1 2 9 0

UNIVERSIDADE D
COIMBRA

Bruno Daniel Oliveira Simões

Diabetic Retinopathy Diagnosis with Bag-of-Features and GPU-based Feature Extraction using SURF

VOLUME 1

Dissertation within the scope of Integrated Master's in Electrical and Computer Engineering, specialization in Computers, oriented by Prof. Doctor Gabriel Falcão Paiva Fernandes and presented to the Faculty of Science and Technology of University of Coimbra, Department of Electrical and Computer Engineering.

September 2022

FCTUC

UNIVERSITY OF COIMBRA
FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Diabetic Retinopathy Diagnosis with Bag-of-Features and GPU-Based Feature Extraction using SURF

Bruno Daniel Oliveira Simões

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering

Supervisor:
Prof. Doctor Gabriel Falcão Paiva Fernandes

Jury:
Prof. Doctor Hélder de Jesus Araújo
Prof. Doctor Luís Alberto da Silva Cruz
Prof. Doctor Gabriel Falcão Paiva Fernandes

Coimbra, September 2022

Agradecimentos

Em primeiro lugar, quero agradecer à minha família: mãe, pai e irmã, por toda a dedicação, sacrifício, apoio, amor e paciência ao longo de toda a minha vida. Sem a ajuda deles a realização deste sonho e a conclusão deste curso não seriam possíveis.

Depois gostaria de agradecer ao meu orientador, professor Gabriel Falcão, por toda ajuda, orientação e motivação que me deu para o desenvolvimento desta dissertação. Sempre disponível para me auxiliar no que fosse preciso.

Este trabalho foi parcialmente financiado pelo Instituto de Telecomunicações e Fundação para a Ciência e a Tecnologia, Portugal, sob as bolsas UIDB/EEA/50008/2020 e EXPL/EEI-HAC/1511/2021.

A todos os meus amigos de curso, o meu obrigado pela amizade durante este anos. Obrigado pelo companheirismo, pelo convívio, pela alegria e pelos momentos de entreajuda que tornaram o meu percurso uma experiência bem melhor.

A toda a gente com a qual convivi nos últimos anos e que de uma forma ou de outra estiveram comigo nesta caminhada e que, não sendo aqui referidos, contribuiram para a minha formação, aprendizagem e experiência de vida o meu agradecimento.

Por fim, lembrar também Coimbra, cidade única e ímpar no percurso académico de alguém. Fica a saudade de um tempo que não volta, mas que vai durar eternamente no meu coração. Coimbra é sonho, é tradição, é história, é fado... e quem passa por aqui não fica indiferente a tudo isso. Que bonito é poder dizer que se foi estudante da Universidade de Coimbra e que se partilhou as ruas desta cidade com este mar de gente vestida de capas negras.

Abstract

The amount of patients with Diabetic Retinopathy (DR) is expected to increase to 191 million by 2030 [1]. Color fundus images of the retina are used to diagnose DR. This analysis can only be done by expert clinicians, which is time-consuming and expensive. Computer vision techniques are essential to automatically evaluate these images and detect DR.

Extracting features from images is a field that has been growing and has many use cases in critical areas of society, such as image classification, object detection and segmentation, self-driving cars, and medical applications. The Bag-of-Features (BoF) method, also called Bag-of-Visual-Words (BoVW), combined with a Machine Learning Classifier can obtain satisfactory results in general image classification. Specifically, the BoF method can be applied to color fundus images of the retina to diagnose DR automatically, saving resources and helping the early treatment of the disease.

The influence of modern computing systems to develop real-time applications led to the adoption of parallel computing, particularly enabled by the increase in capabilities of the Graffic Processing Units (GPUs). Thus, Speeded-Up Robust Features (SURF) algorithm key functions can be implemented in GPU to substantially reduce the necessary time to extract and describe relevant image features.

Throughout this dissertation, the detection and description of local features utilizing SURF are explored. It also discusses and implements the BoF method for image classification. It serves as an input for an SVM to distinguish between classes of data, namely "disease" or "no disease". In addition, a debate on both advantages and disadvantages of the BoF process, when compared to the more recent Convolutional Neural Networks (CNNs), is also presented. It also discusses the concept of GPU-Computing and parallel programming. Its performance is compared against the commonly used CPU programming model.

Keywords

Speeded-Up Robust Features (SURF); Machine Learning (ML); Bag-of-Features (BoF); Convolutional Neural Network (CNN); Diabetic Retinopathy (DR); Computer Vision (CV); Support Vector Machine (SVM); Compute Unified Device Architecture (CUDA); Graphics Processing Unit (GPU);

Resumo

Estima-se que quantidade de pacientes com RD aumente para 191 milhões até 2030 [1]. Imagens coloridas do fundo da retina são usadas para diagnosticar RD. Essa análise só pode ser feita por médicos especialistas, o que é demorado e caro. Técnicas de visão computacional são essenciais para avaliar automaticamente essas imagens e ajudar o médico a detectar RD.

Extrair features de imagens é um campo que tem vindo a crescer e tem muitos casos de uso em áreas críticas da sociedade, como classificação de imagens, detecção de objetos e segementação, carros autónomos, aplicações médicas. O método Bag-of-Features (BoF), também chamado de Bag-of-Visual-Words (BoVW), combinado com um classificador de Aprendizem de máquina, consegue obter resultados satisfatórios na classificação imagens de um modo geral. Especificamente, o método Bag-of-Features pode ser aplicado em imagens coloridas do fundo da retina para diagnosticar DR automaticamente, economizando recursos e auxiliando no tratamento precoce da doença.

A influência dos sistemas modernos de computação para desenvolver soluções em tempo real levou à adopção de computação paralela, viabilizado principalmente pelo aumento das capacidades das Unidades de Processamento Gráfico (GPUs). Assim, as funções-chave do algoritmo Speeded-Up Robust Features (SURF) podem ser implementadas em GPU para reduzir substancialmente o tempo necessário para extrair e descrever características relevantes de imagens.

Ao longo desta dissertação, a detecção e descrição de "image features" locais utilizando o SURF em GPU são exploradas. Também se discute e implementa o método BoF para a classificação de imagens. Ele serve como entrada para máquinas de vetores de suporte (SVMs) de modo a distinguir classes de dados, que são, neste caso: "doença" ou "sem doença". Além disso, também é apresentado um debate sobre as vantagens e desvantagens do processo BoF, quando comparado com as Redes Neurais Convolucionais. Também é discutido o conceito de GPU-computing e programação paralela, sendo o seu desempenho comparado com o modelo de programação de CPU que é amplamente usado.

Palavras Chave

Speeded-Up Robust Features (SURF); Aprendizagem de Máquina (ML); Bag-of-Features (BoF); Rede Neuronal Convulacional; Retinopatia Diabética (RD); Visão Computacional; Máquina de Vetores de Suporte; Compute Unified Device Architecture (CUDA); Unidade de Processamento Gráfico;

"The important thing is not to stop questioning. Curiosity has its own reason for existing. One cannot help but be in awe when he contemplates the mysteries of eternity, of life, of the marvelous structure of reality. It is enough if one tries merely to comprehend a little of this mystery every day."

- Albert Einstein

List of acronyms

SURF	Speeded-Up Robust Features
AI	Artificial Inteligence
CNN	Convolutional Neural Network
ML	Machine Learning
BoF	Bag-of-Features
DR	Diabetic Retinopathy
CV	Computer Vision
SVM	Support Vector Machine
GPU	Graphics Processing Unit

Table of contents

List of figures

List of tables

Chapter 1

Introduction, Context and Motivation

Diabetic retinopathy (DR) is a disease caused by long-standing diabetes. It is the leading cause of blindness in the developed world, even though it can be treated effectively [4]. Of all the people with diabetes mellitus worldwide, approximately one-third have signs of DR, which equals roughly 95 million cases [5]. The diagnosis of DR is very important to diminish its impact on global society. Clinicians can identify DR by the presence of lesions associated with vascular abnormalities caused by the disease. Microaneurysms (red lesions), hard exudates (bright lesions), and neovascularization are the signals that allow us to identify an eye with DR.

Currently, detecting DR is a time-consuming and manual process. It requires a trained clinician to examine and evaluate digital color fundus photographs of the retina. Submission of their reviews occurs a day or two after the exam, leading to delayed results and lost follow-up, miscommunication, and delayed treatment of the patients [6]. Globally, diabetes is a disease that will continue to grow [7] [8], leading to more people being susceptible to developing DR. The human resources and infrastructure needed to diagnose DR and prevent blindness and visual impairments caused by this illness will become even more insufficient. Automatic detection is required to accelerate this process and reduce the health care costs of identifying this condition.

For us humans, it is an easy task to recognize different objects, landscapes, or even other people's faces. But for computers, this process is more complex and non-trivial. To identify objects in images we need to extract useful features which clearly define that same object. It may be a distinct color in the image or a specific shape such as a line, edge, or blob.

The extraction of relevant features in the image, matching the correspondent points between two images can be applied to a variety of fields, among which we highlight: object recognition, 3-D object and scene modeling, video tracking, gesture recognition, panorama stitching as well as robot localization and mapping [9]. Other examples of this can be found in [10], such as: registration of medical images [11], co-registration of very high resolution aerial / satellite images[12] [13] landing of an Unmanned Aerial Vehicle [14] [15], ship extraction [16], object recognition [17], extraction of image correspondences or bundle adjustment [18], [19].

A methodology used for detecting interest points in an image and then matching them is the Speeded-Up Robust Features (SURF) [20], which is both a detector and a descriptor. It evolved from SIFT (Scale-Invariant Features Transform) [21] and introduced some changes to make computations faster.

Nowadays, the filmed media pixel standard adopted up to $4k$ (3840×2160 pixels) resolution, which is exactly four times the pixel count of $Full\ HD$ (1920×1080 pixels). Also, $8k$ (7680×4320 pixels) represents again four times the number of pixels as $4k$ or 16 times that of a Full HD picture. The benefit of higher resolutions, such as 8k, lies in the fact that the human eye begins not to distinguish the pixels from a much closer distance of the screen. Zooming in the picture without resulting in a big loss of detail is also an improvement to take in mind. But as the pixel count goes up, so does the computational effort required to work with these high-resolution images.

The SURF algorithm presents some steps with considerably heavy computations, leading to an increased computational time that scales linearly with the size of images used when computing with Central Process Units (CPUs). When working with large-scale images or video we can no longer keep up with real-time calculations, since the number of processed frames per second (fps) is too low.

In this thesis, we propose to boost SURF performance by taking advantage of GPGPU computing. It uses the Graphics Card or Graphics Processing Unit (GPU) to compute parts of the code where we can apply data-level parallelism. This enables us to build a faster parallel solution for the more resource-intensive tasks of this algorithm. We build a BoF algorithm to help diagnose Diabetic Retinopathy with an automatic procedure, using the GPU-SURF version as the basis of feature extraction required to find key points and feed the BoF model.

CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on GPUs and we will use it as the framework to execute a GPU version of the SURF algorithm.

To illustrate the computational complexity of these algorithms, the image in figure 1.1 with 4k resolution, running the SURF algorithm on CPU, took 2.29 seconds to compute one frame. The processor used was an i7-4720hq on a system with 16GB of RAM, running on Ubuntu. It detected 15171 points of interest.

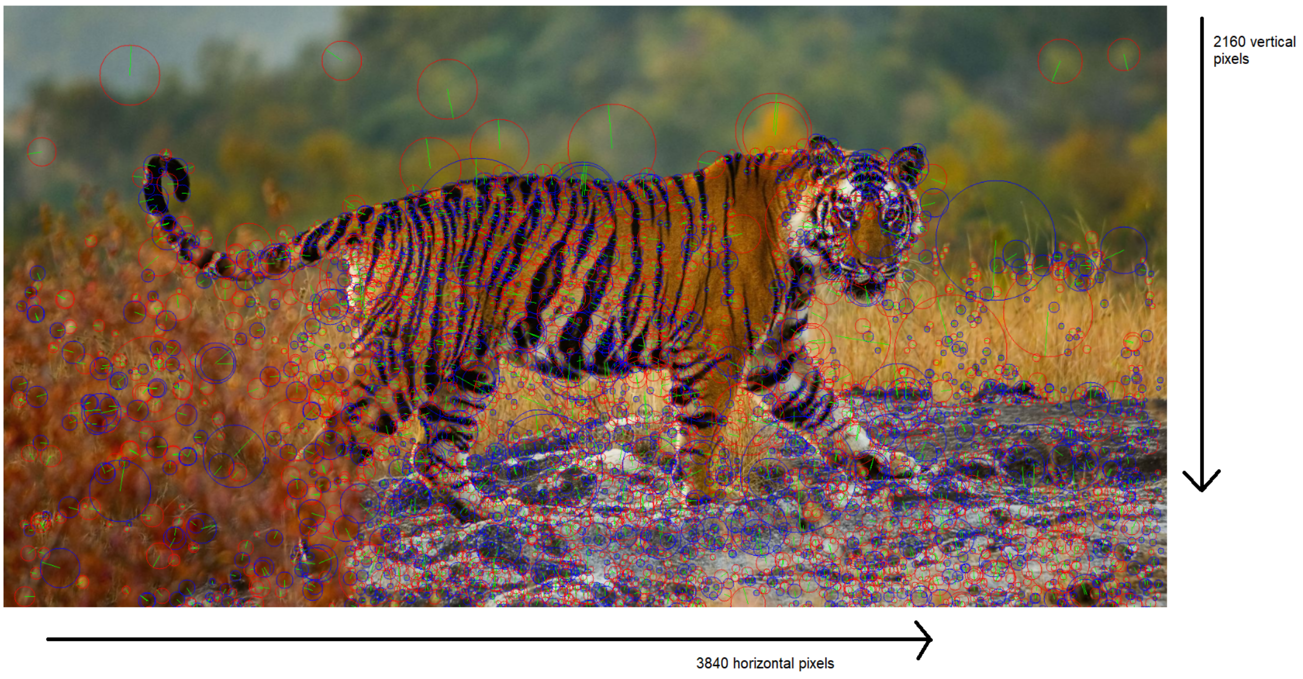


Fig. 1.1 Running SURF on a 4K resolution image

1.1 Objectives

In this thesis, the goal is to automatically diagnose Diabetic Retinopathy and classify high-resolution images of the retina into healthy or with the disease. We also want to accelerate the SURF algorithm using GPU computing to improve the method's performance and efficiency. A BoF model based on GPU-SURF feature extraction algorithm and SVM classifier is constructed. The main objectives are:

- Become familiar with the SURF and BoF algorithms, as well as the SVM classifier.
- Tweak and test a SURF-CPU version.
- Tweak and test a SURF-GPU version.
- Benchmark and compare both implementation performances.
- Implement a BoF method and classify its output using an SVM.

- Train our model with the high-resolution Diabetic Retinopathy images chosen from the EyePACS dataset.
- Further explore the method and optimize its parameters for better and more accurate results.
- Compare the BoF method with the CNN approach for DR automatic detection.

1.2 Dissertation Outline

This thesis is organized into 6 chapters. The content of each chapter is the following:

- ***Chapter 1*** introduces the topic of the thesis, the motivation behind it, and the main goals this work intends to achieve;
- ***Chapter 2*** presents the related work;
- ***Chapter 3*** describes the Bag-of-Features implementation to process our DR color fundus images and dissects the structure of the SURF algorithm.
- ***Chapter 4*** presents and explains the differences between the CPU and GPU architectures. The list of CUDA functions that run in parallel to accelerate GPU-SURF is also mentioned;
- ***Chapter 5*** provides the experimental results and their discussion;
- ***Chapter 6*** concludes this thesis and presents some suggestions for future work;

The main contributions of this dissertation are:

- The implementation of an automatic procedure to diagnose DR with the BoF method;
- Prove that a GPU-SURF-based solution instead of its CPU counterpart can improve the overall execution time and efficiency of the algorithm, especially for high-resolution images;
- Demonstrate that the BoF procedure can be a competitor to CNNs for DR diagnosis under certain conditions and constraints;

Chapter 2

Related Work

This chapter presents the related work of SURF (and SIFT), Bag-of-Features, and Machine Learning techniques to improve feature extraction. It goes through numerous literature publications related to the topics mentioned above, exploring both CPU and GPU-based implementations.

2.1 Feature extraction

Feature extraction is a necessary tool for many computer vision applications, as referred to earlier. Picking out only the most salient points that can be regularly localized across different images, vastly reduces subsequent data processing. However, feature extraction remains a considerable bottleneck for many implementations when the image's resolutions being computed increase to modern levels. For example, SURF running on a CPU cannot process a sufficient number of frames per second to become an alternative for real-time computer vision tasks. One possible solution is to accelerate this algorithm by making use of Graphics Processing Units (GPU).

Recently, GPUs have been extensively studied. They are used as general-purpose processing devices due to their high number of cores and SIMD (single instruction, multiple data) parallel hardware structure [22]. Consequently, developers can exploit GPUs to assure real-time performance in applications related to computer vision that work with large amounts of data.

CUDA is a framework and programming language created by NVIDIA designed for general-purpose computing on the GPU, exploiting its parallel architecture. It minimizes the need for understanding the Graphics pipeline and makes it easier for the programmers to use GPUs as a development tool [23].

2.2 Using SURF and SIFT on CPU for Feature Extraction

Kai Cordes et al. [24] tested several detectors and descriptors for feature extraction. It establishes that the performance of some of them is different on distinct image resolutions. Thus, this is important to consider for future evaluations. SURF and SIFT are ranked $3rd$ out of 6 descriptors examined in their benchmarks.

Diverse papers compare the SIFT and SURF feature extraction methods. SURF is considered to be superior for the most part, but that depends on the assignment. Gowda et al. [25] take advantage of SURF for image stitching and its authors affirm it is more robust and can stitch relative images much more quickly than the SIFT.

The recognition of the eye's iris has the potential for a high level of accuracy due to its uniqueness and stable feature. Mathew et al. [26] perform a comparison of the SIFT and SURF algorithms to analyze their speed and precision on this specific task. Using both the above algorithms combined, finding the matching key points was faster than using the SIFT method alone.

Pui and Mini [27] test these two detectors on face feature extraction. Based on their experiments for this task, SIFT is better in terms of computational speed time. Furthermore, SIFT detects more points of interest. Nevertheless, the points extracted by SURF are located mainly at the distinct facial features, therefore being a more robust descriptor.

SURF's original algorithm code is patented. Even so, since its presentation in 2006 [20] various open-source implementations emerged. Gossow et al. [28] measured different SURF versions written in C++ and compared the results to the one built by SURF authors. The authors also expanded one of these open-source deployments to support multi-threading, resulting in up to 5.1 times faster calculation time on a machine with an 8-core $3.0GHz$ CPU with $16GB$ RAM.

Surekha et al. [29] developed their SURF project and compared the number of interest points found for various scales of octaves, thresholds, and resolutions. The experiments indicate that the increase in the image size will elevate the number of key points detected with different octaves. Moreover, increasing the threshold applied lowers the number of interest points identified.

An improved version of SURF called MU-SURF was created by Motilal Agrawal et al. [30]. It uses a larger window size on the descriptor. The sub-regions used to compute the Haar wavelet responses also increase in size. M. Iturbe and K. Olga [31] judge MU-SURF and SURF performance, utilizing the image set proposed by Mikolajczyk et al. [32]. The descriptor's success is verified for image rotation, scale change, viewpoint change, blur, JPEG compression, and illumination changes. The experiments manifest that MU-SURF performs better in almost all of the above cases and can become an alternative option.

2.3 Using SURF and SIFT on GPU for Feature Extraction

Sinha et al. [22] accelerate SIFT and Kanade-Lucas-Tomasi (KLT), a method for feature tracking, on GPU. Their implementations run 10 to 20 times faster than the corresponding optimized CPU counterparts and enable high-resolution video processing in real time.

Panorama stitching consists of a junction of image frames to create a 360 degree view. Cortés-Gallardo et al. [33] explore this technique in autonomous vehicle driving with the help of several cameras. This strategy requires a fast and robust feature set obtained from the images captured. The choice lies within SIFT, BRISK, and SURF feature extractors. SIFT was discarded due to its lack of velocity when running. When comparing BRISK and SURF, both methods have similar accuracy performance. Despite that, SURF is superior because it is two times faster than BRISK for panorama stitching and reconstruction. Even so, it is still not fast enough. Therefore, their solution was to accelerate the process through parallelization by executing the code on a GPU architecture. To complete this, the CUDA toolkit and OpenCV libraries were used in a C/C++ application, resulting in a speedup of 2 times.

Furgale et al. [34] use the SURF algorithm for real-time robot navigation. Despite it being much faster than SIFT (SURF is about as fast as GPU-accelerated SIFT, according to the authors), it is still not capable of providing real-time performance. Thus, their goal was to benefit from the GPU architecture to accelerate the SURF algorithm for levels of real-time robot navigation. This was the first implementation of SURF to use CUDA, reaching close to a 2.5 speedup in 512×384 images.

Mehrez et al. [35], compare their implementations of SURF and SIFT on GPU to previous CPU and GPU parallel execution of the two algorithms, concluding that they can do all the processing in real-time and still preserve high accuracy. They further achieve higher speedup, frame rate, and SM occupancy than the previous best-known parallel implementations of the two algorithms.

Fingerprint authentication is a growing authentication system, but that requires local invariant feature extraction algorithms to be successful. With an eye on that, Awad et al. [36] utilizes SIFT and SURF to complete that laborious task. CPU and GPU versions of both descriptors are benchmarked, with the spotlight focused on the consumed time of both descriptors. The experimental results proved the superiority of the SURF feature detector compared to SIFT. Moreover, capitalizing on the GPU capabilities for both of them was confirmed to be more efficient. In future work, optimizing the SURF feature detector to work completely on the GPU was considered fundamental to avoid the data transfer between the CPU (host) and the GPU (device) and increase the speedup.

Wanglong Yan et al. [37] adopted an open-source SURF version called OpenSURF [2] to Open Computing Language (OpenCL) and General-Purpose GPU. Their paper depicts the differences in approach of both techniques, accomplishing significant performance improvement compared with the original CPU version.

Novel feature extraction methods were developed in recent years, as is the case of Feature Group Matching (FGM). This modern mechanism is strongly grounded on the SURF and SIFT algorithms. Marinelli et al. [38] forged and tested their CPU and GPU (CUDA) frameworks of SIFT and SURF before implementing FGM, on a system with Intel $i7$ CPU, $8GB$ RAM, and an NVIDIA $gt650M$ GPU, running on Ubuntu with the assistance of CUDA, OpenCV, and SiftGPU libraries. The outcome exhibits a 3.5 speedup factor on SIFT and 1.7 on SURF. Regardless of the higher boost on SIFT, the overall time that it requires to complete is still larger than SURF, therefore it is much more computationally expensive.

Affine-SIFT(ASIFT) is a modified version of the original SIFT algorithm fully invariant to affine image transformations. Valeriu Codreanu [39] built an ASIFT-GPU (CUDA) deployment that achieves up to $70x$ speedup concerning the CPU version when applied to high-resolution $5M$ Pixel images.

Lalonde et al. [40] constructed a real-time eye blink detection program with GPU-based SIFT tracking that reaches about 25 frames per second. Terriberry et al. [41] illustrates the Speeded-Up Robobust Features algorithm's translation to the GPU platform in detail, with several innovative optimizations, obtaining over $30fps$ at HD resolutions and $70fps$ at SD resolutions.

Fassold and Rosner [42] make use of NVIDIA CUDA capabilities and design a SIFT GPU version. Later on, they compare their SIFT GPU work with the reference CPU implementation of SIFT in terms of runtime and quality score. The results reveal that the speedup achieved by the GPU implementation increases with both resolution and the number of key points detected. The experiments unveil a speedup factor of approximately $4-5$ for SD (720×576 pixel resolution) and $5-6$ for Full HD (1920×1080 pixel resolution) video referring to a multi-threaded CPU implementation, permitting them to run the SIFT descriptor extraction algorithm in real-time on SD video.

Cornelis et al. [43] formulated a CUDA-GPU version of the Speeded-Up Robust features algorithm. As a result, their work can be applied to image sequences with 640×480 pixels and pull good results, exceeding 100 frames per second.

2.4 Combining the use of Machine Learning and SURF Feature Extraction

Breast cancer is the first cause of death among women [44]. Fanizi et al. [45] propose a novel method of identifying this type of disease. Most breast cancers coincide with the presence of microcalcifications (MCs) grouped in clusters. The authors of this publication recurred to SURF for extraction of interest points that help to classify a breast region, performed through machine learning techniques, in order to categorize clusters of MCs in digital mammograms.

Horak et al. [46] detected interest points in a set of images by the well-known SURF method. Then they employed supervised learning algorithms to create relevant models of corners in images.

Majumdar et al. [47] operate the SURF (and SIFT) methods for extraction of features in images of leaves taken from a dataset. Then a training stage, a clustering algorithm, and a decision tree are executed to identify different types of leaves.

Dwi Pranata et al. [48] proposes a novel computer-assisted method for automated classification and detection of fracture locations in computed tomography (CT) images for calcaneus fracture using a deep learning algorithm combined with SURF. This Convolutional Neural Network (CNN) deployment can achieve 98% accuracy.

Sergieh et al. [49] investigate reducing the number of generated SURF features to speed up image matching while maintaining the same performance. They use a supervised learning approach that exploits a binary classifier to identify useful keypoints for the matching process, effectively reducing the matching runtime.

2.5 Bag-of-Features for Image Classification

In 2016, Al-dmour et al. [50] describes a novel approach for the offline recognition of handwritten Arabic words, making use of Bag-of-Features and SURF. The classification was performed with a Support Vector Machine using a radial basis function (RBF). The success rate was 85%.

In an attempt to improve blind people's quality of life, Petaitris et al. [51] built an environment recognition system using BoF with a recorded accuracy of 84%.

Murtaza et al. [52] utilized BoF to automatically identify species of plants with 94% accuracy on the LeafScan dataset.

Classification of wood skin defects based on the BoF model was implemented by Yang et al. [53]. Their extraction of the image features was done with a histogram of gradients (HoG) descriptor, achieving 85% of the average recognition rate.

M. Luísa et al. [54] utilized a combined BoF and SVM framework approach for the classification of motile sperm cells. SURF and HoG feature extraction methods performance is compared, achieving 90% and 78% average accuracy respectively, with 2325 clusters used in the k-means clustering and 235 images of each class

Real-time vehicle detection and classification from on-road videos using BoF was created by Sajib et al. [55] with 94% average accuracy.

Hatipoglu et al. [56] implemented a system for gender recognition from Facial images using SURF based BoF Method. It has an average 96% accuracy rate, on the FERET database.

American sign language recognition framework has been successfully implemented on a smartphone platform by Cheok Ming Jin et al. [57]. The average accuracy obtained was 97.13% using SURF and with SIFT it was 92.25%. SURF running speed was also higher in comparison with SIFT.

Jinlin Ma [58] developed a method of protein model classification and retrieval using Bag-of-Visual-Features based on SURF descriptor, achieving 89% matching correctness.

Brain tumor detection and classification into malignant and benign based on BoF and SVM classifier was studied by Irfan Mehmood et al. [59], resulting in average 99% accuracy.

A Bag-of-Features system with SURF descriptor was constructed for Content-Based Image Retrieval (CBIR) for the retrieval of high-resolution satellite images by Bouteldja et al. [60]. The precision highly varies from class to class with an average of 53%.

Bag-of-Features and Support Vector machine-based early diagnosis of skin cancer by a computer-aided detection system was implemented by Ginni Arora et al. [61]. The proposed method shows an accuracy of 85.7% and a training time of 0.8507s in classifying the lesion into malignant or benign, for a quadratic type of SVM kernel.

Sijie Niu et al. [62] applied a Machine Learning (ML) algorithm to detect autism spectrum disorder (ASD) in functional magnetic resonance imaging (fMRI). It is based on Bag-of-Features, SURF, and SVM. Testing 3 different datasets, the highest accuracy scored was 81% and specificity 86%.

A comparison of convolutional neural network and bag of features for multi-font digit recognition was forged by Ibrahim et al. [63]. The recognition accuracy produced by BoF is just slightly lower than the CNN (94% vs 96%).

Breast cancer is one of the leading causes of death by cancer among women[64]. Amara Nedra et al. [65] presented an approach to detect tumors in mammogram images, using BoF based on SURF interest point for the extraction of features from the segmented tumor region. Their results report 99% accuracy in malignant and benign tumors. On the same topic, Falcão Matos et al. [64] proposes a method for discriminating between malignant and benign breast cancer. The method comparatively applies several descriptors for local feature extraction. Testing with SURF extraction for the BoF method and SVM classification shows around 81% accuracy. The highest classification accuracy result was achieved using SIFT-based extraction.

A hand gesture recognition applying SURF-BOF was constructed by Zang et al. [66], reporting 88.5% accuracy on the SVM classification. They also modified SURF by gray threshold segmentation to eliminate invalid feature points. This improved efficiency and performance of the model, resulting in 97.5% accuracy.

Detecting potential aerial threats like drones with computer vision and discriminating them from birds and backgrounds is important for security purposes. A variation of BoF based on SURF was created by Unlu et al. [67], with 78.21% accuracy recorded.

Bhatt et al. [68] produced a content-based image retrieval (CBIR) solution using Bag-of-Features for lung cancer. Their work demonstrates an algorithm with 98.56% testing accuracy results. On the same topic of medical image retrieval, Lahari and Krupa [69] developed a system to classify and search for X-Ray images. Analyzing them with BoF and extracting features using principal component analysis (PCA), wavelet transform, and SURF descriptors. The highest accuracy results (92.25%) were accomplished using SURF on a 5-fold cross-validation approach for SVM classification.

Nguyen et al. built an android application for [70] automatic plant identification method using SURF in combination with BoF and supervised learning with 95.94% accuracy.

A Bag-of-Features (BoF) representation using SURF for classification of images from a different spectral band of melamine-faced panels, which can appear through the production process, was built by Aguilera et al. [71] with a 94% accuracy on detecting panel defects.

An object detection within an X-ray baggage security screening solution was explored by Kundergorski et al. [72], reaching 94% maximum accuracy. Their method compares several feature descriptors to feed a Bag-of-Features, with SURF performing the best and being more computationally efficient.

K. Suh et al. [73] designed a system for sugar beet and volunteer potato classification using the Bag-of-Features (BoF) model and reached 94.5% accuracy with SURF feature extraction and SVM.

Chapter 3

Design and Development of SURF, BoF algorithm and SVM Classifier

In this chapter a detailed explanation of the concepts required to implement the BoF method for automatic DR diagnosis is presented. The designs of the SURF algorithm and the SVM classifier are also reviewed, since they are part of the BoF procedure. The chapter starts by showing the differences between a Normal Retina and a Retina with Diabetic Retinopathy condition as they are the input images for our work.

3.1 Diabetic Retinopathy Signs

While a Normal Retina color fundus image has three main components: optic nerve, macula, and blood vessels, a Retina with signs of DR has other additional details, namely: Hard Exudates (bright lesions), Microaneurysms (red lesions), and New Blood Vessels growth (Neovascularization). Our focus will be on detecting the first two above because the SURF algorithm excels in detecting this kind of image structure, contrary to long edges present in blood vessels. The image below shows a healthy eye and a DR retina:

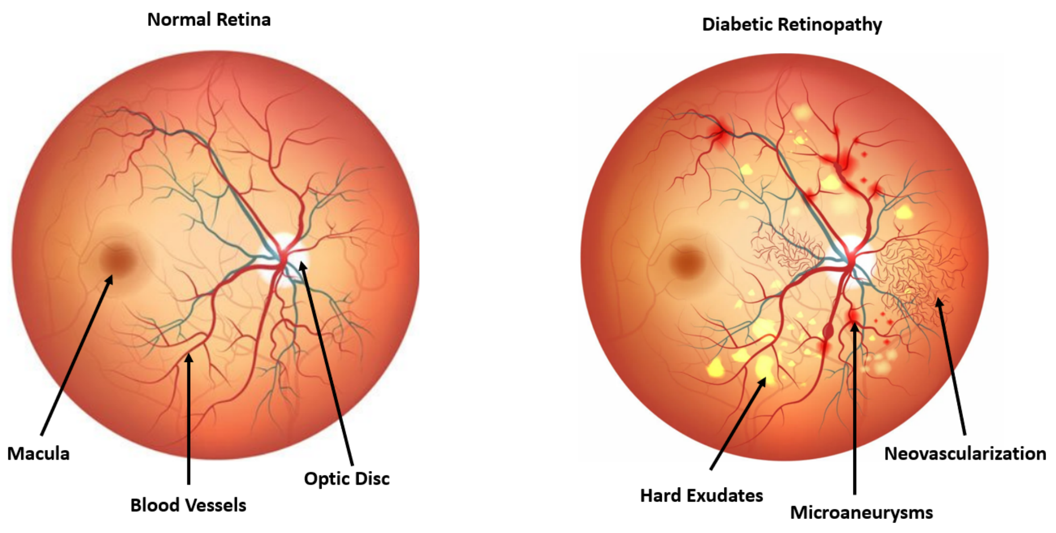


Fig. 3.1 Difference between Normal Retina and an eye with Diabetic Retinopathy

3.2 Bag-of-Features Method

Image classification is applied to label an image according to the class it represents. Bag-of-Features (BoF), also called Bag-of-Visual-Words, is a technique employed for image classification inspired by Bag-of-Words (BoW), a method applied to identify similar text messages. BoW scans the whole document and counts the number of times an individual word is present to construct a word frequency histogram. This histogram is the "signature mark" of the text document and is used to compare different documents. Similar histograms might indicate that we are in the presence of the same text document or the same subject. BoW does not carry any spatial information, hence the "bag" word that relates to unordered space.

In the field of visual information, there is no notion of words as there is in text information. In the digital format, an image is a matrix of numbers representing pixel intensities. To build a structure equivalent to a word in an image, feature extraction is utilized. The goal is to identify a region of neighbor pixels, called patch, that forms an area of interest, helping us to identify the content present in the picture. These regions are the output of the SURF algorithm.

After SURF feature extraction, a given image is now displayed as a variable size set of unordered local features [72]. Nonetheless, a Support Vector Machine (SVM) requires uniform dimensions of feature vectors as its input [57]. Bag-of-Features is therefore implemented. The BoF algorithm for automatic Diabetic Retinopathy diagnosis is simplified in the following diagram:

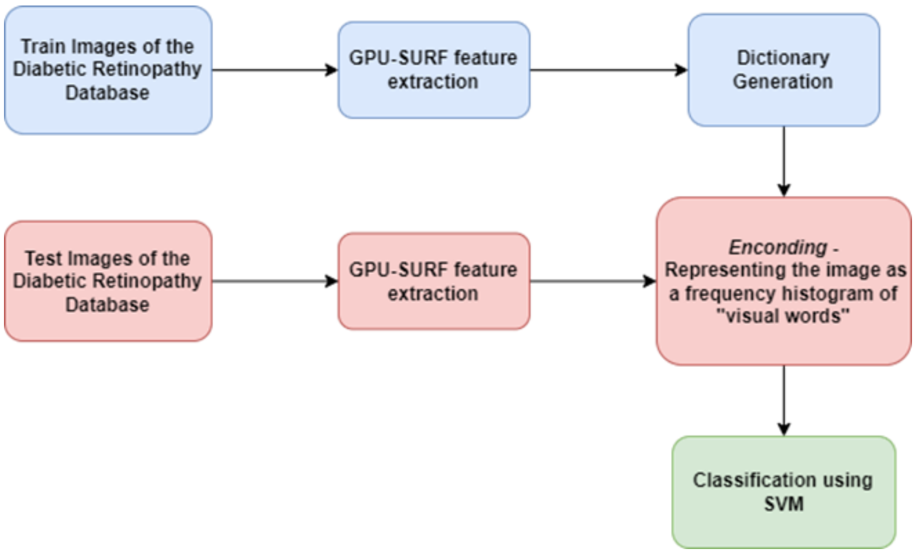

Train Images of the Diabetic Retinopathy Database
GPU-SURF feature extraction
Dictionary Generation
Test Images of the Diabetic Retinopathy Database
GPU-SURF feature extraction
Enconding - Representing the image as a frequency histogram of "visual words"
Classification using SVM

Fig. 3.2 Simplified BoF scheme

The upcoming diagrams detail the four main steps of the BoF approach for automatic DR diagnosis:

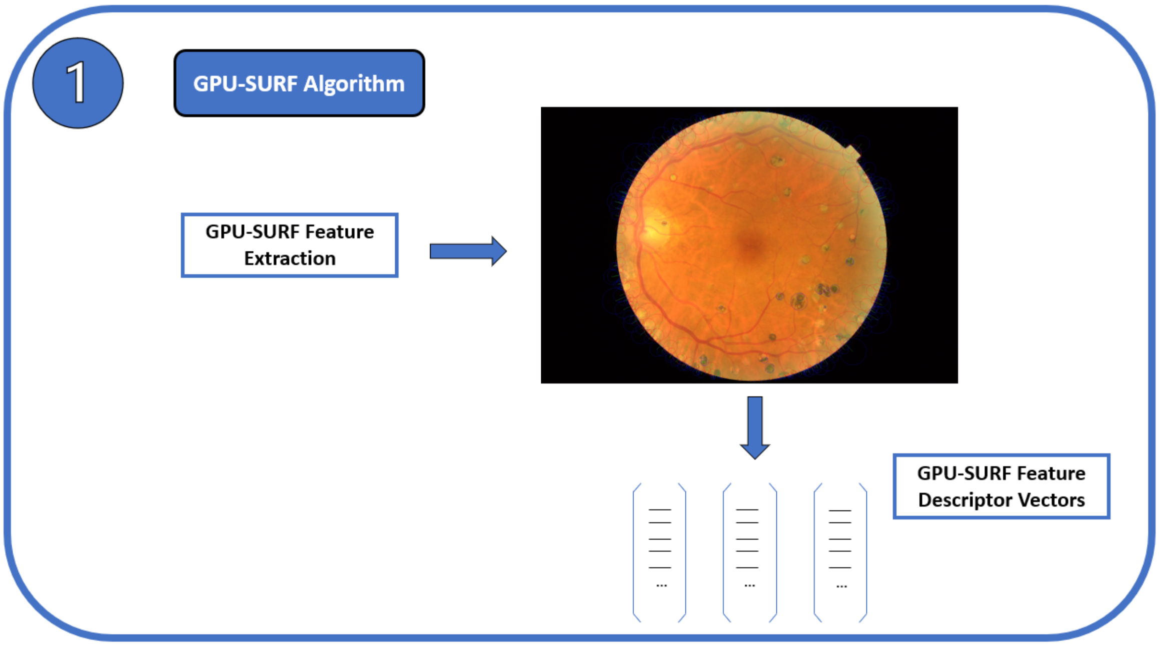

1
GPU-SURF Algorithm
GPU-SURF Feature Extraction
GPU-SURF Feature Descriptor Vectors

Fig. 3.3 GPU-SURF Feature Extraction Stage of BoF

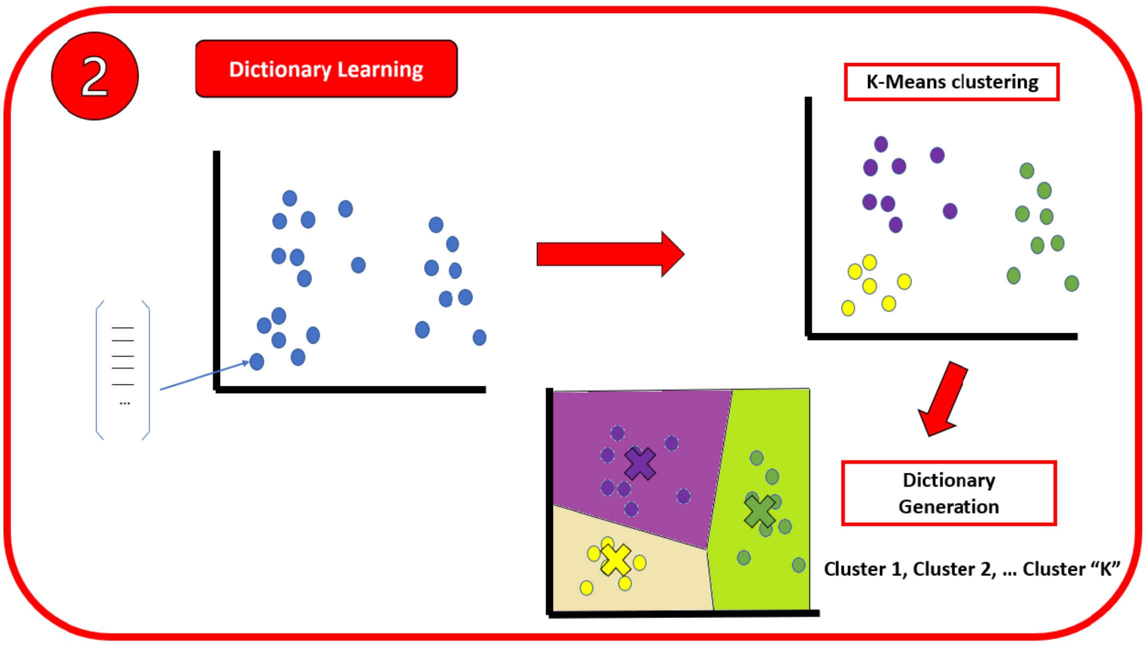


Fig. 3.4 Dictionary Learning Stage of BoF

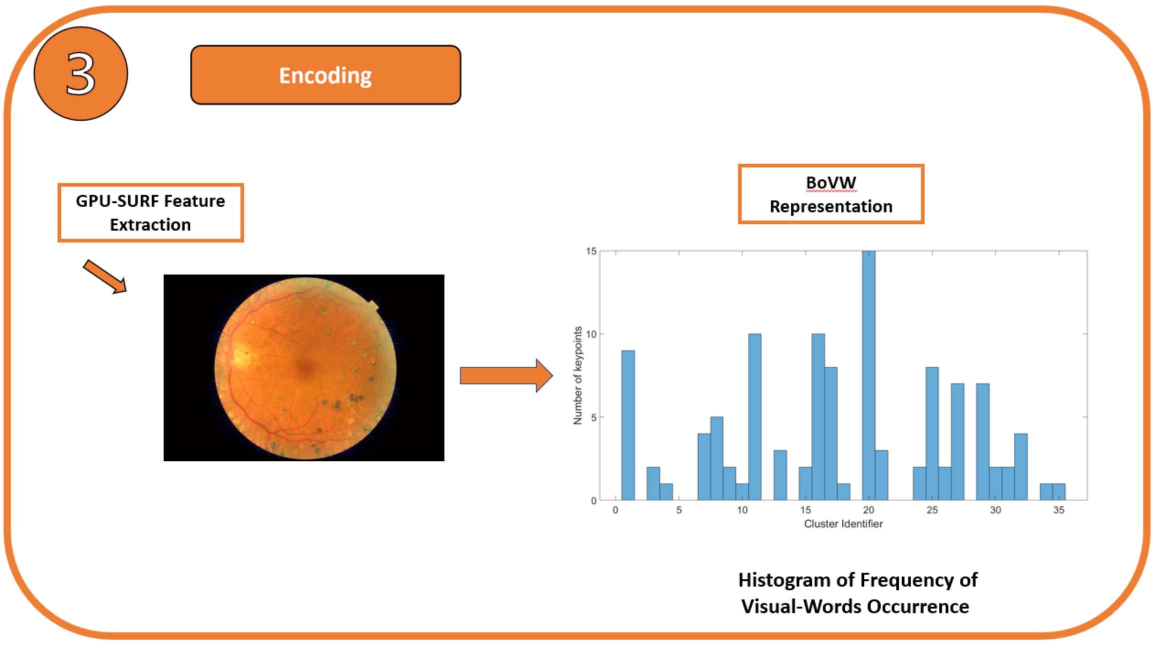

3
Encoding
GPU-SURF Feature Extraction
BoVW Representation
Number of keypoints
Cluster Identifier
Histogram of Frequency of Visual-Words Occurrence

Fig. 3.5 Encoding Stage of BoF

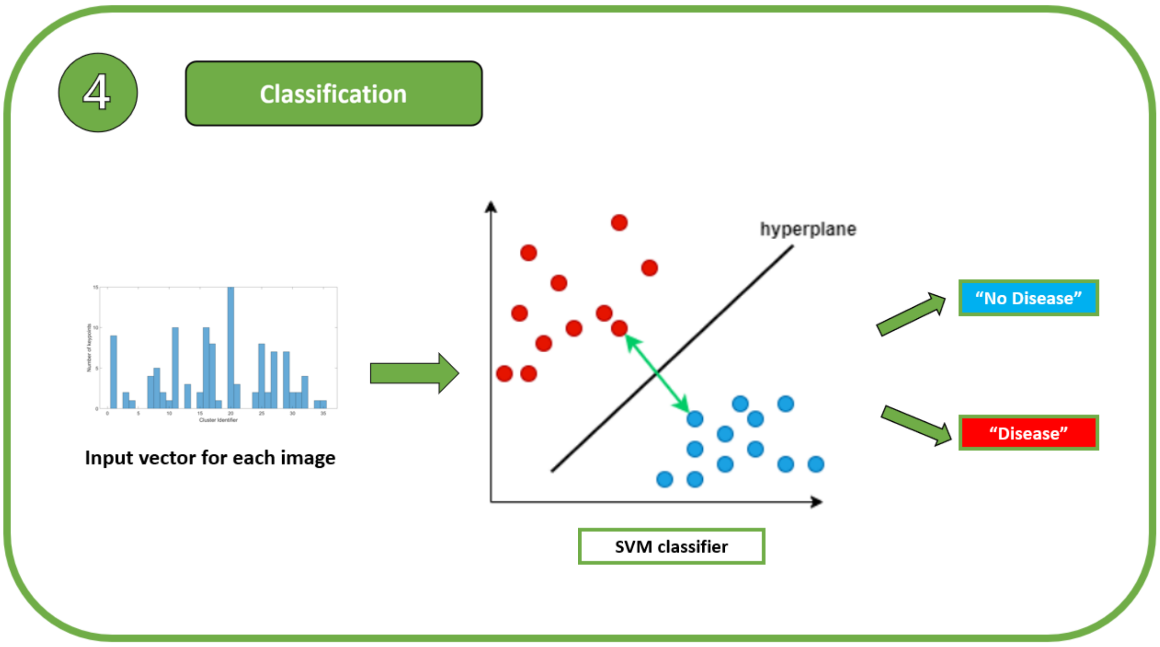


Fig. 3.6 Classification Stage of BoF

Now we will describe in detail the 4 stages of the BoF method (SURF algorithm, Dictionary Learning, Encoding, and Classification).

3.3 SURF Algorithm

The first step of the BoF method is the image feature extraction. Speeded-Up Robust Features (SURF) is both a descriptor and a detector, published at the 2006 European Conference on Computer Vision [20]. Its authors claim to have improved the SIFT (Scale-Invariant Feature Transform) descriptor performance, presented two years earlier [21] by David G. Lowe. SURF has stayed relevant and it is still used nowadays due to its robustness at different image transformations, being invariant to rotations or changes in brightness of some degree and it was our choice to perform feature extraction of DR images. In the coming subsections, this method will be discussed and presented in detail.

3.3.1 Hessian Matrix

The SURF detector is rooted in the determinant of the Hessian matrix. It uses its determinant for selecting both the location and the scale of the key points in the image.

$$H(f(x,y)) = \begin{bmatrix} \frac{\delta^2 f}{\delta x^2} & \frac{\delta^2 f}{\delta x \delta y} \\ \frac{\delta^2 f}{\delta x \delta y} & \frac{\delta^2 f}{\delta y^2} \end{bmatrix} \tag{3.1}$$

As SURF focus on finding keypoints in both space and scale location, then the Hessian matrix can be seen as:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{3.2}$$

where, for example, *Lxx*(*x*, σ) is the convolution of the Gaussian second-order derivative (in *x* direction with a scale of σ) with the image *I* in a point at the location *x*,*y*. The second-order order derivatives can be approximated by Laplacian of Gaussians (LoG). SURF takes this approximation even further, using Box filters [74]. They can be evaluated very fast using integral images, independently of size, as referred later in this thesis.

The determinant of the Hessian is then calculated by this approximation [20]:

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{3.3}$$

Note that *Dxx*, *Dyy*, and *Dxy* are the box filters mentioned above.

In [75] Difference of Guassian (DoG) and Box filters are compared and the results presented demonstrate that the benefits of using box filter representations for the second-order derivative of Gaussians include a considerable increase in efficiency and speed of computation with a negligible accuracy loss. The determinant here is referred to as the blob response at location x = (*x*, *y*, σ). The search for local maxima of this function over both space and scale yields the interest points for an image [2].

3.3.2 Constructing the Scale-Space

The images are progressively smoothed with Gaussians (box filters). The filters increase their size at each iteration, allowing SURF to encounter features at different scales. That is called a Filter Pyramid. For example, a 9×9 filter corresponds to σ = 1.2 (initial scale) and a 27×27 filter corresponds to $\sigma = 1.2 \times 3 = 3.6$ [27]. Due to the nature of box filters and integral images, SURF is a fast algorithm and we will be able to explore parallel filtering computation at different layers of scale-space when we implement our GPU version of the algorithm.

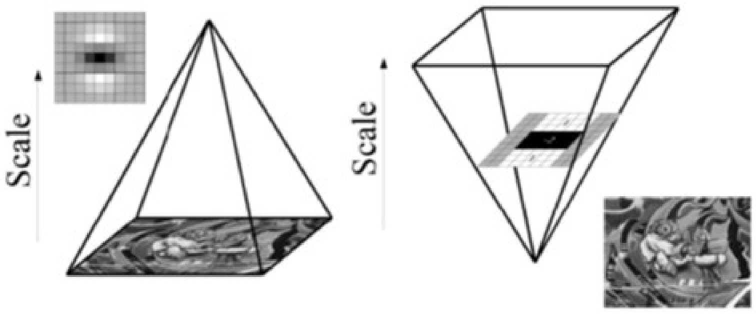


Fig. 3.7 Gaussian and filter pyramids (image from [2])

To create a scale space of images it is generated a set of progressively blurred images. Subsequently, the differences of that set of blurred images are computed. Doing so is called a "Difference of Gaussian" pyramid (similar to the Laplacian of Gaussian).

3.3.3 Keypoint Localization and Scale (Extrema Detection)

The next goal is to find scale and rotation invariant interest points. Initially, the response is thresholded, keeping only the points of interest more important. The threshold value depends on the application. The responses are collected across all scales. Next on the list, a Non-Maximum suppression in a 26 neighborhood around the central pixel is calculated. All the 8 pixels around the pixel at focus in the same scale are checked, together with the 9 pixels in the immediately upper and lower scales. If that pixel is a maximum or minimum in its local neighborhood then it is classified as a point of interest in the image [2].

The final step corresponds to a sub-pixel localization of the interest points, in both scale and space. This is done by expanding the determinant of the Hessian function in a Taylor series, differentiating it, and setting it to zero, accepting only the key points that converge for the interpolated localization of them, creating a more robust and smaller set of features to work.

3.3.4 Orientation Assignment of the Interest Point Descriptor

To compare two distinct key points they must be aligned with each other. Although, most of the time, corresponding features in two images don't have the same orientation, it is necessary to align the regions around the key points to make sure our descriptor is invariant to rotation as proposed by the SURF authors. Starting by considering a region of radius $6 \times s$ in the neighbourhood of the key point, where s is the scale that feature was detected, a

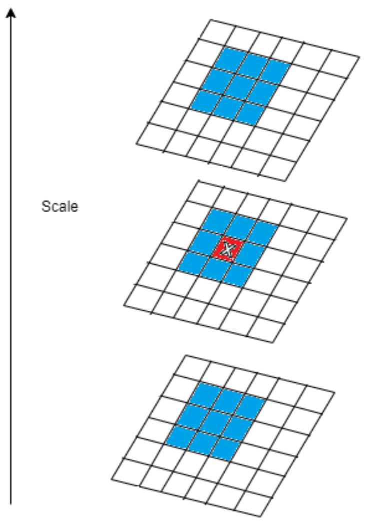


Fig. 3.8 : Non-Maximal Suppression. The pixel marked 'X' is selected as a maximum if it greater than the 26 surrounding pixels immediately above and below.

grid of equally distant points is acquired, forming a region with around 100 points near that interest-point. The specific set of pixels is established by sampling those from within the circle using a step size of σ. This process makes SURF scale-invariant. Afterward, for all of the points in that circular region, its gradients in the x-direction and y-direction are calculated. Haar Wavelet transforms are used to increase robustness and decrease computational time [2].

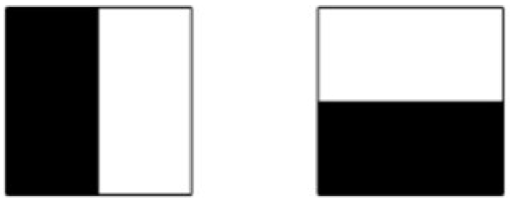


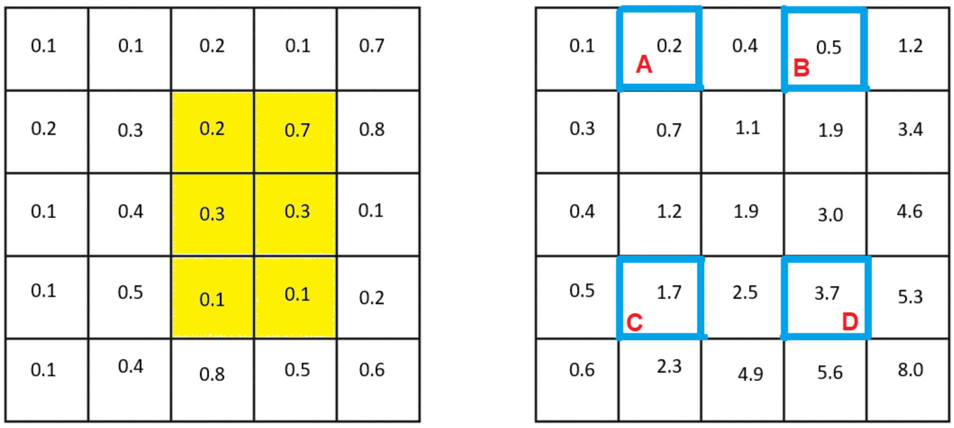
Fig. 3.9 Haar Wavelets for the x-direction and y-direction filtering responses. Black region weight is 1. White region weight is -1. Each wavelet response needs only six operations to be determined when used in conjunction with integral images.

The size of the Haar wavelet is $4 \times s$. It represents the sum of the pixels in the white region minus the sum of the pixels in the dark region.

$$\Delta = white - dark = \frac{1}{n} \sum_{x \varepsilon white}^{n} I(x) - \frac{1}{n} \sum_{x \varepsilon black}^{n} I(x) \tag{3.4}$$

This process is fulfilled by applying the concept of integral image to perform the math. Integral image reduces the time of this operation and also is constant for any size of the Haar Wavelet Filter used because the complexity does not change with the size and alternatively four operations are needed every time:

$$SUM = D - C - B + A \tag{3.5}$$



0.1	0.1	0.2	0.1	0.7
0.2	0.3	0.2	0.7	0.8
0.1	0.4	0.3	0.3	0.1
0.1	0.5	0.1	0.1	0.2
0.1	0.4	0.8	0.5	0.6

0.1	0.2 A	0.4	0.5 B	1.2
0.3	0.7	1.1	1.9	3.4
0.4	1.2	1.9	3.0	4.6
0.5	1.7 C	2.5	3.7 D	5.3
0.6	2.3	4.9	5.6	8.0

Fig. 3.10 Example of working with integral images. On the left is the original image, marked in yellow the region we want to compute. On the right is the integral image. In this case, the results should be calculated from $SUM = 3.7 - 1.7 - 0.5 + 0.2 = 1.7$.

Then, the Haar Wavelet responses obtained earlier are weighted with a Gaussian centered at the point of interest. Gaussians attribute strong influence to the pixels close to the center and less importance to the points further away. Later, the algorithm must cover the circle created around the key point, iterating at steps of $\pi/3$ (or 60 degrees) and sum the orientations for each of the steps. This task is completed for both x and y directions, forming a larger vector. The step with the longest vector of all is selected to be the orientation of the keypoint.

3.3.5 Building the SURF Descriptor

The final stage starts by constructing a square window of size $20 \times s$ around each key point. All the points inside this window will be used to build the features descriptor. Note that this will utilize the orientation computed earlier since this window is orientated in that same direction. Finally, this window is divided into eight 4×4 square smaller windows. Each of these sub-regions is sampled with 5×5 equally spaced points.

For these 25 points in each subregion, Haar Wavelets of size 2 σ are calculated. The gradients obtained in the x-direction, dx, and in the y-direction, dy, of the points-space, will allow us to build the following descriptor:

$$Descriptor = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right] \tag{3.6}$$

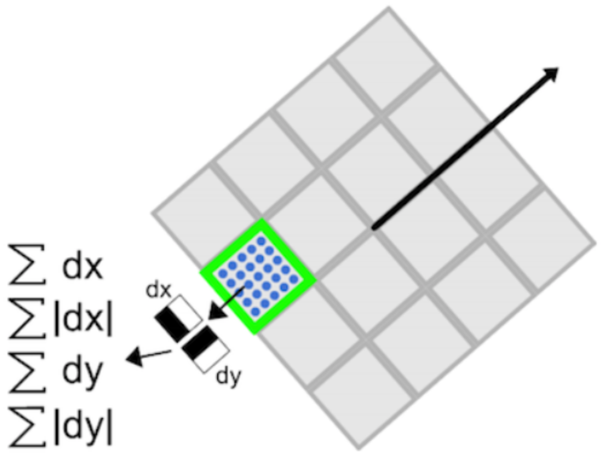


Fig. 3.11 Building the SURF descriptor. Courtesy from [2].

This results in a SURF descriptor vector of length 64 ($4 \times 4 \times 4$) for all 4×4 subregions.

3.4 Dictionary Learning

The second step of the BoF method is to generate the Dictionary of visual-words. The Dictionary (also named Codebook) has a predefined size K of visual words, which is the number of clusters chosen to group the GPU-SURF features. To build the Dictionary, the training image descriptors extracted beforehand are grouped into K clusters using k-means, an unsupervised machine learning algorithm [76].

At the first iteration, the K cluster centers can be initialized randomly or in a fixed location (in our case, we applied random initialization). Note that due to the unsupervised nature of K-means, starting clusters at a random location can lead to a lesser quality clustering division. To avoid this issue, K-means is performed various times with different random starting clusters center positions, keeping the best splitting result and discarding all the rest of the iterations.

To assign each SURF feature descriptor of a given training dataset image to a cluster, they are subsequently cataloged to the nearest cluster centroid utilizing the minimum (squared) Euclidean Distance, forming K groups of data. Finally, to find the new cluster centroid position, the mean (average location) of all feature descriptors present in each cluster is computed and updated. The above steps repeat until convergence is reached (the centroid locations remain the same) or until achieving the maximum number of iterations that were previously defined. Bellow, in the algorithm 1, the pseudo-code for the implementation of the K-means algorithm is presented:

Algorithm 1: K-means algorithm pseudo-code

```
Input: $D = (d1, d2,... dn)$;                // set of $N$ data points to be clustered
Initialization: Randomly assign the clusters centroid locations;
repeat
    • Form $K$ clusters by allocating each data point to the nearest cluster centroid;
    • Update centroids location, replacing the old clusters centers location with the mean
      position of all their data elements;
until Convergence criteria is met;
Output: Set of $K$ number of clusters
```

3.5 Encoding

The third step of the BoF procedure is called Encoding. After building the Dictionary, we can advance to construct the vectors/histograms for each testing image with known labels. The SURF features are extracted for these images and then mapped into the nearest cluster centroid of the Dictionary, based on the (squared) Euclidean distance (or another distance metric). A histogram of counts (frequency of occurrence) is constructed by incrementing a cluster centroid's number of occupants each time a new visual word is placed into it. The result is that each image is represented by a histogram vector of length K [77]. This is called a mid-level image representation [78].

The Euclidean Distance is given by the following equation:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3.7}$$

where q is the centroid location and p is the SURF descriptor vector (both of dimension (n) 64).

3.6 Classification

The final step of the BoF method is the classification stage. Support Vector Machines (SVM), proposed in [79], are utilized for classification purposes. They are binary classifiers used for the task of separating two classes (or more) in a multidimensional space. In our case, the inputs of the SVM are the histograms created by the BoF algorithm that we will try to correctly classify into "Disease" or "No Disease".

SVM's goal is to divide the data into classes. The data points nearest to the margin on both sides of a hyperplane are known as support vectors. The SVM finds the linear hyperplane which maximizes the margin between the support vectors.

The linear separation hyperplane is defined as:

$$f(x) = w^T x + b \tag{3.8}$$

where w is the normal vector to the hyperplane and b is the offset of the hyperplane from the origin along the normal vector w.

SVM can also be used to solve nonlinear classification tasks when the set of samples cannot be separated linearly. By applying kernel functions, these samples are mapped onto a high-dimensional feature space in which linear classification is possible [80]. In this case, different types of kernel functions can be applied, such as linear, polynomial, sigmoid, and RBF (Radial Basis Function) [81]. The main advantage of using SVM is that it performs classification effectively even if the dimension is high [82].

The generalization is obtained with the following equation:

$$f(x) = w^T \phi(x) + b \tag{3.9}$$

where $\phi(x)$ is the kernel function.

For the input testing data, the SVM decision function regarding which class the data belongs to is given by:

$$f(x) = sign(w^T \phi(x) + b) \tag{3.10}$$

The DR image histograms created with the BoF method are labeled according to their respective class. Subsequently, the SVM classifier separates the data into two classes: disease present or no anomaly detected. The SVM classification scheme is the following:

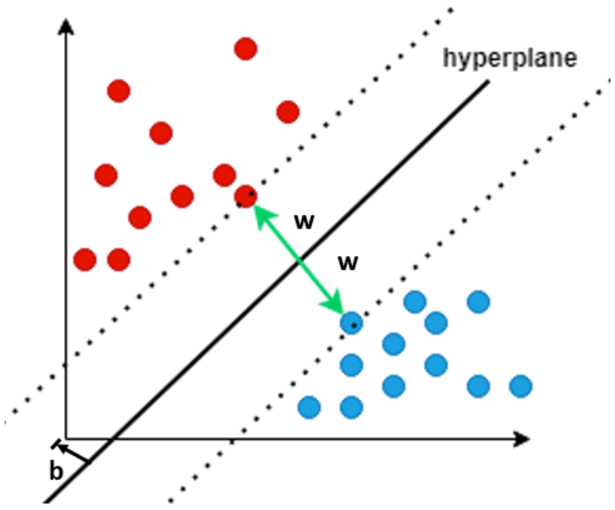


Fig. 3.12 Support Vector Machine concept

The margin is represented by the green two-ways arrow in the figure and the hyperplane by the black line dividing the two classes. The hyperplane equation is:

$$w^T \phi(x) + b = 0 \tag{3.11}$$

The class with red data points is defined by:

$$w^T \phi(x) + b \geq 0 \tag{3.12}$$

And the class with blue data points is given by:

$$w^T \phi(x) + b \leq 0 \tag{3.13}$$

Chapter 4

GPU computing

The main focus of this chapter is to analyze the GPU architectures of modern computing systems and the differences from the common CPU architectures in order to understand their strengths and weaknesses, and how they can be exploited to parallelise the SURF algorithm.

4.1 The CPU architecture

The Central Processing Unit (CPU) is the brain of the computer system as it is in charge of arithmetical and logical operations, controlling the memory instructions and input/output (I/O) operations.

In a single-core system, computing multiple instructions at a given time is impossible since their performance is sequential. The development of multi-core CPU architectures over the years is an attempt by manufacturers to increase performance and to overcome the limitations imposed by the end of Dennard Scaling [83]. This allowed programmers to build solutions utilizing parallel and multi-threading techniques.

Due to the slowdown of Moore's law in recent years, the tendency is for the number of CPU cores to continue growing. Nonetheless, the main focus is to be able to promptly process a wide variety of tasks, especially those where latency or per-core performance is crucial.

4.1.1 CPU memory hierarchy

CPU role requires low latency and, to achieve it, the time to access data present in memory must be as low as possible. To reduce the cost (time spent) to read and write data from/to the main memory (for example, RAM - Random Access Memory or even in the Hard

Drive or SSD) cache mechanims based on SRAM and exploiting time and space locality were proposed. By taking advantage of the principles of locality (spatial and temporal), the cache is used to help the processor to predict the flow of execution.

Nowadays, modern CPUs have a hierarchy of multiple cache levels (L1, L2, often L3, and rarely even L4). The upper levels (ie: L1) of cache are faster but with more limited space available for data, while lower levels of cache are slower but with more room for data storage (faster SRAM memory is more expensive). The CPU tries to avoid fetching data from the main system memories unless it is forced to (in case a cache miss occurs).

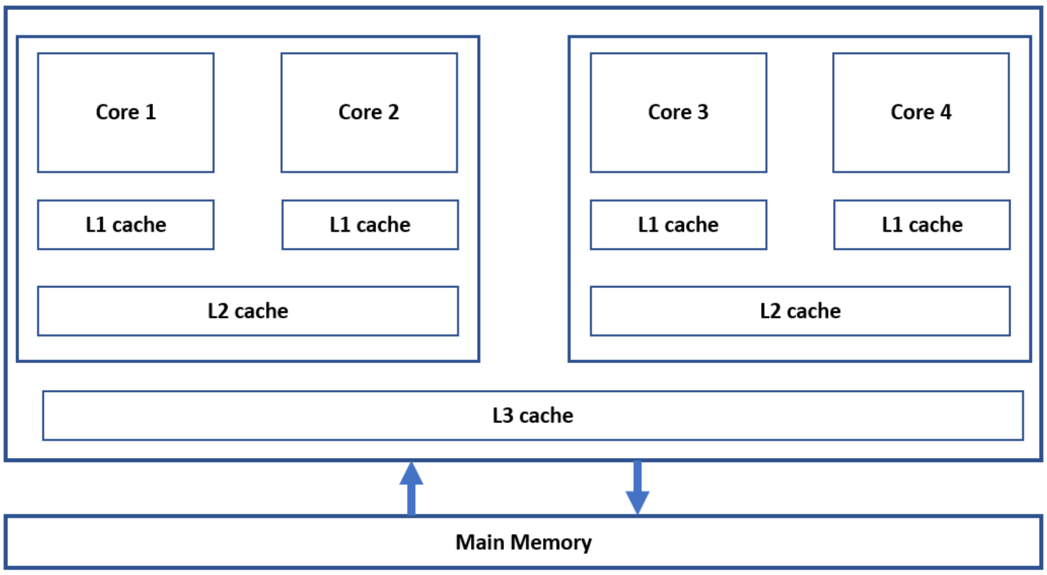


Fig. 4.1 Four cores CPU architecture

4.2 The GPU architecture

By dividing tasks among thousands of small cores, GPUs are ideal to overcome the end of Moore's Law, caused by the physical limits of the hardware. Introduced in the consumer market in 1999 by NVIDIA, GPU's firstly gained popularity due to gaming and multimedia industries.

The graphics processing unit (GPU) is designed to allow parallel computing. The goal is to provide high throughput instead of the low latency of the CPU design. Having hundreds/thousands of small cores, the GPU can handle thousands of threads running simultaneously, thus hiding latency (accessing data in memory) with computation.

Compared to CPUs, the performance gain GPUs allow for parallel processing tasks is significant. This unceasing evolution in the number of cores and bandwidth provided by

GPUs is supporting the development of countless applications in every modern technological field.

Nowadays the parallelism and computational power that GPUs provide is being utilized to accelerate many applications, such as 3D graphics rendering, supercomputing, artificial intelligence, and machine learning.

4.2.1 GPU Memory Hierarchy

A GPU is composed of Streaming Multiprocessors (SM). Each multiprocessor contains hundreds of CUDA cores, fast access registers, and shared memory between all these cores. Some key concepts of the model are presented below:

- **Kernel**: A CUDA kernel is a function that is executed on GPU;
- **Threads**: Threads execute in parallel;
- **Block**: A group of threads is called a CUDA block;
- **Grid**: CUDA blocks are grouped into a grid. A kernel is executed as a grid of blocks of threads;

Threads are organized in blocks and each of these blocks is executed in parallel, one per SM, allowing for massive speedups in some applications.

- **Registers**: Registers are private to each thread and are not visible to other threads. The programmers have no control over the registers as it is the compiler that makes decisions about register utilization.
- **Shared Memory**: Threads within the same Block can communicate with each other, using the shared memory (L1 cache/SMEM). Every SM has a fast, on-chip shared memory.
- **Read-only memory**: Read-only (RO) cache, is one of the components of each SM. They also have an instruction cache, constant memory, and texture memory.
- **L2 cache**: Every thread in every CUDA block can access this memory since the L2 cache is shared across all SMs in the GPU.
- **Global memory**: The device DRAM is used to store data on the GPU side.

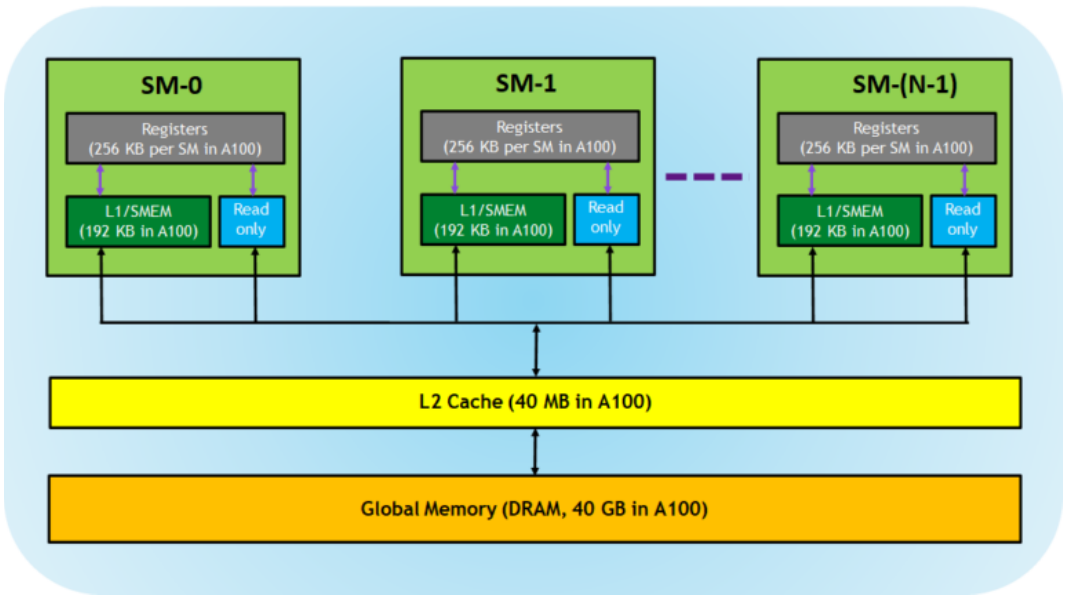


Fig. 4.2 GPU architecture - from NVIDIA webpage

To run a parallel function on the GPU, data must be transferred from the CPU (host) to the GPU (device). After this, execution of the kernel is performed on the GPU streaming multiprocessors. The outputs of the calculations need to be moved back to the CPU to continue/end the execution flow of the program. This is performed with the function *cudaMemcpy*.

4.3 Parallelized GPU-SURF functions

The cost of copying data between host and device impacts performance, introducing an undesired time overhead and latency to the algorithm's execution. This overhead ideally is desired to be zero, as it is time spent (latency) with no useful kernel work done. When a kernel is launched, the data is copied from the host main memory to the device main memory.

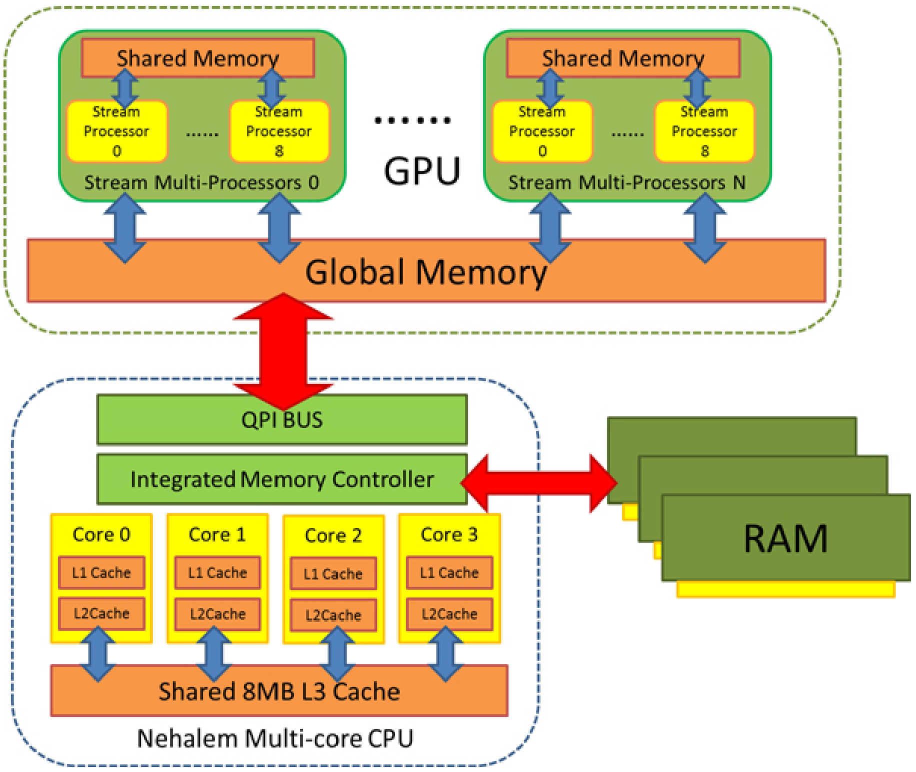


Fig. 4.3 Data transfer between host and device - courtesy from [3]

Thus, not all the functions should be parallelized. The only functions that should be executed on the GPU are the ones where data parallelism in the form of SIMD (Singled Instruction, Multiple Data) and SIMT (Singled Instruction, Multiple Threads) can be applied to increase the program throughput. A summary of the GPU-SURF functions that are executed on the device (GPU) is presented:

- **Build the Determinant;**
- **Build the SURF Descriptor;**
- **Convert RGB to GrayScale;**
- **Detect Interest Points Orientations;**
- **Haar Wavelet Response;**
- **Non Maximum Suppression;**
- **Transpose Matrix;**

Chapter 5

Experimental Results and Discussion

This chapter analyses the results obtained by our implementation of the BoF and GPU-SURF-based DR diagnosis system. The chosen hardware and dataset are addressed and detailed in this chapter. Then a performance comparison between CPU and GPU implementations of the SURF feature extraction algorithm is performed, where the superiority of parallel computing for this task is demonstrated. After this, the post-processing applied to the extracted features is mentioned, improving efficiency and performance. The construction of the Dictionary and the choice of the ideal number of K-clusters, as well as the Encoding stage and construction of the visual-word frequency histogram are discussed. Later on this Chapter, the performance metrics of our model (Accuracy, Precision, Specificity, F1 score, and AUC) are calculated through the use of the confusion matrix. Tests were performed to evaluate different types of SVM kernels for classification. Also the variation of Accuracy as a function of the resolution dataset images or the total computation time required to process and classify each image were analysed. A discussion is also given following the presentation of the results.

5.1 Hardware

The hardware used in this dissertation was a personal laptop computer, with the following specifications:

RAM: 16 GB DDR3

GPU: Nvidia GeForce 860M with 4GB VRAM

CPU: Intel i7 4720HQ @2.6GHz (with 3.4GHz turbo-boost)

Storage: 512 GB of SSD and an external rigid disk of 2TB to store the high-resolution dataset images

5.2 Diabetic Retinopathy Dataset

There are many Diabetic Retinopathy Datasets available online, such as the MESSIDOR and Debrecen. Our choice was the EyePACS dataset. This dataset was obtained from a Kaggle competition named the "Diabetic Retinopathy Detection - Identify signs of diabetic retinopathy in eye images". The competition took place in 2015 but the dataset remains available on their website for downloading.

This dataset contains high-resolution images, in the $4k$ (3840×2160) range, which is ideal for our case. High-resolution images allow for a better diagnosis of DR, detailing information that would disappear or be presented in a dubious state otherwise. The use of GPU computing also benefits from higher resolution images. The results that support these arguments are presented later in this chapter.

This dataset contains healthy color fundus retina images and other images with visible diabetic retinopathy damage. We will name them "no disease" and "disease", respectively. Since the proportion of "no disease" is higher than its counterpart, the SVM classifier may get biased towards the prediction. An example of this showing the confusion matrix is presented later in the results of this thesis. Therefore, this unbalanced dataset should be balanced. The following work is performed with a sub-set of the dataset: 350 images, 175 of each class.

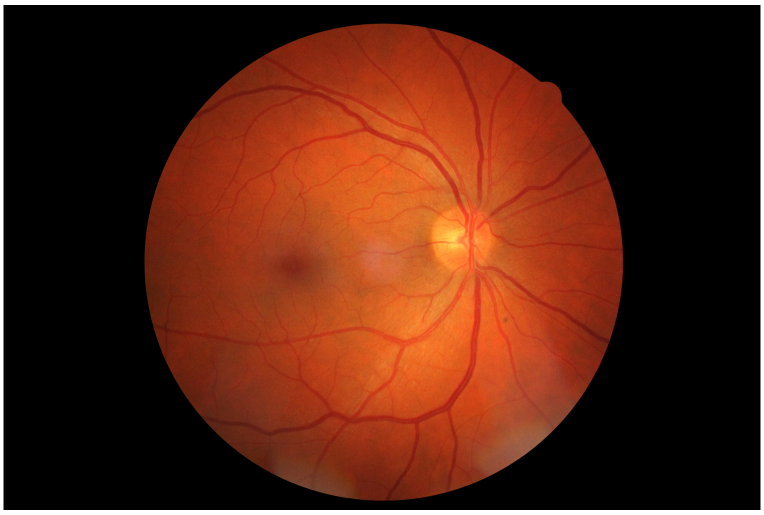


Fig. 5.1 "No Disease"

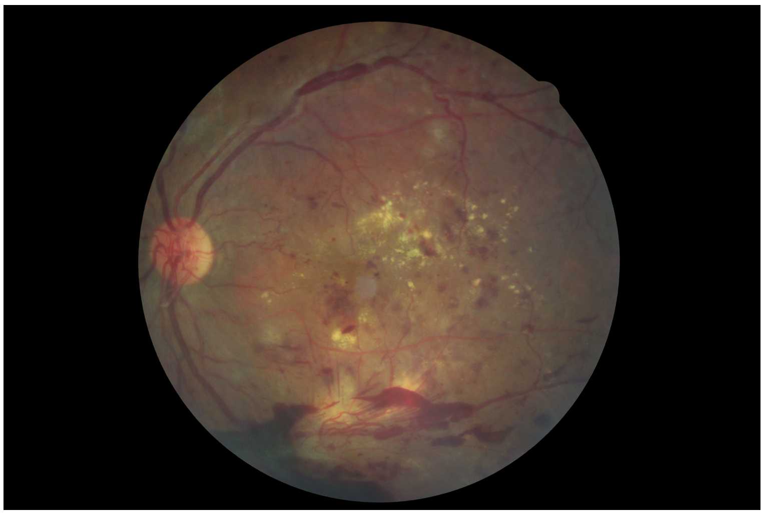


Fig. 5.2 "Disease"

5.3 CPU vs GPU Performance of SURF

As mentioned above, the GPU introduction translates to performance gains and more efficient use of the resources. Comparing the overall time required to complete the SURF feature extraction and save the descriptors we can see that GPU-SURF implementation largely overtakes its CPU-based program counterpart. In the graph below the comparison is made for 350 images, 175 of each class:

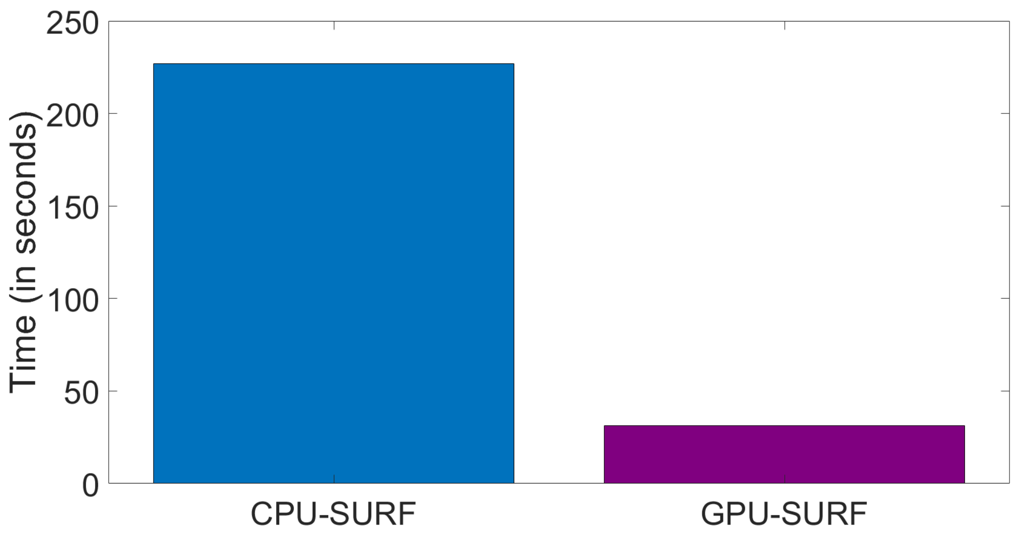


Fig. 5.3 Overall time needed to perform CPU-SURF and GPU-SURF

To measure the speedup obtained by the GPU-SURF implementation against the CPU-SURF algorithm, the mean time for each image execution was calculated with:

$$MeanTime = \frac{OverallExecutionTime}{NumberOfImages} \tag{5.1}$$

In the CPU-SURF case the mean time to process each image was:

$$MeanTime = \frac{226.8}{350} = 0.65s \tag{5.2}$$

The number of frames-per-second (FPS) processed by CPU-SURF was:

$$FPS = \frac{CountedFrames}{NumberOfSeconds} = \frac{350}{226.8} = 1.54 \tag{5.3}$$

And in the GPU-SURF counterpart the mean time to process each image was:

$$MeanTime = \frac{31.4}{350} = 0.09s \tag{5.4}$$

The number of frames-per-second (FPS) processed by GPU-SURF was:

$$FPS = \frac{CountedFrames}{NumberOfSeconds} = \frac{350}{31.4} = 11.14 \tag{5.5}$$

Meaning that the speedup when comparing the GPU and the CPU implementation was:

$$Speedup = \frac{CPU}{GPU} = \frac{0.65}{0.09} = 7.22x \tag{5.6}$$

The graph below shows the individual results:

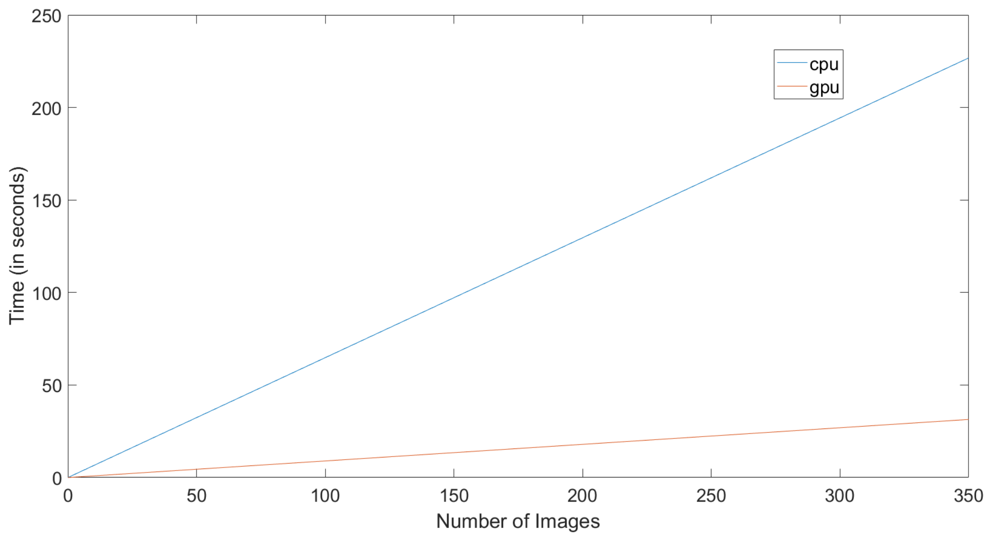


Fig. 5.4 CPU-SURF and GPU-SURF comparison

5.4 Post-processing SURF features

The EyePACS dataset images have noise in the background that can distort the DR classification results as SURF will detect any intensity changes in the pixel values of the images. More so, these undesired SURF interest points are irrelevant as they add no information, and processing them is wasted computational power and time that could be used to process the next image in the BoF method. Using techniques to remove them and maintain only the SURF keypoints that carry information is vital to efficiently run the algorithm and classify DR pathology.

Also, in the border between the retina and the background there is the detection of SURF interest points that are not describing any region of interest but the contrast between a dark region (background) and a colorful region (retina fundus). These key points are also removed.

The strategy adopted to successfully erase the inappropriate keypoints was the following:

- 1. Find the image size (width and length);

- 2. Determine the image center (it corresponds to the center of the retina);
- 3. Find the radius of the circle limit that forms the eye;
- 4. Remove all the keypoints outside the retina (noise) and around the retina border utilizing a threshold value to subtract the radius distance;

The BoF algorithm is then executed with the remaining SURF interest points. The image below shows a comparison of unprocessed and pre-processed frames:

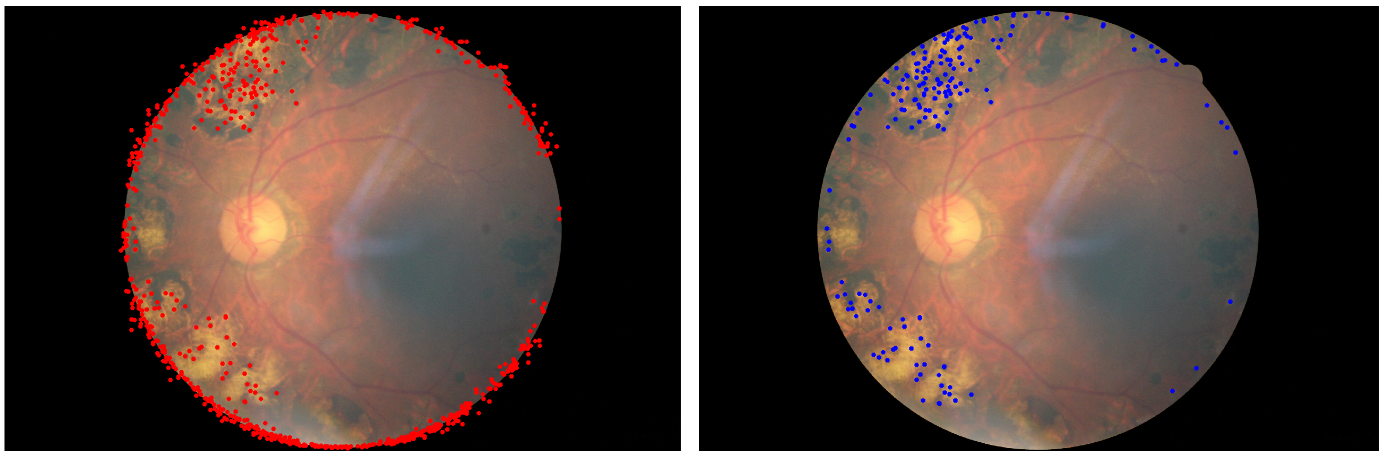


Fig. 5.5 Before processing the DR image (on the left) and after removing the inappropriate keypoints previously detected by GPU-SURF (on the right)

The total number of keypoints detected without pre-processing the GPU-SURF for all the dataset was 96023. After deleting them, it dropped to 19373 representing a 4.96 **times** more efficient data usage.

5.5 Dictionary Buildup: K-means Clustering

The K-means clustering described above was performed using the MATLAB function for the effect. As each SURF descriptor vector has a 64-size dimension, the visualization of the clusters that form the Dictionary becomes difficult. Nonetheless, applying the Principal Component Analysis (PCA) algorithm to reduce the dimensionality of the data allows a better perception of the clustering result. Bellow, there is an example of this, made for 3 clusters only, to simplify the visualization:

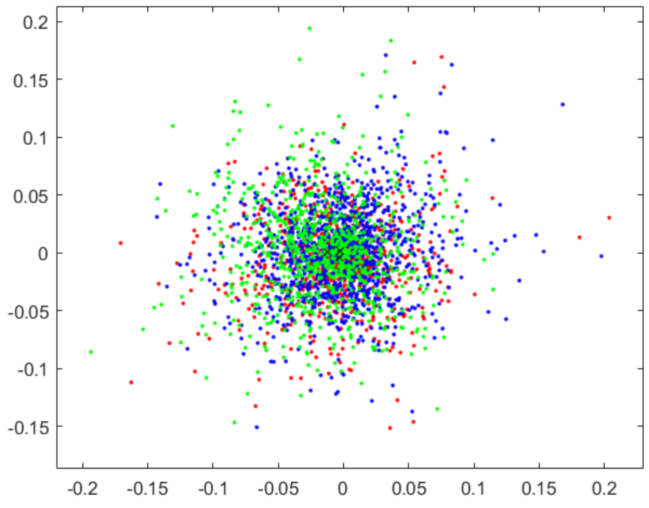


Fig. 5.6 64-Dimension visualization without PCA

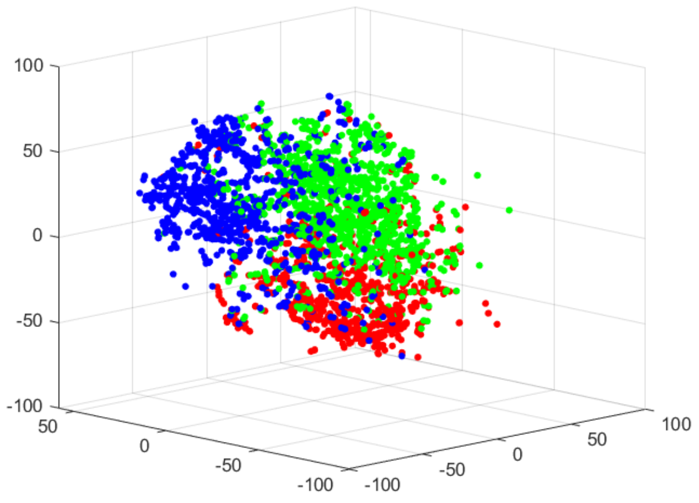


Fig. 5.7 3-Dimension visualization with PCA

Each cluster centroid is a visual-word in the Bag-of-Features algorithm.

The choice of the number of clusters, K, is crucial to accomplish satisfactory performance considering that selecting an inappropriate K may yield poor results. Having a low number of K clusters (visual words) results in under-fitting and high bias (poor discriminatory behavior) while a too high number of visual-words causes overfitting and high variance in the results.

For our case, on the high-resolution DR images of the EyePACS dataset, the value of K chosen was 35. The graph below represents the percentage of correctly labeled images by the SVM classifier when varying the number of visual words in our Dictionary:

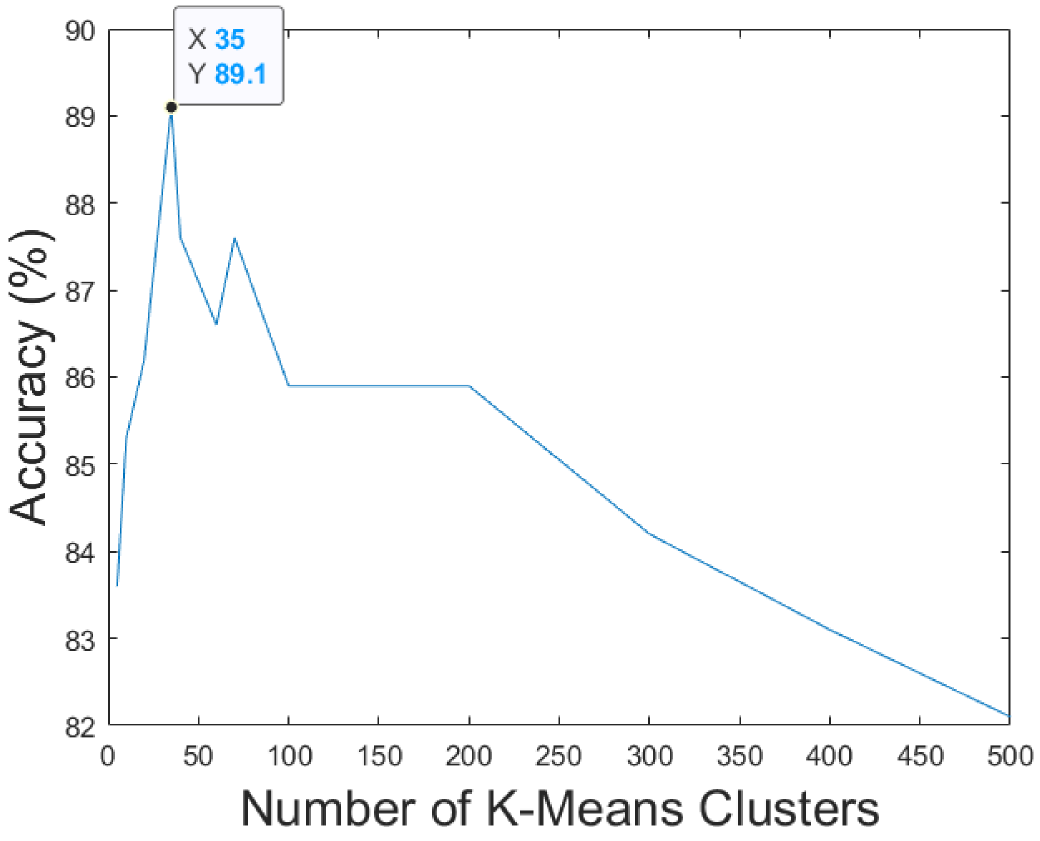


Fig. 5.8 Variance of BoF accuracy results when changing the number of Vocabulary Size

As can be seen, 35 K-means clusters correspond to 89.1% accuracy after running the BoF GPU-SURF-based algorithm. It is the highest value out of all the number of k-means clusters tested, meaning it should be kept and used on future computations to achieve a good split of the input data in the Dictionary buildup.

5.6 Encoding: Frequency of Features Histogram

The dictionary was built with the training DR images. For the testing stage, the SURF extraction is performed again, but this time for the testing DR images, to perform the encoding stage. The (squared) Euclidean Distance between each SURF feature and each cluster centroid is computed and the lowest distance found determines the visual-word index that a determined feature belongs. Repeating this for all testing images and keeping track of the results gives us all the information needed to build the histogram that represents each image.

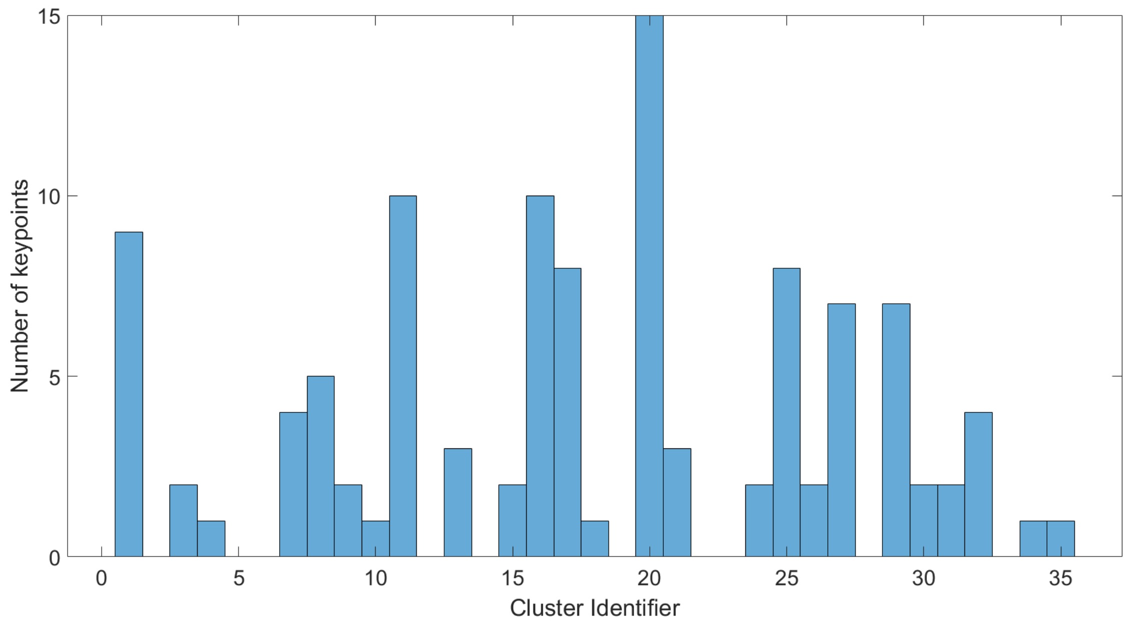


Fig. 5.9 Single DR image histogram example

Note that in this example, there are some empty clusters. It is possible because this histogram of visual-word frequency is a sparse vector representation. Each image has a different number of keypoints detected by SURF, so these histograms are later normalized before sending them to the SVM classifier.

5.7 Confusion Matrix and Metrics

To check if the model is accurate and improve it one can use the concept of Confusion Matrix. This matrix is used to define the performance of a classification algorithm, in our case the Support Vector Machine (SVM). A confusion matrix helps to visualize and summarize the performance of a classification algorithm.

The confusion matrix consists of four fields that are used to define the measurement metrics of the classifier. These four numbers are:

TP (True Positive): TP represents the number of images that have been properly classified to have aneurysms or hard exudates, meaning they have the DR disease.

TN (True Negative): TN represents the number of correctly classified images that have no disease (healthy).

FP (False Positive): FP represents the number of miss-classified images. They were diagnosed with the DR disease but are actually healthy pictures.

FN (False Negative): FN represents the number of images miss-classified as healthy but that actually are showing symptoms of the disease.

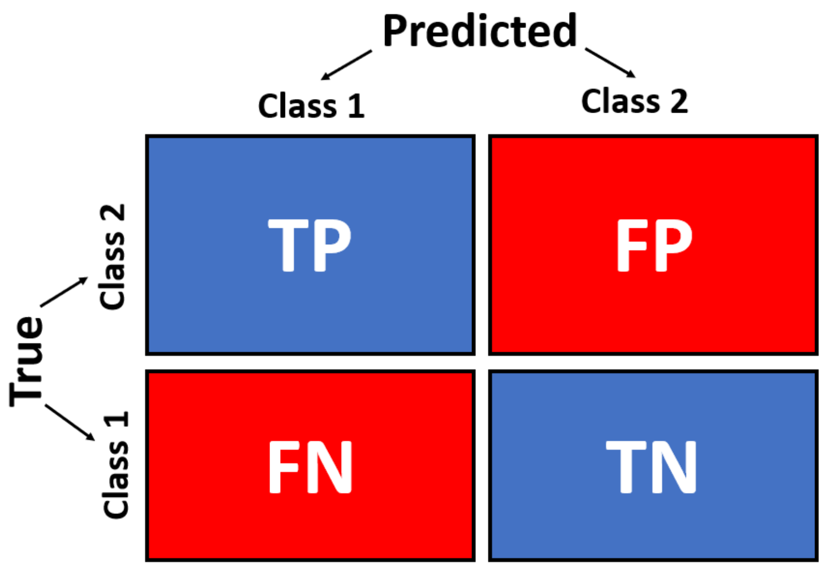


Fig. 5.10 Generic Confusion Matrix for binary classification

In our case, the confusion matrix below indicates that the "No Disease" class images were almost all correctly classified, resulting in 170 out of 175 correctly labeled by the SVM. In terms of the "with Disease" class, we notice that some of the images (33 out of 175) were miss-classified as healthy when in the reality they had DR signs. This could be improved by increasing the miss-classification cost of this class. However, it would decrease the accuracy and overall metrics of the model so we opted not to change it.

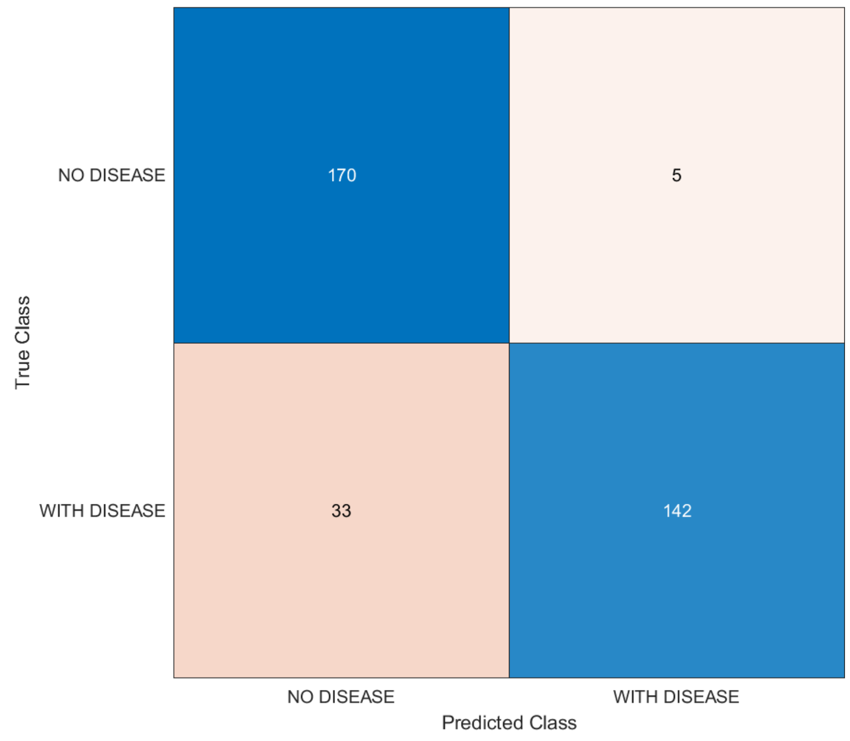


Fig. 5.11 Confusion Matrix. Blue for correct decisions, pink for wrong classifications

The performance metrics computed from the Confusion Matrix are the following:

Accuracy is the metric that counts the proportion of correct predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.7}$$

Sensitivity (also called **True Positive Rate** or **Recall**) tells us what proportion of the positive class got correctly classified. In our case, it shows what proportion of the Diabetic Retinopathy class "Disease" images was correctly detected by the SVM.

$$Sensitivity = \frac{TP}{TP+FN} \tag{5.8}$$

False Negative Rate (FNR) tells us what proportion of the positive class the SVM classified incorrectly.

$$FNR = \frac{FN}{FN+TP} \tag{5.9}$$

The goal is to have a higher **TPR** and a lower **FNR** since we want to correctly classify the presence of the disease.

Specificity tells us what proportion of the negative class ("no disease") got correctly classified.

$$Specificity = \frac{TN}{TN + FP} \tag{5.10}$$

Specificity determines the proportion of healthy people"(no disease") who were correctly identified by the SVM model.

F1 score shows the equilibrium between the precision and the recall.

$$F1score = \frac{2 \times sensitivity \times accuracy}{sensitivity + accuracy} \tag{5.11}$$

As said above, the goal is to have both sensitivity and specificity as close to 1 as possible. The Receiver Operating Characteristics curve (ROC curve) is a 2 dimension plot used to measure the performance of the classification. In the ROC space, the upper left corner represents perfect classification while a diagonal line represents random classification [84].

ROC curves plot the TPR (true positive rate) against the FPR (false positive rate) by varying the threshold (probability of membership to a class) or score [84]. The ROC curve was determined with the help of the MATLAB Toolbox for Machine Learning.

5.8 Our Metric Results

Using a Linear Kernel for the SVM classifier we achieved 89.1% **accuracy**, 97.1% **precision**, 81.1% **specificity** and 0.923 **F1 score**, resulting in a **AUC** of 95%. This is in line with other publications. For instance, Antal et al. [85], achieved 91% Specificity and 90% Sensitivity. The British Diabetic Association recommendation is 95% Specificity and 80% Sensitivity [86].

Validation estimates model performance on new data compared to the training data. Cross-Validation protects the data against overfitting by partitioning the dataset into folds and then estimating accuracy on each fold. We used this validation method to examine the predictive accuracy of our model, dividing the data into 5 folds. The table below shows the resume of the above metrics presented in subchapter 5.8 for our implementation:

Accuracy	89.1%
Sensitivity	97.1%
FNR	2.85%
Specificity	81.1%
F1 score	0.923

Table 5.1 Our model metrics results.

The accuracy obtained was (89.1%) meaning our model is well designed and working as expected. But accuracy can be misleading if used with imbalanced datasets, as mentioned in previous chapters. We balanced the subset of the dataset used for the experiments as mentioned earlier to guarantee there is no bias towards a class from the classifier perspective. We also computed the other metrics based on the confusion matrix for evaluating performance.

The results show that the goal of obtaining high sensitivity (97.1%) and low false negative rate (2.85%) were achieved. The F1 score was also high (0.923) (its maximum cap value is at 1.0), showing good relation between precision and recall. The specificity value, despite being slightly lower than its counterparts, still achieves a considerable score of (81.1%). This value can be improved and the miss-classification of images of the class "Disease" can be reduced at the cost of inferior accuracy, as shown in the ROC curve below in figure 5.12.

The Support Vector Machine type of kernel influences the final classification results. Here is a comparison of the accuracy of the types of kernels for our automatic DR diagnosis:

Type of SVM kernel	*Accuracy (%)*
Linear	89.1%
Quadratic	84.9%
Cubic	88.0%
Fine Gaussian	80.6%
Medium Gaussian	86.3%
Coarse Gaussian	68.9%

Table 5.2 Type of SVM accuracy results.

The Linear Kernel equation provides the highest accuracy.

A Receiver Operating Characteristic curve (ROC curve) is a graph that plots the True Positive Rate and the False Positive Rate. It shows the performance of a classification model when varying the classification thresholds. In an SVM binary classifier, the choice of a threshold depends on the importance of TPR and FPR for the classification problem. If there is no external concern about low TPR or high FPR, one option is to weight them equally by choosing the threshold that maximizes TPR-FPR. Another option is to choose a threshold that yields a low FPR while keeping a reasonable TPR (for example, to increase the number of patients correctly diagnosed with DR).

The Area Under Curve (AUC) tells us how much the model is capable of correctly labeling classes. The higher it is, the better the model is at distinguishing between patients with the disease and no disease. A theoretical perfect model has the AUC equal to 1. The figure 5.12 shows the ROC curve for the automatic DR diagnosis from the confusion matrix obtained in the section 5.7 of this chapter (figure 5.11) and demonstrates that our model obtained an AUC of 0.95.

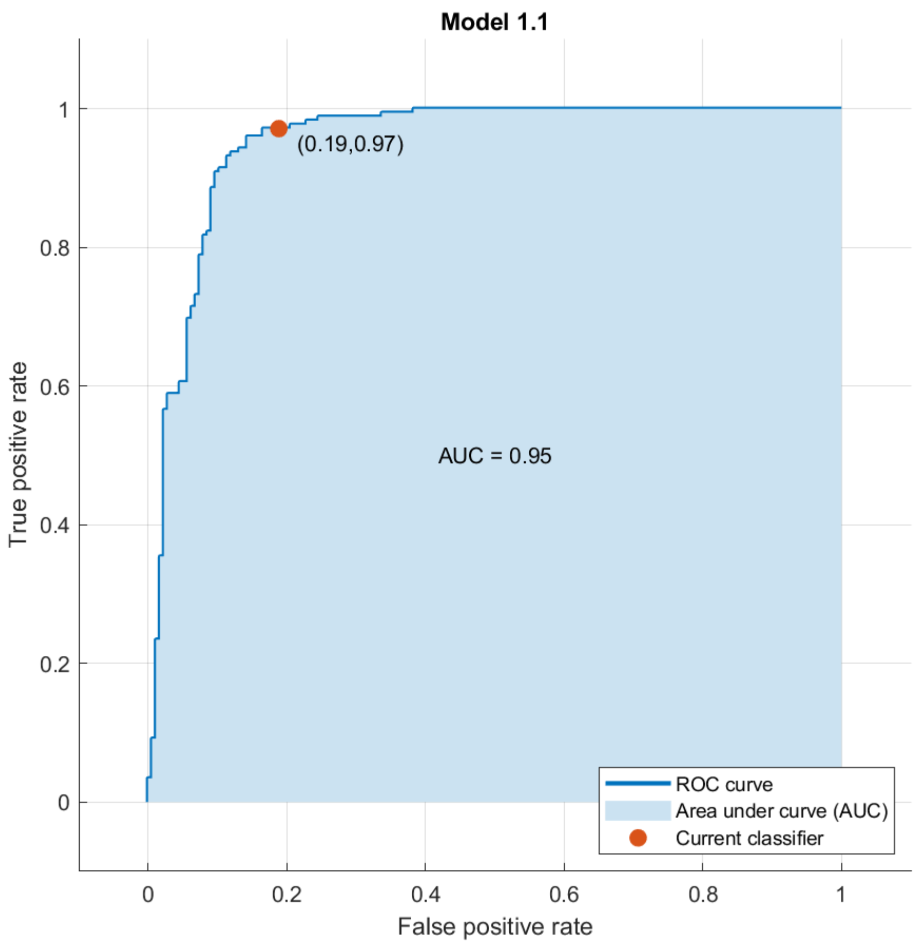


Fig. 5.12 ROC Curve and AUC

The impact of image resolution on this work is shown in the figure below. It shows that increased detail in a DR image allows for a better description of features at the level of the SURF algorithm and consequently a better diagnosis of the disease using the BoF method.

DR image resolution	*Accuracy (%)*
3840×2160	89.1%
1920×1080	85.9%
960x540	83.6%
480x270	78.4%

Table 5.3 Image resolution impact on the accuracy.

The results of our automatic DR diagnosis method shown above when compared with the results of other state-of-the-art methods published in recent papers show that we are in line with other authors' publications. A brief summary is presented:

Authors	*Accuracy (%)*	*method*
Our implementation	89.1%	BoF
Sundaram et al. [87]	92.0%	BoF
Islam et al. [88]	94.4%	BoF
I. Sadek et al. [89]	97.2%	BoF
Mukti et al. [90]	84.6%	BoF
Erciyas et al. [91]	99.1%	CNN

Table 5.4 A summary of other authors' publications on this topic.

5.9 Overall Computation Time for our BoF implementation

To measure the time a single image takes to be processed we must sum the times of GPU-SURF feature extraction, Bag-of-Features creating the visual words, and the SVM classification stage. We saw before in section 5.3 that the GPU feature extraction takes roughly 0.09s per image. The total time to run the BoF and SVM methods in the Matlab script code for our 350 test images and 50 training images (visual vocabulary formation) is

$150.19s$. Resulting in a total average time of 0.52 seconds per test image for the automatic DR diagnosis.

If we exclude the training time for creating the visual dictionary (it only runs once at the first iteration) the code execution takes $68.88s$, resulting in a total average time of 0.29 seconds per test image.

Chapter 6

Conclusion and Future Work

The goals proposed for this dissertation in chapter 1 were achieved. The main goal was to implement a solution to automatically diagnose DR. Other goals proposed were to explore and adapt both CPU-based SURF and GPU-based SURF solutions and, consequently, to compare their performance. After this, the objective was to build a Bag-of-Features method to correctly classify color fundus retina images. This algorithm was implemented in MATLAB.

This solution demonstrates that despite BoF adoption for image classification problems being mostly abandoned and replaced with Convolutional Neural Networks, it can still compete with more modern and expansions techniques for some tasks and specific scenarios.

The advantage lies in the need for less training data and consequently lower training time. Also, CNN's input image resolution needs to be low (256×256 for example), as the calculations complexity grows exponentially to process the higher pixel count from the 3 color channels (RGB). On lower image resolutions, some fine details are lost, as is the case of DR hard exudates or small aneurysms. In higher resolution images (for example 4K) these minutiae are preserved, helping to the DR disease diagnosis. The BoF method combined with GPU-SURF can process these high-quality images quite easily. Compared to the CPU version of the SURF algorithm, the GPU-SURF obtained a speedup of 7.22x on our subset of the EyePACS dataset, being more efficient and taking less computational time. The automatic classification of DR images using our BoF model obtained a 89.1% **accuracy** and 97.1% **precision** and 81.1% **specificity** with an SVM linear kernel.

However, it should be noted that a solution built based on BoF presents weaknesses, such as:

1. By the nature of DR images it is hard to choose a SURF threshold that suits all images. Some of them are darker and others are lighter. Some are blurrier while others are

more focused and nitid. The signs to detect DR vary in a lot of them: hard exudates, aneurysms, and growth of abnormal blood vessels have a distinct format, size, color tone, and orientation from patient to patient and they are not always present all at the same time. By not capturing some of these key points by using the wrong threshold there can be considerable degradation in the final result.

2. Blood vessels and the optic nerves are regions of the image where SURF often detects keypoints but they do not contribute to the DR diagnosis with this method that focuses on detecting anomalies like aneurisms and hard exudates. The keypoints present in these areas of the image often contribute to poor Dictionary build-up in the training stage of the BoF method or to the wrong histogram of visual-words frequency result, affecting the SVM and leading to misclassification data.

3. In a deep learning CNN solution for DR automatic classification, the model benefits from having large amounts of data in the training stage. Having more images means that even if the two classes we are trying to separate are very similar (which is the case of the Diabetic Retinopathy dataset) we will start to build more correct heights and recognize patterns that allow us to correctly classify new images. However, the BoF method training stage can achieve satisfactory results with a low quantity of images, and increasing the number of training images (and consequently, features/keypoints) doesn't necessarily translate to better results. Outliers and overfitting can negatively impact the experiments. If the data in the training set is descriptive enough, a correct choice of the number of k-means clusters (or another clustering algorithm) is more important to achieve good precision and recall.

We conclude that the BoF approach presents advantages in some aspects and drawbacks in others. In the resume, figure 6.1 shows a summary diagram of the conclusions presented during this chapter.

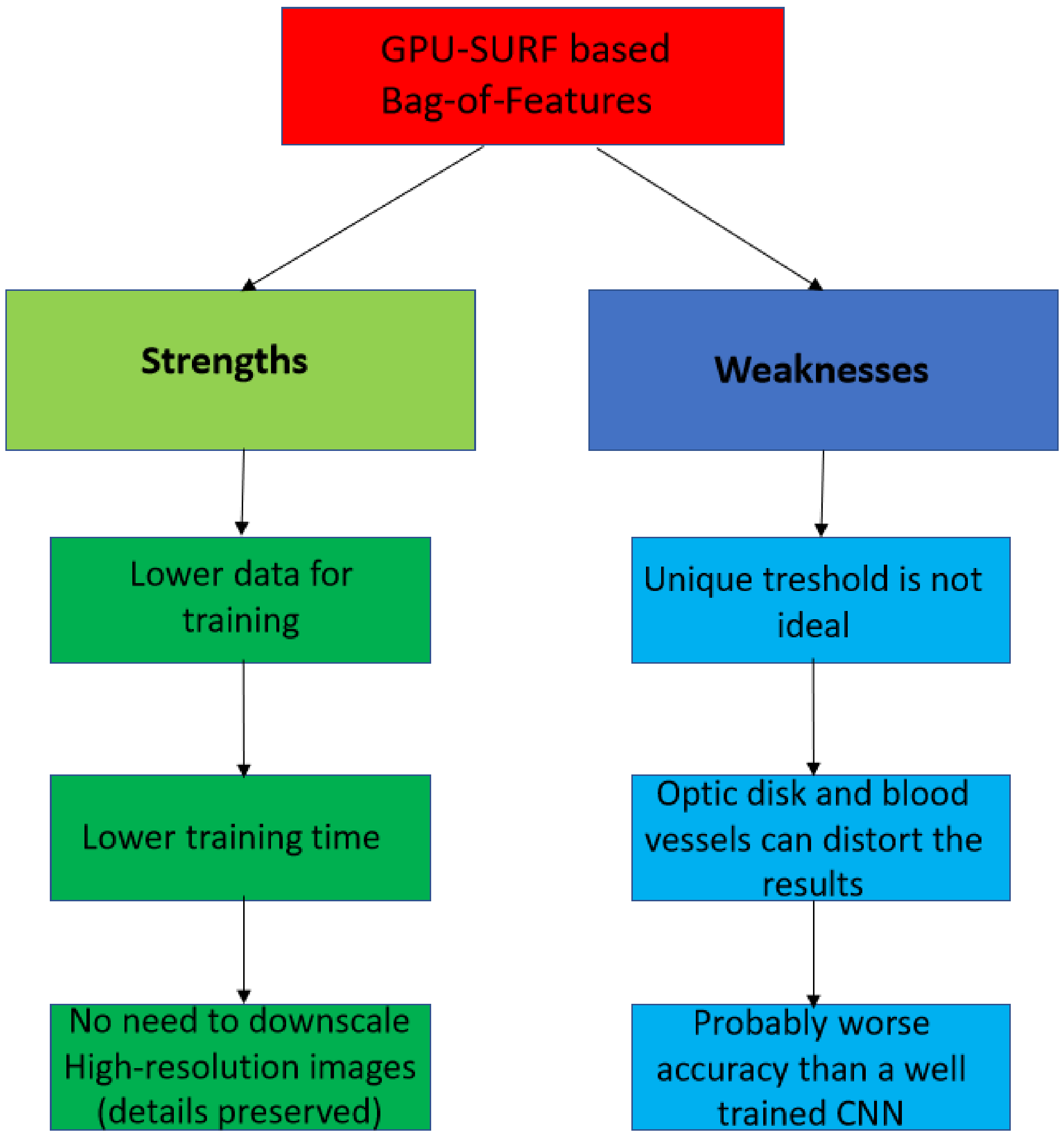

GPU-SURF based Bag-of-Features
Strengths
Weaknesses
Lower data for training
Lower training time
No need to downscale High-resolution images (details preserved)
Unique treshold is not ideal
Optic disk and blood vessels can distort the results
Probably worse accuracy than a well trained CNN

Fig. 6.1 Strengths and weaknesses of GPU-SURF based Bag-of-Features.

6.1 Future work

Considering the work developed and the results obtained, there are several possibilities for future work. For the problem mentioned in **1)**, a solution could be utilizing an automatic threshold method as Kirchner did in [92] for the SIFT descriptor for example (no literature was found for the SURF case). For the **2)** issue reported, in 2019 Memari et al. [93] managed to perform Retinal Blood Vessel Segmentation by using matched filtering and fuzzy c-means clustering. These segmented blood vessels can then be utilized to identify abnormal growth of smaller blood vessels and contribute to diagnosing DR by focusing on these features instead of focusing on bright/red lesions (hard exudates, aneurysms). Or using another strategy, make use of the information about the location of blood vessels to remove all the SURF keypoints detected in that location. Regarding the topic of the optic nerve negatively impacting the results, Kemal Akyol et al. [94] implemented a method for automatic detection of the optic disc in a retinal image by using keypoint detection, texture analysis, and visual dictionary techniques (other methods for optic disc identification are also mentioned in their paper). Their work can be used to improve computer-aided eye disease diagnosis systems, namely Diabetic Retinopathy. The optic nerve detection is fairly simple in normal retina images, but in the case of retina images with DR disease, it becomes harder, as this region is often miss classified and confused with hard exudate information. Localization of the optic disc can be a solution to significantly improve the Bag-of-Features classification score. Relative to **3)** there is not a solution per se as the problem reported is a consequence of the method chosen. In general, for image classification tasks, deep learning (the CNN approach) will perform better, as shown in [95], but for this specific case, further studies should be performed.

We believe that the inclusion of the above-proposed solutions will improve the overall quality of a Bag-of-Features implementation. In a future work, to evaluate the feasibility of automatic DR diagnosis using the BoF method, it would be a good idea to also implement a diagnosis method using a CNN and, with the same dataset, perform a comparative evaluation between the two implementations. This would give a more reliable view of the advantages and disadvantages of both techniques. This said, GPU-SURF-based BoF for automatic DR diagnosis justifies its implementation in medical applications where the use of high-resolution images is chosen to improve the results and where computational and time efficiency is a requirement.

Appendix A

Code

The code used to run the CPU version of SURF is open source, implemented in C/C++, and also contains OpenCV libraries. It is an adaptation of the code from the original authors of SURF by Chris Evans of the University of Bristol. The GPU version of the SURF algorithm used was from the Max Planck Institut based in Germany. This is based on Evans' code, mentioned above. To run the code it is necessary to have the following software dependencies installed: CUDA Toolkit and SDK, OpenCV (version 2.4), CUDPP, and Cmake library.

Both versions have been modified a bit to fit our needs, to evaluate performance, and to work properly in our specific case.

Regarding the Bag-of-Features algorithm itself, its implementation was done in a MATLAB script, receiving as input a list of text files where the complete information of the SURF feature descriptors for each of the images in our dataset was stored. This method results in a table where each row is a histogram with a K dimension. To implement the classification using SVM, the MATLAB Toolbox "Machine Learning and Deep Learning" was used with the previous table as input. The source code can be supplied upon request.

References

[1] M. Rama, K. Mookiah, U. R. Acharya, C. Kuang, C. Min, E. Y. K. Ng, and A. Laude, "Computer-aided diagnosis of diabetic retinopathy : A review," vol. 43, pp. 2136–2155, 2013.

[2] C. Evans, "Notes on the OpenSURF Library SURF : Speeded Up Robust Features," *University of Bristol Tech Rep CSTR09001 January*, no. 1, p. 25, 2009. [Online]. Available: http://www.mendeley.com/research/notes-opensurf-library-surf-speeded-up-robust-features/

[3] X. Su, J. Xu, and K. Ning, "Parallel-META : efficient metagenomic data analysis based on high-performance computation," *BMC Systems Biology*, vol. 6, no. Suppl 1, p. S16, 2012. [Online]. Available: http://www.biomedcentral.com/1752-0509/6/S1/S16

[4] A. N. Kollias and M. W. Ulbig, "Diabetic Retinopathy," *Dtsch Arztebl International*, vol. 107, no. 5, pp. 75–84, 2010. [Online]. Available: https://www.aerzteblatt.de/int/article.asp?id=67627

[5] R. Lee, T. Y. Wong, and C. Sabanayagam, "Epidemiology of diabetic retinopathy , diabetic macular edema and related vision loss," *Eye and Vision*, pp. 1–25, 2015. [Online]. Available: http://dx.doi.org/10.1186/s40662-015-0026-2

[6] "Diabetic retinopathy detection - identify signs of diabetic retinopathy in eye images," https://www.kaggle.com/c/diabetic-retinopathy-detection, accessed: 2022-05-30.

[7] C. Bommer, V. Sagalova, E. Heesemann, J. Manne-goehler, and R. Atun, "Global Economic Burden of Diabetes in Adults : Projections From 2015 to 2030," vol. 41, no. May, pp. 963–970, 2018.

[8] P. Hossain, B. Kawar, and M. El Nahas, "Obesity and diabetes in the developing world — a growing challenge," *New England Journal of Medicine*, vol. 356, no. 3, pp. 213–215, 2007, pMID: 17229948. [Online]. Available: https://doi.org/10.1056/NEJMp068177

[9] T. Lindeberg, “Image Matching Using Generalized Scale-Space Interest Points,” *Journal of Mathematical Imaging and Vision*, vol. 52, no. 1, pp. 3–36, 2015.

[10] M. Marinelli, A. Mancini, and P. Zingaretti, “GPU acceleration of feature extraction and matching algorithms,” *MESA 2014 - 10th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Conference Proceedings*, 2014.

[11] D. Ni, Y. Qul, X. Yang, Y. P. Chui, T. T. Wong, S. S. Ho, and P. A. Heng, “Volumetric ultrasound panorama based on 3D SIFT.” *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 11, no. Pt 2, pp. 52–60, 2008.

[12] N. M. Suaib, M. H. Marhaban, M. I. Saripan, and S. A. Ahmad, “Performance evaluation of feature detection and feature matching for stereo visual odometry using SIFT and SURF,” *IEEE TENSYMP 2014 - 2014 IEEE Region 10 Symposium*, pp. 200–203, 2014.

[13] L. Cheng, M. Li, Y. Liu, W. Cai, Y. Chen, and K. Yang, “Remote sensing image matching by integrating affine invariant feature extraction and RANSAC,” *Computers and Electrical Engineering*, vol. 38, no. 4, pp. 1023–1032, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2012.03.003

[14] A. Cesetti, E. Frontoni, A. Mancini, A. Ascani, P. Zingaretti, and S. Longhi, “A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 61, no. 1-4, pp. 157–168, 2011.

[15] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, “Vision-based autonomous navigation and landing of an unmanned aerial vehicle using natural landmarks,” *2009 17th Mediterranean Conference on Control and Automation, MED 2009*, no. March 2014, pp. 910–915, 2009.

[16] A. Mancini, A. N. Tassetti, A. Cinnirella, E. Frontoni, and P. Zingaretti, “A novel method for fast processing of large remote sensed image,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8157 LNCS, no. PART 2, pp. 409–418, 2013.

[17] H. Fadaifard, G. Wolberg, and R. Haralick, “Multiscale 3D feature extraction and matching with an application to 3D face recognition,” *Graphical Models*, vol. 75, no. 4, pp. 157–176, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.gmod.2013.01.002

[18] B. Alsadik, F. Remondino, F. Menna, M. Gerke, and G. Vosselman, “Robust extraction of image correspondences exploiting the image scene geometry and approximate camera orientation,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W1, no. February, pp. 1–7, 2013.

[19] G. Fangi and C. Nardinocchi, “Photogrammetric processing of spherical panoramas,” *Photogrammetric Record*, vol. 28, no. 143, pp. 293–311, 2013.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[21] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” vol. 60, no. 2, pp. 91–110, 2004.

[22] S. N. Sinha, J.-m. Frahm, M. Pollefeys, and Y. Genc, “GPU-based Video Feature Tracking And Matching,” vol. 012, no. May, pp. 1–15, 2006.

[23] J. Kim, E. Park, X. Cui, H. Kim, and W. A. Gruver, “A fast feature extraction in object recognition using parallel processing on CPU and GPU,” *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, no. November 2015, pp. 3842–3847, 2009.

[24] K. Cordes, L. Grundmann, and J. Ostermann, “Feature evaluation with high-resolution images,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9256, pp. 374–386, 2015.

[25] V. G. D, “Image Stitching Using Speeded Up Robust Features,” no. June, pp. 3514–3519, 2015.

[26] S. Mathew, C. Science, E. Division, C. Science, and E. Division, “A COMPARISON OF SIFT and SURF ALGORITHM FOR THE RECOGNITION OF AN EFFICIENT IRIS BIOMETRIC SYSTEM,” pp. 37–42, 2016.

[27] P. S. Pui and J.-l. Minoi, “Keypoint Descriptors in SIFT and SURF for Face Feature Extractions Keypoint Descriptors in SIFT and SURF for Face Feature Extractions,” no. February 2019, pp. 0–10, 2018.

[28] D. Gossow, P. Decker, and D. Paulus, “An Evaluation of Open Source SURF Implementations An Evaluation of Open Source SURF Implementations,” no. January, 2010.

[29] J. N. Surekha, Y. S. Chakrapani, and M. Kamaraju, “Implementation of High Performance Speeded Up Robust features Detection,” no. November, 2020.

[30] M. Agrawal, K. Konolige, and M. R. Blas, “CenSurE : Center Surround Extremas for Realtime CenSurE : Center Surround Extremas for Realtime Feature Detection and Matching,” no. October, 2008.

[31] M. Iturbe and K. Olga, “SURF and MU-SURF descriptor comparison with application in soft-biometric tattoo matching applications.”

[32] K. Mikolajczyk and C. Schmid, “A Performance Evaluation of Local Descriptors,” vol. 27, no. 10, pp. 1615–1630, 2005.

[33] E. Cortes-Gallardo, C. F. Moreno-Garcia, A. Zhu, D. Chipuli-Silva, J. A. Gonzalez-Gonzalez, D. Morales-Ortiz, S. Fernandez, B. Urriza, J. Valverde-Lopez, A. Marin, H. Perez, J. Izquierdo-Reyes, and R. Bustamante-Bello, “A comparison of feature extractors for panorama stitching in an autonomous car architecture,” *Proceedings - 2019 International Conference on Mechatronics, Electronics and Automotive Engineering, ICMEAE 2019*, pp. 50–55, 2019.

[34] P. Furgale and C. H. Tong, “ECE1724 Project Speeded-Up Speeded-Up Robust Features,” no. May, 2009.

[35] A. Mehrez, A. A. Morgan, and E. E. Hemayed, “Speeding up spatiotemporal feature extraction using GPU Speeding up spatiotemporal feature extraction using GPU,” *Journal of Real-Time Image Processing*, no. October 2018, 2019.

[36] A. I. Awad, “Fingerprint Local Invariant Feature Extraction on GPU with CUDA Preliminaries,” vol. 37, pp. 279–284, 2013.

[37] W. Yan, X. Shi, X. Yan, and L. Wang, “Computing OpenSURF on OpenCL and General Purpose GPU Computing OpenSURF on OpenCL and General Purpose GPU Regular Paper,” vol. 10, pp. 1–12, 2013.

[38] M. Marinelli, A. Mancini, and P. Zingaretti, "GPU Acceleration of Feature Extraction and Matching Algorithms."

[39] V. Codreanu, F. Dong, B. Liu, J. B. T. M. Roerdink, D. Williams, and P. Yang, "GPU-ASIFT : A Fast Fully Affine-Invariant Feature Extraction Algorithm," no. January 2014, 2013.

[40] M. Lalonde, D. Byrns, L. Gagnon, N. Teasdale, D. Laurendeau, and S. West, "Real-time eye blink detection with GPU-based SIFT tracking Real-time eye blink detection with GPU-based SIFT tracking," no. May 2014, 2007.

[41] T. B. Terriberry, L. M. French, and J. Helmsen, "GPU Accelerating Speeded-Up Robust Features."

[42] H. Fassold, "A Real-time GPU Implementation of the SIFT Algorithm for Large-Scale Video A Real-time GPU Implementation of the SIFT Algorithm for Large-Scale Video Analysis Tasks," no. February, 2015.

[43] N. Cornelis and K. U. Leuven, "Fast scale invariant feature detection and matching on programmable graphics hardware Fast Scale Invariant Feature Detection and Matching on Programmable Graphics Hardware," no. July, 2008.

[44] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, 2018. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21492

[45] A. Fanizzi, T. M. A. Basile, L. Losurdo, R. Bellotti, U. Bottigli, V. Didonna, A. Fausto, R. Massafra, M. Moschetta, P. Tamborra, S. Tangaro, and D. L. Forgia, "Open Access A machine learning approach on multiscale texture analysis for breast microcalcification diagnosis," *BMC Bioinformatics*, vol. 21, no. Suppl 2, pp. 1–11, 2020. [Online]. Available: http://dx.doi.org/10.1186/s12859-020-3358-4

[46] K. Horak, J. Klecka, O. Bostik, D. Davidek, and A. S. Learning, "Classification of SURF Image Features by Selected Machine Learning Algorithms," pp. 636–641, 2017.

[47] J. Majumdar and A. Mahato, "Comparison of SIFT SURF Corner Detector as Features and other Machine Learning Techniques for Identification of Commonly used Leaves," pp. 387–392, 2018.

[48] Y. Dwi, K.-c. Wang, J.-c. Wang, I. Idram, J.-y. Lai, J.-w. Liu, and I.-h. Hsieh, "Computer Methods and Programs in Biomedicine Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images," vol. 171, pp. 27–37, 2019.

[49] H. M. Sergieh and D. Lyon, "Improving SURF Image Matching Using Supervised Learning."

[50] A. Al-dmour and M. A. Abuhelaleh, "ARABIC HANDWRITTEN WORD CATEGORY CLASSIFICATION USING BAG OF FEATURES," no. July, 2016.

[51] T. Petraitis, R. Maskeliūnas, R. Damaševičius, D. Połap, and M. Woźniak, "Environment Recognition based on Images using Bag-of-Words," no. Ijcci, pp. 166–176, 2017.

[52] F. Murtaza and M. H. Yousaf, "Plant Species Identification using Discriminant Bag of Words (DBoW)," 2017.

[53] F. Yang and Y. Wang, "Study on Image Recognition and Classification of Wood Skin Defects Based on BOW Model," vol. 154, no. Meees, pp. 80–84, 2018.

[54] M. Luisa, D. Garcia, D. A. P. Soto, and L. S. Mihaylova, "A Bag of Features Based Approach for Classification of Motile Sperm Cells," 2017.

[55] S. R. Sajib, "A Feature Based Method for Real Time Vehicle Detection and Classification From On-Road Videos," no. Vdl, pp. 22–24, 2017.

[56] B. Hatipoglu and C. Kose, "A Gender Recognition System from Facial Images using SURF based BoW Method," vol. 5, pp. 6–10.

[57] C. M. Jin and Z. Omar, "A Mobile Application of American Sign Language Translation via Image Processing Algorithms," pp. 104–109, 2021.

[58] J. Ma, Z. Ma, B. Kang, and K. Lu, "A Method of Protein Model Classification and Retrieval Using Bag-of-Visual-Features," vol. 2014, 2014.

[59] I. Mehmood, M. Sajjad, K. Muhammad, S. Inayat, and A. Shah, "An efficient computerized decision support system for the analysis and 3D visualization of brain tumor," pp. 12 723–12 748, 2019.

[60] S. Bouteldja, T. H. Boumediene, A. Kourgli, and T. H. Boumediene, “High Resolution Satellite Image Indexing And Retrieval Using SURF Features And Bag of Visual Words,” no. March, 2017.

[61] G. Arora, A. Kumar, D. Zainul, A. Jaffery, and A. Rocha, “Bag of feature and support vector machine based early diagnosis of skin cancer,” *Neural Computing and Applications*, vol. 0123456789, no. iii, 2020. [Online]. Available: https://doi.org/10.1007/s00521-020-05212-y

[62] Y. Chen, “Bag-of-Features Model for ASD fMRI Classification using SVM,” no. Dl, pp. 52–57, 2021.

[63] N. Husna, M. Kadir, S. Nur, S. Mohd, N. Hidayah, and Z. Ibrahim, “Comparison of convolutional neural network and bag of features for multi-font digit recognition,” vol. 15, no. 3, pp. 1322–1328, 2019.

[64] C. E. Falc, J. Carvalho, S. D. Almeida, C. Silva, and S. Vieira, “Diagnosis of breast tissue in mammography images based local feature descriptors,” pp. 12 961–12 986, 2019.

[65] A. Nedra and M. Shoaib, “Detection and Classification of the Breast Abnormalities in Digital Mammograms via Linear Support Vector Machine .” pp. 141–146, 2018.

[66] R. Zhang, Y. Ming, and J. Sun, “Hand gesture recognition with SURF-BOF based on Gray threshold segmentation,” pp. 118–122, 2016.

[67] E. Unlu, E. Zenou, E. Belin, N. Riviere, and E. Belin, “Ordered Minimum Distance Bag-of-Words Approach for Aerial Object Identification,” 2017.

[68] S. D. Bhatt, “Image Retrieval using Bag-of-Features for Lung Cancer Classification,” pp. 531–536, 2021.

[69] M. V. Lahari and N. K. B, “Intelligent Content Based X-Ray Image Retrieval using Speeded up Robust Feature Descriptors,” no. December, pp. 18–19, 2017.

[70] Q.-K. Nguyen, T.-L. Le, and N.-H. Pham, “Leaf based plant identification system for android using surf features in combination with bag of words model and supervised learning,” pp. 404–407, 2013.

[71] C. A. Aguilera, C. Aguilera, and A. D. Sappa, “Melamine Faced Panels Defect Classification beyond the Visible Spectrum,” pp. 1–10, 2018.

[72] M. E. Kundegorski and S. Akc, "On using Feature Descriptors as Visual Words for Object Detection within X-ray Baggage Security Screening," vol. 44, no. June, pp. 0–6, 2017.

[73] H. K. Suh, J. W. Hofstee, J. Ijsselmuiden, and E. J. V. Henten, "ScienceDirect Sugar beet and volunteer potato classification using Bag-of-Visual-Words model , Scale-Invariant Feature Transform , or Speeded Up Robust Feature descriptors and crop row information," *Biosystems Engineering*, vol. 166, pp. 210–226, 2017. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2017.11.015

[74] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001.

[75] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, no. July 2006, pp. 404–417, 2006.

[76] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k -means clustering algorithm," vol. 36, pp. 451–461, 2003.

[77] D. Schmitt and N. Mccoy, "Object Classification and Localization Using SURF Descriptors," pp. 1–5, 2011.

[78] M. Oquab and L. Bottou, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks."

[79] C. Vapnik, "Support-Vector Networks," vol. 297, pp. 273–297, 1995.

[80] Y. Lei, "Individual intelligent method-based fault diagnosis," pp. 67–174, 12 2017.

[81] P. Govindaraj, "Performance comparison of various feature descriptors in object category detection application using SVM classifier," no. January, 2019.

[82] A. James, "Object Recognition and Classification Based on Improved Bag of."

[83] H. Esmaeilzadeh, E. Blem, R. St, A. Karthikeyan, and S. Doug, "Dark Silicon and the End of Multicore Scaling."

[84] K. Ataman and W. N. Street, "Optimizing Area Under the ROC Curve using Ranking SVMs."

[85] B. Antal and A. Hajdu, "Knowledge-Based Systems An ensemble-based system for automatic screening of diabetic retinopathy," vol. 60, pp. 20–27, 2014.

[86] B. D. Association *et al.*, "Retinal photography screening for diabetic eye disease," *London: BDA*, 1997.

[87] R. Sundaram and K. Ravichandran, "An automated eye disease prediction system using bag of visual words and support vector machine," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 4025–4036, 2019.

[88] M. Islam, A. V. Dinh, and K. A. Wahid, "Automated Diabetic Retinopathy Detection Using Bag of Words Approach," pp. 86–96, 2017.

[89] I. Sadek, D. Sidibé, and F. Meriaudeau, "Automatic discrimination of color retinal images using the bag of words approach," vol. 9414, p. 94141J, 2015. [Online]. Available: https://doi.org/10.1117/12.2075824

[90] F. Mukti, C. Eswaran, and N. Hashim, "Detection and classification of diabetic retinopathy anomalies using bag-of-words model," *Journal of Medical Imaging and Health Informatics*, vol. 5, 09 2015.

[91] A. Erciyas, "An Effective Method for Detecting and Classifying Diabetic Retinopathy Lesions Based on Deep Learning," vol. 2021, 2021.

[92] M. R. Kirchner, "Automatic thresholding of sift descriptors."

[93] N. Memari, A. Rahman, R. M. Iqbal, B. Saripan, S. Mashohor, and M. Moghbel, "Retinal Blood Vessel Segmentation by Using Matched Filtering and Fuzzy C - means Clustering with Integrated Level Set Method for Diabetic Retinopathy Assessment," *Journal of Medical and Biological Engineering*, vol. 39, no. 5, pp. 713–731, 2019. [Online]. Available: https://doi.org/10.1007/s40846-018-0454-2

[94] K. Akyol, F. Baha, and F. J. Bay, "Automatic Detection of Optic Disc in Retinal Image by Using Keypoint Detection , Texture Analysis , and Visual Dictionary Techniques," vol. 2016, 2016.

[95] S. Loussaief, A. Abdelkrim, and F. Detection, "Deep Learning vs . Bag of Features in Machine Learning for Image Classification," no. March, 2018.