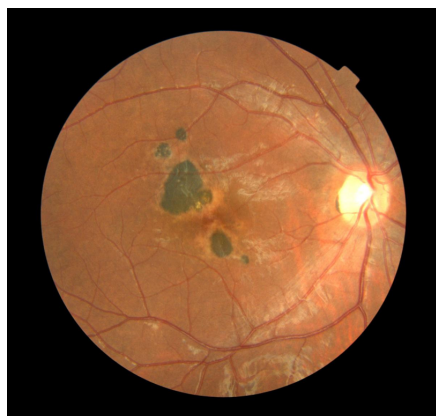
(e)



(f)

Figure A.5: NOEX example images.

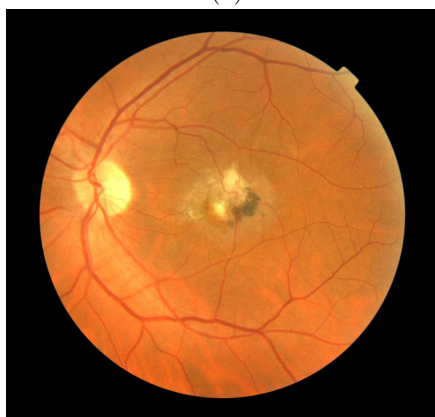
Dataset VBR (22 retinal fundus images)



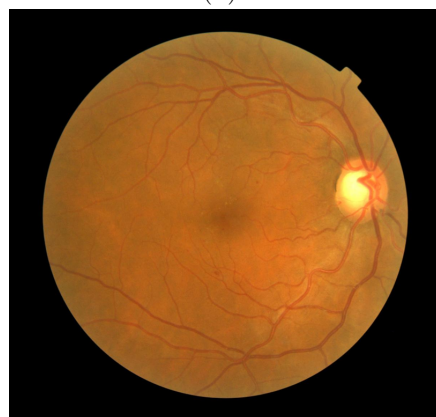
(a)



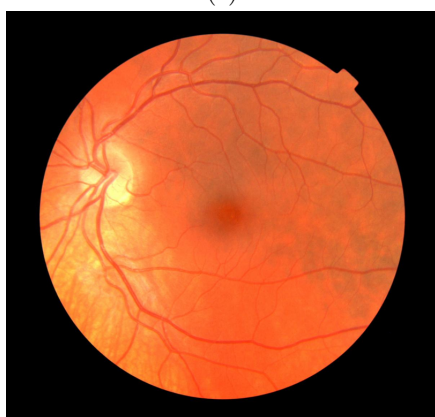
(b)



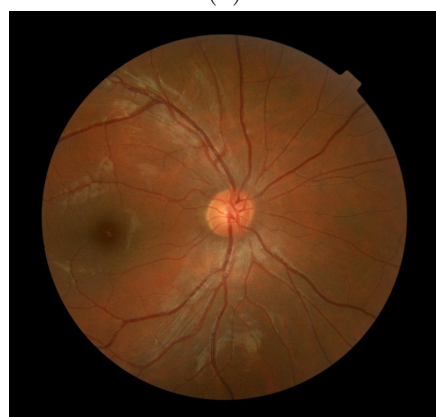
(c)



(d)



(e)



(f)

Figure A.6: VBR example images.



Dataset 50HP (100 retinal fundus images)



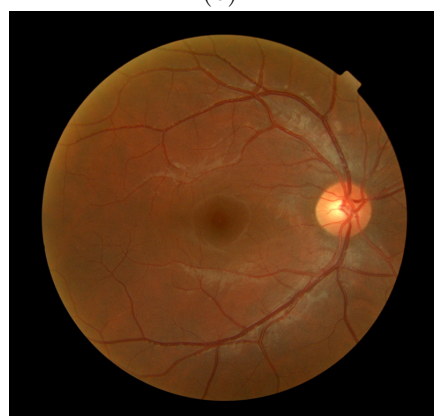
(a)



(b)



(c)



(d)



(e)



(f)

Figure A.7: 50HP example images.

Dataset FAZD (60 retinal fundus images)

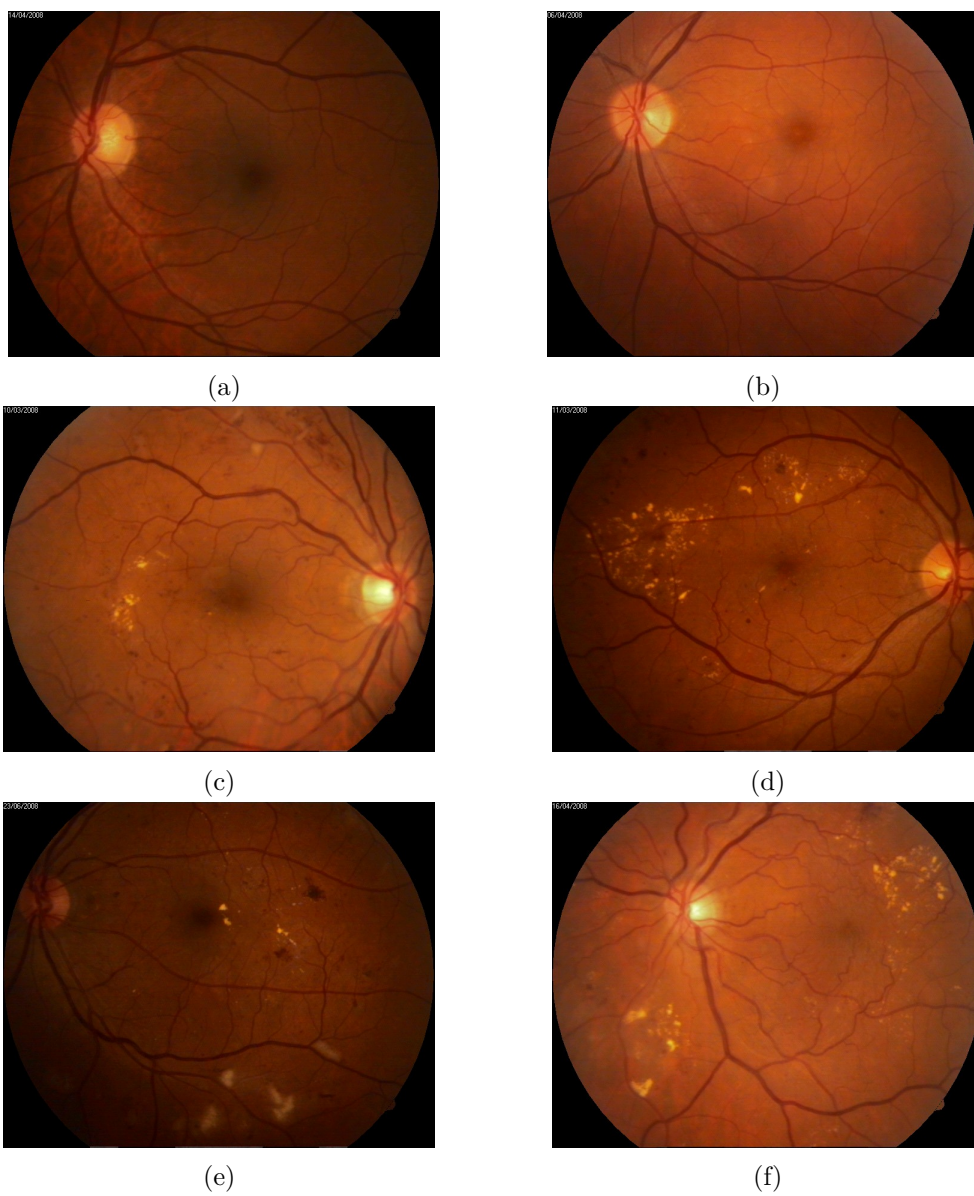


Figure A.8: FAZD example images.

Dataset SCR (622 retinal fundus images)

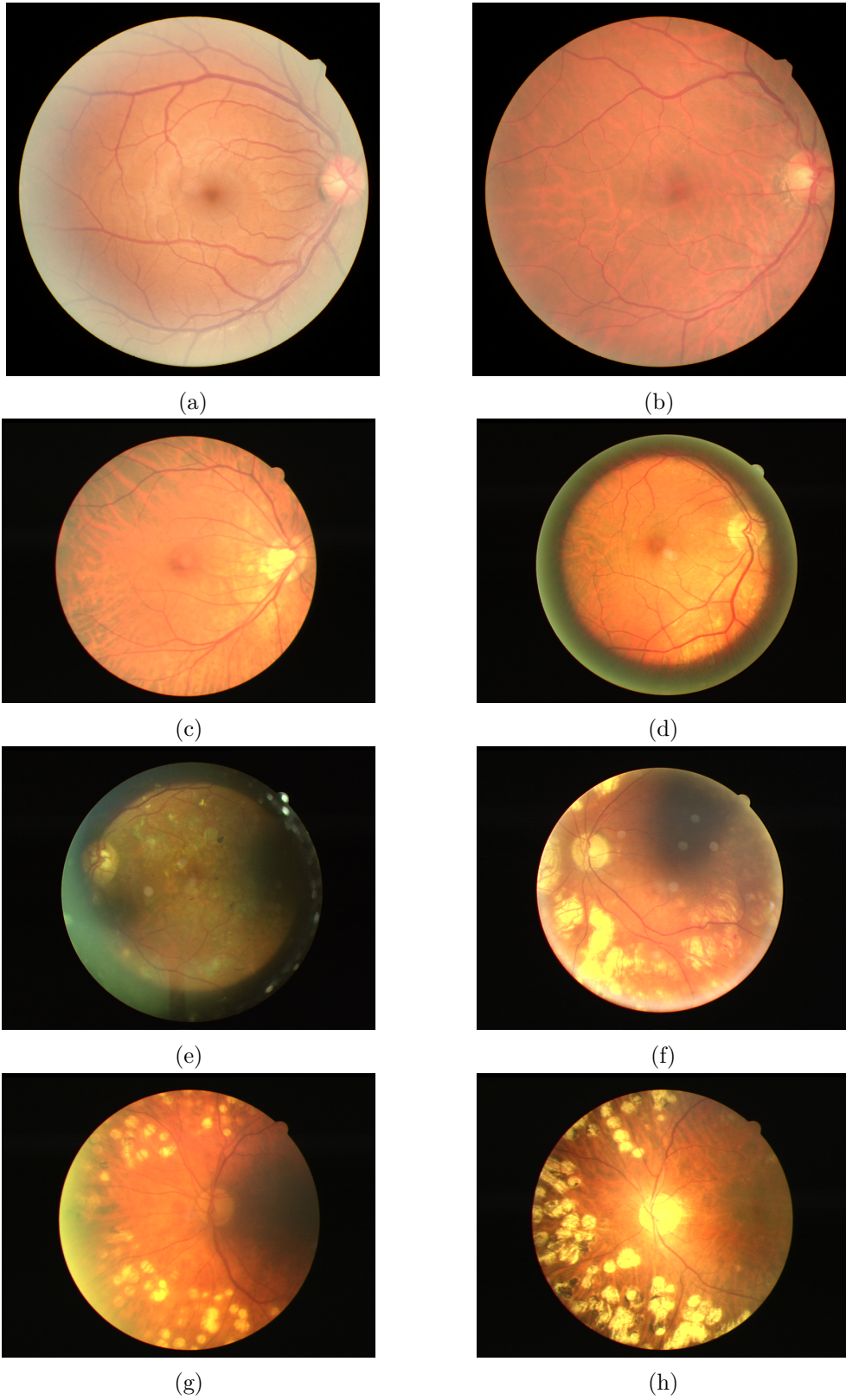


Figure A.9: SCR example images ((a) to (d) "No Laser", (e) to (h) "Laser").

**Dataset BAT (49 retinal fundus images)**

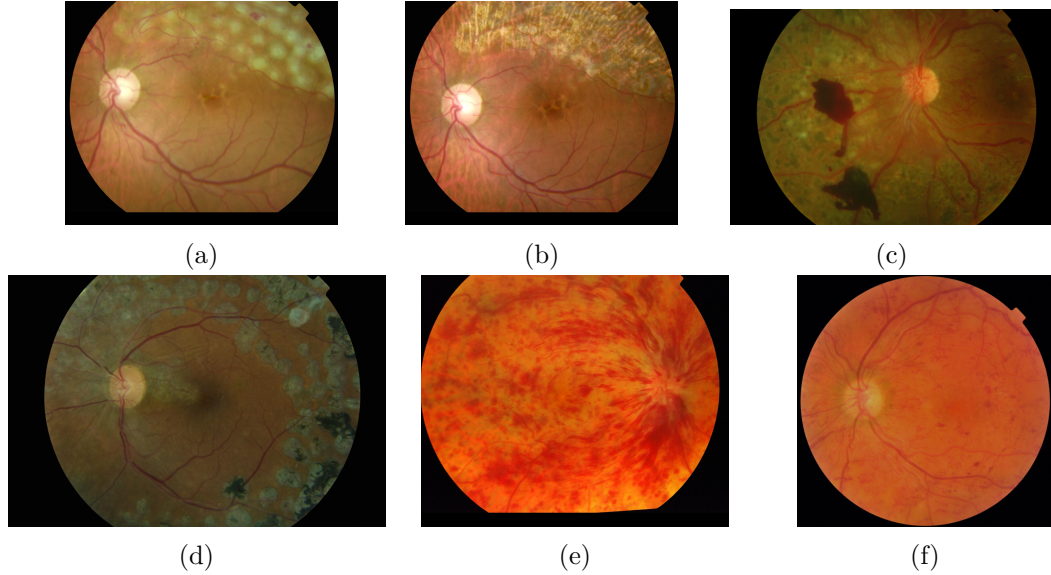


Figure A.10: BAT example images.

As stated in section 2.6 of Chapter 2 the BAT dataset has a specific constitution representing nine different patients. The structure of this dataset can be summarized by table A.2

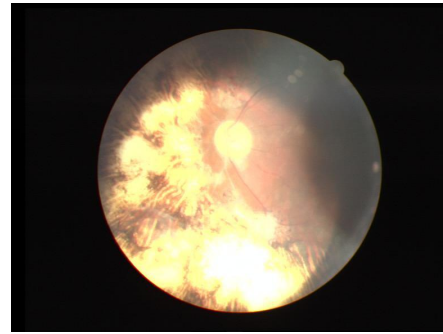
Patient	Images	
	"No Laser"	"Laser"
#1	2	4
#2	2	6
#3	2	3
#4	1	3
#5	1	8
#6	2	1
#7	2	4
#8	1	2
#9	2	3

Table A.2: BAT Dataset discrimination.

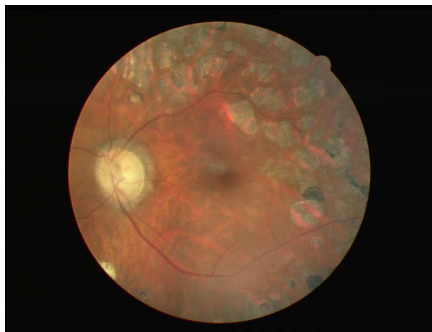
Dataset PDJD (101 retinal fundus images)



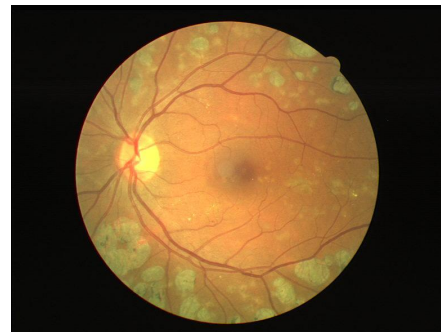
(a)



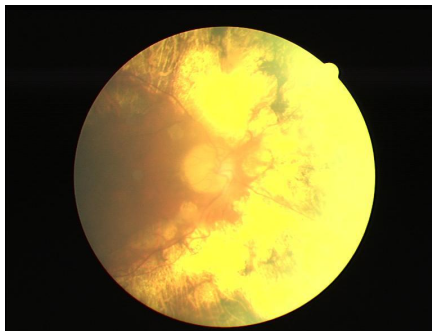
(b)



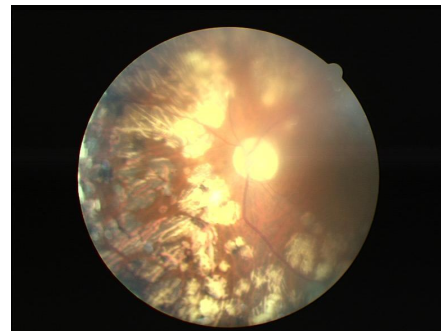
(c)



(d)



(e)



(f)

Figure A.11: PDJD example images.

Dataset SCR15 (99 retinal fundus images)



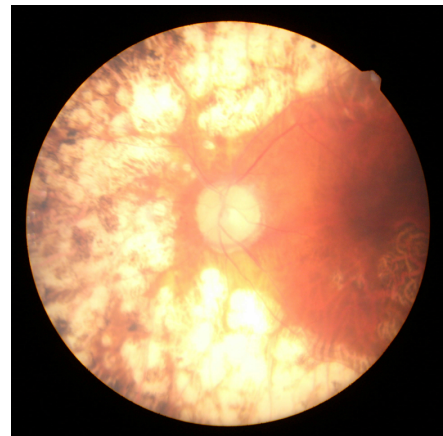
(a)



(b)



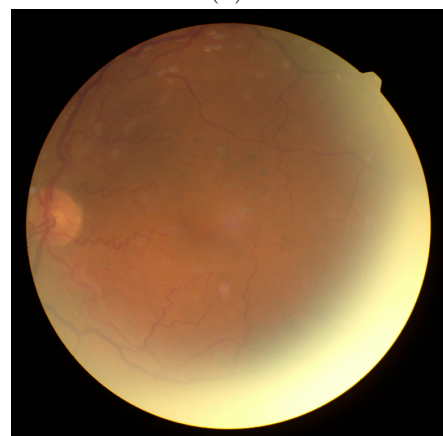
(c)



(d)



(e)



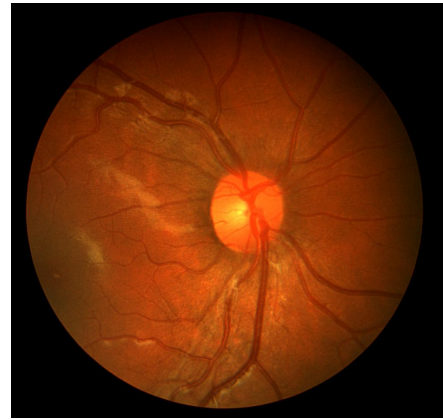
(f)

Figure A.12: SCR15 example images.

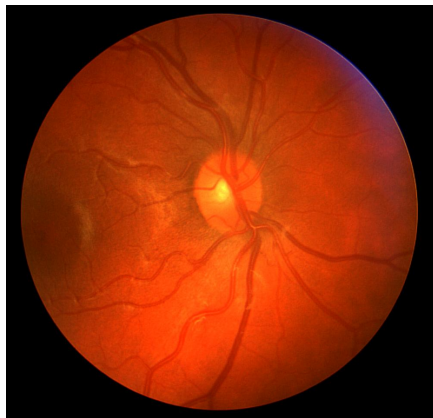
Dataset CDB (28 retinal fundus images)



(a)



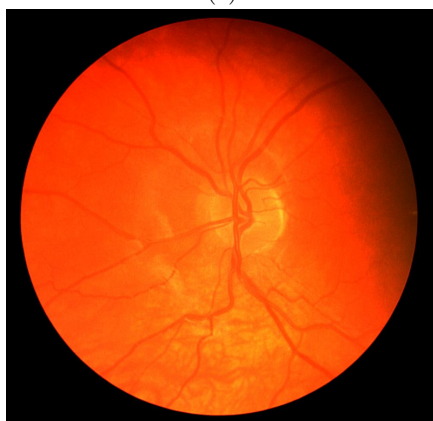
(b)



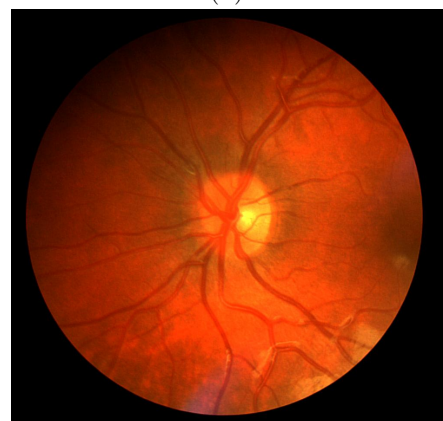
(c)



(d)



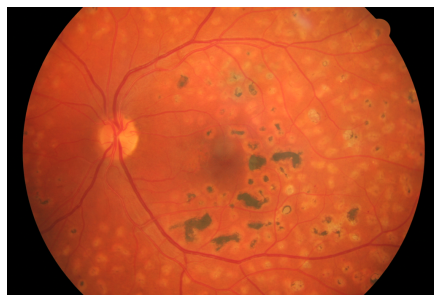
(e)



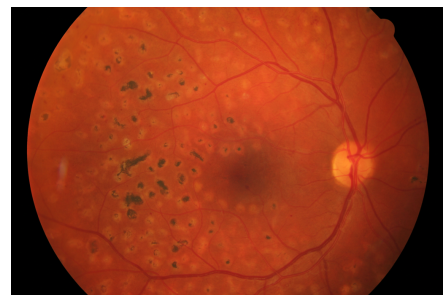
(f)

Figure A.13: CDB example images.

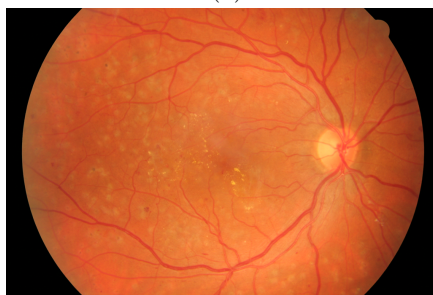
Dataset HRF (45 retinal fundus images)



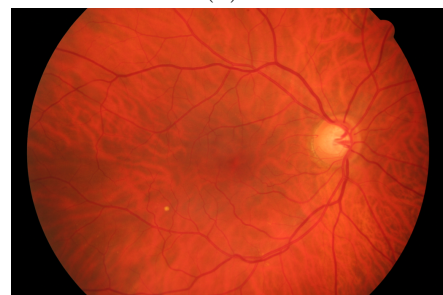
(a)



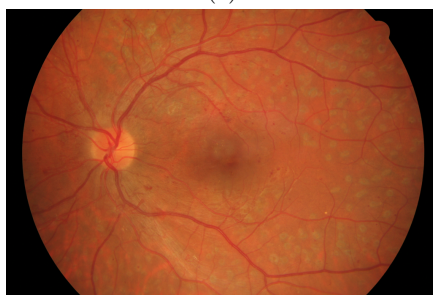
(b)



(c)



(d)



(e)



(f)

Figure A.14: HRF example images.



## Appendix B

# Initial Algorithm Considerations

Table B.1: Kernels and Constants used.

Constant Name	Constant Value	Constant meaning
constant tiles	8 x scaling constant	Used in Adaptive Histogram Equalization
constant luminance	28 x scaling constant	Kernel size for obtaining luminance map
constant median	5 x scaling constant	Kernel size of median filter
constant vessels	ROI/37 x scaling constant	Dilation constant of the vascular network
constant min circles	$D/23$	Minimum circle radius considered by CHT
constant max circles	$D/4$	Maximum circle radius considered by CHT
constant LMS	15 x scaling constant	Disk structuring element radius
constant area LMS	100 x scaling constant	Pixel threshold for removing objects in LMS
constant FVF	100 x scaling constant	Pixel threshold for removing objects in FVF

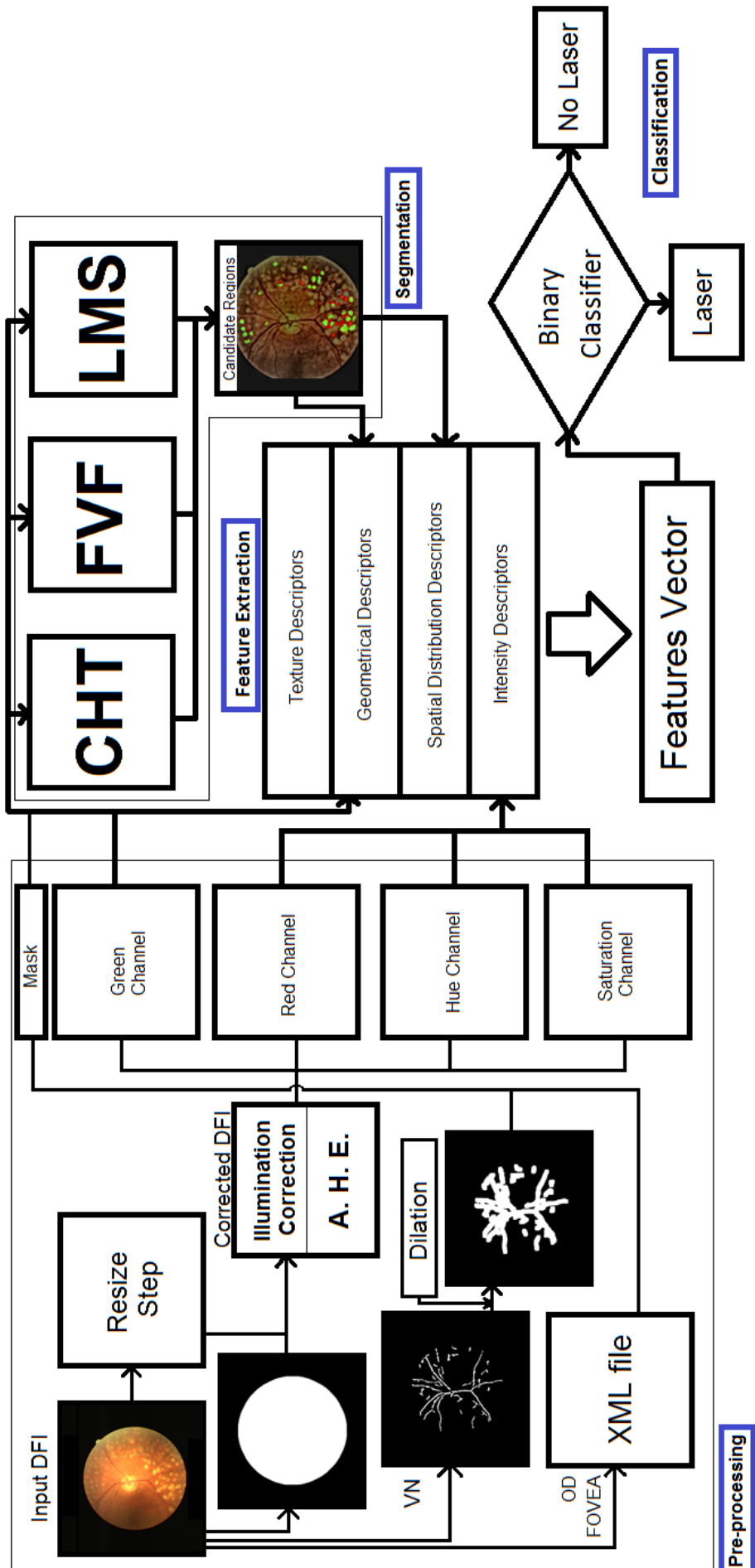


Figure B.1: Algorithm to be optimized.

# Computed Features

Feature	Description	Feature	Description
1	number_CHT	34	max_laser_sat
2	area_CHT	35	variance_var_green
3	weighted_area_CHT	36	variance_var_red
4	likelihood_CHT	37	variance_var_hue
5	number_LMS	38	variance_var_sat
6	area_LMS	39	avg_gray_level
7	number_FVF	40	avg_contrast
8	area_FVF	41	smoothness
9	likelihood_FVF	42	skewness
10	weighted_area_FVF	43	uniformity
11	radius_mean	44	entropy
12	radius_var	45	autocorrelation
13	hull	46	contrast
14	hull_area	47	dissimilarity
15	moran_i	48	homogeneity
16	moran_null_hipot	49	correlation
17	trace	50	cluster_prominence
18	determinant	51	cluster_shade
19	max_eigen	52	energy
20	point_density	53	entropy_glcM
21	m_dist_total	54	max_prob
22	v_dist_total	55	variance
23	mean_laser_green	56	sum_avg
24	var_laser_green	57	sum_var
25	max_laser_green	58	sum_entropy
26	mean_laser_red	59	diff_var
27	var_laser_red	60	diff_entropy
28	max_laser_red	61	inv_diff
29	mean_laser_hue	62	inv_diff_norm
30	var_laser_hue	63	inv_diff_moment_norm
31	max_laser_hue	64	info_corr_1
32	mean_laser_sat	65	info_corr_2
33	var_laser_sat		

Figure B.2: Computed Features.

## Appendix C

### Code Sections Efficiency

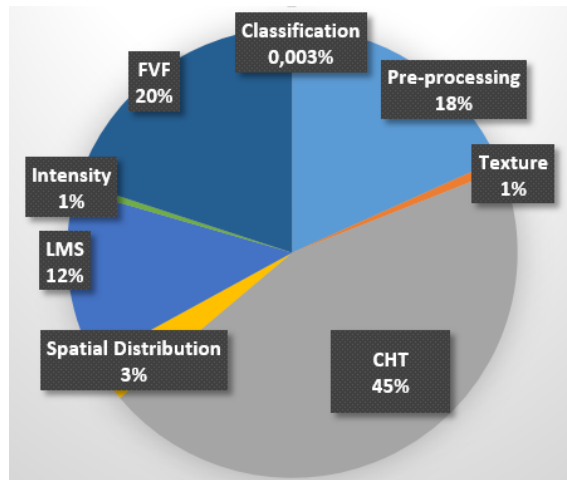


Figure C.1: Initial algorithm mean processing time, per section.

Table C.1: Initial algorithm classification with Sousa *et al.* DT classifier.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1175	1175	12	0	7	98,99	-	98,99
MA	0	148	148	0	0	0	100,00	-	100,00
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	34	34	1	0	1	97,14	-	97,14
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	99	99	1	0	1	99,00	-	99,00
FAZD	0	57	57	3	0	3	95,00	-	95,00
BAT	21	14	35	1	13	14	93,33	61,76	71,43
PDJD	98	0	98	0	3	3	-	97,03	98,03
Total	119	1596	1715	18	16	34	98,89	88,15	98,06
<b>Specificity: 98.89%, Sensitivity: 88.15%, Accuracy: 98.06%</b>									

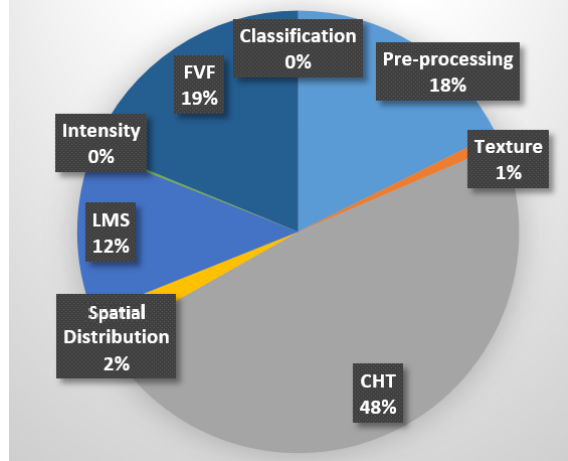


Figure C.2: Algorithm mean processing time, per section, after preprocessing improvements.

Table C.2: Algorithm classification with Sousa *et al.* DT classifier, after preprocessing modifications.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1171	1171	16	0	16	98,65	-	98,65
MA	0	148	148	0	0	0	100,00	-	100,00
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	34	34	1	0	1	97,14	-	97,14
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	99	99	1	0	1	99,00	-	99,00
FAZD	0	59	59	1	0	1	98,33	-	98,33
BAT	19	13	32	2	15	17	86,67	55,88	65,31
PDJD	100	0	100	0	1	1	-	99,01	99,01
Total	119	1593	1712	21	16	37	98,70	88,15	97,88
<b>Specificity: 98.70%, Sensitivity: 88.15%, Accuracy: 97.88%</b>									

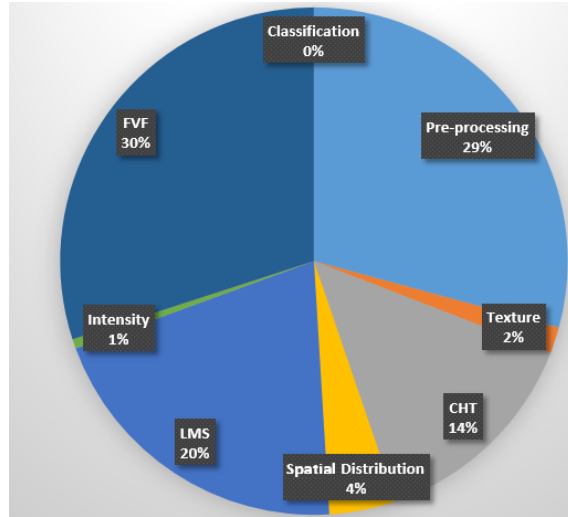


Figure C.3: Algorithm mean processing time, per section, after CHT improvements.

Table C.3: Algorithm classification with Sousa *et al.* DT classifier, after CHT modifications.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1176	1176	11	0	11	99,07		99,07
MA	0	148	148	0	0	0	100,00		100,00
EX	0	47	47	0	0	0	100,00		100,00
NOEX	0	33	33	2	0	2	94,29		94,29
VBR	0	22	22	0	0	0	100,00		100,00
50HP	0	99	99	1	0	1	99,00		99,00
FAZD	0	60	60	0	0	0	100,00		100,00
BAT	16	13	29	2	18	20	86,67	47,06	59,18
PDJD	98	0	98	0	3	3	0,00	97,03	97,03
Total	114	1598	1712	16	21	37	99,01	84,44	97,88
<b>Specificity: 99.01%, Sensitivity: 84.44%, Accuracy: 97.88%</b>									

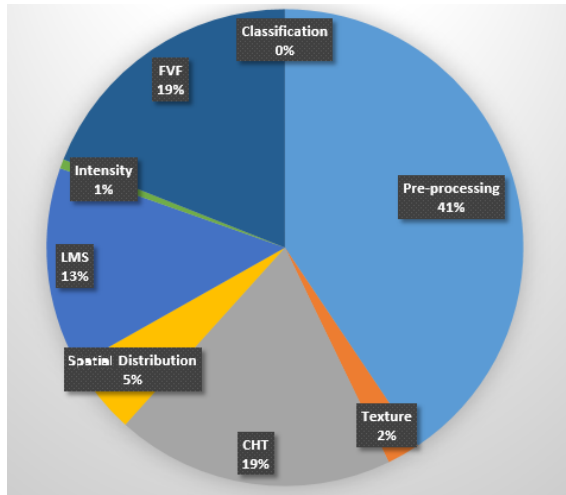


Figure C.4: Algorithm mean processing time, per section, after FVF and LMS improvements.

Table C.4: Algorithm classification with Sousa *et al.* DT classifier, after FVF and LMS modifications.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1176	1176	11	0	11	99,07		99,07
MA	0	148	148	0	0	0	100,00		100,00
EX	0	47	47	0	0	0	100,00		100,00
NOEX	0	33	33	2	0	2	94,29		94,29
VBR	0	22	22	0	0	0	100,00		100,00
50HP	0	99	99	1	0	1	99,00		99,00
FAZD	0	60	60	0	0	0	100,00		100,00
BAT	16	13	29	2	18	20	86,67	47,06	59,18
PDJD	98	0	98	0	3	3	0,00	97,03	97,03
Total	114	1598	1712	16	21	37	99,01	84,44	97,88
<b>Specificity: 99.01%, Sensitivity: 84.44%, Accuracy: 97.88%</b>									

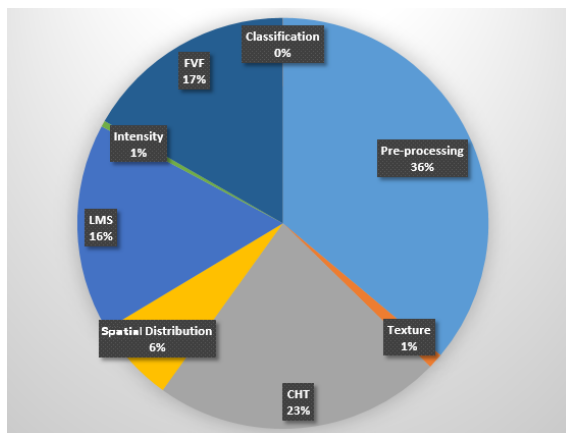


Figure C.5: Algorithm mean processing time, per section, after code-based improvements.

# Appendix D

## Feature Selection

Entropy, in in a classification context, is the expected necessary information to classify a sample.

A common information theory measure used in the training of Decision Tree Classifiers and entropy-based evaluator is the **Information Gain** of a feature  $F$ , which is the change in entropy  $H$  or the difference between two entropy values:

$$IG(F) = H(S) - H(F) \quad (\text{D.1})$$

where  $S$  is a training set that is composed by  $s$  data samples with  $m$  classes, which are given by

$$H(S) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (\text{D.2})$$

$$H(F) = - \sum_{j=1}^n H(S) \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} \quad (\text{D.3})$$

where:

- $p_i$  is the probability that a random sample belongs to class  $C_i$  which is equal to  $\frac{s_i}{s}$ .
- $s_{ij}$  is the number of samples of class  $C_i$  in a subset  $S_j$



**Gain Ratio** is a modification of the Information gain measure where the issue involving the bias of this latter attribute evaluator (where attributes with a larger number of attributes were favored) was solved by normalizing Information Gain with the Intrinsic Information of a feature. This is the reason why the improved C4.5 Decision Tree algorithm uses this measure instead of Information Gain used in its predecessor, ID3.

The Gain Ratio measure of a feature is the Information Gain measure of that feature divided by its intrinsic information:

$$GR(F) = \frac{IG(F)}{SI_S(F)} \quad (D.4)$$

where the intrinsic information measure is given by

$$SI_S(F) = - \sum_{j=1}^n \frac{|S_j|}{|S|} \log_2 \left( \frac{|S_j|}{|S|} \right) \quad (D.5)$$

Gain Ratio decreases as the intrinsic information gets larger, hence the prioritizing of wider ranged features does not happen with Gain Ratio, which can in turn reduce the probability of overfitting.

Table D.1: IG-filtered and GR-filtered feature subsets and their respective individual value.

GR		IG	
moran_null_hipot	0.367	number_CHT	0.438
moran_i	0.325	weighted_area_CHT	0.417
determinant	0.313	area_CHT	0.385
trace	0.299	max_laser_green	0.385
likelihood_CHT	0.296	moran_i	0.383
hull_area	0.294	m_dist_total	0.378
max_eigen	0.292	likelihood_CHT	0.343
number_CHT	0.288	hull_area	0.333
max_laser_green	0.281	v_dist_total	0.311
weighted_area_CHT	0.278	moran_null_hipot	0.308
likelihood_FVF	0.257	area_FVF	0.303
area_CHT	0.257	weighted_area_FVF	0.297
v_dist_total	0.257	number_FVF	0.292
radius_var	0.255	variance_var_hue	0.287
m_dist_total	0.252	determinant	0.286
area_FVF	0.216	trace	0.277
var_laser_green	0.213	max_eigen	0.274
weighted_area_FVF	0.212	area_LMS	0.269
number_FVF	0.209	variance_var_green	0.258
variance_var_hue	0.206	variance_var_sat	0.256
hull	0.204	variance_var_red	0.250
area_LMS	0.202	max_laser_hue	0.249
var_laser_red	0.200	number_LMS	0.246
variance_var_green	0.197	radius_var	0.245
max_laser_hue	0.191	var_laser_hue	0.240
number_LMS	0.189	max_laser_red	0.233
variance_var_sat	0.188	point_density	0.233
var_laser_sat	0.186	radius_mean	0.220
variance_var_red	0.183	likelihood_FVF	0.220
point_density	0.183	max_laser_sat	0.211
cluster_prominence	0.181	var_laser_green	0.210
radius_mean	0.175	hull	0.203
max_laser_sat	0.172	mean_laser_red	0.202
mean_laser_hue	0.169	mean_laser_green	0.202
max_laser_red	0.167	var_laser_red	0.200
var_laser_hue	0.166	var_laser_sat	0.186
mean_laser_red	0.150	mean_laser_sat	0.184
mean_laser_green	0.147	bright_lesions	0.173
smoothness	0.145	mean_laser_hue	0.169
avg_contrast	0.145	solidity_LMS	0.167
solidity_LMS	0.144	skewness	0.138
mean_laser_sat	0.143	cluster_shade	0.112
roundedness_LMS	0.131	-	-

## Appendix E

# Other Classification Results

After a stratified 5-fold cross validation using an DT classifier with the features **IG\_DT** on the images of the training dataset, the following results were obtained:

<b>IG_DT</b>	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
NOMASCR	159	636	795	16	44	60	97,55	78,33	92,98

The trained classifier resulted in an DT model with 6 leaves and a total of 11 nodes (Appendix F), which WEKA needed 0.01 seconds to build, and that uses only 3 of the initial 8 features. By testing this classifier in all the test datasets, that total 1749 images (135 "Laser" and 1614 "No Laser"), the following results were obtained:

Table E.1: **IG\_DT** classifier tested on test datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1180	1180	7	0	7	99,41	-	99,41
MA	0	148	148	0	0	0	100,00	-	100,00
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	34	34	1	0	1	97,14	-	97,14
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	100	100	0	0	0	100,00	-	100,00
FAZD	0	56	56	4	0	4	93,33	-	93,33
BAT	19	14	33	1	15	16	93,33	55,88	67,35
PDJD	97	0	97	0	4	4	-	98,02	98,02
Total	116	1601	1719	18	10	28	99,19	85,93	98,28
<b>Specificity: 99.19%, Sensitivity: 85.93%, Accuracy: 98.28%</b>									

After a stratified 5-fold cross validation using a DT classifier with the features **GR\_DT** on the images of the training dataset, the following results were obtained:

<b>GR_DT</b>	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
NOMASCR	160	644	804	8	43	51	98,77	78,82	94,04

The trained classifier resulted in an DT model with 10 leaves and a total of 19 nodes (Appendix F), which WEKA needed 0.02 seconds to build. By testing this classifier in all the test datasets, that total 1749 images (135 "Laser" and 1614 "No Laser"), the following results were obtained:

Table E.2: **GR\_DT** classifier tested on test datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1172	1172	15	0	15	98,73	-	98,73
MA	0	147	147	1	0	1	99,32	-	99,32
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	33	33	2	0	2	94,29	-	94,29
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	100	100	0	0	0	100,00	-	100,00
FAZD	0	54	54	6	0	6	90,00	-	90,00
BAT	19	14	33	1	15	16	93,33	55,88	67,35
PDJD	98	0	98	0	3	3	-	97,03	97,03
Total	117	1589	1706	25	18	43	98,45	86,66	97,54
<b>Specificity: 98.45%, Sensitivity: 86.66%, Accuracy: 97.54%</b>									

After a stratified 5-fold cross validation using a ADT classifier with the features **IG/GR\_ADT** on the images of the training dataset, the following results were obtained:

<b>IG/GR_ADT</b>	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
NOMASCR	162	639	801	13	41	54	98,01	79,80	93,68

The trained classifier resulted in an ADT tree model with 21 prediction nodes and 10 decision nodes using 6 of the 10 selected features (Appendix F), which WEKA needed 0.21 seconds to build, using 7 of the initial 11 features. This classifier represents two of the subsets found during feature selection phase. This is because ADT-based wrapper returned the same features for each filtered subsets with IG and GR. By testing this classifier in all the test datasets, that total 1749 images (135 "Laser" and 1614 "No Laser"), the following results were obtained:

Table E.3: **IG/GR\_ADT** classifier tested on test datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1178	1178	9	0	9	99,24	-	99,24
MA	0	146	146	2	0	2	98,65	-	98,65
EX	0	46	46	1	0	1	97,87	-	97,87
NOEX	0	34	34	1	0	1	97,14	-	97,14
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	100	100	0	0	0	100,00	-	100,00
FAZD	0	54	54	6	0	6	90,00	-	90,00
BAT	20	13	33	2	14	16	86,66	58,82	67,35
PDJD	97	0	97	0	4	3	-	96,04	96,04
Total	117	1593	1710	21	18	39	98,70	86,67	97,77
<b>Specificity: 98.70%, Sensitivity: 86.67%, Accuracy: 97.77%</b>									

After a stratified 5-fold cross validation using an SVM classifier with the features **IG\_SVM** on the images of the training dataset, the following results were obtained

<b>IG_SVM</b>	TP	TN	T	FP	FN	F	SPE	SENS	ACC
NOMASCR	160	641	801	11	43	54	98,31	78,82	93,68

The trained classifier resulted in a SVM model which WEKA needed 0.28 seconds to build. By testing this classifier in all the test datasets, that total 1749 images (135 "Laser" and 1614 "No Laser"), the following results were obtained:

Table E.4: **IG\_SVM** classifier tested on test datasets.

Dataset	TP	TN	T	FP	FN	F	SPE(%)	SEN(%)	ACC(%)
MESSIDOR	0	1181	1181	6	0	6	99,49	-	99,49
MA	0	148	148	0	0	0	100,00	-	100,00
EX	0	47	47	0	0	0	100,00	-	100,00
NOEX	0	34	34	1	0	1	97,14	-	97,14
VBR	0	22	22	0	0	0	100,00	-	100,00
50HP	0	100	100	0	0	0	100,00	-	100,00
FAZD	0	56	56	4	0	4	93,33	-	93,33
BAT	21	14	35	1	13	14	93,33	61,76	71,38
PDJD	97	0	97	0	4	4	-	96,04	96,04
Total	118	1602	1720	12	17	29	99,26	87,41	98,34
<b>Specificity: 99.26%, Sensitivity: 87.41%, Accuracy: 98.34%</b>									

## Appendix F

# Classification Models

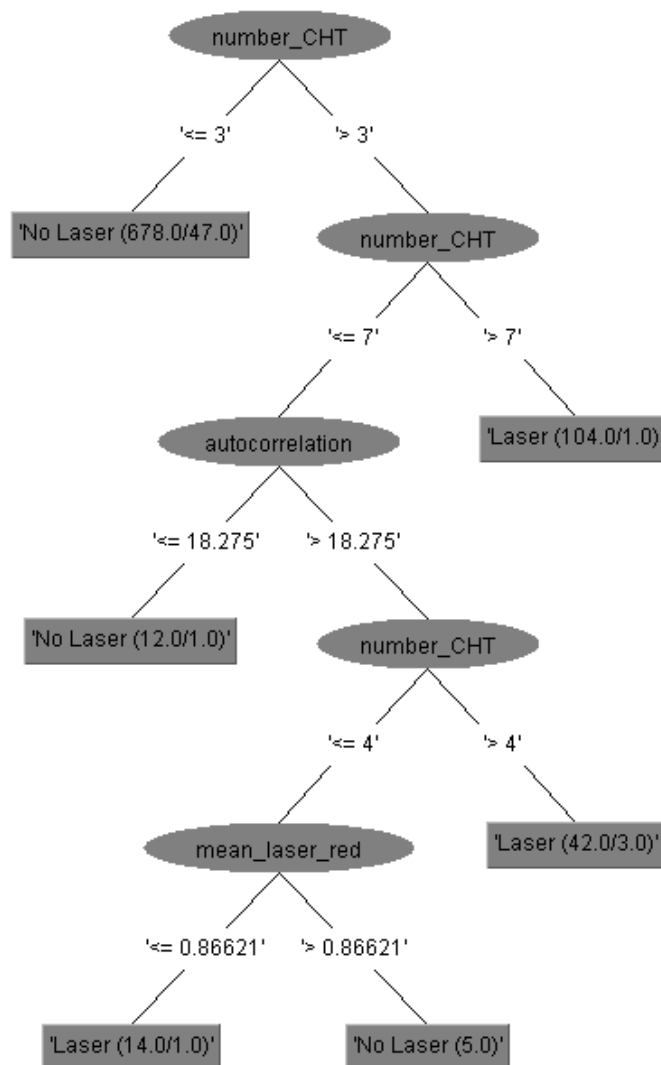


Figure F.1: Decision Tree Classifier trained using the features **IG\_DT**.

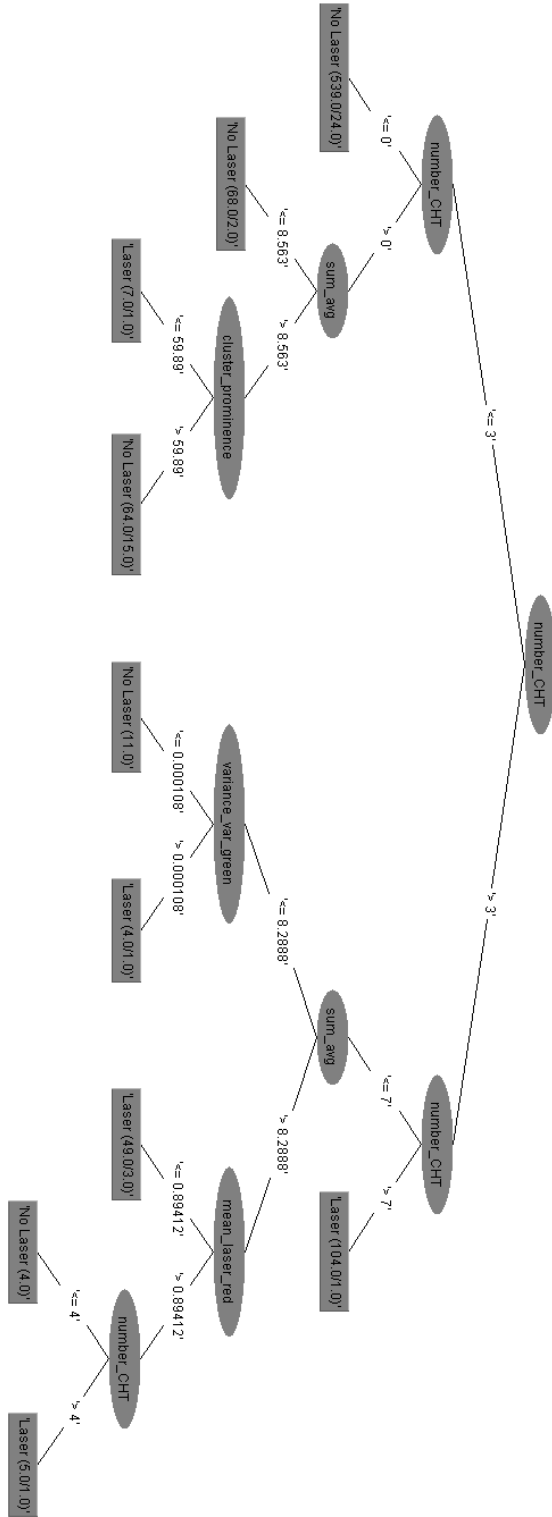


Figure F.2: Decision Tree Classifier trained using the features **GR\_DT**.



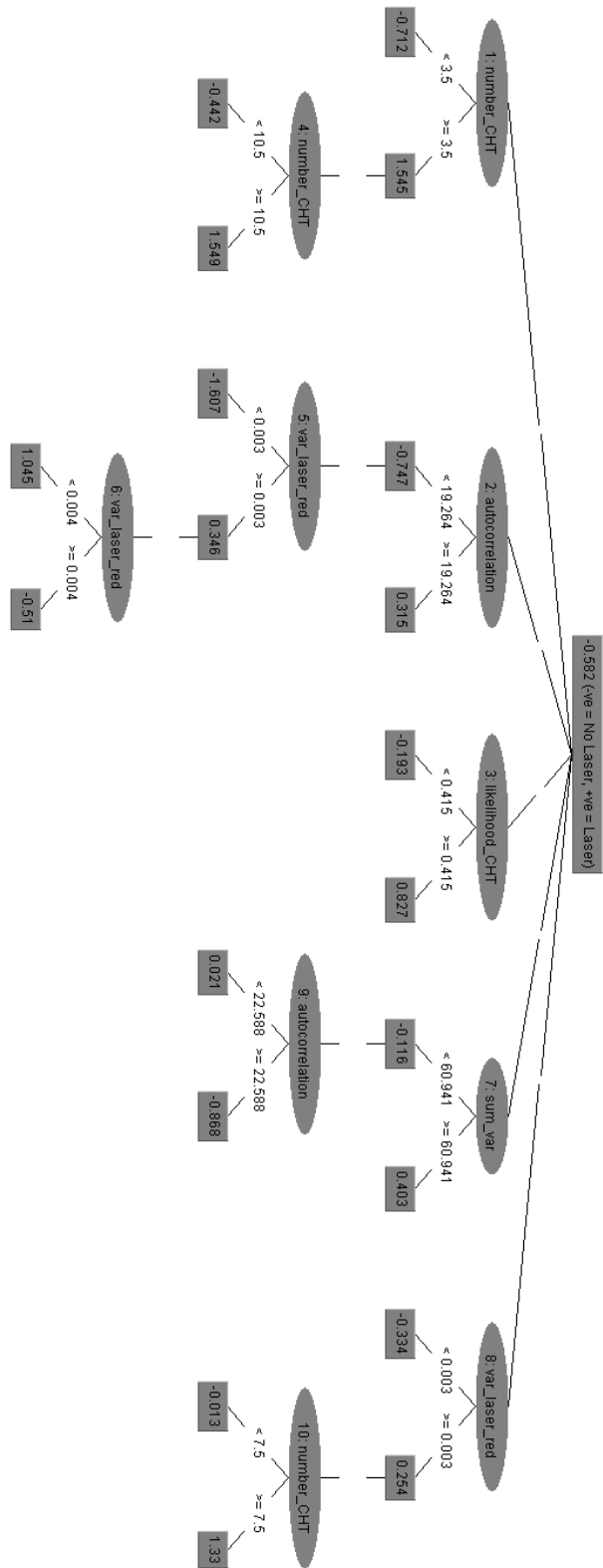


Figure F.3: Alternating Decision Tree Classifier trained using the features IG\_ADT/GR\_ADT.

# Bibliography

- [1] I. D. Federation, *IDF Diabetes Atlas, 6th edn.* Brussels, Belgium: International Diabetes Federation, 2013.
- [2] S. Garg and R. M. Davis, “Diabetic retinopathy screening update,” *Clinical Diabetes*, vol. 27, no. 4, pp. 140–145, 2009.
- [3] J. W. Yau, S. L. Rogers, R. Kawasaki, E. L. Lamoureux, J. W. Kowalski, T. Bek, S.-J. Chen, J. M. Dekker, A. Fletcher, J. Grauslund, *et al.*, “Global prevalence and major risk factors of diabetic retinopathy,” *Diabetes care*, p. DC\_111909, 2012.
- [4] A. Foster and S. Resnikoff, “The impact of vision 2020 on global blindness,” *Eye*, vol. 19, no. 10, pp. 1133–1135, 2005.
- [5] R. J. Tapp, J. E. Shaw, C. A. Harper, M. P. De Courten, B. Balkau, D. J. McCarty, H. R. Taylor, T. A. Welborn, and P. Z. Zimmet, “The prevalence of and factors associated with diabetic retinopathy in the Australian population,” *Diabetes care*, vol. 26, no. 6, pp. 1731–1737, 2003.
- [6] D. A. Sim, P. A. Keane, A. Tufail, C. A. Egan, L. P. Aiello, and P. S. Silva, “Automated retinal image analysis for diabetic retinopathy in telemedicine,” *Current diabetes reports*, vol. 15, no. 3, pp. 1–9, 2015.
- [7] J. G. R. d. A. e. Sousa, “Contributions to the automatic detection of laser marks in retinal digital fundus images,” Master’s thesis, Physics Department, University of Coimbra, Coimbra, Portugal, 2015.
- [8] “<http://www.drbourke.com.au/anatomy-of-the-eye>.”
- [9] J. G. F. Dowler, “Laser management of diabetic retinopathy,” *Journal of the Royal Society of Medicine*, vol. 96, pp. 277–279, June 2003.
- [10] E. Jr, M. Michelessi, and G. Virgili, “Laser photocoagulation for proliferative diabetic retinopathy,” *Cochrane Database Syst Rev.*, November 2014.

- [11] M. M. K. Muqit, J. Denniss, V. Nourrit, G. R. Marcellino, D. B. Henson, I. Schiessl, and P. E. Stanga, "Spatial and spectral imaging of retinal laser photo-coagulation burns.," *Investigative ophthalmology & visual science*, vol. 52, pp. 994–1002, Feb. 2011.
- [12] J. Pires Dias, C. Manta Oliveira, and L. da Silva Cruz, "Detection of laser marks in retinal images," in *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pp. 532–533, June 2013.
- [13] E. Decenciere, X. Zhang, G. Cazuguel, B. Laÿ, B. Cochener, C. Trone, P. Gain, J.-R. Ordóñez-Varela, P. Massin, A. Erginay, *et al.*, "Feedback on a publicly distributed image database: the messidor database," *Image Analysis and Stereology*, pp. 231–234, 2014.
- [14] F. Tahir, M. Akram, M. Abbass, and A. Khan, "Laser marks detection from fundus images," in *Hybrid Intelligent Systems (HIS), 2014 14th International Conference on*, pp. 147–151, Dec 2014.
- [15] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.
- [16] W. A.-N. S. Sekhar and A. K. Nandi, "Automated localisation of retinal optic disk using hough transform," *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1577–1580, May 2008.
- [17] N. M. Salem, S. a. Salem, and A. K. Nandi, "Segmentation of retinal blood vessels based on analysis of the hessian matrix and clustering algorithm," *European Signal Processing Conference, Poznan, Poland*, pp. 428–432, 2007.
- [18] D. Marin, A. Aquino, M. Gegundez-Arias, and J. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *Medical Imaging, IEEE Transactions on*, vol. 30, pp. 146–158, Jan 2011.
- [19] E. Moghimirad, S. H. Rezatofghi, and H. Soltanian-Zadeh, "Multi-scale approach for retinal vessel segmentation using medialness function," in *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pp. 29–32, IEEE, 2010.

- [20] X. Zhu, R. Rangayyan, and A. Ells, *Digital Image Processing for Ophthalmology: Detection of the Optic Nerve Head*. Morgan/Claypool, 2011.
- [21] D. Welfer, J. Scharcanski, C. M. Kitamura, M. M. Dal Pizzol, L. W. B. Ludwig, and D. R. Marinho, "Segmentation of the optic disk in color eye fundus images using an adaptive morphological approach," *Comput. Biol. Med.*, vol. 40, pp. 124–137, Feb. 2010.
- [22] M. Niemeijer, M. Abramoff, and B. van Ginneken, "Segmentation of the optic disc, macula and vascular arch in fundus photographs," *Medical Imaging, IEEE Transactions on*, vol. 26, pp. 116–127, Jan 2007.
- [23] S. Roy Chowdbury, "Automated Segmentation and Pathology Detection in Ophthalmic Images," *Dissertation to obtain PhD Degree*, pp. 133–138, 2014.
- [24] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.
- [25] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. a. Viergever, "Multiscale vessel enhancement filtering," *Medical Image Computing and Computer-Assisted Intervention - MICCAI'98. Lecture Notes in Computer Science*, vol. 1496, pp. 130–137, 1998.
- [26] M. Staring, C. S. Xiao, D. P., and B. C. Stoell, "Pulmonary Vessel Segmentation using Vessel Enhancement Filters," no. 1, pp. 1–8.
- [27] D. Jimenez-Carretero, A. Santos, S. Kerkstra, R. D. Rudyanto, and M. J. Ledesma-Carbayo, "3D Frangi-based lung vessel enhancement filter penalizing airways," *Proceedings - International Symposium on Biomedical Imaging*, pp. 926–929, 2013.
- [28] J. Deng and X. Xie, "3D Interactive Coronary Artery Segmentation Using Random Forests and Markov Random Field Optimization Peninsular Radiology Academy, Plymouth University Schools of Medicine & Dentistry Plymouth Hospitals NHS Trust, United Kingdom Department of Computer S," pp. 942–946, 2014.
- [29] Y. Zheng, "Machine learning based vesselness measurement for coronary artery segmentation in cardiac CT volumes," *SPIE Medical Imaging 2011: Image Processing*, pp. 79621K–79621K–12, 2011.

- [30] M. Helmberger, M. Urschler, M. Pienn, Z. Bálint, A. Olschewski, and H. Bischof, “Pulmonary vascular tree segmentation from contrast-enhanced CT images,” *CoRR*, vol. abs/1304.7140, 2013.
- [31] M. Sofka and C. V. Stewart, “Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures,” *IEEE Transactions on Medical Imaging*, vol. 25, no. 12, pp. 1531–1546, 2006.
- [32] R. M. Rangayyan and S. Mukhopadhyay, “Detection of Blood Vessels in Retinal Fundus Images,” *Computer Science Journal of Moldova*, vol. 22, no. 2, pp. 155–185, 2014.
- [33] P. Mukhopadhyay and B. B. Chaudhuri, “A survey of hough transform,” *Pattern Recogn.*, vol. 48, pp. 993–1010, Mar. 2015.
- [34] S. A. Amiri, H. Hassanpour, M. Shahiri, and R. Ghaderi, “Detection of Microaneurysms in Retinal Angiography Images Using the Circular Hough Transform,” *Journal of Advances in Computer Research*, vol. 3, no. 1, pp. 1–12, 2012.
- [35] J. Pinão and C. M. Oliveira, “Fovea and optic disc detection in retinal images with visible lesions,” in *Technological Innovation for Value Creation*, pp. 543–552, Springer, 2012.
- [36] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
- [37] “<http://eucleas.blogspot.pt/2015/09/finding-total-coin-using-kmeans-and.html>.”
- [38] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [39] Y. Freund and L. Mason, “The alternating decision tree learning algorithm,” in *icml*, vol. 99, pp. 124–133, 1999.
- [40] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [41] “<http://www.cise.ufl.edu/ddd/cap6635/fall-97/short-papers/2.htm>.”
- [42] R. Kohavi and J. R. Quinlan, “Data mining tasks and methods: Classification: decision-tree discovery,” in *Handbook of data mining and knowledge discovery*, pp. 267–276, Oxford University Press, Inc., 2002.

- [43] B. Pfahringer, G. Holmes, and R. Kirkby, “Optimizing the induction of alternating decision trees,” in *Advances in Knowledge Discovery and Data Mining*, pp. 477–487, Springer, 2001.
- [44] E. Decencière, G. Cazuguel, X. Zhang, G. Thibault, J.-C. Klein, F. Meyer, B. Marcotegui, G. Quellec, M. Lamard, R. Danno, *et al.*, “Teleophta: Machine learning and image processing methods for teleophthalmology,” *IRBM*, vol. 34, no. 2, pp. 196–203, 2013.
- [45] “Exudate detection in color retinal images for mass screening of diabetic retinopathy,” *Medical image analysis*, vol. 18, no. 7, pp. 1026–1043, 2014.
- [46] M. Golabbakhsh and H. Rabbani, “Vessel-based registration of fundus and optical coherence tomography projection images of retina using a quadratic registration model,” *Image Processing, IET*, vol. 7, no. 8, pp. 768–776, 2013.
- [47] T. Mahmudi, R. Kafieh, H. Rabbani, M. Akhlagi, *et al.*, “Comparison of macular ocs in right and left eyes of normal people,” in *SPIE Medical Imaging*, pp. 90381W–90381W, International Society for Optics and Photonics, 2014.
- [48] S. H. M. Alipour, H. Rabbani, M. Akhlaghi, A. M. Dehnavi, and S. H. Javanmard, “Analysis of foveal avascular zone for grading of diabetic retinopathy severity based on curvelet transform,” *Graefe’s Archive for Clinical and Experimental Ophthalmology*, vol. 250, no. 11, pp. 1607–1614, 2012.
- [49] A. Budai, J. Odstreilik, R. Kolar, J. Hornegger, J. Jan, T. Kubena, and G. Michelson, “A public database for the evaluation of fundus image segmentation algorithms,” *Investigative Ophthalmology & Visual Science*, vol. 52, no. 14, pp. 1345–1345, 2011.
- [50] Y. A. Moe, *Machine learning appraoch to automatic exudate detection in retinal images of diabetic patients*. PhD thesis, Asian Institute of Technology, 2007.
- [51] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, “Textural features for image classification,” *Systems, Man and Cybernetics, IEEE Transactions on*, no. 6, pp. 610–621, 1973.
- [52] “<http://radio.feld.cvut.cz/matlab/toolbox/images/linfilt4.html>.”
- [53] A. A. Youssif, A. Z. Ghalwash, and A. S. Ghoneim, “Comparative study of contrast enhancement and illumination equalization methods for retinal vasculature

- segmentation,” in *Proceedings of the Third Cairo International Biomedical Engineering Conference (CIBEC'06)*, pp. 1–5, 2006.
- [54] H. A. Rahim, A. S. Ibrahim, W. Zaki, and A. Hussain, “Methods to enhance digital fundus image for diabetic retinopathy detection,” in *Signal Processing & its Applications (CSPA), 2014 IEEE 10th International Colloquium on*, pp. 221–224, IEEE, 2014.
- [55] S. H. Rasta, M. E. Partovi, H. Seyedarabi, and A. Javadzadeh, “A comparative study on preprocessing techniques in diabetic retinopathy retinal images: Illumination correction and contrast enhancement,” *Journal of medical signals and sensors*, vol. 5, no. 1, p. 40, 2015.
- [56] Y. Hatanaka, T. Nakagawa, Y. Hayashi, T. Hara, and H. Fujita, “Improvement of automated detection method of hemorrhages in fundus images,” in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 5429–5432, IEEE, 2008.
- [57] W. Wen-cheng and C. Xiao-jun, “A Segmentation Method for Uneven Illumination Particle Images,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 5, pp. 1284–1289, February 2013.
- [58] N. A. Ibraheem, M. M. Hasan, R. Z. Khan, and P. K. Mishra, “Understanding Color Models : A Review,” *ARPJN Journal of Science and Technology*, vol. 2, no. 3, pp. 265–275, 2012.
- [59] A. G. Marrugo Hernández, “Image analysis algorithms for feature extraction in eye fundus images,” 2009.
- [60] R. Bernardes, P. Serranho, and C. Lobo, “Digital ocular fundus imaging: a review,” *Ophthalmologica*, vol. 226, no. 4, pp. 161–181, 2011.
- [61] G. Jie and L. Ning, “An improved adaptive threshold canny edge detection algorithm,” in *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering - Volume 01, ICCSEE '12*, (Washington, DC, USA), pp. 164–168, IEEE Computer Society, 2012.
- [62] C. Pal, A. Chakrabarti, and R. Ghosh, “A brief survey of recent edge-preserving smoothing algorithms on digital images,” *arXiv preprint arXiv:1503.07297*, 2015.

- [63] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 629–639, Jul 1990.
- [64] K. He, J. Sun, and X. Tang, “Guided image filtering,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [65] K. N. Chaudhury, D. Sage, and M. Unser, “Fast bilateral filtering using trigonometric range kernels,” *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3376–3382, 2011.
- [66] C.-C. Kao, J.-H. Lai, and S.-Y. Chien, “Vlsi architecture design of guided filter for 30 frames/s full-hd video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 513–524, March 2014.
- [67] Z. Lin and X. Wang, “Dehazing for image and video using guided filter,” *Open Journal of Applied Sciences*, vol. 2, no. 04, p. 123, 2013.
- [68] D. Du, H. Lu, L. Zhang, and F. Li, “Visual tracking via guided filter,” in *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 1781–1785, Sept 2015.
- [69] F. Porikli, “Constant time  $o(1)$  bilateral filtering,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.
- [70] K. N. Chaudhury, “Acceleration of the shiftable  $o(1)$  algorithm for bilateral filtering and nonlocal means,” *Image Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 1291–1300, 2013.
- [71] T. Jerman, F. Pernuš, B. Likar, and Ž. Špiclin, “Beyond frangi: an improved multiscale vesselness filter,” in *SPIE Medical Imaging*, pp. 94132A–94132A, International Society for Optics and Photonics, 2015.
- [72] M. Gütlein, E. Frank, M. Hall, and A. Karwath, “Large-scale attribute selection using wrappers,” *2009 IEEE Symposium on Computational Intelligence and Data Mining, Nashville, USA*, pp. 332–339, April 2009.
- [73] C. S. Tan, S. R. Sadda, and S. M. Hariprasad, “Ultra-widefield retinal imaging in the management of diabetic eye diseases,” *Ophthalmic Surgery, Lasers and Imaging Retina*, vol. 45, no. 5, pp. 363–366, 2014.



[74] “<http://retinacuritiba.com.br/cirurgias/fotocoagulacao-a-laser/>.”

[75] “<http://www.ootech.nl/shop/lasers/navigated-laser/navilas.html>.”

[76] “<https://www.flickr.com/photos/mutiraoitabuna/5176530381>.”