



**ÇANKAYA UNIVERSITY  
FACULTY OF ENGINEERING  
COMPUTER ENGINEERING DEPARTMENT**

**Project Report**  
Version 2

**CENG 408**  
Innovative System Design and Development II

**P201709**  
**DETECTION OF OBSTRUCTIONS IN THE VESSELS  
IN FUNDUS IMAGES**

*Aykut ER*  
*201311018*  
*Egeberk ÖZBERK*  
*201411045*

Advisor: *Roya CHOUPANI*

# Table of Contents

---

Table of Contents .....	ii
Abstract .....	v
Özet: .....	v
1. Introduction.....	6
1.1 Problem Statement .....	6
1.2 Motivation .....	6
1.3 Solution Statement .....	6
1.4 Contribution.....	7
2. Literature Search .....	7
2.1 Gabor Filter Method.....	7
2.2 Ant Colony System Algorithm.....	8
2.3 Rician Denoise Algorithm & Thresholding.....	9
2.4 Image Matting Technique.....	10
3. Summary .....	12
3.1 Technology Used.....	12
4. Software Requirements Specification .....	12
4.1 Introduction .....	12
4.1.1 Purpose .....	12
4.1.2 Scope of Project.....	12
4.1.3 Glossary.....	13
4.1.4 Overview of Document .....	13
4.2 Overall Description .....	13
4.2.1 Product Perspective .....	13
4.2.2 User Characteristics.....	14
4.2.3 Constraints.....	14
4.2.4 Risks .....	14
4.2.5 Assumptions .....	14
4.3 Requirements Specification.....	14
4.3.1 External Interface Requirements .....	14
4.3.2 Functional Requirements.....	15
4.3.3 Performance Requirements .....	15
4.3.4 Software system attributes.....	15
5. Software Design Document .....	16
5.1 Introduction .....	16

5.1.1	Purpose .....	16
5.1.2	Scope .....	16
5.1.3	Glossary .....	16
5.1.4	Overview of document .....	17
5.2	Architecture Design.....	17
5.2.1	Simulation Design Approach ... ..	17
5.3	Architecture Design of Obstruction Detection System .....	18
5.3.1	Options Menu... ..	18
5.3.2	Find Congestion.....	18
5.4	Use case realizations .....	19
5.4.1	Brief Description of Figure 5.4.1.....	19
5.5	User Interface .....	20
5.5.1	Overview of User Interface .....	20
6.	Test Plan.....	27
6.1	INTRODUCTION .....	27
6.1.1	Version Control .....	27
6.1.2	Overview .....	27
6.1.3	Scope .....	27
6.1.4	Terminology .....	27
6.2	FEATURES TO BE TESTED.....	27
6.2.1	Graphical User Interface (GUI).....	27
6.2.2	Segmentation Process.....	28
6.2.3	Emboli Detection Process.....	28
6.3	ITEM PASS / FAIL CRITERIA .....	28
6.3.1	Exit Criteria .....	28
6.4	REFERENCES .....	28
6.5	TEST DESIGN SPECIFICATIONS .....	28
6.5.1	Graphical User Interface (GUI).....	28
6.5.2	Segmentation Process (SP).....	30
6.5.3	Emboli Detection Process (EDP) .....	31
6.6	Detailed Test Cases .....	32
6.6.1	GUI.UPLIMG_BTN.01.....	32
6.6.2	GUI.FIND_BTN.01.....	32
6.6.3	GUI.MARK_BTN.01 .....	33
6.6.4	GUI.MARK_BTN.02.....	33
6.6.5	GUI.SAVE_BTN.01.....	34
6.6.6	GUI.IMDISP.01 .....	34

6.6.7	GUL.RESDISP.01 .....	35
6.6.8	SP.PE_TPF.01 .....	35
6.6.9	SP.PE_FPF.01 .....	36
6.6.10	SP.PE_ACC.01 .....	36
6.6.11	SP.PE_TIME.01 .....	37
6.6.12	EDP.PE.01 .....	37
7.	Test Results .....	38
7.1	Individual Test Results .....	38
7.2	Summary of Test Results .....	39
7.3	Exit Criteria .....	39
7.4	Known Problems .....	39
7.5	Conclusion .....	40
8.	Conclusions .....	40
9.	User Manual .....	40
9.1	General Briefing .....	40
9.2	Software Information .....	41
9.3	System Overview .....	43
10.	Compilation / Installation Guide .....	43
	Acknowledgement .....	44
	References .....	45

## Abstract

As we progress in computer technologies, the number of solutions and software that we produce to facilitate our lives is increasing at the same rate. There has been a significant increase in the number of software solutions developed for the medical field over the past decade, especially because of the increased knowledge we have gained through studies conducted in the medical field. The goal of our work is to facilitate the ophthalmologist's work within the overcrowded hospital because of the growing human population and therefore the increased need for medical attention. The crowded hospital creates a stress that cannot be ignored on the patients waiting in the hospital and the doctor in charge of dealing with them. For this reason, on a busy day, the time to leave for a single patient is reduced, and the doctor may see a decrease in the diagnostic accuracy due to stress. The aim of the project we have developed is to make software that will enable the ophthalmologist to make a quick and well accurate diagnosis even under the conditions mentioned above. This software is a system that can determine the locations of the retinal vessels in a fundus image and, if exists, the locations of the emboli and make an estimated diagnosis.

**Keywords:** Fundus Image, Ophthalmology, Emboli

## Özet

Bilgisayar teknolojilerinde ilerledikçe, hayatımızı kolaylaştırması için ürettiğimiz çözümlerin ve yazılımların sayısı da aynı oranda artmaktadır. Özellikle, medikal alanda yapılan çalışmalar ile artmış olan bilğimiz dolayısıyla, geçtiğimiz yıllarda medikal alan için geliştirilen yazılım çözümlerinin sayısında gözle görülür bir artış olmuştur. Bizim çalışmamızın hedefi ise, artan insan nüfusu ve dolayısıyla artan medikal ilgi ihtiyacı sebebi ile normalden fazla kalabalıklaşan hastane içerisinde göz doktorun işini kolaylaştırmaktır. Kalabalıklaşan hastane içerisinde bekleyen hastalar ve onlar ile ilgilenmekle sorumlu doktor üzerinde göz ardı edilemeyecek bir stres oluşturmaktadır. Bu sebepten ötürü yoğun bir günde hem tek bir hasta için ayrılacak süre azalmaktadır hem de stres sebebi ile doktorun teşhis doğruluğunda azalma görülebilir. Bizim geliştirdiğimiz projenin amacı ise, göz doktorunun yukarıda bahsedilen koşullar altında bile hızlı ve dengeli teşhis yapabilmesini sağlayabilecek bir yazılım yapmaktır. Yazılım göz damarlarının yerlerini ve varsa, damarlarda embolinin olduğu yerleri belirten ve sonucunda ise tahmini bir teşhis yapabilecek bir sistemdir.

**Anahtar Kelimeler:** Fundus Fotoğrafı, Göz ve Göz Hastalıkları Bilimi, Emboli

# **1. Introduction**

## **1.1 Problem Statement**

The eye is a vital organ for the human being. There is almost no work in our life we do not use our eyes. For this reason, we give our eyes the necessary attention and we go to the eye doctor immediately in the slightest problem that arises in our eyes. With the same respect, ophthalmologists need a high degree of attention and precision so that they can diagnose the problems that arise in our eyes accurately. Any slightest mistake made during the diagnosis process will cause a problem that may corrupt the patient's sight or even cause blindness. Unfortunately, under difficult working conditions, the ophthalmologist may miss out on an important detail in the retinal image that he looked at in the slightest lack of attention. The problem is that the eye veins are too thin and hard to see. It is even more difficult to see the obstructions in the eye veins. We aim to facilitate the ophthalmologist's work with software that can determine any of these vessels and any congestions occurring inside them.

## **1.2 Motivation**

We are a group of senior students in computer engineering department who are interested in image processing. As a group, in this project we aimed to help the doctors in the treatment of the patients. Since we do not have knowledge about medical science, we received help from an ophthalmologist working in this area and we read the documents about this topic to find out the diseases in the eye. In order to find out where the illnesses in the eye are, we have been working on image processing and we have searched how to use Matlab to develop the necessary processes for this topic.

## **1.3 Solution Statement**

In this section, we will share our solutions to the problems that are mentioned in section of Problem Statement.

First of all, if we're going to work with retinal vessels, we have to know exact location and features of the vessels. Normally however, we were only going to find locations of the congestion spots in the retinal image in order to reduce computational time. Later on that discussion, we have come to a conclusion that we need immense precision and sensitivity about the location data of those congestion spots as mentioned in problem statement section. The fact is, retinal images are too unstable, too changeable to rely on static constants like congestion color or vein thickness thus, we have to rely on something more precise, more certain. Therefore, we'll start our process with a segmentation method which will retrieve sensitive data of vessel locations. We need to know the coordinates of the vessels because we can't be sure whether the congestions found by the software are actually congestions or not. There is a huge risk of confusing a found spot with a noise or some other part of the eye flora.

In order to eliminate such risk, we'll develop a congestion detection method which will find any congestion spot that it can find then check the congestion spot with the coordinates extracted by the segmentation method which also we'll develop and if the congestion spot

coordinate matches one of the coordinates of the vessels, or in other words, if a congestion spot is actually inside of a vein only then, it'll be considered as congestion point. Even after processing image, our software will run a feedback mechanism afterwards to be sure that there will be absolutely no mistakes by allowing the ophthalmologist to mark every single congestion spot whether it is actually congestion or a mistake. With this technique, software will achieve accurate results leading the ophthalmologist to diagnose faster.

## **1.4 Contribution**

With to this software, the ophthalmologist can diagnose multiple patients in the time needed to diagnose a single patient. In addition, early and more accurate diagnosis will prevent the disease to advance any further thus, it may even save some from going blind.

## **2. Literature Search**

### **2.1 Gabor Filter Method**

In 2007, Alauddin Bhuiyan, Baikunth Nath, Joselito Chua and Ramamohanarao Kotagiri proposed a texture based vessel segmentation method to overcome the problem that is the division of the entire vessel is not easy due to the changes in the constructions in the small vessels.

The proposed method is based on blood vessel segmentation and texture classification of vessels and non-vessels in color retinal images [1]. One of the reasons for developing this method is that the texture analysis performs well in determining the vessel and non-vessel parts of the images. Another reason is that the original images are contained in RGB color pixels and the addition of color to the texture analysis due to the color distributions in the images makes the process even easier. 3D RGB space does not contain colors that people can perceive [1]. So, uniform color spaces are used along with RGB color channels to find the textures.

Due to high contrast for texture analysis,  $E$  and  $E\lambda$  in the Gaussian color space,  $L$  luminance in  $L^*a^*b$  color space, and green channel  $G$  in RGB color space are selected. Adaptive Histogram Equalization (AHE) method [2] is applied to these four color space images. For these color channels, a Gabor filter with twenty-four directions and three wavelengths is applied to determine the textures. After the texture clusters are analyzed and the texture image is formed, a feature vector is generated for each pixel in the texture image. These vectors are classified as a vessel or background part [1] using unsupervised FCM clustering algorithms [3-4]. The result of this algorithm is to create a 2D matrix and transform the cluster numbers to binary values to generate a segmented image.

The maximum response value in the color channels is reduced the complexity in the training on data and the length of the feature vector. They also generate an image for histogram analysis and for each color channel to determine the cluster number. For each pixel from these images, the feature vector of length of twelve elements is classified according to vessel and non-vessel using FCM [3-4] algorithm [1].

The values of this technique is calculated using the DRIVE [5] database. For performance evaluation, we detected the vessel center line in our output segmented images and hand labeled ground truth segmented (GT) images applying the morphological thinning operation. Since the vessel width in the GT images does not always coincide with the position of the vessel, the images in which the segmentation process is performed cannot be compared with the GT images. In their work, five skeletonized images are compared with GT images and they assume a maximum of two pixels in the images. In general, 84.37% sensitivity and 99.61% specificity were obtained. But they did not specify the accuracy rate in their work. So we do not know exactly about the performance of the work. According to their performance gives good results compared to other studies, like Hoover et al. [6] method.

In this method, vessel width and vessel bifurcation and crossover detection calculations are not carried out, so extra procession is performed on the images for these calculations, but those who develop this method continue to work on these problems.

## **2.2 Ant Colony System Algorithm**

In 2013, another study related for vessel segmentation [7] has done Ahmed Hamza Asad, Ahmad Taher Azar, Mohamed Mostafa M. Fouad, Aboul Ella Hassanien. Their approach is mainly based on an algorithm called ant colony system and they've also used a modified version of their old heuristic function for improvement. Their proposed method consists of 4 steps and these are Preprocessing, Features Computation, ACS Based Segmentation and Post-Processing.

In preprocessing step, green channel of the original image is extracted since it provides the best contrast harmony between vessel and the background and because of the fact that green channel image will be used in further processing, its contrast is enhanced. After enhancements, reflection of the optic disc is removed as its overlapping with some of the central vessels' lengths.

The features computation process aims to distinguish vessels from the background by correcting varying illumination in background. Correction is achieved by computing homogenized background and gray-level based features ( $f_2$ ,  $f_5$ ) [7]. After obtaining gray-level based features, vessel-enhanced image is computed its gray-level ( $I_{ve}$ ) is used for used for computing Hu moment-invariants based feature ( $Hu_1$ ) (detailed in [8]) as well as it's selected feature by CFS (detailed in [9]) [7].

The main method used for segment blood vessels in this approach is an improved version of ant colony system algorithm proposed by Dorigo et al. [10]. Purpose of this algorithm was solving the travelling salesman (TSP) and it's based on behavior of real ants.

Background for ACS: In nature when real ants are searching for foods, multiple ants are going out in random paths. As the ant is moving, it deposits a chemical substance which is called pheromone on its moving path for guiding other subsequent ants to its path. As the time goes, the pheromone is evaporating. So as the path is shorter as its pheromone concentration



remains more time and more other ants are attracted to it. Thus the shortest path is the only one which attracts other ants [7].

More detailed version of the ACS could be found in [11]. As for the ACS based segmentation part, according to their pheromone level and heuristic function value, each pixel is classified as either vessel pixel or background pixel.

In post-processing step, firstly disjoined pixels are handled. If a pixel is surrounded with at least four neighboring pixels, its value is assigned as 1 and 0 if otherwise. Secondly, regions which have area less than 20 pixels are removed. Finally, a median filter of size 3\*3 eliminates all remaining isolated noisy pixels [7].

Proposed method needs to be improved as the accuracy they've achieved is not much competitive against the state-of-the-art methods however, they've achieved quite good sensitivity because of the simplicity of its used features.

Database used in these experiments is the STARE [12] database. In the future, experiments should include the database DRIVE [5] and by focusing more on false positives reduction, performance in terms of specificity and accuracy should be improved in order to be competitive against state-of-the-art methods.

## **2.3 Rician Denoise Algorithm & Thresholding**

In 2016, another study about blood vessel segmentation has been proposed by Kimmy Mehta and Navpreet Kaur from Punjabi University. They obtained background image with morphological operations and performed the segmentation using thresholding method.

Like the research presented in the previous works, they also used green channel for the detection and segmentation of retinal blood vessels from the retinal image because the green channel is high sensitive to the blood vessels [13]. In other words, it has been considered in existing vessel segmentation studies since it has higher contrast compared to other channels [14-15]. Detection of green channels, using basic morphological processes and thresholding methods, is used for the emergence of vascular structure in RGB retinal images and segmentation of blood vessels.

They wanted to get the original image by removing the background retinal image from the image. The background image is estimated [16] using morphological operations [17]. In this study they've used morphological closing operation to find the background retinal image. Since closing operation suppresses dark details smaller than the structuring element [16]. They tried to produce the background by removing the vessels in the retinal image.

However, because of the still existing noise from original image was a problem, they had to remove even the slightest noise in the image to get the highest quality possible. The image denoising process involves image transformation that can be easily identified by noise, and then is reversed to reduce noise [15]. They've used Rician Denoise algorithm [18], which is a total variation based algorithm [16] that tries to remove noises, to overcome this problem.

The vascular structure of the retinal image is divided using the thresholding method proposed by Phansalkar et al. [19] in 2011. To adapt this method to their problems, they attempted to solve their problems by changing the thresholding scheme globally instead of locally [16]. This is done by changing the size of the local search area according to whole image. Then a calculated threshold is used for the vascular structure.

After these operations, the image of the segmentation process does not give sufficient determination to examine any disease. They did two post-processing for this problem and to improve the performance of this algorithm. First one for removing the spur pixels for final enhancements and second one for removing any region which is not belong to the actual vascular structure. From these, area filtering is an application used to erase the spur pixels that do not belong to the vascular structure. This method is done using the Connected Component Analysis (CCA) technique, in which the image is labeled according to the pixel connections in the different components [18] and these connections are 4 or 8 connectivity. In this study, 8-way connectivity was used.

The performance of the proposed algorithm is evaluated in two databases for automatic segmentation. Databases were DRIVE database [5] and the STARE database [12], which includes half of the number of images inside the DRIVE. The developed method is evaluated with sensitivity, specificity and accuracy metrics.

The average accuracy for the DRIVE [5] database is 94.35%, the average specificity is 98.13% and the sensitivity is 68.15%. The average accuracy for the STARE [12] database is 94.49%, the average specificity is 72.12% and the sensitivity is 97.01%.

This method performed better when compared to previous years' work and gave similar results. This simplicity, robustness, rapid implementation and effectiveness of this algorithm can be suitable method for pre-screening systems for early detection.

## **2.4 Image Matting Technique**

In 2017, another study related for vessel segmentation has done by Zhun Fan, Jiewei Lu, Wenji Li, Caimin Wei, Han Huang, Xinye Cai and Xinjian Chen. They actually come up with a different technique so called image matting and their proposed method is called hierarchical image matting model. They argued that image matting has never been used to segment blood vessels [20].

Image matting aims to extract the foreground of the image accurately. More specifically, model the input image as a linear combination of background and foreground images.

It is vital to enhance the image before any process has applied in order to perform segmentation efficiently [21]. They've used two efficient filters to enhance the input image which are Morphologically Reconstructed Filter [22] and the other one is the Isotropic Undecimated Wavelet Filter [23]. Both of these filters extract the green channel image since green channel image provides the best vessel-background contrast and it has much less noise compared to the red and blue channels [24].

They've used argued that creating a user-specified trimap is very time consuming and requires high amount of effort [20] therefore, they've decided to develop another process for generating the trimap automatically.

Process of generating trimap consists of two main steps, image segmentation and vessel skeleton extraction. Image segmentation part deals with the image which was enhanced with the morphologically reconstructed filter whereas the vessel skeleton extraction part deals with the image which was enhanced with the undecimated wavelet filter. At image segmentation part, enhanced image is segmented into three regions which are unknown regions, background regions and preliminary vessel regions. Because of preliminary vessel regions still contains regions with unwanted noise, image has to be denoised, resulting the denoised preliminary vessel regions. At skeleton extraction part, for the purpose of providing more information about vessel skeleton and unknown regions, firstly by applying global thresholding to the enhanced image, binary image is obtained. Then binary image is divided into three parts according to the area feature. Finally extraction [25] process is applied in order to obtain skeleton of blood vessels. After applying both skeleton extraction and image segmentation parts, trimap is generated with combining background regions, unknown regions and foreground regions which is the union of resulting the denoised preliminary vessel regions and skeleton of blood vessels.

For the purpose of handling pixels at the unknown regions, they've implemented a hierarchical image matting model to check if the pixels in the unknown region are background pixel or a vessel pixel. This implementation includes 2 steps. First one is stratifying the unknown pixels. In this step, for each pixel in the unknown region a distance value is obtained. This value is the distance between the unknown region pixel and the nearest vessel pixel. Then, distance values are sorted in ascending order and same distance values are grouped into one hierarchical layer in order to create a hierarchical order set. Needless to say, which is also in ascending order?

Second step is hierarchical update where new labels are assigned to each pixel in each hierarchical set member by checking whether it is a background pixel or a vessel pixel. We already have a labeled pixel set consist of background and vessel pixels returned from the trimap generation process. We'll now use those sets for updating our unknown pixels in each hierarchy by calculating their correlations to its neighboring labeled pixels (9x9). After those calculations we chose the pixel with nearest correlation value and give the label of it to the unknown pixel. This method also has a post-processing method where they remove misclassified vessel regions in order to obtain highest accuracy value possible.

Their experiments show that their model is vastly competitive. Image matting model has been tested with three different databases, DRIVE [5], STARE [12] and CHASE\_DB1, which includes 28 fundus images with large variety. Accuracy percentages they measured are %96, %95.7 and %95.1 for DRIVE, STARE and CHASE\_DB1 respectively. They achieved fairly low computing time for such accuracy values.

### **3. Summary**

#### **3.1 Technology Used**

This software will communicate with Matlab to run its image processing functions. Software itself will be developed with C# language. Software also requires a fundus camera to obtain high quality retinal images for best performance and this device is determined to be a Zeiss VISUCAM 500, capable of obtaining fundus images with resolution up to 5MP.

The target platform will be Microsoft Windows and development environment will be Microsoft Visual Studio 2013. Functions and methods for processing fundus image will be implemented on Matlab 2016.

### **4. Software Requirements Specification**

#### **4.1 Introduction**

This part of the document provides detailed information about the requirements of obstructions detection system software. It will explain the purpose and features of the proposed method, the working principles of this method and the general information about the difference of the proposed method from previous studies. This document is intended for both stakeholders and developers who are working on such work.

##### **4.1.1 Purpose**

This project's main purpose is to find congestions inside vessels in eye angiographic images therefore, primary objective is to reduce the effort spent by the doctor to detect eye diseases.

The main motivation for the ophthalmologist to use this software is to allow the software to interpret the image much more quickly with immense accuracy and precision to reduce the time spent on a single patient so that the doctor can deal with more patients in the same time.

##### **4.1.2 Scope of the Project**

This software system will be an Obstruction Detection System for an ophthalmologist. This system will be designed to maximize the ophthalmologist's productivity by providing tools to assist in automating the congestion detection process inside retinal vessels, which would otherwise have to be performed manually. By maximizing the ophthalmologist's work efficiency and production the system will meet the ophthalmologist's needs while remaining easy to understand and use. More specifically, the system is designed to automatically identify obstructions however, in case of any possible error, the ophthalmologist is also asked to provide a feedback about the congestion spots that system has already found.

###### **4.1.2.1 Benefits**

Thanks to this software, the ophthalmologist can diagnose multiple patients in the time needed to diagnose a single patient. In addition, early and more accurate diagnosis will prevent the disease to advance any further thus, it may even save some from going blind.

### 4.1.3 Glossary

Term	Definition
Fundus Image	Fundus photographs are visual records which document the current ophthalmoscopic appearance of a patient's retina [26].
Ophthalmologist	Doctor who specialized in eye and sight care.
Fluorescein	Chemical used for enhancing the brightness of the vessels.
SRS	Software requirements specification.
Obstruction	Congestion resulting eye strokes.
Stakeholder	People with any interest in project's outcome.
Diagnose	Identify the nature of the medical condition.
Emboli	Something that travels through the bloodstream, lodges in a blood vessel and blocks it. Examples of emboli are a detached blood clot, a clump of bacteria, and foreign material such as air.

Table 4.1.1 Glossary of SRS

### 4.1.4 Overview of the Document

The rest of this paper is organized as follows:

Section 2 describes the definition, specification of the project and properties of the method and its application for users who use the system and read the document. The constraints and risks of this method are mentioned. Section 3 is mainly written for developers of this system and describes in technical terms the details of the requirements of this system. Section 4 describes the specifications of actors and stakeholders and their interactions with the developed method are presented. The functions used by the user in order to use the project software and the tasks of these functions are described.

## 4.2. Overall Description

### 4.2.1 Product Perspective

The software described in this SRS allows the doctor mark congestions inside the veins by a specified geometrical shape and color in order to see them easily. After the software done its marking process, doctor can choose between two diagnosis option which are diagnose the disease himself/herself or letting the software to interpret the situation. Software will have pre-prepared results for certain values. Those result outputs are diagnostics collected from doctors prior to the development phase of the software. By selecting automatic diagnosis option, doctor can either accept the result or make changes on it. Either way, he/she accelerates the process of diagnosis.

### **4.2.2 User Characteristics**

Expectation from the users who read the document should have knowledge about the concepts related to image processing and in the eye area in medical science.

Expectation from the users who use the system should have knowledge about the topics and concepts related to medical science. That is, the user is expected to be an ophthalmologist.

### **4.2.3 Constraints**

Type of the image to be processed is a constraint for this software. Image needs to be in DICOM standards as most of the medical images share this standard.

This software is also constrained by the capacity of the storage of computer the software is running on as the software allows ophthalmologist to save results.

### **4.2.4 Risks**

While transferring the fundus image into the software, image quality may be reduced or can pick unwanted noise. Both will cause software to interpret the fundus image with poor accuracy. Software may mark spots without any congestion occurring or miss spots where the congestion actually exists.

### **4.2.5 Assumptions**

The accuracy and precision percentage of the software finding the congestion spots in the veins in fundus images will be in direct proportion to the quality of the photograph and also automatic diagnosis computed by this software will be generating outputs as similar as ones that the doctor himself/herself does.

## **4.3. Requirements Specifications**

### **4.3.1 External Interface Requirements**

#### ***4.3.1.1 User Interfaces***

The only UI shown in this project is the main form divided into sub parts which are:

- Upper horizontal menu including file operations (upload/save).
- Division where the pre or post processed image is shown.
- Result division for output and diagnosis to be shown.
- Toolbar to allow user to select marking shape and color of the congestion.
- Checkbox to indicate whether diagnose operation will be automatic or manual.

#### ***4.3.1.2 Hardware Interfaces***

No hardware interfaces needed to run this software however, fundus camera is recommended and required for gathering retinal images with maximum quality thence software will run on its maximum efficiency.

The fundus camera is required to be Zeiss VISUCAM 500 or newer.

#### ***4.3.1.3 Software Interfaces***

Software presented in this SRS does not need any other software interface than the operating system itself.

#### ***4.3.1.4 Communications Interfaces***

No internet connection is required to run this software thus there will not be any communication interfaces.

### **4.3.2 Functional Requirements**

- User shall be able to upload fundus images to the software.
- User shall be able to change color and geometrical shape of the marks.
- User shall be able to save processed image to specific location with specified name along with the results formatted in a .txt file.

### **4.3.3 Performance Requirements**

- Congestion marking process shall take no more than a minute.
- Accuracy percentage of the output image shall be higher than %60.

### **4.3.4 Software System Attributes**

#### ***4.3.4.1 Availability***

Users can use the system on computer environment. To be able to use the system on computer environment, the system is clicked and started by user.

#### ***4.3.4.2 Security***

Because there is no critical information to be kept, there are no security constraints.

#### ***4.3.4.3 Portability***

Because the system will be developed on computer environment, it can work only on computer.

#### ***4.3.4.4 Usability***

- Software shall accept image formats of .jpg, .png and .tif
- Software shall be able to maintain images with any size bigger than 565x565 but reject and display a warning message if smaller.

#### ***4.3.4.5 Scalability***

There is no scalability requirement because the system has only one user.

#### 4.3.4.6 Ease of Use

Since the developed system is a medical oriented project, it must be a doctor-friendly user interface and this interface should be simple and easy to use.

## 5. Software Design Document

### 5.1. Introduction

This part of the document provides detailed information about the design of the obstructions detection system software. It will explain the working principles of the proposed method, and design of the user interface of this method. This document prepared to guide to users.

#### 5.1.1 Purpose

The purpose of this document is providing the details of the proposed project "*Detection of obstruction in the vessels in fundus images*".

Fundus images (aka eye angiogram images) are the photographs that have been taken by fundus cameras after injecting a chemical called fluorescein inside patient's veins to provide better visibility by enhancing brightness of the vessels. Even with the latest technology is used to take angiographic photographs, the amount of human error and lack of precision is considerable while trying diagnosing the disease.

Our goal is to show the obstructions in vessels visible in the fundus images to the doctor who uses the software better than he/she can see himself/herself. The software to be developed will also be able to interpret the image to the user with the help of pre-prepared diagnostic estimates gathered from doctors who are our project partners.

#### 5.1.2 Scope

During the work hours, considering the frequency of patient arrivals is excessive, a doctor has a very short period to commentate every single image brought in front of him/her. Main aim here in this project is to reduce the time spent on diagnosis to almost an instant.

The main motivation for the ophthalmologist to use this software is to allow the software to interpret the image much more quickly with immense accuracy and precision to reduce the time spent on a single patient so that the doctor can deal with more patients in the same time.

Image processing algorithms and methods that software needs will be implemented by using Matlab and the software itself will be developed using C#.

#### 5.1.3 Glossary

Term	Definition
Fundus Image	Fundus photographs are visual records which document the current ophthalmoscopic appearance of a patient's retina [26].



Fluorescein	Chemical used for enhancing the brightness of the vessels.
Ophthalmologist	Doctor who specialized in eye and sight care.
SDD	Software Design Document.
Obstruction	Congestion resulting eye strokes.
Block Diagram	The type of schema that the components of the system are displayed in blocks
Matlab	Program to calculate mathematical operations and functions.

**Table 5.1.1 Glossary of SDD**

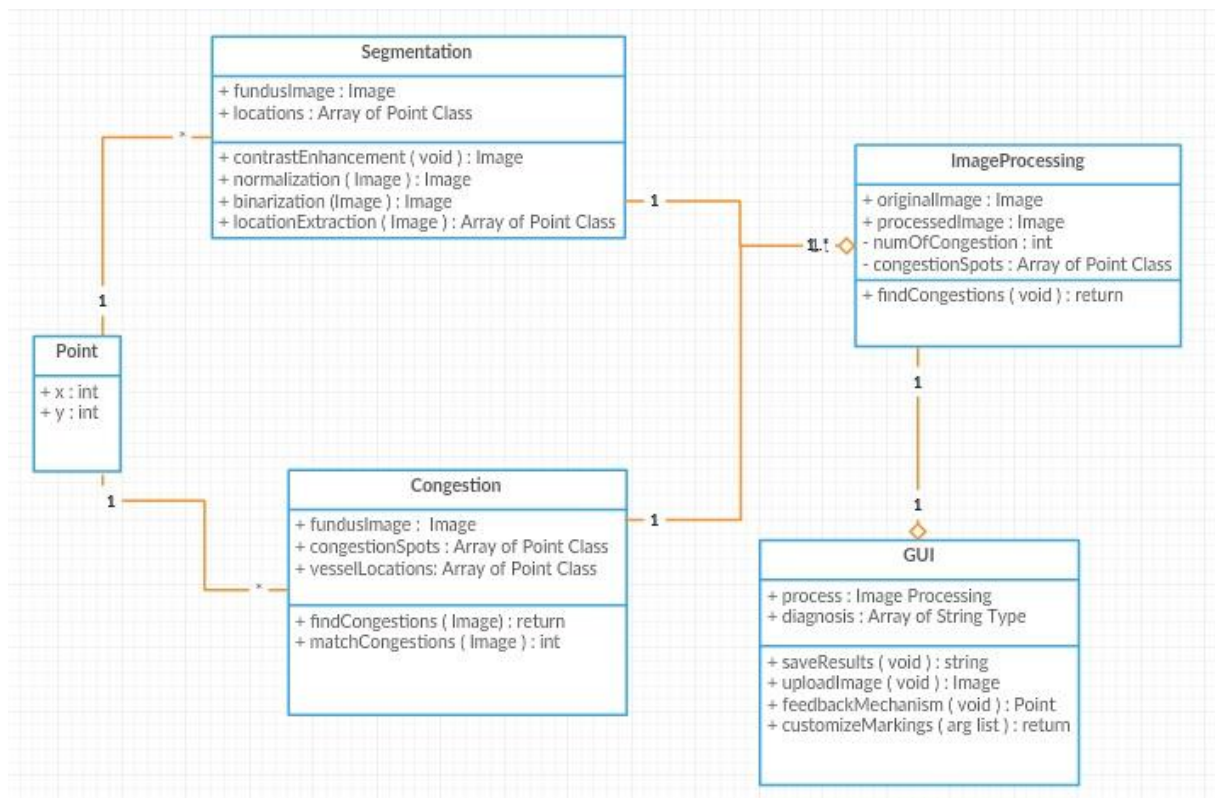
## 5.1.4 Overview of the Document

This section provides information about the contents of the rest of the document as follows: Section 2 describes the problem and details the design of this project along with the class architecture. Section 3 displays and explains the block diagram of the system, which is designed according to use cases in SRS document.

## 5.2 Architecture Design

### 5.2.1 Simulation Design Approach

#### 5.2.1.1 Class Diagram



**Figure 5.2.1 Class Diagram of Obstruction Detection System**

Figure 5.2.1 displays classes used in this system. GUI, master class here, is responsible of all main operations. It also runs main image processing operations segmentation and congestion

detection. GUI class also generates a diagnosis for the processed image and creates a result file to be saved. The ImageProcessing class includes Segmentation and Congestion which are responsible for their individual image processing methods. Segmentation class runs first in order to give the location map of the vessels to the Congestion class. Congestion then processes the image with the vessel map and its congestion spot map to return processed image to the ImageProcessing class. Both Segmentation and Congestion classes will run their functions with the help of Matlab. Lastly, point class represents a single coordinate for the congestion spots.

## **5.3 Architecture Design of Obstruction Detection System**

### **5.3.1 Options Menu**

**Summary:** This system is used by participant. Participant can upload a new image file to the main screen, save the image and change marking shape and color.

**Actor:** User / doctor.

**Precondition:** Software system must be running.

**Basic Sequence:**

1. User can upload a new image file by selecting new image button.
2. User can save the image by selecting save button.
3. User can customize markings by selecting customize marking button.
4. User can customize color and shape of the marking by selecting shape or color button individually under customize marking button.

**Exception:** Save the image without any operations performed.

**Post Conditions:** None.

**Priority:** Medium

### **5.3.2 Find Congestion**

**Summary:** This system is used by participant. Participant wants to see congestion spots in the fundus image.

**Actor:** User / doctor.

**Precondition:** Image must be uploaded.

**Basic Sequence:**

1. User pressed find congestions button.
2. System shows each congestion point.
3. Feedback mechanism starts running.
4. User can select each of the congestion point to give feedback.

**Exception:** No image has uploaded to perform operation.

**Post Conditions:** Feedback mechanism starts.

**Priority:** High.

## 5.4 Use Case Realizations

### Obstruction Detection System Project

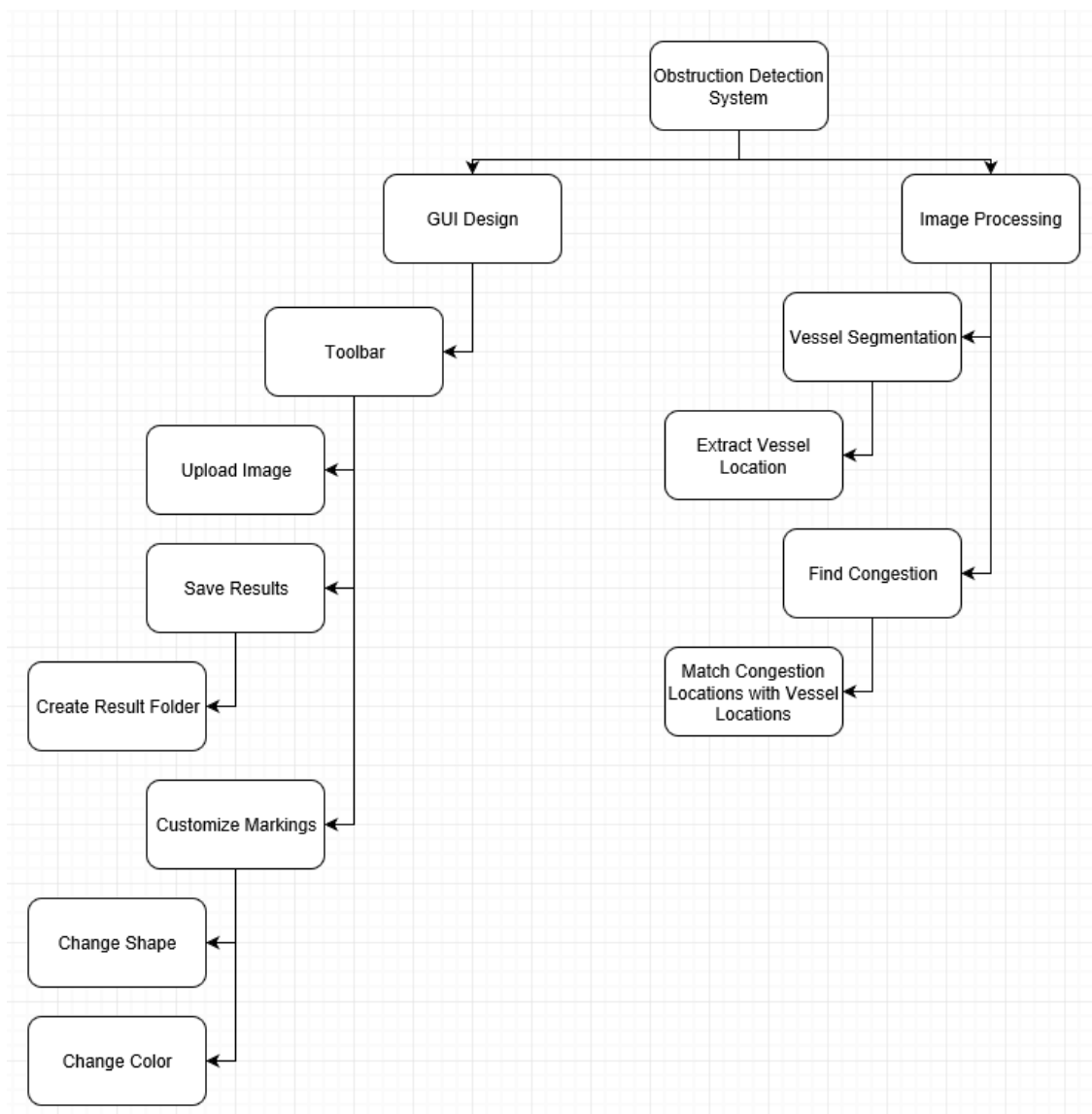


Figure 5.4.1 Project Components of Obstruction Detection System

#### 5.4.1 Brief Description of Figure 5.4.1

Components of the Obstruction Detection System Project are shown in the Figure 5.4.1. All designed systems of the system are displayed in the block diagram in the figure. There are two main components of the system which have their own sub-systems.

### ***5.4.1.1 GUI Design***

GUI design is responsible for interaction between the actors and the system. There is one subsystem in this design which has Main Menu and Main Menu has its own subsystem called Toolbar and Toolbar has three subsystems which are Customize Markings, Upload Image, and Save Results which also has its own subsystem called Create Result Folder and also finally Customize Markings has its own subsystems which are change shape and change color.

Main Menu is a start page, participant can do various operations which are uploading a new image file to be processed, change markings and save the results after the congestion detection operation has done.

### ***5.4.1.2 Image Processing***

Image Processing is responsible for all sorts of main image operations and composed of two main parts which are segmentation part and congestion detection part. Vessel Segmentation has its own subsystem called Extract Vessel Location and Find Congestion has its own subsystem called Match Congestion Locations with Vessel Locations.

Vessel Segmentation simply returns a vector of coordinates of the vessels which will be used by congestion detection function. Find Congestion first finds all spots where possible congestion occurs then uses the vector returned from Vessel Segmentation to eliminate all congestion spots which are not inside of any blood vessel.

## **5.5 User Interface**

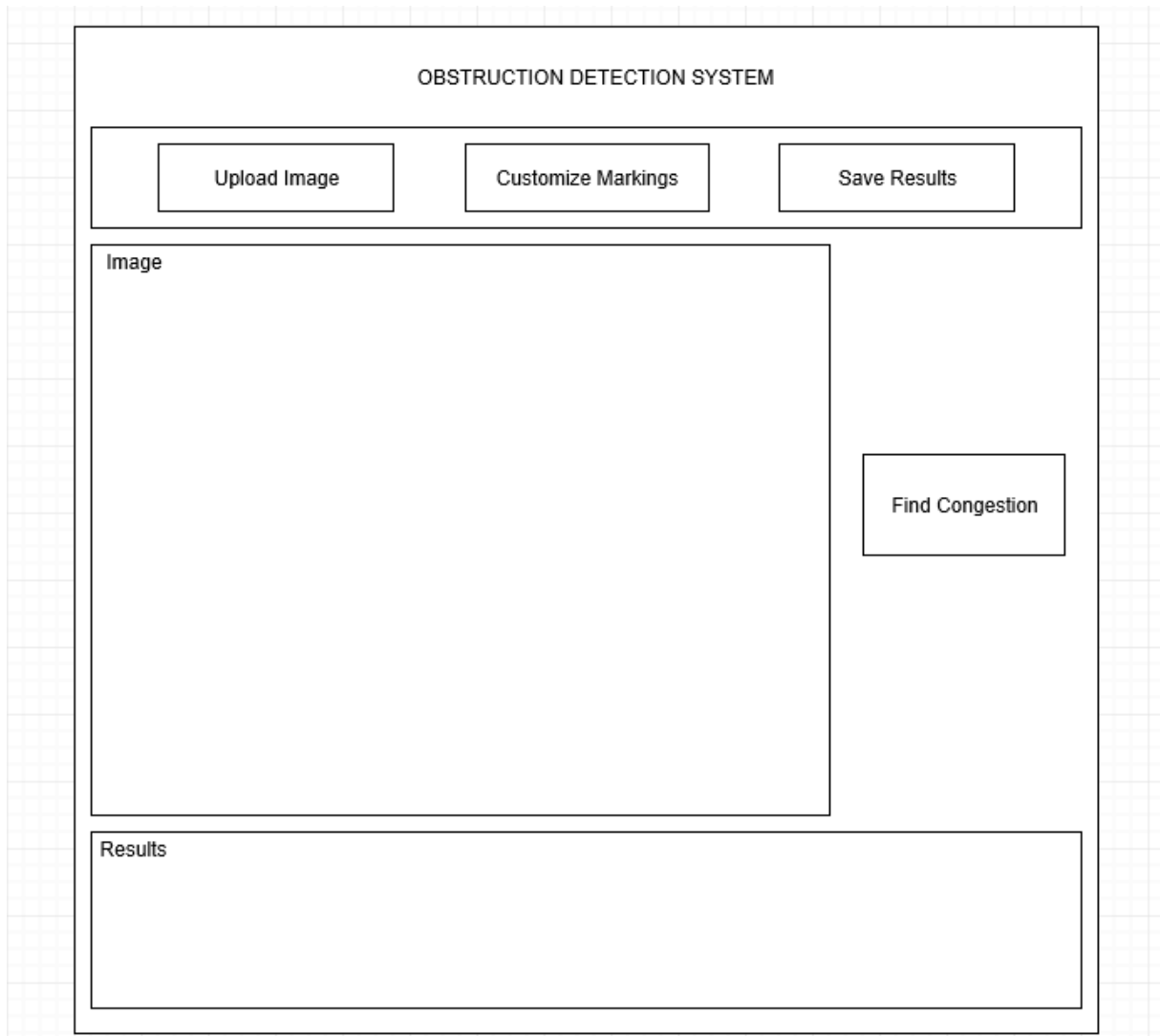
In this section of the document, buttons and layout of the main menu is presented in below figures.

### **5.5.1 Overview of User Interface**

#### ***5.5.1.1 Main Menu Layout***

The main menu is the interface that the user of the software will see when opening the program. The user can access all the main operations through this user-friendly interface. The upper part of the interface contains basic operations. The middle pane contains the image with a button that will trigger the main action "Find Congestion". Finally, the bottom of the interface is responsible for showing the results to the user.

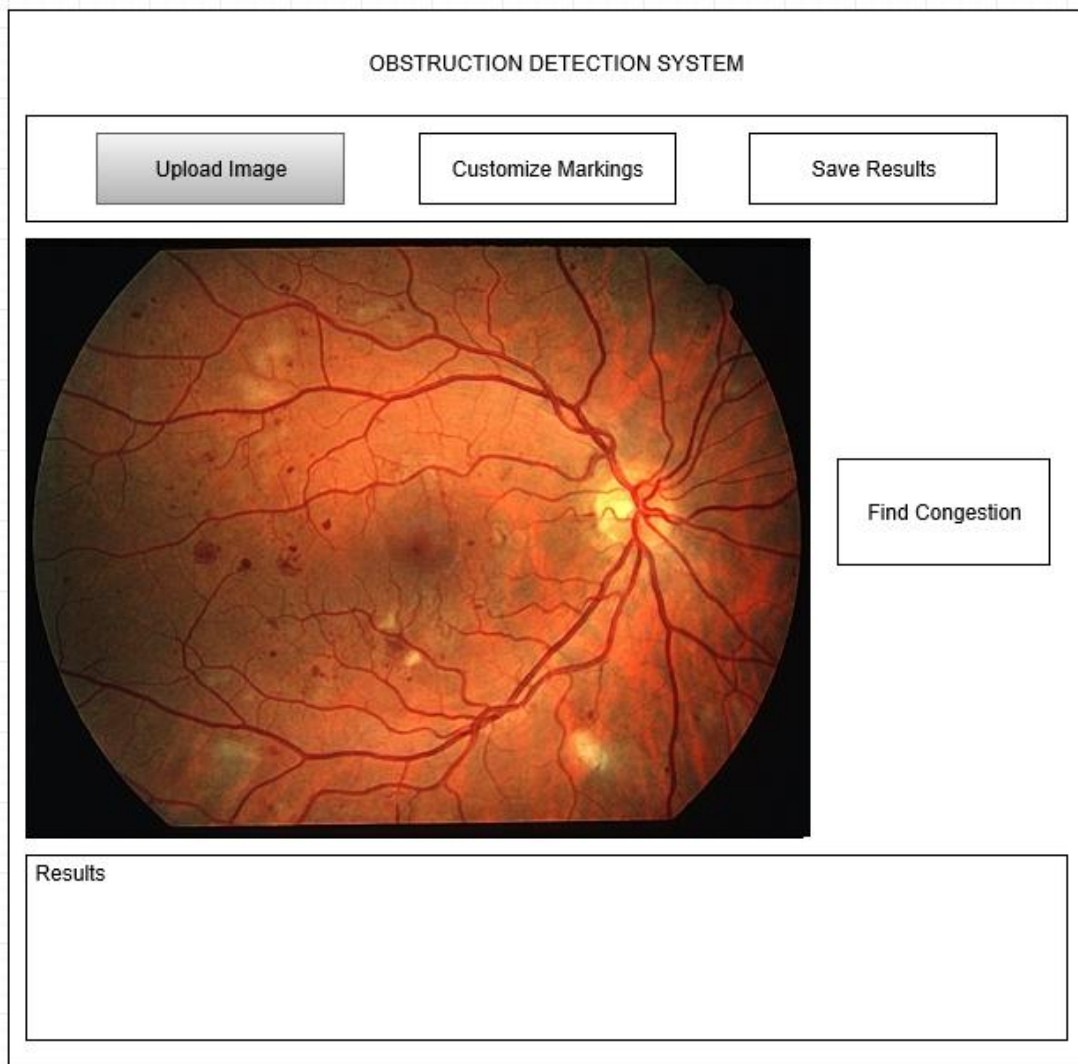
Main menu layout is shown below.



**Figure 5.5.1 Main Menu Layout of the System**

#### ***5.5.1.2 Upload Image Button***

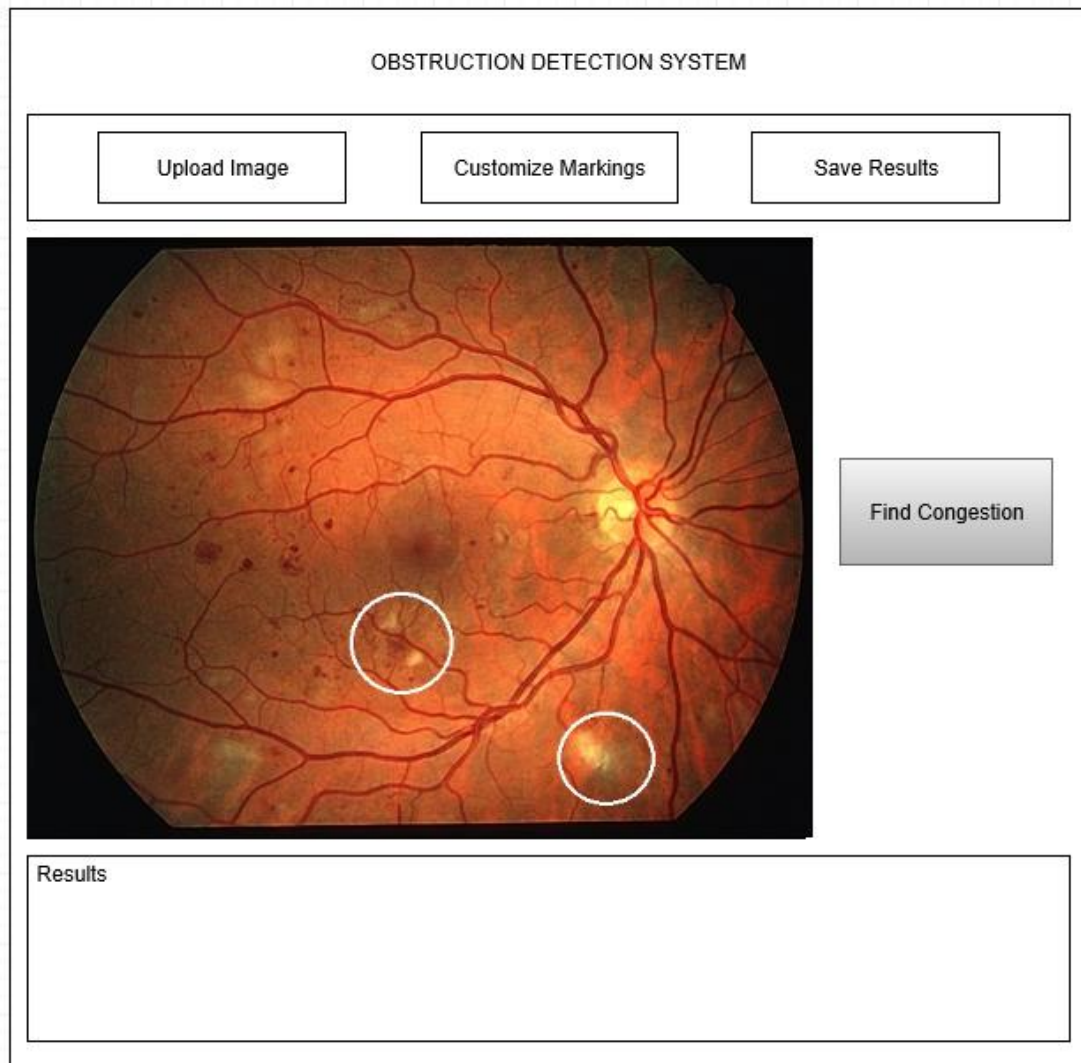
The upload image button allows the user to upload the fundus image. This operation will not be completed unless the image format is appropriate to the software.



**Figure 5.5.2 Upload Image Button**

### ***5.5.1.3 Find Congestion Button***

The find congestion button allows the user to initiate the main operation. The operation will not start unless an image has already been loaded into the software. When the process is complete, the software will display the processed image, and in this image, the congestions will be marked like shown in below image, if any.



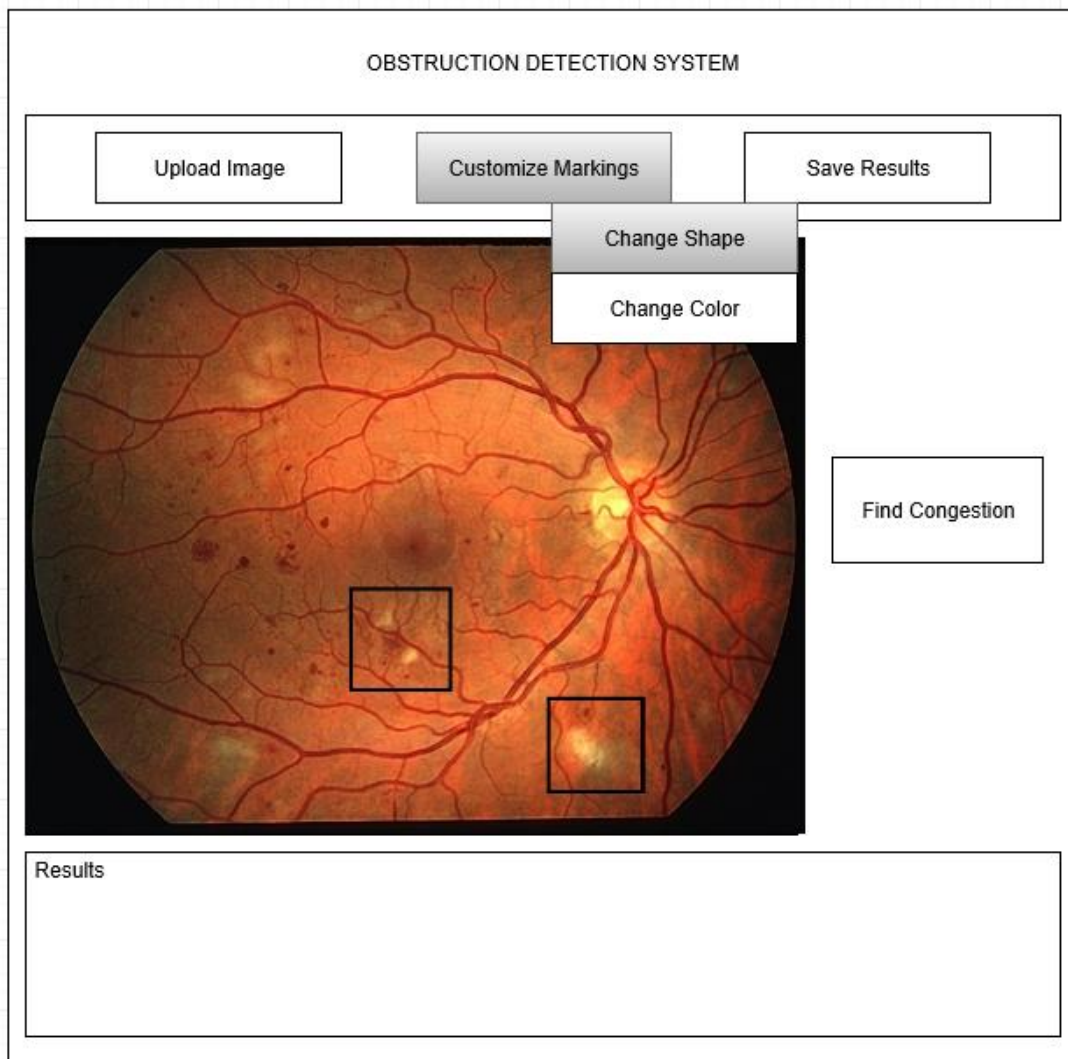
**Figure 5.5.3 Find Congestion Button**

#### ***5.5.1.4 Customize Markings Button***

The general purpose of the button is to allow user to select the geometric shape and color of the markings of the point of congestion when the obstruction is present in the vein.

##### ***5.5.1.4.1 Change Shape Button***

The purpose of the button is to ask the user what geometrical shape he/she wants software to show the obstruction. Shapes include triangle, circle and square.

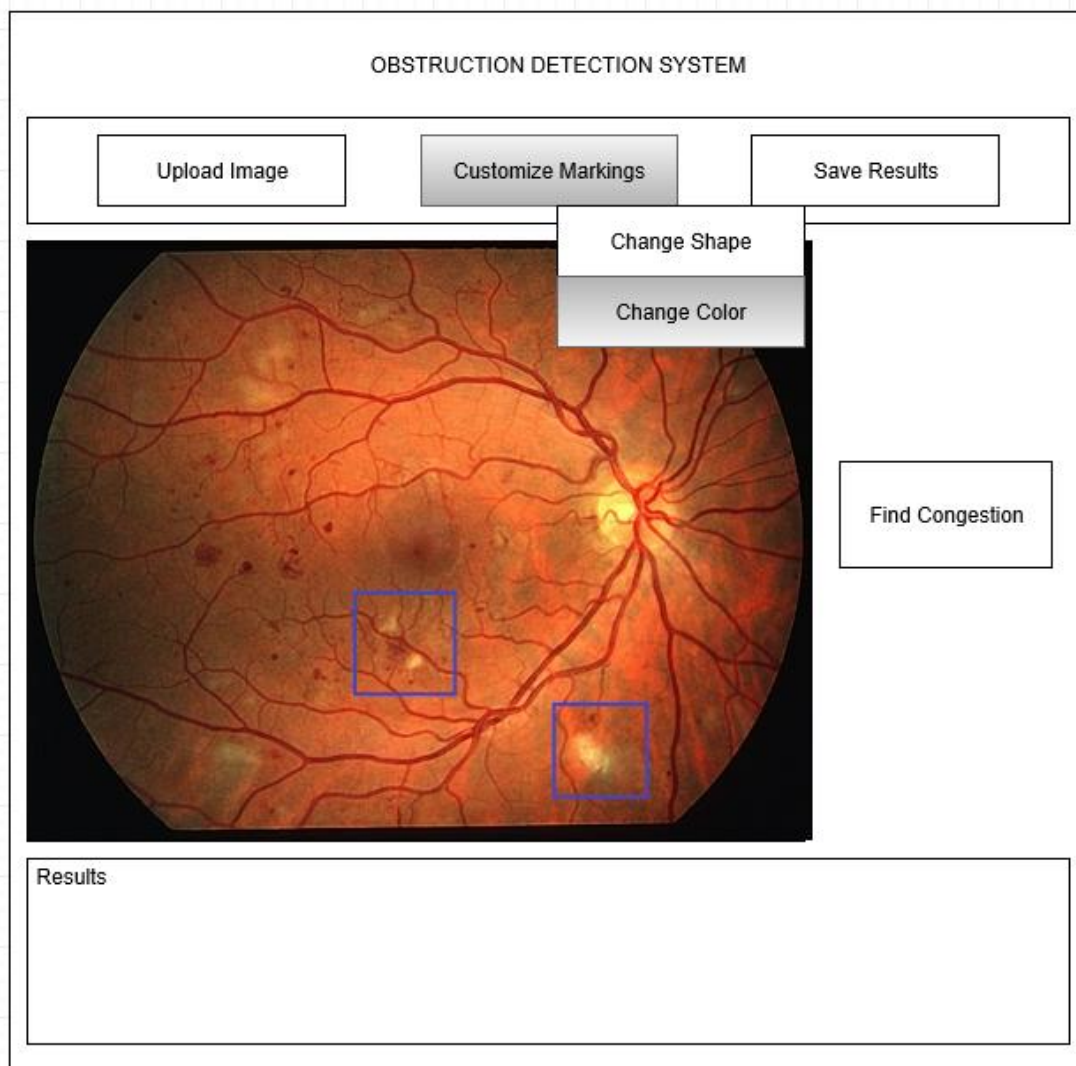


**Figure 5.5.4 Change Shape Button**

#### ***5.5.1.4.2 Change Color Button***

The purpose of the button is to ask the user what color for the shape he/she wants software to show the obstruction. Any RGB supported color will be selectable.

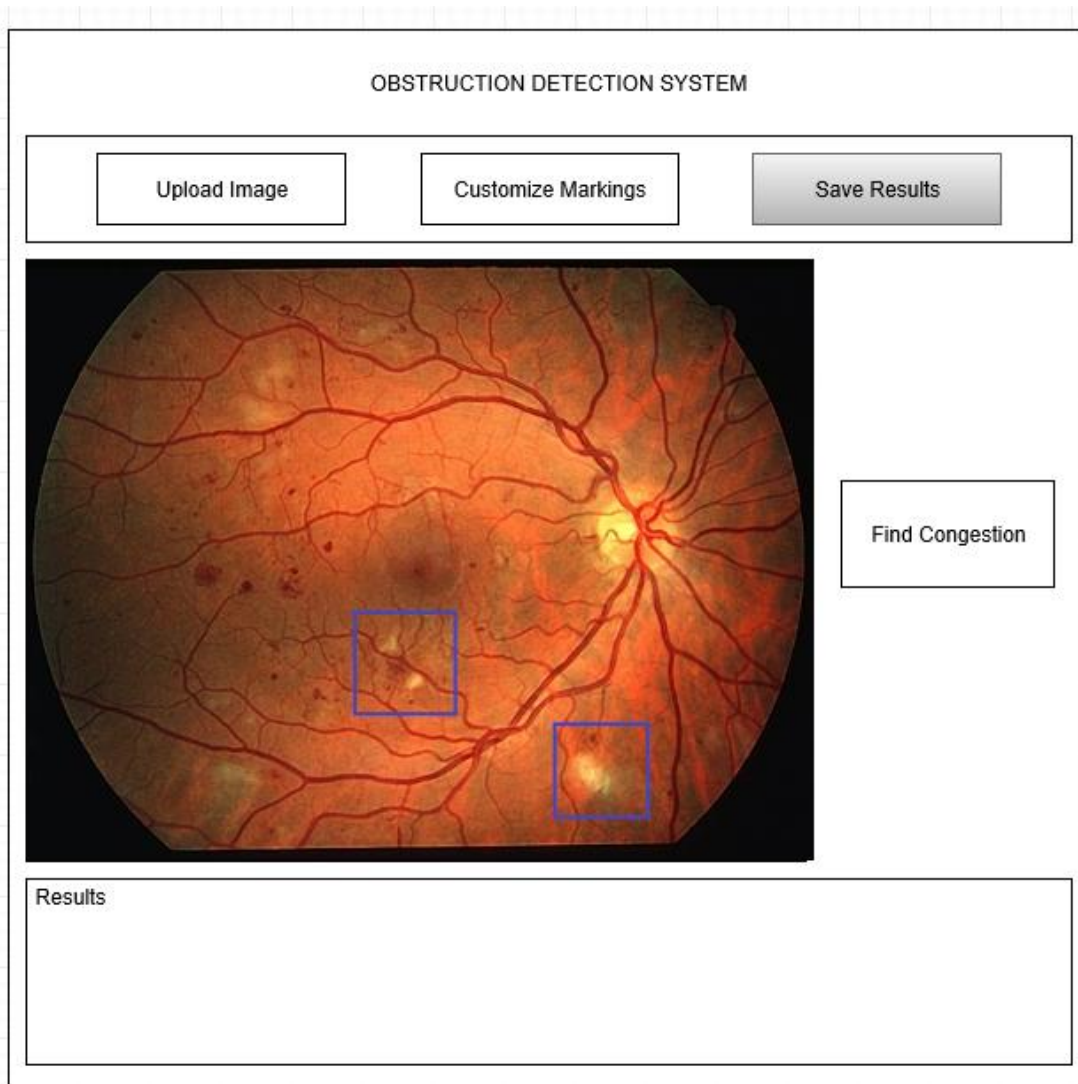




**Figure 5.5.5 Change Color Button**

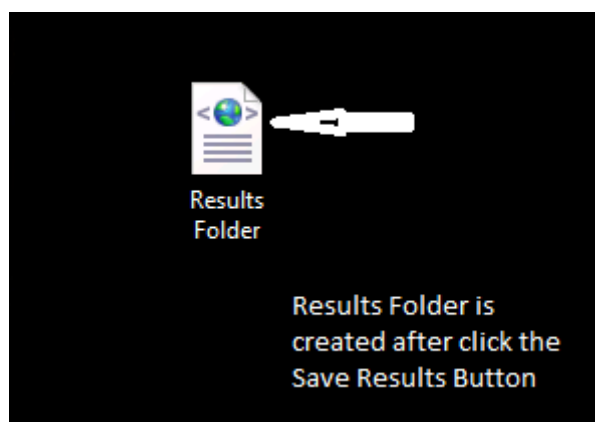
#### ***5.5.1.5 Save Results Button***

The Save Results button allows user to save the results in a desired location and with a desired name after the image is processed.



**Figure 5.5.6 Save Results Button**

When the save process is performed, the software creates a folder including a text file in which the diagnostic and results are stored, and the processed image.



**Figure 5.5.7 Results Folder**

## 6. Test Plan

### 6.1 Introduction

#### 6.1.1 Version Control

Table 6.1.1 Version Control

Version No	Description of Changes	Date
1.0	First Version	Mar 13, 2018
2.0	Second Version	May 13, 2018

#### 6.1.2 Overview

The use case of Detection of Obstructions in the Vessels in Fundus Images system users namely participant and admin which had been determined in SRS document will be tested.

#### 6.1.3 Scope

This document encapsulates the test plan of the use cases, test design specifications and the test cases correspond to test plan.

#### 6.1.4 Terminology

Table 6.1.2 Terminology

Acronym	Definition
GUI	Graphical User Interface
SP	Segmentation Process
EDP	Emboli Detection Process

## 6.2 Features to Be Tested

This section lists and gives a brief description of all the major features to be tested. For each major feature there will be a Test Design Specification added at the end of this document.

### 6.2.1 Graphical User Interface (GUI)

In project, graphical user interface components are used. The GUI has only one major part which is the Main Menu and its divided into sub parts. Those sub parts include three buttons

which consist of Find Emboli Button (Congestion), Upload Image Button, Change Markings Button and finally Save Results Button and one image display area where end user (ophthalmologist) can analysis operation results.

### **6.2.2 Segmentation Process**

The segmentation process is a helper process for the main operation, emboli detection. This process is used for extracting the vessel locations and therefore increasing the main operation's accuracy. Testing of the stated requirements will occur in this document.

### **6.2.3 Emboli Detection Process**

The emboli detection process is the main operation of our software. After the segmentation process is done, emboli detection process uses the extracted vessel skeleton to identify whether the detected emboli is inside one of the vessels or not and marks congestion spots according to the desired marking shape and color. Testing of the stated requirements will occur in this document.

## **6.3 Item Pass/Fail Criteria**

### **6.3.1 Exit Criteria**

- 100% of the test cases are executed
- 100% of the test cases passed
- All High and Medium Priority test cases passed

## **6.4 References**

- [1] GroupP201709\_SRS\_V2.0, March 13, 2018
- [2] GroupP201709\_SDD\_V2.0, March 13, 2018

## **6.5 Test Design Specifications**

### **6.5.1 Graphical User Interface (GUI)**

#### ***6.5.1.1 Upload Image Button (GUI.UPLIMG\_BTN)***

Participant can upload image with a DICOM format by selecting the “Upload Image” button.

#### ***6.5.1.2 Find Emboli (Congestion) Button (GUI.FIND\_BTN)***

Participant can detect occurrences of retinal emboli by selecting the “Find Emboli” button.

#### ***6.5.1.3 Customize Markings Button (GUI.MARK\_BTN)***

Participant can display marking customization options by selecting the “Customize Markings” button.

##### ***6.5.1.3.1 Change Shape Button (GUI.MARKSHP\_BTN)***

Participant can change shape of the markings by selecting the “Change Shape” button.

#### **6.5.1.3.2 Change Color Button (GUI.MARKCLR\_BTN)**

Participant can change color of the markings by selecting the “Change Color” button.

#### **6.5.1.4 Save Results Button (GUI.SAVE\_BTN)**

Participant can save the results by selecting the “Save Results” button.

#### **6.5.1.5 Image Display Area (GUI.IMDISP)**

System displays initial and modified image here.

#### **6.5.1.6 Result Display Area (GUI.RESDISP)**

System displays results of the image processing operations and various warning messages here.

#### **6.5.1.7 Test Cases**

Here list all the related test cases for this feature

**Table 6.5.1 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.UPLIMG_BTN.01	3.3.5	M	Select “Upload Image” button, after selecting system browser will be displayed.

**Table 6.5.2 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.FIND_BTN.01	3.1.1	M	Select “Find Congestion” button, after selecting main image processing operation will start.

**Table 6.5.3 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.MARK_BTN.01	3.2	M	Select “Change Shape” button, after selecting system displays various geometric shapes for the selection.
GUI.MARK_BTN.02	3.2	M	Select “Change Color” button, after selecting system displays various colors for the selection.

**Table 6.5.4 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.SAVE_BTN.01	3.1.1	M	Select “Save Results” button, after selecting system displays system browser for the save location selection.

**Table 6.5.5 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.IMDISP.01	3.1.1	L	Image is uploaded.

**Table 6.5.6 Test Cases**

TC ID	Requirements	Priority	Scenario Description
GUI.RESDISP.01	3.1.1	L	Image processing operations result and warning messages are displayed.

## 6.5.2 Segmentation Process (SP)

### 6.5.2.1 Performance Evaluation (True Positive Fraction) (SP.PE\_TPF)

System tests true-positive fraction (TPF) value for the segmentation process with the equation:  $TPF = TP / (TP + FN)$  where TP denotes true positive and FN denotes false negative.

### 6.5.2.2 Performance Evaluation (False Positive Fraction) (SP.PE\_FPF)

System tests false-positive fraction (FPF) value for the segmentation process with the equation:  $FPF = FP / (FP + TN)$  where FP denotes false positive and TN denotes true negative.

### 6.5.2.3 Performance Evaluation (Accuracy) (SP.PE\_ACC)

System tests the accuracy value for the segmentation operation with dividing the TPF value to total number of pixels in the image.

### 6.5.2.4 Performance Evaluation (Time) (SP.PE\_TIME)

System tests the process time taken for each of the image inside the DRIVE dataset and calculates an average computational time.

### 6.5.2.5 Test Cases

Here list all the related test cases for this feature

**Table 6.5.7 Test Cases**

TC ID	Requirements	Priority	Scenario Description
SP.PE_TPF.01	3.3.1	H	After the segmentation process TPF value must indicate %85 of the total pixels.

**Table 6.5.8 Test Cases**

TC ID	Requirements	Priority	Scenario Description
SP.PE_FPF.01	3.3.1	M	After the segmentation process FPF value must indicate % 10 of the total pixels

**Table 6.5.9 Test Cases**

TC ID	Requirements	Priority	Scenario Description
SP.PE_ACC.01	3.3.1	H	After the segmentation process accuracy value must be at least 85%.

**Table 6.5.10 Test Cases**

TC ID	Requirements	Priority	Scenario Description
SP.PE_TIME.01	3.3.1	H	After the each segmentation process average time spent must be under five minutes.

## 6.5.3 Emboli Detection Process (EDP)

### 6.5.3.1 Performance Evaluation (EDP.PE)

System compares detected emboli occurrences with the ground truth to check whether the detected congestions are actually congestion spots or not.

### 6.5.3.2 Test Cases

Here list all the related test cases for this feature

**Table 6.5.11 Test Cases**

TC ID	Requirements	Priority	Scenario Description
EDP.PE.01	3.3.1	H	All congestion spots must be detected less than 3 minutes.

## 6.6 Detailed Test Cases

### 6.6.1 GUI.UPLIMG\_BTN.01

Table 6.6.1 GUI.UPLIMG\_BTN.01

<b>TC_ID</b>	GUI.UPLIMG_BTN.01
<b>Purpose</b>	Upload a DICOM image.
<b>Requirements</b>	3.3.5
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 1 Minutes.
<b>Dependency</b>	None.
<b>Setup</b>	A DICOM image should exist in host computer.
<b>Procedure</b>	[A01] Click “Upload Image” button.
	[V01] System displays system browser panel.
	[A02] User selects an image.
	[A03] Click on the “Select” button.
	[V02] System displays selected image.
<b>Cleanup</b>	New image.

### 6.6.2 GUI.FIND\_BTN.01

Table 6.6.2 GUI.FIND\_BTN.01

<b>TC_ID</b>	GUI.FIND_BTN.01
<b>Purpose</b>	Start the main process.
<b>Requirements</b>	3.1.1
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 1 minutes.
<b>Dependency</b>	None.
<b>Setup</b>	DICOM image should be loaded to the software.
<b>Procedure</b>	[A01] Click on the “Find Congestion” button.
	[V01] Observe that the operation is successful and the results displayed.
<b>Cleanup</b>	New image.



### 6.6.3 GUI.MARK\_BTN.01

Table 6.6.3 GUI.MARK\_BTN.01

<b>TC_ID</b>	GUI.MARK_BTN.01
<b>Purpose</b>	Selecting shape of the mark.
<b>Requirements</b>	3.2
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 1 minutes.
<b>Dependency</b>	None.
<b>Setup</b>	None.
<b>Procedure</b>	[A01] Click on the “Change Shape” button.
	[V01] Geometrical shape list appears.
	[A02] Select a shape.
	[V02] Observe that the selection applied and confirmation message appears.
<b>Cleanup</b>	Close Customize Markings menu.

### 6.6.4 GUI.MARK\_BTN.02

Table 6.6.4 GUI.MARK\_BTN.02

<b>TC_ID</b>	GUI.MARK_BTN.02
<b>Purpose</b>	Selecting color of the mark.
<b>Requirements</b>	3.2
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 1 minutes.
<b>Dependency</b>	None.
<b>Setup</b>	None.
<b>Procedure</b>	[A01] Click on the “Change Color” button.
	[V01] Color list appears.
	[A02] Select a color.
	[V02] Observe that the selection applied and confirmation message appears.
<b>Cleanup</b>	Close Customize Markings menu.

## 6.6.5 GUI.SAVE\_BTN.01

Table 6.6.5 GUI.SAVE.01

<b>TC_ID</b>	GUI.SAVE.01
<b>Purpose</b>	Saving operation results.
<b>Requirements</b>	3.1.1
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 2 minutes.
<b>Dependency</b>	Find congestion and Segmentation process test cases should pass.
<b>Setup</b>	Find congestion operation should be finished.
<b>Procedure</b>	[A01] Click on the “Save Results” button.
	[V01] System browser appears.
	[A02] Select a save location.
	[V02] Observe that the save operation is successful and system browser disappears.
<b>Cleanup</b>	Close system browser.

## 6.6.6 GUI.IMDISP.01

Table 6.6.6 GUI.IMDISP.01

<b>TC_ID</b>	GUI.IMDISP.01
<b>Purpose</b>	Display the uploaded image.
<b>Requirements</b>	3.1.1
<b>Priority</b>	Low.
<b>Estimated Time Needed</b>	< 1 minutes.
<b>Dependency</b>	None.
<b>Setup</b>	An image should be uploaded.
<b>Procedure</b>	[A01] Go to upload image menu.
	[A02] Select an image.
	[A03] Click upload button.
	[V01] Observe that the image is displayed properly.
<b>Cleanup</b>	None.

## 6.6.7 GUI.RESDISP.01

Table 6.6.7 GUI.RESDISP.01

<b>TC_ID</b>	GUI.RESDISP.01
<b>Purpose</b>	Display the results.
<b>Requirements</b>	3.1.1
<b>Priority</b>	Low.
<b>Estimated Time Needed</b>	< 1 minutes.
<b>Dependency</b>	Find congestion and Segmentation process test cases should pass
<b>Setup</b>	Find Congestion operation should be started.
<b>Procedure</b>	[A01] Start Find Congestion Process.
	[V01] Observe that the results are displayed properly.
<b>Cleanup</b>	None.

## 6.6.8 SP.PE\_TPF.01

Table 6.6.8 SP.PE\_TPF.01

<b>TC_ID</b>	SP.PE_TPF.01
<b>Purpose</b>	Extracting the vessel skeleton.
<b>Requirements</b>	3.3.1
<b>Priority</b>	High.
<b>Estimated Time Needed</b>	< 1 Minutes.
<b>Dependency</b>	None.
<b>Setup</b>	An image should be uploaded and the “Find Congestion” button is pressed.
<b>Procedure</b>	[A01] Click on “Find Congestion” button.
	[V01] System starts segmentation.
	[V02] Vessel skeleton is extracted.
	[V03] Observe that the segmentation operation is successful and the performance values are adequate.
<b>Cleanup</b>	None.

## 6.6.9 SP.PE\_FPF.01

Table 6.6.9 SP.PE\_FPF.01

<b>TC_ID</b>	SP.PE_FPF.01
<b>Purpose</b>	Extracting the vessel skeleton.
<b>Requirements</b>	3.3.1
<b>Priority</b>	Medium.
<b>Estimated Time Needed</b>	< 1 Minutes.
<b>Dependency</b>	None.
<b>Setup</b>	An image should be uploaded and the “Find Congestion” button is pressed.
<b>Procedure</b>	[A01] Click on “Find Congestion” button.
	[V01] System starts segmentation.
	[V02] Vessel skeleton is extracted.
	[V03] Observe that the segmentation operation is successful and the performance values are adequate.
<b>Cleanup</b>	None.

## 6.6.10 SP.PE\_ACC.01

Table 6.6.10 SP.PE\_ACC.01

<b>TC_ID</b>	SP.PE_ACC.01
<b>Purpose</b>	Extracting the vessel skeleton.
<b>Requirements</b>	3.3.1
<b>Priority</b>	High.
<b>Estimated Time Needed</b>	< 1 Minutes.
<b>Dependency</b>	None.
<b>Setup</b>	An image should be uploaded and the “Find Congestion” button is pressed.
<b>Procedure</b>	[A01] Click on “Find Congestion” button.
	[V01] System starts segmentation.
	[V02] Vessel skeleton is extracted.
	[V03] Observe that the segmentation operation is successful and the performance values are adequate.
<b>Cleanup</b>	None.

### 6.6.11 SP.PE\_TIME.01

Table 6.6.11 SP.PE\_TIME.01

<b>TC_ID</b>	SP.PE_TIME.01
<b>Purpose</b>	Extracting the vessel skeleton.
<b>Requirements</b>	3.3.1
<b>Priority</b>	High.
<b>Estimated Time Needed</b>	< 5 Minutes.
<b>Dependency</b>	None.
<b>Setup</b>	An image should be uploaded and the “Find Congestion” button is pressed.
<b>Procedure</b>	[A01] Click on “Find Congestion” button.
	[V01] System starts segmentation.
	[V02] Vessel skeleton is extracted.
	[V03] Observe that the segmentation operation is successful and the performance values are adequate.
<b>Cleanup</b>	None.

### 6.6.12 EDP.PE.01

Table 6.6.12 EDP.PE.01

<b>TC_ID</b>	EDP.PE.01
<b>Purpose</b>	Finding retinal emboli occurrences.
<b>Requirements</b>	3.3.1
<b>Priority</b>	High.
<b>Estimated Time Needed</b>	3 Minutes.
<b>Dependency</b>	Segmentation process test case should pass
<b>Setup</b>	Segmentation process should be finished.
<b>Procedure</b>	[V01] Segmentation process finishes.
	[V02] System starts congestion detection process.
	[V03] System evaluates performance results.
	[V04] Observe that the performance values are adequate.
<b>Cleanup</b>	None.

## 7. Test Results

### 7.1 Individual Test Results

Table 7.1.1 Individual Test Results

TC ID	Priority	Date Run	Run By	Result	Explanation
GUI.UPLIMG_BTN.01	M	11.05.2018	Aykut Er	Pass	Images are uploaded.
GUI.FIND_BTN.01	M	11.05.2018	Aykut Er	Pass	Emboli detection process is started.
GUI.MARK_BTN.01	M	11.05.2018	Aykut Er	Fail	Mark operations will be added in the next release.
GUI.MARK_BTN.02	M	11.05.2018	Aykut Er	Fail	Mark operations will be added in the next release.
GUI.SAVE_BTN.01	M	11.05.2018	Aykut Er	Pass	Result folder is saved.
GUI.IMDISP.01	L	11.05.2018	Aykut Er	Pass	Images are displayed.
GUI.RESDISP.01	L	11.05.2018	Aykut Er	Pass	Results are displayed.

Table 7.1.2 Individual Test Results

TC ID	Priority	Date Run	Run By	Result	Explanation
SP.PE_TPF.01	H	11.05.2018	Egeberk Özberk	Pass	True Positive ratio is as expected.
SP.PE_FPF.01	M	11.05.2018	Egeberk Özberk	Fail	False Positive ratio is higher than expected.
SP.PE_ACC.01	H	11.05.2018	Egeberk Özberk	Pass	Expected accuracy value is achieved.
SP.PE_TIME.01	H	11.05.2018	Egeberk Özberk	Pass	Process is done at expected amount of time.

Table 7.1.3 Individual Test Results

TC ID	Priority	Date Run	Run By	Result	Explanation
EDP.PE.01	H	11.05.2018	Egeberk Özberk	Pass	Emboli occurrences are detected.

## 7.2 Summary of Test Results

Table 7.2.1 Test Results

Priority	Number of TCs	Executed	Passed
H	4	4	4
M	6	6	3
L	2	2	2
<b>Total</b>	<b>12</b>	<b>12</b>	<b>9</b>

We have executed 12 test cases and 9 test cases are passed. Also, all of high and half of the medium priority test cases are passed. Exit criteria is met.

## 7.3 Exit Criteria

We have executed all test cases and 75% of test cases are passed. Also, 100% of high and 50% medium priority test cases are passed. Software development activities are completed within the anticipated cost. Software development activities are completed within the anticipated timeline. Exit criteria is met.

Table 7.3.1 Exit Criteria

Criteria	Met or Not
100% of test cases are executed	M
75% of the test cases are passed	M
100% of High and 50% Medium Priority test cases passed.	M
No high priority or severe bugs are left outstanding.	M
Verify if software development activities are completed within projected cost.	M
Verify if software development activities are completed within projected timelines.	M

## 7.4 Known Problems

Mark shape button and mark change button parts of the GUI are not prepared in time. Due to this fact, test cases are failed. The problem will be solved when these buttons are prepared also, segmentation process still takes large amount of time compared to other supervised methods due to our low computational power resulting inadequate neural network. This problem will be also fixed on next release.

Amount of false positives are higher than expected again due to poor network, which will be fixed within the segmentation time fix with next releases.

## **7.5 Conclusion**

This section includes the test results of the project “Detection of Obstruction in Vessels in Fundus Images”. We have created our test plan and exit criteria to the project. We implemented the test cases and have compiled the test results. According to the results, all test cases and 75% of test cases passed. Also, 100% of high and 50% medium priority test cases passed. Severe bugs are left outstanding. Software development activities are completed within the projected cost. Software development activities are completed within the projected timelines.

Our project were met all exit criterias so our product is ready for use. We will try to solve the failed test case to make it more quality.

## **8. Conclusions**

This document includes wide information about our project that titled as "Detection of Obstructions in Vessels in Fundus Images". In this project, we aimed to help doctors in the treatment of patients by automatically locating obstructions in vessels for doctors.

To develop project, we have made a lot of research about image processing, Matlab and similar projects. We have analyzed the similar projects, and have tried to understand the processes we have to learn and do in order to be able to develop this project. After the research part, we have received requirements from advisor and the doctor who helped us in this project, and after these requirements, SRS documentation is prepared. After the requirements are specified, the design of the developing project is prepared and this design is explained in the SDD document. During this period, we designed the architecture of the project.

The advantage of the project is that it automatically finds the locations of all obstructions and shows them to the doctor so that they can find all the obstructions even when they are tired or busy. However, according to the quality of the incoming image, marking can also be done in places without obstruction in the images. But we also find a solution to this problem by receiving feedback from the doctor on whether or not is obstruction in the software system as a future work.

## **9. User Manual**

### **9.1. General Briefing**

BVS software aims to speed up the diagnosis process by quickly segmenting the vessels and detecting any emboli occurrences with more accuracy and faster than the actual doctor. Intended end user for this document and software is expected to be an ophthalmologist.

Software does not require any internet connection nor have any hardware dependencies.



## **9.2. Software Information**

### **9.2.1 Overall Briefing**

The software is composed of 4 interactive buttons for the end user handling certain operations and 6 distinct display areas.

Four of which are the image display zones;

- Original fundus image,
- Ground truth,
- Segmented blood vessels,
- Detected emboli occurrences.

Remaining display areas are reserved for the results indicating software's performance or current state at any moment with errors encountered if ever exists and list of the images uploaded by the user.

Buttons inside software's GUI has different purposes listed below;

- Uploading an image,
- Segmentation process,
- Emboli Detection,
- Save Results.

Functionalities of the components will be explained later in this document.

### **9.2.2 Getting Started**

#### ***9.2.2.1 Uploading an image***

Button named "Upload Image" that resides upper left handles the upload operation.

Pressing the button will trigger the file explorer enabling you to select a folder where all the retinal fundus images are stored.

User must select between "Emboli" or "Segmentation" folder for different operations. Both folders can be found in the project folder.

Emboli folder has only one photo due to lack of data. Segmentation folder however, has twenty photos with their ground truths.

### **9.2.3 Operational Functionality**

#### ***9.2.3.1 Segmentation Process***

Button named "Segmentation" residing at top section of the GUI handles the vessel segmentation.

Outcome will be a binary image displaying at "Segmented Image" display zone (see addendum No: 1) and result bar indicating the accuracy of the operation or the error message regarding to addendum No: 1.

Process of segmentation *is necessary* for the emboli detection operation in order to achieve higher accuracy and precision for the detection operation.

**\*Addendum:**

1. If there are no images loaded or selected, pressing segmentation button will generate an error message in result display area indicating the process requires an image to operate.

### ***9.2.3.2 Emboli Detection***

Button named "Emboli Detection" residing at top section of the GUI handles the emboli detection process, and will not appear if emboli folder isn't being selected.

Pressing the button will display the processed version of the original image in emboli detection display area (see addendum No: 1). Process button will not appear until segmentation operation is performed.

Finally result display area will display either a success message or an error message depending on the state of the software, along with the number of emboli occurrences.

**\*Addendum:**

1. If there are no images loaded or selected, pressing segmentation button will generate an error message in result display area indicating the process requires an image to operate.

### **9.2.4 Save Operation**

Button named "Save Results" residing at the rightmost top section of the GUI saves the results and will not appear if no operation had been performed.

Pressing the button will trigger the file explorer enabling user to chose where he/she wants to save (see addendum No: 1).

For saving user must choose a folder for each of the result independently.

Saving operation is handled successfully if the results display area pops a message indicating that the operation has been handles successfully.

**\*Addendum:**

1. If there has never been an operation, file explorer will not be triggered and an error message will be displayed in results display area.
  - 1.1. If there has never been a emboli detection operation, detection results **will not be included** in results folder for saving.

### **9.3. System Overview**

- The software is based on the Windows 7, 8, 10 or variants.
- Performance is hardware dependent.
  - Higher CPU speed or higher RAM capacity the better.
- Operational Status :
  - Partially operational.
  - Under development.

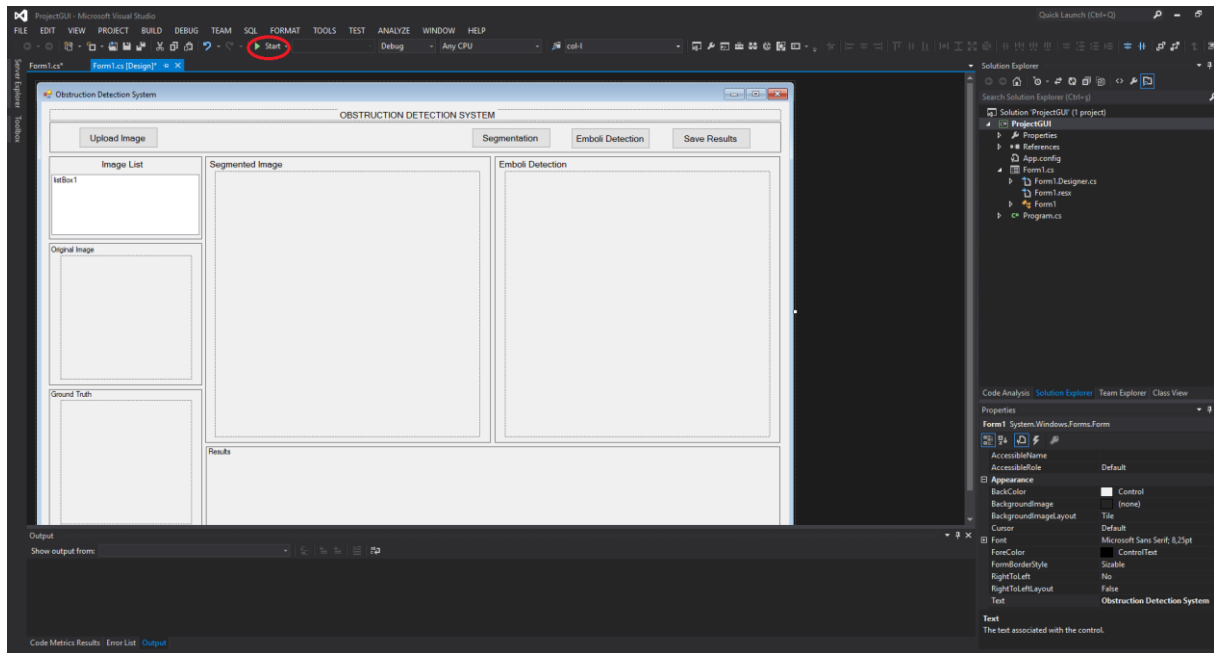
## **10. Compilation / Installation Guide**

### **10.1 Prerequisites / Tools**

- Visual Studio 2012 or higher should be installed on computer to compile and run the project.
- Final version of the project is available at the link <https://github.com/CankayaUniversity/ceng-407-408-project-blood-vessel-segmentation/releases/tag/v1.0.0>. It can be downloaded as a .zip file. Extract the files from .zip folder.

### **10.2 Opening the Project in Visual Studio**

- Open Visual Studio.
- When Visual Studio is opened, click Open Project and then choose the ProjectGUI folder from the downloaded project folder.
- After the project is opened in Visual Studio, as seen in Figure 1, click Start button



**Figure 10.2.1 Visual Studio page to run the GUI**

- After the Start button is clicked, then the program is built and run.

## 10.3 Installation

- From the guide above, after the program is started, the GUI of the project can be used.

## Acknowledgement

We are grateful for guidance we have received from Assist. Prof. Dr. Roya CHOUPANI and Assoc. Prof. Dr. Reza ZARE HASSANPOUR. The help we received from them was a great asset to improve this project and ourselves.

## References

1. Bhuiyan, A., Nath, B., Chua, J., & Kotagiri, R. (2007). Blood Vessel Segmentation from Color Retinal Images using Unsupervised Texture Classification. *2007 IEEE International Conference on Image Processing*. doi:10.1109/icip.2007.4379880.
2. Tingbiao Chen, Liangzheng Xia, "Digital image processing", Beijing: Posts & Telecommunications Press, 1994.
3. J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57
4. J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York
5. Research Section, Digital Retinal Image for Vessel Extraction (DRIVE) Database. Utrecht, the Netherlands, Univ. Med. Center Utrecht, Image Sci. Inst. [Online]. Available: <http://www.isi.uu.nl/Research/Databases/DRIVE>
6. A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19(3), pp. 203–210, 2000.
7. Ahmed. H. Asad, A. T. Azar, M. M. M. Fouad and A. E. Hassanien. (2013). An Improved Ant Colony System for Retinal Blood Vessel Segmentation. *Computer Science and Information Systems: Proceedings of the Fedcsis*. (pp. 199-205). Krakow: IEEE.
8. M.K. Hu, "Visual pattern Recognition by Moment Invariants," *IRE. Trans. Inform. Theory.* , vol. 8, no. 2, 1962.
9. M.A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," *ICML*, 2000.
10. M. Dorigo and L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.* vol. 1, no. 1, pp. 53–66, 1997.
11. Ahmed. H. Asad, A. T. Azar, and A. E. Hassanien, "Integrated features based on gray-level and Hu moment invariants with ant colony system for retinal blood vessels segmentation," *Int. J. Syst. Boil. Biomed. Tech.*, vol. 1, no. 4, pp. 61-74, 2012.
12. STARE ProjectWebsite. Clemson, SC, Clemson Univ. [Online]. Available: <http://www.ces.clemson.edu/>
13. Raja, D. S., Vasuki, S., & Kumar, D. R. (2014). Performance Analysis of Retinal Image Blood Vessel Segmentation. *Advanced Computing: An International Journal*, 5(2/3), 17-23. doi:10.5121/acij.2014.5302
14. Joao V. B. Soares, Jorge J. G. Leandro, Roberto M. Cesar Jr., Herbert F. Jelinek, and Michael J. Cree, "Retinal Vessel Segmentation Using the 2- D Gabor Wavelet and Supervised Classification", *IEEE Trans. Med. Imag.*, 2010.
15. Dalwinder Singh, Dharamveer, Birmohan Singh, "New Morphology based Approach for Blood Vessel Segmentation in Retinal Images,"
16. Mehta K, and Kaur N.: "An Enhanced Segmentation Technique for Blood Vessel in Retinal Images.". *International Journal of Computer Applications*, vol. 150, no. 6, 2016, pp. 4–5., doi: 10.5120/ijca2016911548.
17. Rosenfeld, A., and Pfaltz, J. L. (1996) „Sequential operations in digital picture processing“, *Journal of the ACM*, Vol. 13 No. 4, pp. 471–494.
18. P. Getreuer, M. Tong, and L. A. Vese. A variational model for the restoration of MR images corrupted by blur and rician noise. In *Proc. ISVC*, pages 686–698, 2011.

19. Phansalkar, Neerad, Sumit More, Ashish Sabale, and Madhuri Joshi. "Adaptive local thresholding for detection of nuclei in diversity stained cytology images." In Communications and Signal Processing (ICCSP), 2011 International Conference on, pp. 218-220. IEEE, 2011.
20. Fan Z., Lu Z., Li W., Wei C., Huang H., Cai X., Chen X.: A Hierarchical Image Matting Model for Blood Vessel Segmentation in Fundus images, 2017.
21. M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, "Blood vessel segmentation methodologies in retinal images—a survey," *Computer Methods and Programs In Biomedicine*, vol. 108, no. 1, pp. 407–433, 2012.
22. M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, "An ensemble classification-based approach applied to retinal blood vessel segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2538–2548, 2012.
23. P. Bankhead, C. N. Scholfield, J. G. McGeown, and T. M. Curtis, "Fast retinal vessel detection and measurement using wavelets and edge location refinement," *PloS One*, vol. 7, no. 3, p. e32435, 2012.
24. D. Marín, A. Aquino, M. E. Gegundez-Arias, and J. M. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Transactions on Medical Imaging*, vol. 30, no. 1, pp. 146–158, 2011.
25. L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.
26. Yang-Williams, K. (2002). *Ophthalmic Photography: Retinal Photography, Angiography, and Electronic Imaging*, 2nd ed. *Optometry and Vision Science*, 79(8), 478. Doi: 10.1097/00006324-200208000-00008.
27. Roychowdhury, S., Koozekanani, D.D., Parhi, K.K.: Blood vessel segmentation of fundus images by major vessel extraction and sub-image classification. *IEEE J. Biomed. Health Inform.* 99 (2014). doi:10.1109/JBHI.2014.2335617.
28. B. Sujithkumar, and V. Singh, (2012) "Automatic Detection of Diabetic Retinopathy in Non-dilated RGB Retinal Fundus Images", *International Journal of Computer Applications*, Vol. 47, No. 19, pp. 26-32.
29. F. Zana, J. Klein, A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform, *IEEE Trans. Med. Imaging* 18 (5) (1999) 419–429.
30. T. A. Ciulla, A. G. Amador, and B. Zinman, (2003) "Diabetic Retinopathy and Diabetic Macular Edema: Pathophysiology, screening, and novel therapies", *Diabetes Care*, Vol. 26, No. 9, pp. 2653-2664.
31. S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, M. Goldbaum, Detection of blood vessels in retinal images using two-dimensional matched filters, *IEEE Trans. Med. Imaging* 8 (3) (1989) 263–269.
32. R. Nekovei, Y. Sun, Back-propagation network and its configuration for blood vessel detection in angiograms, *IEEE Trans. Neural Networks* 6 (1) (1995) 64–72.
33. Júnior, S. B., & Welfer, D. (2013). Automatic Detection of Microaneurysms and Hemorrhages in Color Eye Fundus Images. *International Journal of Computer Science and Information Technology*, 5(5), 21-37. doi:10.5121/ijcsit.2013.5502.
34. Y. Hatanaka, T. Nakagawa, Y. Hayashi, T. Hara, and H. Fujita, (2008) "Improvement of Automated Detection Method of Hemorrhages in Fundus Images", *Proceedings: 30th Annual International IEEE EMBS Conference*, pp. 5429 - 5432.
35. T. Kauppi, V. Kalesnykiene, J. K. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kalviainen, and J. Pietila, (2007), "DIARETDB1: diabetic retinopathy

database and evaluation protocol,” Technical report, Faculty of Medicine, University of Kuopio, Finland.

36. IEEE. “*IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*”. IEEE Computer Society. October 20, 1998.