UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA DELL' INFORMAZIONE
CORSO DI LAUREA SPECIALISTICA IN BIOINGEGNERIA

TESI DI LAUREA

ALGORITHMS FOR IMAGE ANALYSIS OF NEURAL TISSUE FOR THE EVALUATION OF HIV THERAPIES

RELATORE: Ch. mo Prof. Alfredo Ruggeri (Università di Padova)

CORRELATORE: Prof. Bjarne Kjær Ersbøll (Danmarks Tekniske Universitet)

LAUREANDO: *Fabio Lissa*

ANNO ACCADEMICO 2009 - 2010

“Whatever you can do,
or dream you can,
begin it.
Boldness has genius,
power
and magic
in it.”

J. W. von Goethe

Prefazione e Ringraziamenti

Questa tesi rappresenta il progetto finale del mio percorso di studi nella Laurea Specialistica in Bioingegneria dell'Università degli Studi di Padova.
Il progetto è stato svolto da Gennaio a Giungo 2010 presso il Department of Informatics and Mathematical Modelling della Technical University of Denmark (DTU) sotto la supervisione del professor Bjarne K. Ersbøll e del professor Alfredo Ruggeri (Università degli Studi di Padova) e in collaborazione con Visiopharm, un'azienda danese che opera nel campo dell'analisi di immagini da microscopio.

In questa breve sezione desidero ringraziare tutti colori che, in vari modi e tempi, hanno reso possibile lo svolgimento di questo progetto.
I professori Alfredo Ruggeri e Bjarne K. Ersbøll, e l'ing. Micheal Grunkin per avermi dato la possibilità di lavorare in Visiopharm e per aver creduto nelle mie capacità.
Il mio company supervisor Michael Friis Lippert, che mi ha guidato passo dopo passo attraverso questo progetto e la cui professionalità e disponibilità nei miei confronti sono state per me modello da seguire, e da cui prender spunto in ogni momento.
I dipendenti e le persone con cui ho collaborato in Visiopharm, per la gentilezza con cui sono stato accolto e per avermi fatto sentire parte integrante del gruppo di lavoro.
Gli amici che mi hanno aiutato negli ultimi giorni di progetto e tutte le persone da ogni parte del mondo che ho conosciuto in Danimarca e che spero di incontrare ancora, un giorno, sulla mia strada.
Gli amici di sempre, quelli della compagnia, quelli dell'appartamento e dell' università e tutte le persone che hanno fatto parte dei momenti più importanti e belli della mia vita, dal gruppo parrocchiale alle attività sportive a quelle lavorative: grazie, nel mio girovagare per il mondo siete sempre nei miei pensieri.
Valeria, per la sua dolcezza, per il suo sorriso.
La mia famiglia, i parenti e tutte le persone che mi vogliono bene e mi sono state vicino e che mi fanno sentire il loro affetto.
Grazie a tutti, vi voglio bene.

Padova, 8 Luglio 2010 Fabio Lissa

Sommario

L'HIV è una malattia causata dal virus dell'immunodeficienza umana che distrugge gradualmente il sistema immunitario del corpo umano, per il quale diventa più difficile combattere le infezioni. Il cervello è uno degli organi più colpiti dall' HIV di tipo 1, e i neuroni sono uno dei pochi tipi di cellule del corpo umano che non sopportano la replicazione del virus. Gli effetti di tale progressiva degenerazione del tessuto neurale può essere simulata su culture di neuroni (in vivo e in vitro), somministrando composti a base di proteine virali neuro-tossiche, come il 3NP, che causano danni da stress ossidativo e produzione di radicali liberi portando i neuroni a morte per apoptosi. É possibile quantificare la capacità di composti terapeutici a base di Paroxetina di ridurre gli effetti di degenerazione e morte del tessuto neurale per mezzo della valutazione dello stato di salute di culture di neuroni, trattate in precedenza con composti a base di tossine. Le immagini delle culture di neuroni sono poi ottenute per mezzo di un microscopio a fluorescenza. L'obbiettivo di questo progetto é di analizzare questo tipo di immagini, con lo scopo di quantificare importanti caratteristiche nel tessuto neurale. Le immagini analizzate sono divise in tre gruppi principali, che rappresentano rispettivamente culture di neuroni in normali condizioni di salute (*control group*), culture trattate con tossina (*3NP*) e culture trattate con tossine ed agente terapeutico (*3NP plus Paroxetine*). Confrontando le misurazioni delle caratteristiche più importanti di tali differenti gruppi di immagini é stato possibile dimostrare se le culture trattate con Paroxetina mostravano migliori condizione di salute rispetto a quelle trattate solamente con tossine, in cui gli effetti di tali composti avevano diminuito la reazione di fluorescenza corrispondente all'attività neurale. Tale quantificazione è data dalla misurazione di tre principali caratteristiche di ogni immagine: l'area totale dei corpi centrali dei neuroni (*neuron cell bodies*), la lunghezza totale dei neuriti (*neurite branches*) e il numero totale di macchioline (*small beads*) risultanti da processi di degenerazione neuritica. Dal momento che l'obbiettivo dell'analisi - dal punto di vista medico - è la valutazione dell'effetto protettivo asso-dendritico di composti a base di Paroxetina, è stato anche calcolato e valutato un parametro che quantizzi l'effetto di crescita neuritica per ogni neurone. É stato implementato un programma di analisi di immagine utilizzando il SDK di VIS, il software per analisi di immagini da microscopio della Visiopharm, l'azienda presso la quale questo progetto di tesi è stato svolto. Tale programma esegue le misurazioni dei tre principali componenti dell'immagine, implementando un protocollo di analisi composto di metodi di processing comunemente usati per immagini di campioni biologici. È stata quindi compiuta una validazione statistica dei risultati ottenuti. Da una valutazione di tali risultati, è stato possibile determinare che le culture trattate con Paroxetina erano quantitativamente differente da quelle trattate con sole tossine, ed erano più simili a quelle appartenenti al *control group*. Sulla base dei risultati di questo programma è possibile confermare che i composti a base di Paroxetina usati hanno un effetto protettivo asso-dendritico sulle culture di neuroni.

Parole chiave: HIV, neuroni, degenerazione dendritica, crescita dendritica, Paroxetina, microscopia a fluorescenza, analisi di immagine, misurazione di caratteristiche, effetto protettivo asso-dendritico

Preface

This thesis has been realized as a requirement of the Master Science Degree in Bioengineering of the University of Padova, Italy.
This 30 ECTS project work has been carried out from January 2010 to June 2010 at the Department of Informatics and Mathematical Modelling (IMM) at the Technical University of Denmark (DTU) under the supervision of professor Bjarne K. Ersbøll (IMM-DTU) and professor Alfredo Ruggeri (University of Padova) and in collaboration with Visiopharm, the Danish microscopy imaging company where I have functioned as a part of the project group.

In the following I would like to thank all the people that could make this project possible.
The project supervisors, prof. Bjarne K. Ersbøll and prof. Alfredo Ruggeri.
Micheal Grunkin for trusting in my abilities and for the useful suggestions received during the project.
Michael Friis Lippert for his constant aid and careful supervision. His continuous availability for listening all the issues suddenly appearing, his calm and lucidity in the several discussions we had daily, that were source of great inspiration to me. His ability of quickly finding solutions to every matter impressed me. His precious suggestions and his guide drove me literally through this project.
The employees at Visiopharm for their support and their interest, especially Kim Anders Bjerrum and Thomas Ebstrup. And my "colleagues", Helene Hvidegaard Højrup and Mikael Ott-Ebbesen. In this context I really want to thank the whole company for the kindness and the warm treatment they always gave me and for making me feel as a part of the team.
The people at the IMM, especially dr. Line Katrine Harder Clemmensen and dr. Anders Dahl. Martin Petrella, José Quaresma, Matteo Zatti: thank you for helping me during the last days of the thesis project.
The people from all over the world I met here in Denmark and which I have been living together with: you have been the greatest experience of my life, so far.
Thank you, again.

Lyngby, June 18th, 2010 Fabio Lissa

Abstract

HIV infection is a disease caused by the human immunodeficiency virus (HIV) that gradually destroys the immune system, which makes it harder for the body to fight infections. The brain is a major target organ for HIV-type 1 infection and neurons are one of the few cell types in the human body that do not support HIV type-1 replication.
The effects of progressive neural tissue degeneration can be emulated by administering compounds based on neuro-toxic viral proteins to neuron cultures (in vivo and in vitro), like the 3NP, that causes oxidative stress damage and free radical production, which trigger an apoptotic cell death in the neurons . The capability and the power of Paroxetine, a therapeutic agent, in reducing the effects of degeneration and death of neuronal tissue, can be quantified by screening the health status of the neuron cultures, previously treated with toxins, after the administration of such therapeutic agent. The neuron cultures can then be imaged under a fluorescence microscope.
The goal of this project is to evaluate this kind of images, by quantifying the most important features of the neuron tissue. The images that are to be analyzed are divided into three main groups, representing respectively: normal healthy neuron tissue (the control group), cultures treated with toxins (3NP) and cultured treated with toxins and a therapeutic agent (3NP plus Paroxetine). Comparing the measured features from each of these images, it was possible to determine if the cultures of the 3NP plus paroxetine groups showed actual improvement over the 3NP groups, in which the effect of the toxin had decreased the fluorescent response corresponding to the neuronal activity. This quantification was given by the measurement of three main features: The total area of neuron cell bodies, the total length of neurites (both axons and dendrites) branches and the number of small beads resulting from the degeneration process of a neurite. Since the objective of the analysis, from a medical standpoint was the evaluation of the axo-dendritic protective effect of the Paroxetine-based compound, a parameter for evaluating the effect of neuritic growth per neuron was calculated and evaluated.
An images analysis program was implemented using the SDK of VIS, the image analysis software of Visiopharm, the company this project was carried out at. This program performs feature measurements on the images, by implementing an image analysis protocol. Finally, a quantitative and statistical analysis of the results has been performed. From an evaluation of the results, it was possible to determine that the neuron cultures submitted to an administration of both toxin and therapeutic agent were quantitatively different from the neuron cultures that were treated with toxin only, and were more similar than the latter ones to the cultures of the control group. Based on this image analysis program, it is possible to confirm that the Paroxetine based compounds used has actually had an axo-dendritic protective effect on the samples of neuronal tissue.

Keywords: HIV, neurons, neuritic degeneration, neuritic growth, Paroxetine, fluorescence microsopy, Image Analysis, features measurement, axo-dendritic protective effect.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction to the Project

This chapter gives a general overview of the project, along with a description of the medical background of the project and the technological instruments for image analysis that have been used in order to achieve the final goal. This section is part and parcel of the entire project as the remarks made herein provide the reader with the foundation of the project itself and give the guidelines about how the project goal comes to be achieved and how to read the thesis, as well.

1.1 General Introduction of the project

HIV infection is a disease caused by the human immunodeficiency virus (HIV) that gradually destroys the immune system, which makes it harder for the body to fight infections [1]. AIDS (acquired immune deficiency syndrome) is the final and most serious stage of HIV disease, which causes severe damage to the immune system [1]. The brain is a major target organ for HIV-type 1 (HIV-1) infection. The virus enters the central nervous system and may cause a wide range of neurological disorders including cognitive motor impairment and HIV-associated dementia. Neurons are one of the few cell types in the human body that do not support HIV type-1 (HIV-1) replication [2]. Neurological injury caused by HIV-1 is mediated mainly by a direct way, through administration of neuro-toxic viral proteins[3]. These cause dendritical and axonal pruning, finally resulting in death of neurons.

It is possible to reproduce the effects of progressive neural tissue degeneration by administering neuron cultures (in vivo and in vitro) compounds based on those neuro-toxic viral proteins, like the toxin 3NP, which is 3-nitropropionic acid. This compound is toxic because it is a mitochondrial electron chain transport inhibitor (Complex 2) that causes oxidative stress damage and free radical production, which trigger an apoptotic cell death in the neurons cultures.

The project which forms the context of this master thesis is concerned with the evaluation of the capability and the power of therapeutic agents to reduce the effects of degeneration and death of neuronal tissue, by demonstrating both their neuro-protective and axo-dendritic effect. The project is carried out by Dr. Joseph Steiner and people on his staff, at the Department of Neurology, The Johns Hopkins University School of Medicine of Baltimore, MD. By administering compounds based on Paroxetine, a therapeutic agent, to the neuronal cultures, they have already demonstrated that it has a neuro-protective effect, both in vitro and in vivo in neuronal cultures. It has not yet been evaluated whether paroxetine also has axo-dendritic protective effect, that means a capability of minimize the effect of axons and dendrites pruning, for every neuron. This aspect is what Dr. Joseph Steiner is currently investigating on. He applied to Visiopharm A/S (the company this master thesis was carried out at) to perform an analysis of images of neuronal cultures that were treated with that agent,

in order to evaluate the presence and the health status of the neurons on the samples of neuronal tissue which those images were about. The main goal of this master thesis project is to develop a program implementing an image analysis protocol aimed to quantify the presence of neurons, dendrites and axons on the neuronal cultures the images are referred to, based on extraction and measurement of particular features of images. In particular, it has to quantify the presence of dendrites and axons in relation to the number of neurons cell bodies on the same image, by performing a comparison of such value in the three kind of images that were delivered over, depicting neuronal cultures that have been either untreated, treated with toxins and treated with toxins and, afterward, with therapeutic agent. By evaluating this value in each of the images referring to a particular treatment process in different areas of a neuronal tissue, it should be possible to quantify the effects of such extremes of treatment (quantity of toxin and quantity of therapeutic agent).
This introduction to the topic is followed by a more detailed description of medical background and informatics tolls used during the project period. Throughout the chapter the motivation for this thesis is pointed out and summed up in the end of this section.

1.2 Medical Background of the project

This section gives a brief overview of the medical and biological background of the project**.** It details about the effects of HIV virus on neurons and how it is possible to evaluate results of administration of therapeutic agents to neuronal cultures that have been treated whit toxins in order to emulate such effects on neuronal cultures.

The HIV is a lentivirus member of the retrovirus family. Infection with HIV occurs by the transfer of blood, semen, vaginal fluid, pre-ejaculate, or breast milk. Within these bodily fluids, HIV is present as both free virus particles and virus within infected immune cells. It is an RNA virus that is replicated in a host cell via the enzyme reverse transcriptase to produce DNA from its RNA genome. The DNA is then incorporated into the host's genome by an integrase enzyme. The virus thereafter replicates as part of the host cell's DNA. HIV infects primarily vital cells in the human immune system such as helper T cells (to be specific, $CD4^+$ T cells), macrophages, and dendritic cells. These cells are the white blood cells crucial to maintaining the function of the human immune system. Notably, HIV-1 infects a restricted number of cell types in the brain. Neurons are one of the few cell types in the human body that do not support HIV type-1 (HIV-1) replication. Although brain tissue from patients with human immunodeficiency virus (HIV) and/or AIDS is consistently infected by HIV type 1 (HIV-1), only 20 to 30% of patients exhibit clinical or neuro-pathological evidence of brain injury. Extensive HIV-1 sequence diversity is present in the brain, which may account in part for the variability in the occurrence of HIV-induced brain disease [3]. Neurological injury caused by HIV-1 is mediated mainly both by an indirect way (through excess production of host molecules by infected or activated glial cells) and a direct way, that is through administration of neurotoxic viral proteins, most notably the Tat and gp120 containing the V1V3 or C2V3 envelope region from non-clade B, brain-derived HIV-1 sequences. Dr. Joseph Steiner his staff are currently leading laboratory studies aimed to determinate whether

compounds based on therapeutic agents, as Paroxetine, have axo-dendritic protective effect on samples of neuronal tissue, which come from cerebral cortex and from hippocampus of rodents. The neuronal samples underwent a process of preparation in which they were also stained with compounds of fluorescent proteins (e.g. GFP or tdTomatoe) which exhibit bright colored fluorescence when exposed to the fluorescence microscope blue light. Afterward, neuronal cultures were treated with 3-NP, a toxin that first causes a dendritic and axonal pruning to every neuron which it get in contact with, and finally causing apoptotic neuronal cell death. Once the toxins take effect, degeneration process involving neurites (that is a general term with which both dendrites and axons can be identified) cause their structure to be torn into shreds, which also involves they to loss of their colored fluorescence. Finally, compounds based on Paroxetine, which is the test therapeutic agent of this project, were administered. After treating the neuronal samples with such compounds, images of those cultures were taken by means of an image acquisition station, composed of a fluorescence microscope and a camera.
Paroxetine is an antidepressant [4] that is a selective serotonin reuptake inhibitor, and it also has neuro-protective effect. By means of an analysis of those images of neuron cultures treated with compounds based on this substance, it is possible to determinate whether such therapeutic agent also has an axo-dendritic protective effect.
A deeper explanation of the sample preparation and image acquisition can be found in the first part of chapter 2-"Data Material". A more detailed description of both Paroxetine and 3-NP, which are the main substances used in treatments of the neuronal cultures of this project, can be found in Appendix B and Appendix C.

1.3 Informatics tools of the project

As the developed program is ideally to be used as an extension to the Visiopharm microscopy imaging system, VIS, the main program is programmed in the same language as VIS, that is C++. It also uses some of the functions from the program code behind VIS, Imagining Utilities, that are mainly functions for image handling, basic image manipulation and image displaying. Images were converted into objects of the CImage class, in which an image is regarded as the result of the overlapping of three layers:

- the original images (in which every pixel intensity is given by the combination of its values in the main color bands)

- a layer named "label" of the same size of the original image (same number of rows and columns). For each pixel of the original image**,** it has initial value of 0 (that means "no label"). That value can be set to another value (that means another color) by using a specific function.

- a layer named "mask", which permit particular images operations to be performed only on a defined area of such image

As the latter one has not been used, every displayed image must be considered as the result of the overlapping of the original image and the label layer that is made semi-transparent so that also the underlying image can be viewed. The labeling of particular features in the images is a fundamental concept that has been widely used in different steps of the implementation of the image analysis protocol. Finally, a part of the project period (one month) was spent on the implementation of a C++ sourcecode which allows the integration of a Matlab executable file into a C++ code, for the latter to be able to use such function as it was a C++ code built-in function (*). Therefore, the main goal to achieve was the creation of common data type that could easily be handled and shared by both programming languages. This integration was performed since Matlab was deemed to be a quicker way to implement functions handling data structures like matrices and arrays which the images were converted into, and because it is also possible to use several tools and libraries for image analysis. The integration of generic functions from Matlab to C++ had been carried out by using the Matlab Runtime Compiler, according to the steps described in the Matlab integration tutorial, from the Mathworks website [5]. It was installed, tested and run with simple functions. As it was always possible to achieve the desired result by using VIS' Imaging Utilities functions and by implementing functions in C++, it was never actually necessary to use Matlab. Nevertheless the program developed has an inactive part that, activated, can call a Matlab function, to compute a series of operations from input parameters and return an output, both in common data type. A detailed explanation of how this integration has been carried out is given in Appendix G.

1.4 Project Outcomes

During the project period, different sets of images depicting different conditions of treatments were delivered over in different times, and those images were used as training set. Once the final two extremes of treatment were delineated (which quantity of which therapeutic agent and toxin), the final dataset, that was the project's Test Set, was made available. The images of the test set were divided into three main groups, depending in the treatment processes they were submitted to. The following groups were named:

- Control group (untreated cultures)
- Lesioned Cultures (cultures to which toxins were administered)
- Lesioned Cultures + Paroxetine (cultures to which toxins were administered, first, and a potential axo-dendritic protective compounds based on Paroxetine, afterward)

(*) The programming softwares were Matlab R2009b and Microsoft Visual Studio C++ 2005.

Taking into account the previous considerations and the information in the introduction section, it is now possible to point out the main project objectives. At the end of the project period, the goal was to have a program that performs a quantitative analysis of the feature of the images and gives as output the values of the following main measured parameters:

- Total Length of the Neurites Branches (intended as sum of dendrites and axons, see the blue circled components in Figure 1.1)
- Total area of the Neuron Cell Bodies (see the green circled components in Figure 1.1)
- Total number of the Small Beads (intended as the remaining of the neurites structures, after they have suffered a degeneration process, see the yellow circled components in Figure 1.1)

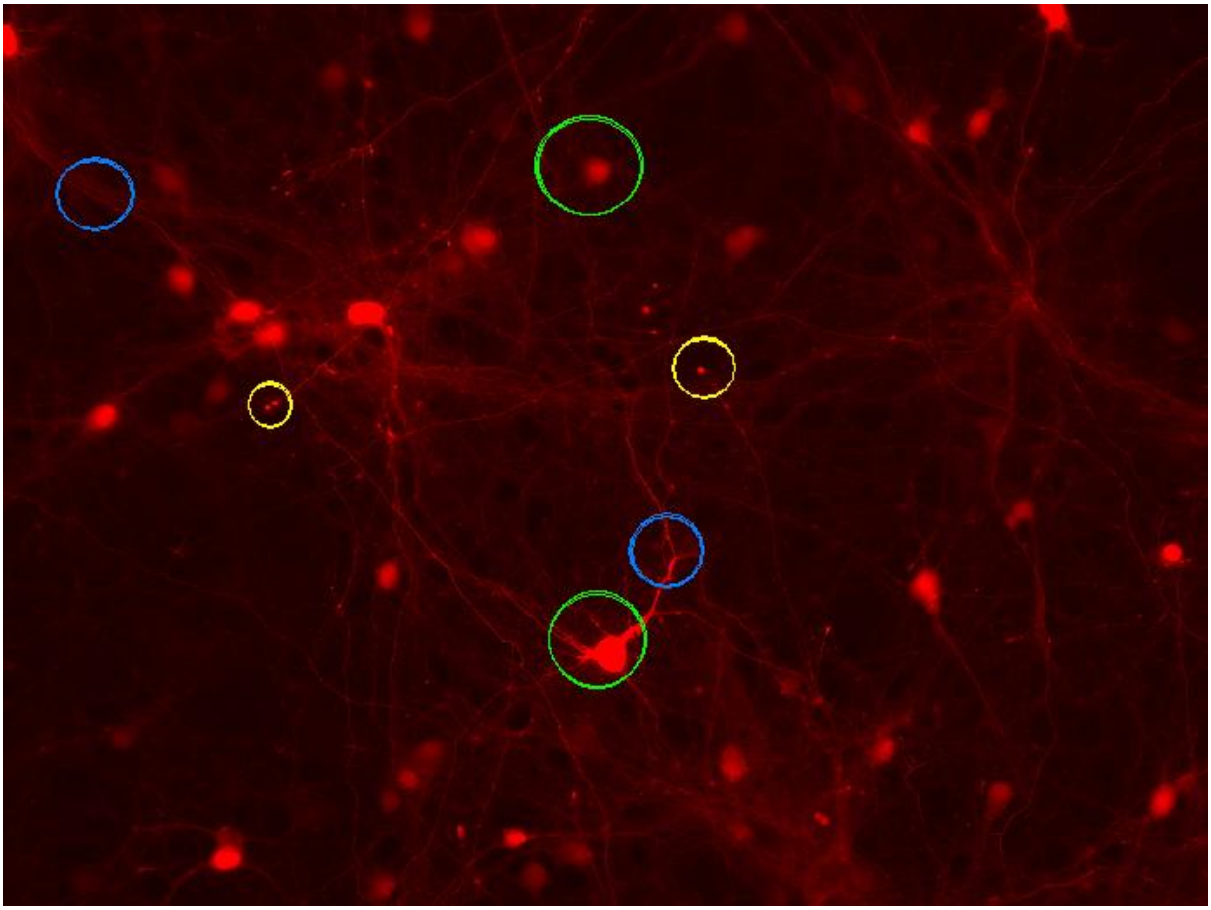


Figure 1.1 – Instances of the three main components of images, whose measurement are the program main outputs

By means of the comparison of these values on the three main groups of images it is possible to have a step forward to determine whether the administration of a given compound, based on Paroxetine as therapeutic agent, has actually been able to reduce the effect of the neuronal degeneration triggered by the toxins, or not. That conclusion also involves that such compound has an axo-dendritic protective capability.
Other parameters were calculated by the program. They are:

- Total number of the Neuron Cell Bodies
- Total area of the Small Beads

Finally, it is important to remark that, in order to correctly evaluate the effect of a treatment on each neuron, only the absolute values of the three main parameters were not sufficient. In fact, the quantity of neurites must always be evaluated in relation to the quantity of neuron cell bodies on a given image, to get an indicative value of what was the actual effect of the treatment, in mean, in terms of each neuron.
Therefore, a parameter including both the length of the neuron and a value concerning the neuron cell bodies (which can be both the area and the number) was calculated. Further description of the content of the images that might help in getting a clearer view of the meaning of the listed parameters is given in the following chapter.

1.5 Project outcomes - application to the medicine

Dr. Steiner and his staff are setting up the assays this project is aimed to quantify and also attempting to validate a new assay system in order to better evaluate new and novel neuroprotective and therapeutics agents for neurologic disorders. This assay and quantization of axo-dendritic degeneration is a good and early measure of neurodegeneration that occurs in multiple neurologic disorders, more specifically in Alzheimers, Parkinsons, ALS, stroke and Huntington's disease. Of course, in these neurodegenerative disorders, different populations of neurons are vulnerable and may become dysfunctional-cholinergic, dopaminergic to motor neurons or to cortical/striatal neurons or also to medium spiny neurons in the corpus striatum. A common feature of these neurodegenerative disorders is the axodendritic dysfunction and degeneration, that will be possible to quantify with the project's algorithm. Compounds that are regarded as neuroprotective and axonally protective against neuritic degeneration are good candidate compounds to treat these neurologic disorders. Two such compounds, paroxetine and fluconazole, which were found to protect neurons against HIV neurotoxic proteins in vitro and in mouse models of neurodegeneration, will shortly be entering human clinical trials as adjunctive neuroprotective agents to treat neurocognitive impairment resulting from HIV infection. Additional compounds, which Dr. Steiner will screen against axodendritic degeneration (quantified by the project's program), which penetrate the blood brain barrier and achieve neuroprotective brain levels, will then be evaluated for neuroprotective efficacy in an animal model of the clinical indication. Novel

drug-like compounds with in vitro and in vivo efficacy may likewise be used as therapies to treat these same neurodegenerative diseases.

1.6 Reading the thesis

This section states what can be found in the next chapters of the thesis and lists the abbreviations used in the report.

1.6.1 Thesis overview

The report of this thesis is divided into four parts:

PART I – Description of the Data : Introducing the reader to the dataset used in the thesis, consisting of images captured by microscope of cultures of neurons, which have been submitted either to treatment based on administration of toxins or potential axo-dendtritic protective compounds or both or none of the former. A brief description of their contents in terms of the tree main components is also given, in some of these.

PART II – Analysis of Methods: This part describes the image analysis methods that have been chosen, in order to obtain the best result on the three steps of the Image Analysis Protocol used, that are pre-processing, processing and post-processing. It is possible to evaluate the results obtained by the application of each method, and what is the final result for a given image, starting from an image named TestImage.

PART III - Methods Implementation: This part describes how the methods have been implemented in language C++, by using pseudo-code and showing what is the meaning of the value of the parameters of the functions.

PART IV – Results and Evaluation: Values of the outputs obtained from the application of the implemented methods to the images of the Test Set are given and analyzed. A statistical validation of the results and comparison with what was expected to obtain are used to draw a final conclusion. Some suggestions for future work will also be provided based on the observations made and the experience achieved during this project

1.6.2 List of Abbreviations

NCB	Neuron Cell Bodies
NB	Neurite Branches
SB	Small Beads
IAP	Image Analysis Protocol
PI	Pixel Intrensity
OI	Original Image
FI	Filtered Image
PLLF	Polynomial Local Linear Filter
MF	Median Filter
FI1	Final morphological-post-processed PLLF filtered Image
FI2	Final morphological-post-processed MF filtered Image
FCI	Final Classification Image

PART I

Description of the Data

Chapter 2

Data Material

This chapter gives an overview of the data analyzed in the project, showing differences between the various kind of images in terms of contents, relying most on their acquisition modality, samples staining and image saving format. The chapter also introduces and describes the exemplary TestImage, which will be used throughout the steps of the Image Analysis Protocol. All the images were captured by Dr. Joseph Steiner and people on his staff, at the Department of Neurology, The Johns Hopkins University School of Medicine, Baltimore, MD.

2.1 Description of the Data

2.1.1 Sample provenance and preparation

Here follows a detailed description of the preparation of the two kinds of neuronal cultures utilized in this project, which the database's images were afterward taken from. Neuronal cultures come from rodent cerebral cortex and from hippocampus of *Sprague Dawley* rat, which is an outbred multipurpose breed of albino rat used extensively in medical research. Cultures were prepared from embryonic day 18 Sprague–Dawley rats. Neuronal tissues were dissociated by gentle tituration with a fire-polished glass pipette in calcium-free Hank's balanced salt solution. The single cell suspension was centrifuged at 1000 g and re-suspended in minimal essential medium containing 10% heat-inactivated fetal bovine serum and 1% antibiotic solution (Sigma, St. Louis, MO, USA). For more pure neuronal cultures, cells were allowed to attach for 3 hours before the media was replaced with serum-free neurobasal medium containing 2% B-27 supplement (Gibco, Rockville, MD, USA) and 1% antibiotic anti-mytotic mix (10^4 U of penicillin G/mL, 10 mg Streptomycin/mL and 25 mg Amphotericin B/mL, Sigma, St. Louis, MO).Rodent neuronal cultures were used between 10 and 14 days in vitro and were > 98% neurons as observed by immune-fluorescent staining for MAP-2; the remainder of the cells were predominantly astrocytes as observed by immune-staining for glial fibrillary acidic protein. Alternatively, rat mixed hippocampal neurons were generated from freshly cultured rat hippocampi in neurobasal media containing 5% fetal bovine serum and 2% B27 supplement. Mixed rat hippocampal cultures were plated into 96 well plates at a density of $4x10^5$ cells per mL. Three hours after plating the hippocampal cultures, the media was replaced with Neurobasal media containing 5% heat-inactivated FBS, 1% B-27 supplement (Gibco, Rockville, MD, USA). Cultures were used between 10 and 14 days in vitro. These cells contain neurons and astrocytes in nearly 1:1 ratio, with a percent or two of microglia in them. Neurons are then labeled in cultures by transfecting the cultures either with a red or green fluorescent protein under the expression of a neuron

specific promoter (Amaxa's Nucleofector is used to achieve this). Then neurons are allowed to mature in the incubator at 37 degrees C until they have been in culture for 11-14 days. For the experiments, the cultures are usually pretreated with the neuroprotective agent for 1 hour prior to adding the toxin. Then following toxin addition (2-5 mM 3NP, 100-500 nM Tat, 100-300 pM gp120), cultures are incubated for 18 hours at 37 degrees C.

2.1.2 Image Acquisition

The previously described cultures were imaged by using a *Zeiss AxioObserver Z1* fluorescence microscope with live imaging capabilities (which is described in Chapter 1, sub-chapter 1.). The live imaging system is used to capture phase contrast images along with fluorescent images of the cells and neurites.The Microscope was equipped with a *Maerzhaeuser* digitizing control stage (manual and software-controlled mode) for preselecting and storing points of interest and live imaging capability and with a low-light digital image capture camera *Zeiss AxioCam HRm* (with both color and grayscale images capturing capability). The Zeiss inverted fluorescence microscope is used in experiments involving living cell cultures over extended periods of time. The system is completely automated and can be programmed for long-term experiments on living cells.
During this experiment:

- the Zeiss AxioObserver Z1 collecteded 9 images objectively in an automated fashion from each well of a 96 wells plate of the microtiter plate, where the mixed hippocampal cultures have been grown;

- the specimens were illuminated with light of the specific wavelengths of either 2nm or 5nm.

- Most of images in the project database were taken using a lens with a magnification of 20X, a few of them were captured at 10X at 40X. (*)

A further description of fluorescence microscopy technique can be found in Appendix F.

2.1.3 The contents of the images and their variation

It might be opportune to introduce some general concept that might be helpful in order to understand the following image description (see Fig. 2.1 for specific explanation). A neuron is composed mainly of:

Neuron cell body, the central part of the neuron (often called the soma)

(*) In all regular microscopes, the magnification effect provided by the built-in lens (which magnify image by x 10) must be multiplied by the external lens value to get the final magnification value. This means that using for example a 20x lens value (as above) actually magnifies the image by x 200.

- Dendrites, the branched projections of a neuron that act to conduct the electrochemical stimulation received from other neural cells to the cell body, or soma, of the neuron from which the dendrites project;

- Axon, a long slender projection of the neuron that conducts electrical impulses away from the neuron's cell body.

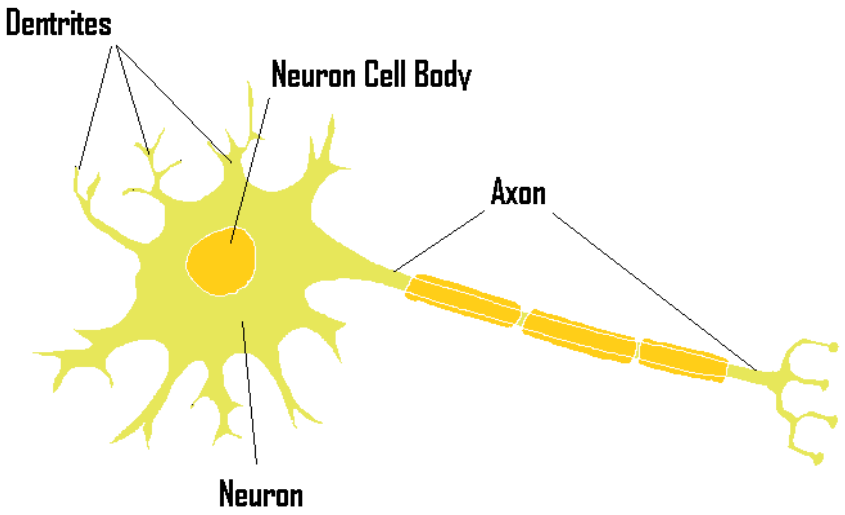


Figure 2.1 – Structure of a neuron, where its main components are showed

Neurons cultures were stained with compounds of fluorescent proteins (e.g. GFP or tdTomatoe) which exhibit bright colored fluorescence when exposed to blue light. Thus the presence of such fluorescent protein components in the image is a good (quantitative & qualitative) index to quantify the health status (or the degeneration status) of a neuronal tissue. Every image depicts a sample of hippocampal tissue and features a variable number (or even none) of each of the previously described biological components. A fourth component, named Small Beads is present in the image. These small beads are considered the result of a dendritical (or axonal) degeneration process and it is however difficult to reconstruct their former layout. Hence it is possible to do the following assumptions:

- It is not possible (nor it is important) to distinguish axons from dendrites; they both will be named “neurites”.

- The more neurites that are present in an image the better the neuronal tissue health status is.

- The presence of small beads is an indicator of the degeneration process.

In the following image, (named TestImage, which is the one used throughout the steps of the Image Analysis Protocol) instances of each component listed below can be observed:

- Neuron Cell body (see green circled areas in Fig. 2.2)
- Neurites (see blue circled areas in Fig. 2.2)
- Small beads (see yellow circled areas in Fig. 2.2)

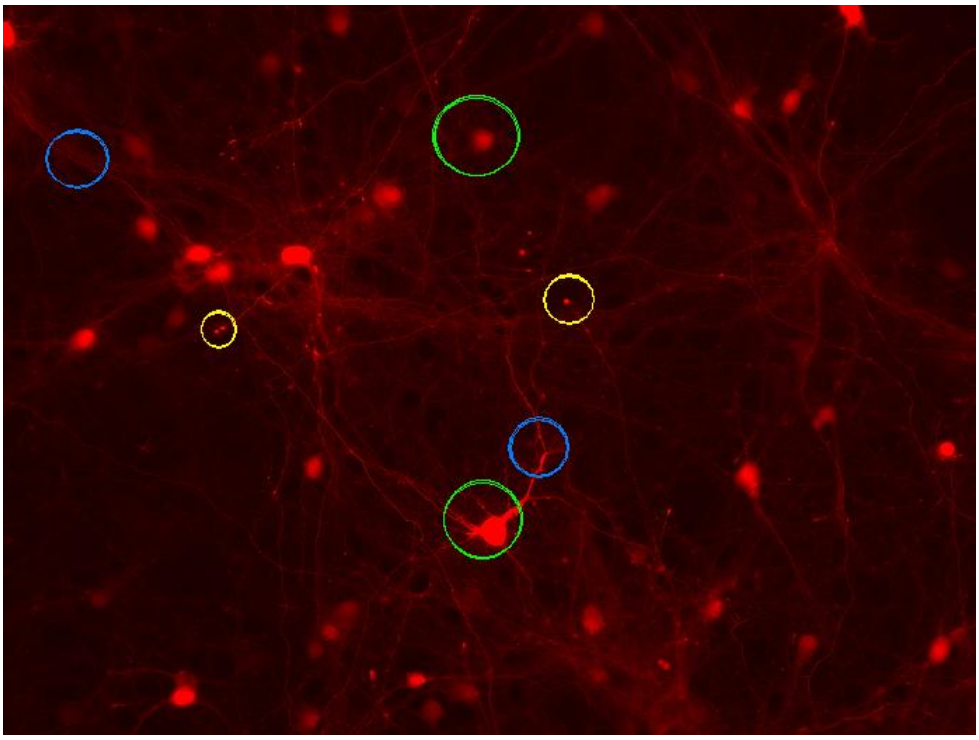


Figure 2.2 – Instances of the three main different components of images

Here follows a gallery of images that show the diversity of the amount of each of the three main components in the different kinds of images this project is aimed to analyze, as these are the parts that this implementation is most concerned with.Images are divided into two main groups, Training Set and Test Set. This is also the temporal order in which these images were analyzed, as the Test Set has been made available only later in the project period. Therefore, most of the work of algorithm implementation has been lead on images of the training set, which were anyway qualitatively similar to the test set ones. In fact they featured the same variance in terms of:

1) Magnifications (images of both groups were either 10x, 20x or 40x magnified)
2) Background textures
3) Different total amount of the three main components of neuronal tissue relied to a normal tissue condition or to the presence of toxins, therapeutic agent or both

Images of the Training set

A more detailed desription of what each image does depitcs can be read beside the image.These images feature also the following common property:

- most of the information is either on the green band (figures. 2.3) or on the red band (figures 2.4)
- there are no substantial variations of contrast in different areas of the same image

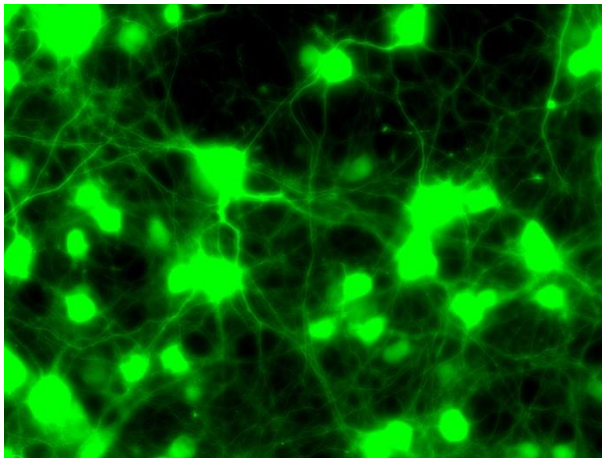
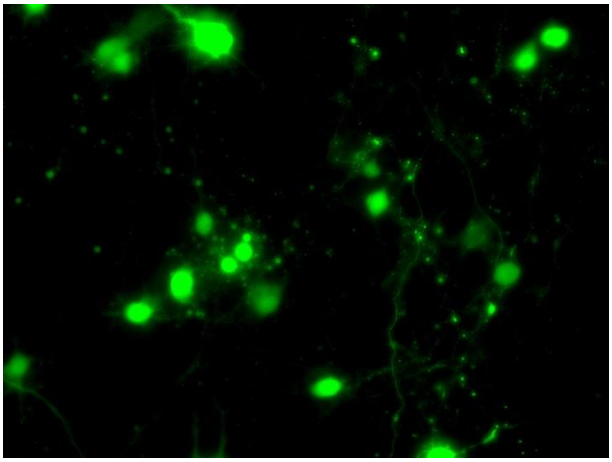


Figure 2.3a – A 40x magnified image, depicting a neuronal tissue in normal health conditions (that means neither toxins nor therapeutic agent application). Most information is in the green band. By a brief visual inspection, it is possible to locate a lot of neuron cell bodies, neurites (both thick and thin structures) and a few small beads. Black background permits an easy distinction of the main neuronal tissue components.

Figure 2.3b - A 40x magnified images in which the 3-NP toxin application took effect. Most information is in the green band.

The variation of total amount of both neuron cell bodies and neurites from the former image is clear. It looks like the neurites branches were pruned. Furthermore, The amount of small beads seems to be bigger than the one in the former image. Black background.



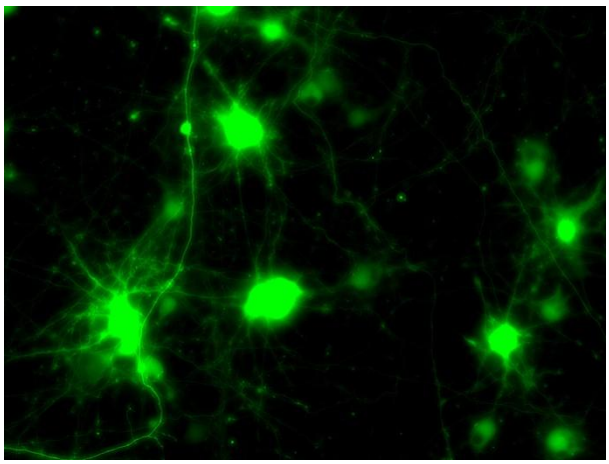


Figure 2.3c - A 40x magnified images in which the 3-NP toxin and a Paroxetine based treatment have been applied, in that chronological order. It looks like the values of total amounts of the three main components lie in between the values of the same amounts of former images and that a neuritic growth is present. Most information is in the green band. Black Background

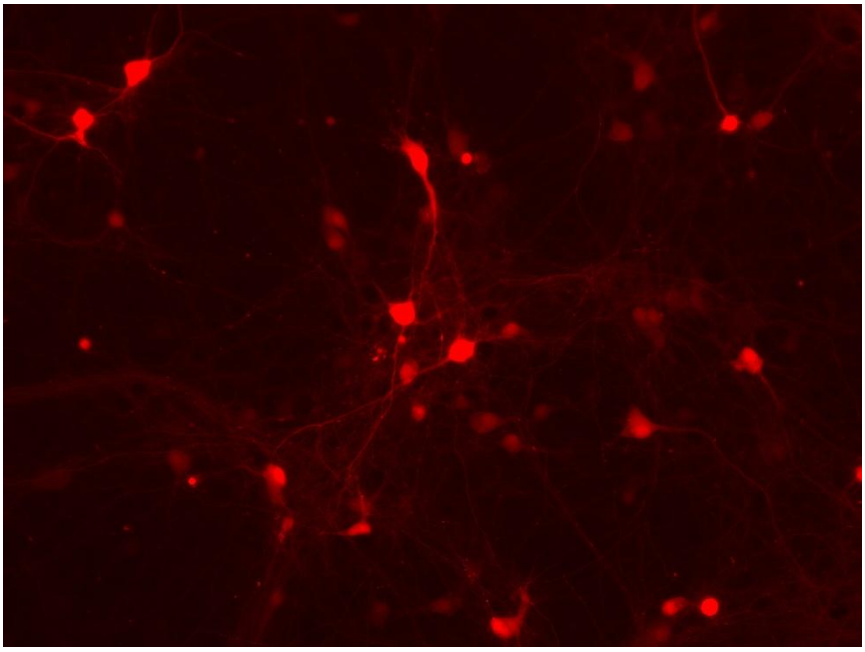


Figure 2.4a – A 20x magnified image, depicting a neuronal tissue in normal health conditions (again means neither toxins nor therapeutic agent application). By a brief visual inspection, it is possible to locate a lot of neuron cell bodies, neurites (both thick and thin structures) and a few small beads. Most information is in the Red band. Black background.

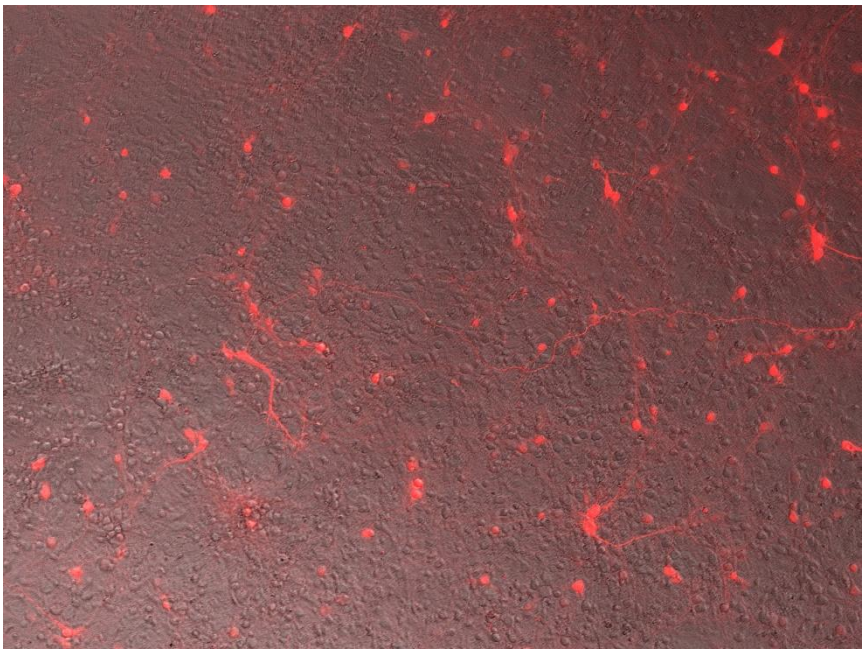


Figure 2.4b – A 10x magnified image, where fluorescent images have been overlaid onto the phase images. It depicts a neuronal tissue in normal health conditions. It is possible to locate a lot of neuron cell bodies, neurites (both thick and thin structures) but detections of small beads seem more difficult, also because of the phased background. It is also possible to notice that the rightmost-upper corner is a darker, where there is also more contrast between the neuronal components (red objects) and the background.

In all of next images, the fluorescent images have been overlaid onto their phase image, therefore a phased background which has components in all of the three images' bands is present. Most of the images' information is in the red band.

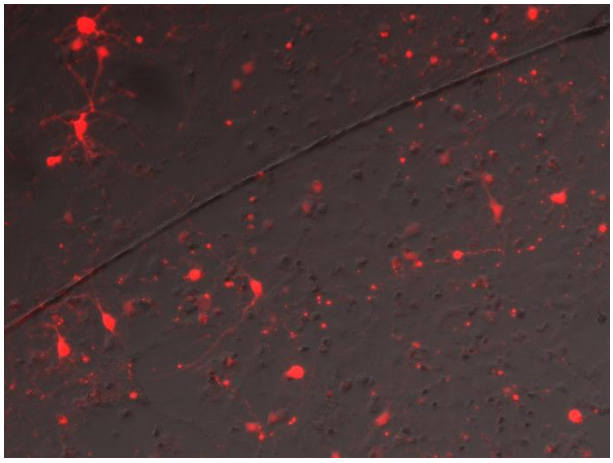


Figure 2.5a - A 20x magnified image in which a 5mM 3-NP toxin application took effect. By a brief visual inspection it is possible to locate neuron cell bodies, neurites (both thick and thin structures) and several beads. It is also possible to notice that there is a dark linear structure (an artifact, perhaps an edge of some object or a ruined surface) and that a darkened area is present on the leftmost-top corner.

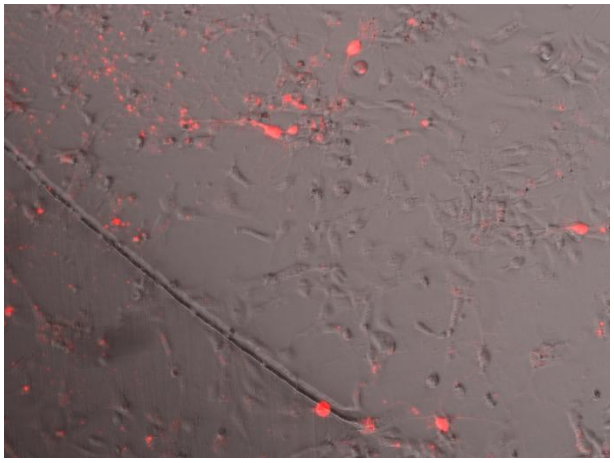


Figure 2.5b - A 20x magnified image in which a 2mM 3-NP toxin application took effect. The same considerations for the former image are valid here, as well. There is a darker corner on the leftmost-bottom corner and it looks like this in this image there is less contrast between the neuronal components and the background, especially on the left part of the image.

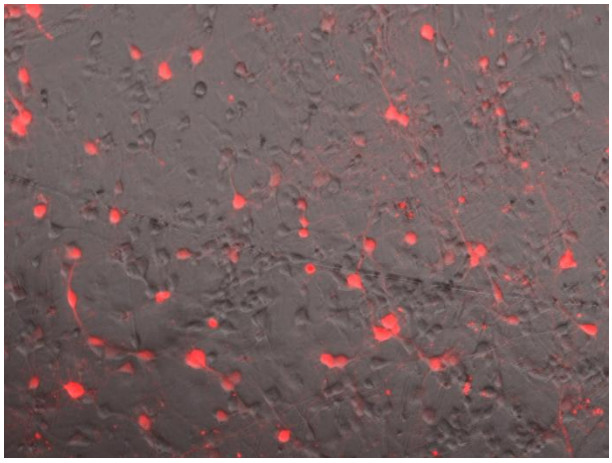
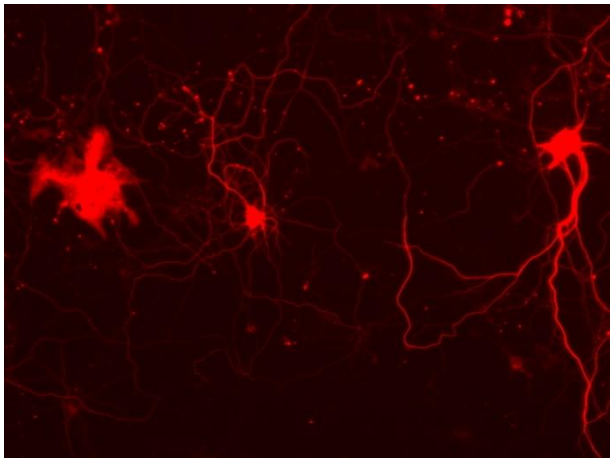


Figure 2.5c - A 20x depicting a neuronal tissue in normal health conditions (again, it means neither toxins nor therapeutic agent application). There is a darker corner on the leftmost-bottom corner, a linear structure appears and it looks like in this image there is less contrast between the neuronal components and the background, especially on the left-upper part of the image.

Images of the Test set

Figure 2.6a - A 20x magnified image in which a 2mM 3-NP toxin application took effect. By a brief visual inspection it is possible to locate neuron cell bodies, neurites (both thick and thin structures) and several beads. No phased background nor undesired artifact, as darker linear structures are present, and the image look brighter since a different staining protein has been used.



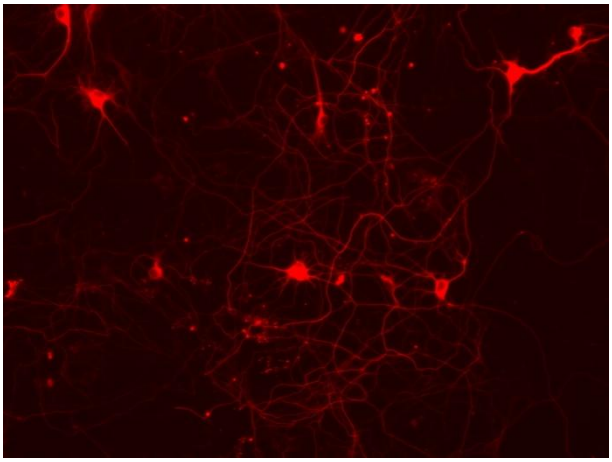
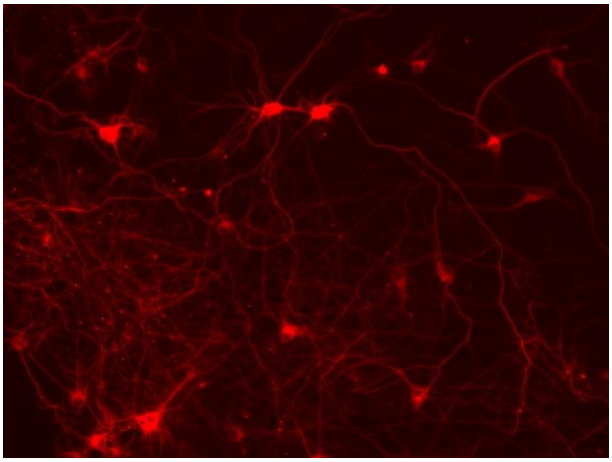


Figure 2.6b - A 20x magnified image in which a 2mM 3-NP toxin application took effect and afterward an administration of a therapeutic agent were made (Paroxetine) The same considerations made for the former image are valid here, as well.

Figure 2.6c - A 20x depicting a neuronal tissue in normal health conditions (again, it means neither toxins nor therapeutic agent application). The same considerations made for the former image are valid here, as well.



2.1.4 General comments

Here follow some comments about the image features and some indication about what should be improved in order to obtain an input image in which features are as detectable as possible:

- some neuron cell bodies overlap other or others neuron cell bodies, thus it is very difficult to distinguish one cell body from another and to count them.

- phased background and artifacts as dark linear structures have components in each main color band (RGB), therefore they represent an issue in order to get the actual linear structures to be correctly detected, being itself composed of both linear structures and round objects

- darkened corned (created by the light of the microscope) can be influential on the identification of the desired features; that effect must be compensated

- the images of the Test set were generated using a slightly different vector, a Tau-tdTomatoe fusion protein which labels the neurites more sharply than with just tdTomatoe alone, used for the Training Set.

2.1.5 An overview of the data

The data which formed the database of this project, both training and test set ones, can be futher categorized into the following table, detailing how the data were structured and which data was used for which purpose. Every image has a size of 1388 (width) per 1040 (height) pixels and a depth of 1 frame.

Set	Main Color Band	Magnification	Format - Resolution	Substances Administered			
				CONTR.	TREAT.	LES.	TOT.
Training Set	Green	40x	JPG 96 DPI	16	4	2	22
		20 x	ZVI 49618 DPI	-	8	10	18
	Red	20x	JPG 72 DPI	-	10	5	15
		10x	JPG 72 DPI	-	5	-	5
Test Set	Red	20x	JPG 72 DPI	5	20	10	35

Table 2.1 – An overview of the data and its main properties. Images of test set, that was analyzed only on the last period of the project, had the same properties of at least one kind of images that were already be present in e training set. The values of format and resolution were used to set the parameters of post-processing methods.

Where:

- "**CONTR.**" = Control Cultures, no administration of toxins or therapeutic compound
- "**LES.**" = Lesioned Cultures: administration of toxins (3NP)
- **"TREAT"** = Lesioned and Treated cultures: administration of Toxins+therapeutic compound based on a therapeutic agent (Paroxetine)

As can be seen from the table above, the training set covered all of the possible variants in terms of main color band, magnification, format and resolution. As will be explained in detail in the following chapters (Part II, Analysis of Methods), each final image configuration resulting from the combination of those technical features of the images involves:

- the use of specific parameters for the methods performing operations of image pre-processing, processing and post processing.
- a particular ordered sequence of morphological post-processing steps (see par 6.3)

Therefore when the Test Set was made available, the Image Analysis protocol was already able to figure out what were the right operating parameters to be used in order to obtain the best result from each of its steps and to finally obtain the best image feature classification and - consequentially - the best desired parameters as output.
The intention was to implement a protocol whose functioning was depending as much as possible on the image content itself, avoiding the use (as far as possible) of extern inputs which were not relied from the image content. The images magnification - one of the most important parameter - was not always derivable from the image itself (*) but its values was written on the image's file name, instead. Finally, the reason why the Test Set was sent late in the project period was that Dr. Joseph Steiner (and his staff) had been working on the identifications of numerous potential neuro-protective compounds from their screening assays during the thesis project period. In the images belonging to the Test set depicting results of administration of toxin plus therapeutic agent to the neuronal tissue, the best-resulting two extremes of treatment (therapeutic agent and toxin) were used. The test set was sent once such extremes were delineated, to see if the compounds were able to block neuritic degeneration, based on the Image Analysis Protocol outcomes.

2.1.6 The illustrative TestImage

The image of Fig. 2.2 was chosen as input image to be used throughout the different steps of the illustrative running of the Image Analysis Protocol, whose results are displayed and explained in the next chapters. Therefore, it is also possible to track the image features extraction and enhancement process paths, from the original image to the final classified one.

(*) Further explanations of this choice are given in sub-chapter 3.2.1"An external input: the image magnification value" and chapter 7.

PART II

Analysis of Methods

Chapter 3

The Image Analysis Protocol

The following chapters give an overview of the computational methods used to perform the analysis of the image of the dataset, in order to obtain the protocol to output the desired measured values of the image features. The Images Analysis Protocol is made of typical steps and methods commonly used to perform feature analysis on images of biological samples taken by microscope, like the images of neuronal cultures of this project.

Step	Task Name	Main Methods	Objectives
1	**PRE-PROCESSING** **FILTERING**	**Polynomial Local Linear (1D and 2D) , Laplacian** **Median**	- Structure Shapes Enhancement - Noise Reduction/ Suppression - Localize Linear structures - Localize Round structures
2	**PROCESSING** **SEGMENTATION / CLASSIFICATION**	**K-means Classification** **Bayesian Classification** **Adaptative Thresholding**	- Separate the different structures - Locate objects and boundaries
3A	**POST-PROCESSING** **MORPHOLOGICAL POST-PROCESSING**	**Skeletonization** **Erosion and Dilation** **Area Condition** **Length Condition** **Irregularity Condition**	- Morphological Operations on Neuron Cell Bodies and Neurites - Eliminate smallest neurites - Pattern/Shape recognition - Shape Editing - Separate Linear structures from NON-Linear circular structures
3B	**POST-PROCESSING** **OBJECT FEATURES / MEASUREMENTS AND OUTPUT GENERATION**	**Neuron Cell Bodies Area Measurement function** **Neurites - Total Length Measurement function** **Small Beads - Total Number and Area Measurement function**	- Characterization of Neuron Cell Bodies and Neurites - Eliminate smallest neurites - Pattern/Shape recognition - Shape Editing Calculate: -**NEURON CELL BODIES AREA AND NUMBER** -**NEURITES LENGTH** - **SMALL BEADS AREA AND NUMBER**

Table 3.1 - Image Analysis Protocol Plan. At the end of each step, the listed objective must be achieved

3.1 Image Analysis Protocol (IAP) – general purposes

The following chapters of Part II, "Image Processing Methods" provide detailed description of the image processing methods listed in the "Main Methods" column of Table 3.1, giving a theoretical explanation of how they work. The final subchapter of every chapter of Part II shows the results of the application of the method that was chosen to perform each step of the IAP on a given input image. The goal of the IAP program is to measure the following features on each image:

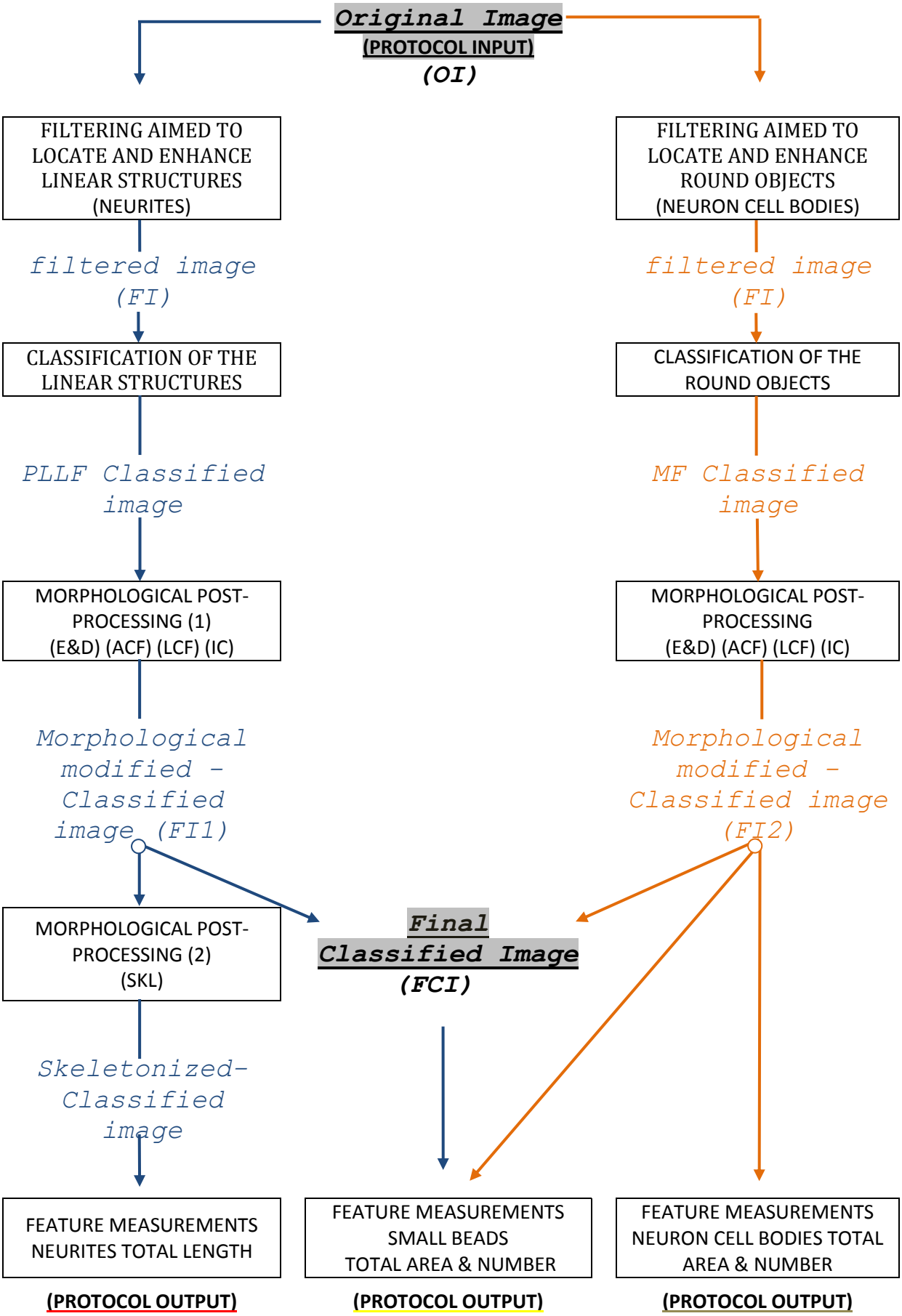
- Neuron Cell Bodies Total Area and Total Number
- Neurite Branches Total Length
- Small Beads Total Area and Total Number

In order to carry out the measuring process, the program runs through an ordered sequence of methods. It starts off by performing an image pre-analysis, passing through the pre-processing, classification and morphological post-processing of the features till the final post-processing task of measurement, where the three desired features are calculated. The scheme 3.1 shows the ordered sequence of methods that has been performed, starting from a given input image, in order to get the protocol to measure such outputs on the final classified image. Furthermore, in order to create a protocol suitable to process various kinds of image of neuronal tissue, the intention was to develop algorithms which functioning and parameters were depending as much as possible on the image itself.

3.1.1 The IAP implementation overview

The implementation of IAP's methods by meaning of computational algorithms has been carried out in the programming language C++, since the program developed is to be used as an extension to the Visiopharm microscopy imaging system (VIS) which is based on C++. Most of the utilized functions are contained in Imaging Utilities library, which is a part of the program code behind VIS. These functions are mainly methods of the previously described CImage class, which permit basic operations of image handling, manipulation and displaying and functions for advanced image post-processing.

Some of the utilized functions were used only by giving them the input parameters (e.g. the filtering functions) without modifying their source code, whereas other Imaging Utilities functions have been modified and fine tuned to perform the specific task they were aimed at (e.g. Area , Length and Irregularity condition functions). Further functions have been implemented in order to carry out image pre-analysis, parameters passing between other functions and other particular tasks of the protocol (e.g. multiple thresholding classification). Chapter 7 ("Implementation and running of the Image Analysis Protocol") goes more into detail about how the image computational methods have been implemented in C++ and about the functioning of the algorithms and the meaning of the input parameters.



Scheme 3.1 – Functions and fluxes of images through the different steps of the IAP

3.1.2 Choice of the best-result-giving method and optimal parameter values

The best-result-giving method was chosen for every step of the IAP and the research of optimal parameter values for the methods was another important task, as well. Each method was performed with its optimal parameter values which guarantee that the final result of every step was the best that could be achieved. For every step of the IAP, the decision of the best-result-giving method and the research for optimal parameters have been explained in detail, reporting the practical reasons of such decisions and a comparison with other methods and non-optimal parameter values. In general, in order to establish whether a method (or a parameter value) gave better results compared to another one, it was often necessary to evaluate the goodness of their output results by the comparison of:

1. the capability of achieving the desired goal;

2. the ease in obtaining good results from the next IAP step

From point 2., it appears clear that the final algorithm has been chosen after the comparisons of several combinations of methods belonging to different stages of the IAP, especially between the classification and the morphological post-process steps and between the different methods performing the morphological post-processing task.

3.1.3 An overview through the steps

In the following chapters, in order to have a connection between the different steps of the IAP, these have been performed in an ordered sequence on a unique given image (which is named TestImage) so that it is possible to track the image features extraction and enhancement process paths, from the original to the final classified one. It is also possible to see clearly what does every method perform and improve, starting either from image or parameters outputted from the former IAP's step. Choices for the best-result-giving method and optimal parameter values have been discussed and compared with the others possibilities. All of the comments and displayed results are qualitatively similar to those obtained by applying the IAP to any other image of the database. Each measurement method has been applied first to the training set previously described in Chapter 2 ("Data material") in order to figure out which was the best-result-giving method and to achieve a fine-tuning of the parameters of the methods. The sequence of best-result-giving methods, with the fine-tuned parameters has been finally applied to the test set, when it was made available, and, as the images of the training set covered all of the possible variants in terms of main color band, magnification, format and resolution, no further adjustments of methods sequence or parameters was made.

3.2 Pre-analysis stage

As the protocol was supposed to work on different kinds of images (see Chapter 2, par. 2) a pre-analysis stage has been implemented and run before the IAP's steps listed above, in order to extract the following important information about the image format and content:

- Main Color Band (Red or Green)
- Magnification Value
- DPI Values

On the basis of those different values, the following general modifications have been applied to the images protocol post-processing stage:

- different parameters are used for algorithm
- methods may be applied in different sequences

Both the values of the parameter used for the methods and the method application sequence are fundamental points; a wrong application of such points can lead to a wrongly classified image as output, that substantially differs from a well classified one because of the wrong feature classification.

3.2.1 An external input: the image magnification value

A Further kind of pre-analysis had initially been performed between the classification step and the morphological post-classification step of the IAP. It consists of the calculation of

- Neuron Cell Bodies Area Mean Value

It was calculated by measuring such values on the filtered image, which aims to localize the neuron cell bodies structures (its functioning and properties are better and wider explained in Chapter 7 ("Implementation and running of the Image Analysis Protocol").
The importance of knowing such a value right before the morphological post-processing stage, is given by the fact that it provides indirect (approximate) information about the size of neurites and beads. Thus it is possible to know a priori (*), for each of the three main feature:

- how much shape of such objects can be modified without deleting them completely

(*) Such information was obtained by a visual inspection and analysis of features samples on the three different kinds of images

- how much shape of such objects can be altered without causing their former (original) shape to be no longer rebuildable.

It appears obvious that the knowledge of the exact value of such parameter is basic for the achievement of a good final classification result. Although the intention of developing algorithms whose operating parameters were depending as much as possible on the image content itself, it was observed that so calculated value was heavily depending on the health status of the neuronal tissue (depending on the application of toxin or therapeutic agent or both, or none). For instance, the following values of such calculated parameter were fairly similar:

- image featuring a magnification of 20X of a tissue sample where toxin plus therapeutic agent administration was applied

- image featuring a magnification of 20X of a tissue where no administration was applied.

A reliable value of mean area of the neuron cell bodies was not obtainable from such a method. Hence, it was decided to adopt a more safe solution to give that parameter value as input to the protocol step aimed to perform the morphological post-processing methods on the image components. The chosen solution consists of giving as input to the algorithm the magnification value (written on each image's file name) as text, typing it on the command window (as previously described in sub-chapter 2.1.5 - "An overview of the data"). A more detailed explanation of this aspect can be found in chapter 6 - "Morphological Post-processing". Finally, the idea of using the value of image's DPI (that can be given as image properties of any common software for image displaying) was not regarded as a good alternative. Although its value in images taken with magnification value of 40X was different, from example, from images taken with magnification value of 20X, such value was the same for both images taken with magnification value of 20X and 10 X.

Chapter 4

Image Pre-Processing

The main goal in image pre-processing is to perform image features enhancement, detection and extraction. It is possible to achieve such a result by applying a filtering operation on the original image. Filtering application to a time-varying input signal is a very common technique in electronics, digital signal processing and mechanical engineering, by means of which it is possible, for example, to eliminate unwanted frequencies from the input signal or to select a desired frequency among many others.

4.1 Linear Filtering - a brief overview.

The simplest way to perform a linear filtering is by applying a linear operator (linear filtering) to a digital image, by means of which it is possible to replace each pixel by a linear combination of its neighbors. The prescription for the linear combination is called the *convolution kernel*, which is explained in the follow. A 2D filter matrix is used for every pixel of the given image, and the sum of products is taken. Each product is the color value (pixel intensity value) of the current pixel or a neighbor of it, with the corresponding value of the filter matrix. The center of the filter matrix has to be multiplied with the current pixel, the other elements of the filter matrix with corresponding neighbor pixels.

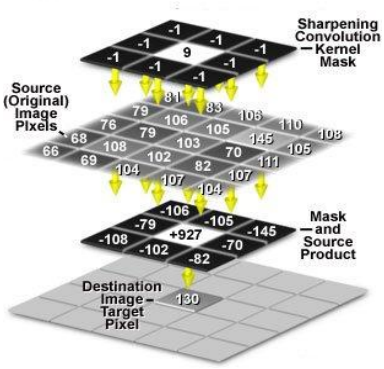


Figure 4.1 - An example of Convolution Kernel Process by using a Sharpening Convolution Mask

This operation where the sum of products of elements from two 2D functions is taken and one of the two functions moves over every element of the other function, is called *Convolution*, or *Correlation.* The difference between Convolution and Correlation is that for Convolution the filter matrix has to be mirrored, but usually it is symmetrical anyway so there is no difference. The filters with convolution are relatively simple. More complex filters, that can use more fancy functions, exist as well, and can do much more complex

things. The 2D convolution operation requires a 4-double loop, so it is not extremely fast, unless a small filter matrix is used. Here are a few rules about the filter:

- Its size has to be uneven, so that it has a center, for example 3x3, 5x5 and 7x7.
- To get the resulting image to have the same brightness as the original, the sum of all elements of the filter should be 1.
- If the sum of the elements is larger than 1, the result will be a brighter image, and if it's smaller than 1, a darker image. If the sum is 0, the resulting image isn't necessarily completely black, but it might be very dark.

The image has finite dimensions, and for example if a value pixel on the left side must be calculated, there are no more pixels to the left of it while these are required for the convolution, it is possible either to use value 0 or wrap around to the other side of the image. The resulting pixel values after applying the filter can be negative or larger than 255 and if that happens it is possible to truncate them so that values smaller than 0 are made 0 and values larger than 255 are set to 255. For negative values, it is also possible to take the absolute value instead. In the Fourier Domain or Frequency Domain, the convolution operation becomes a multiplication instead, which is faster. In the Fourier Domain, much more powerful and bigger filters can be applied faster, especially if the Fast Fourier Transform is used [6].

4.2 Median Filter

The median filter is a non-linear digital filtering technique, often used as pre-processing step. Under certain conditions it allows isolated noise suppression without blurring sharp edges. Specifically, the median filter replaces a pixel by the median of all pixels in the neighborhood as follows. Being:

1) x the input image (OI)
2) y the resulting image (FI)
3) w the neighborhood centered around location (m, n) in the OI

The median filtered image's pixel's value at the position [m, n] is computed as

$$y[m,n] = \text{median} \{ x[i, j], (i, j) \in w \}$$

An example of 1D median filter follows: consider a [1x5] window sliding over a 1D array (either horizontal or vertical) of pixels. Assume that the five pixels currently inside the windows are:

The middle pixel with value 200 is an isolated out-of-range noise. The median of these five values can be found by sorting the values (in either ascending or descending order). The middle value is the median:

80 80 90 120 200

The original pixel value 200 is replaced by the median 90. The window of a 2D median filter can be of any central symmetric shape, a round disc, a square, a rectangle, or a cross. The pixel at the center will be replaced by the median of all pixel values inside the window [7].

4.3 The Local Linear Polynomial Filter

[M] A Local Linear Polynomial Filter performs a linear structures detection and location on a given image, taking advantages of knowledge of local properties such as pixel intensities gradient and curvature property. Other typical applications of a local approximation to a sampled signal are smoothing and differentiation. The coefficients of the local polynomials which the filter is based on can be interpreted as a convolution filters. It is a better approach than more simplistic approach to assessing such properties such as e.g. forward or backward differencing, which are generally sensitive to noise and therefore often not useful. A simple computational form of the filter coefficients is derived and presented in the following.

4.3.1 1D-Polinomial Local Linear Filter

Given the sampled signal *g(x)* with sampling density *Δx* which define a window around a given sampling point, the filtering purpose consist of the approximation of the sampled function within the given window with a polynomial of a given order *P*. Without loss of generality, the sampled values in the interval $x \in [-K\Delta x, K\Delta x]$ are now considered. In order to approximate the sampled signal with a polynomial of a given order *P*, it is possible to proceed as follows:

$$f_p(x) = \sum_{I=0}^{P} \theta_I x^I = \Theta_T Z(x) \tag{1.1}$$

where a vector notation is employed as follows

$$\Theta^T = [\Theta_0, \Theta_1, \ldots, \Theta_P] \qquad Z(x)\ T = [1, x, \ldots, x^P] \tag{1.2a, 1.2b}$$

The coefficients of the polynomial are estimated using the Least-Squares (LS) estimator, which minimizes the objective functional

$$S(\Theta) = \sum_{I=-K}^{K} (g(i \cdot \Delta_x) - f(i \cdot \Delta_x))^2 = \sum_{i=-K}^{K} (g(i \cdot \Delta_x) - \Theta Z(i \cdot \Delta_x))^2 \tag{1.3}$$

Differentiating the objective functional with respect to Θ and solving S(Θ)=0, the normal equation is obtained:

$$\widehat{\Theta} = \left(\sum_{i=-K}^{K} \mathrm{Z}(\mathrm{i} \cdot \Delta_{\mathrm{x}}) \mathrm{Z}(\mathrm{i} \cdot \Delta_{\mathrm{x}})^{T} \right)^{-1} \left(\sum_{i=-K}^{K} \mathrm{g}(\mathrm{i} \cdot \Delta_{\mathrm{x}}) \mathrm{Z}(\mathrm{i} \cdot \Delta_{\mathrm{x}}) \right) \tag{1.4}$$

In the following, the inverse coefficient matrix B will be employed. It is defined as

$$B = \left(\sum_{i=-K}^{K} (\mathrm{Z}(\mathrm{i} \cdot \Delta_{\mathrm{x}}) \mathrm{Z}(\mathrm{i} \cdot \Delta_{\mathrm{x}})^{T} \right)^{-1} = \begin{bmatrix} b_{00} & \cdots & b_{0P} \\ \vdots & \ddots & \vdots \\ b_{P1} & \cdots & b_{PP} \end{bmatrix} \tag{1.5}$$

Indices are chosen consistently with the natural powers in the approximating polynomial. They are also chosen incidentally consistently with the natural indices in C/C++ arrays. This should minimize the risk of indexation errors in the implementation. The following examples may come in handy in order to understand how the filter works.

Example 1

A second-order polynomial with support in a window of size *[-K;K]* is considered. For notational simplicity it is assumed that $\Delta x = 1$. Using the vector and matrix notation previously introduced:

$$\Theta^{\mathrm{T}} = \quad \theta_0, \theta_I, \ldots, \theta_P \qquad\qquad \mathrm{Z}(\mathrm{x})^{\mathrm{T}} \quad = \quad [\mathbf{1}, \mathbf{x}, \mathbf{x^2}]$$

and

$$\mathrm{B} = \left(\sum_{i=-K}^{K} (\mathrm{Z}(\mathrm{i}) \mathrm{Z}(\mathrm{i})^{T} \right)^{-1} = \left(\sum_{i=-K}^{K} \begin{bmatrix} 1 & i & i^2 \\ i & i^2 & i^3 \\ i^2 & i^3 & i^4 \end{bmatrix} \right)^{-1}$$

it is possible to write the inverse matrix B as

$$B = \begin{bmatrix} 2K+1 & 0 & \frac{1}{3}K(K+1)(2K+1) \\ 0 & \frac{1}{3}K(K+1)(2K+1) & 0 \\ \frac{1}{3}K(K+1)(2K+1) & 0 & \frac{1}{15}K(K+1)(2K+1)(3K+3K^2-1) \end{bmatrix}$$

$$= \begin{bmatrix} 3\dfrac{3K^2+3K-1}{(2K+3)(2K+1)(2K-1)} & 0 & \dfrac{-15}{(2K+3)(2K+1)(2K-1)} \\ 0 & \dfrac{3}{K(2K+1)(K+1)} & 0 \\ \dfrac{-15}{(2K+3)(2K+1)(2K-1)} & 0 & \dfrac{45}{(2K+3)(2K+1)(2K-1)(K+1)} \end{bmatrix}$$

which can be used along with equation (1.4) in order to obtain the following estimators for the coefficients of the polynomial:

$$\theta_2 = \sum_{\mathrm{I}=-\mathrm{K}}^{\mathrm{K}} \frac{15(3i^2 - K(K+1))}{K(K+1)(2K+1)(2K-1)(2K+3)}\, g(i)$$

$$\theta_1 = \sum_{\mathrm{I}=-\mathrm{K}}^{\mathrm{K}} \frac{3i}{K(K+1)(2K+1))}\, g(i)$$

$$\theta_0 = \sum_{\mathrm{I}=-\mathrm{K}}^{\mathrm{K}} 3\,\frac{3K^2+3K-1-5i^2}{(2K+1)(2K-1)(2K+3)}\, g(i)$$

It is possible to notice that each of estimators is actually a convolution mask. This result can be generalized as follows. Given:

- the signal *g(x)*
- a convolution filter with support in the interval *[-K; K]*, defined as

$$\Phi = \langle \phi_{-K}, \phi_{-K+1}, \cdots, \phi_0, \cdots, \phi_{K-1}, \phi_K \rangle \qquad (1.6)$$

it is convolved with the signal as

$$G_{\Phi}(X) = \Phi \otimes \mathrm{g(x)} = \sum_{\mathrm{i}=-\mathrm{K}}^{\mathrm{K}} \mathrm{g(x + i \cdot \Delta_x)} \cdot \phi_i \qquad (1.7)$$

Each coefficient in the polynomial can be expressed as a convolution mask with i=0, 1,..,P, containing 2K+1 filter coefficients as shown in Equation (1.6). Each filter coefficient is computed as

$$\Theta_{\mathrm{m}}(\mathrm{j}) = \left(\frac{1}{\Delta_x}\right)^{\mathrm{m}} \sum_{\mathrm{i}=0}^{\mathrm{P}} \mathrm{B_{mi}(j)^i} = \mathrm{B_m Z(j)} \quad \text{where} \quad m=0,...,P \quad \text{and} \quad j=-K,...,K \qquad (1.8)$$

The i^{th} order derivative of the signal, in the center of the window in which the signal is approximated, is computed using

$$\frac{d^i}{dx^i} f(x) = w(i) \cdot \theta_i(x) \quad \text{and} \quad w(i) = \begin{cases} i \ for \ i > 0 \\ 1 \ for \ i = 0 \end{cases} \tag{1.9}$$

θi is the result of applying the filter kernel Θ_i to the signal in the window $x \in [-K;K]$:

$$\theta_1(x) = \Theta_i \otimes g(x) = \sum_{i=-K}^{K} g(x + i \cdot \Delta_x) \cdot \phi_i \tag{1.10}$$

The zero-order derivative will yield the predicted value in the center of the window, which corresponds to smoothing of the signal.

Example 2

Assume a polynomial of order P=4 with K=3 and Δx=1. Using Equation (1.5) and Equation (1.8), the following filter coefficients are found.

Order	j=-3	-2	-1	0	1	2	3
0	0.02164	-0.1298	0.32467	0.56709	0.32467	-0.1298	0.02164
1	0.08730	-0.26587	-0.2301	0	0.23010	0.26587	-0.08730
2	-0.04924	0.25378	-0.07196	-0.26515	-0.07196	0.25378	-0.04924
3	-0.02777	0.02777	0.02777	0	-0.02777	-0.02777	0.02777
4	0.01136	-0.02651	0.00378	0.02272	0.00378	-0.02651	0.01136

The convolution filter corresponding to order zero is a smoothing filter, while the first- and second order derivatives are gradient and curvature (Laplacian) filter masks. This is easily recognized considering the shape of these convolution masks (see Fig. 4.2)

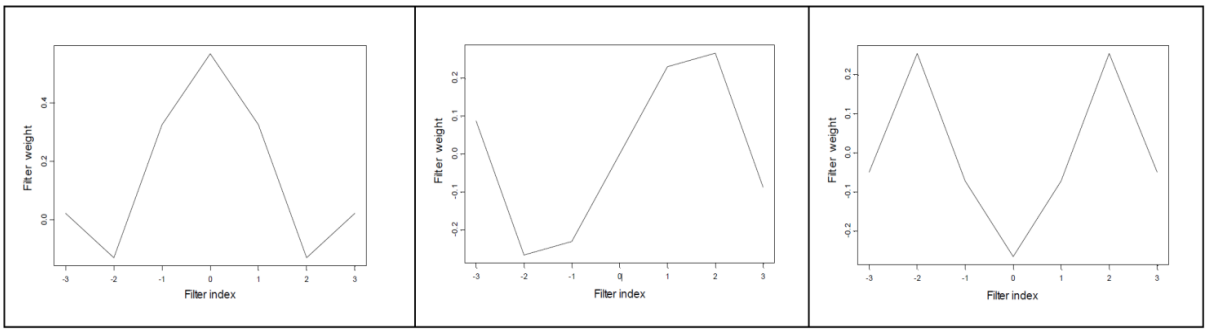


Figure 4.2 - The convolution masks corresponding to the are shown above for a polynomial of order four. The small window size (K=3) gives the convolution masks a somewhat coarse appearance.

Example 3

A sinusoid embedded in Gaussian noise was sampled in the interval [0, 2π] with $\Delta_x = \pi/50$

By Using a polynomial of order P=2 and K=10 it is possible to obtain a smoothed version of the sampled noise as well as the first order derivative of the noisy signal, as shown in Figure 2.

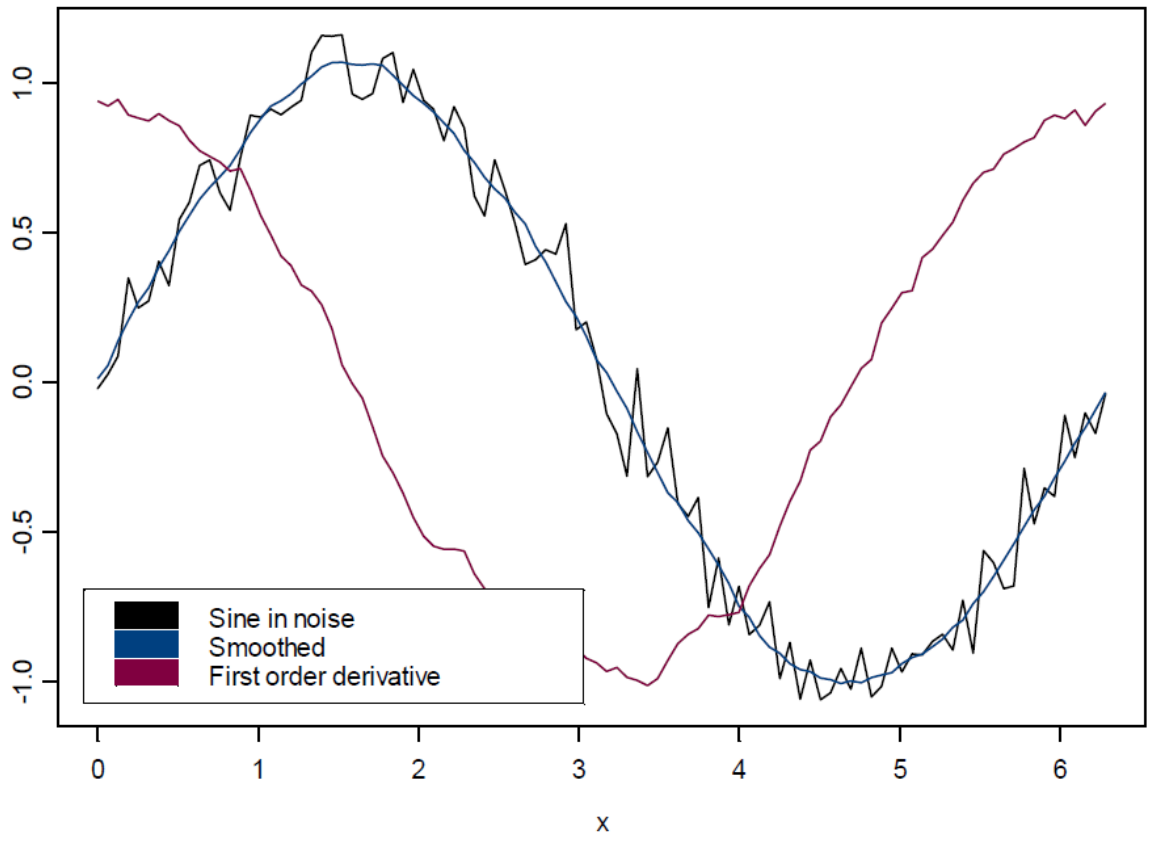


Figure 4.3 – A sinusoid embedded in Gaussian white noise is sampled equidistantly in the interval [0,2π]. An approximating polynomial of order 2 is used to smooth the sampled signal, employing a window size of K=10. The same approximating polynomial is used for obtaining the first order derivative.

4.3.2 Extension to 2D

The results presented above can be easily extended to 2-D. Assuming that the spatial signal (e.g. an image), $g(x,y)$ sampled with s sampling densities of Δx and Δy respectively, it is possible to approximate the sampled signal with a smooth function in (e.g., for simplicity) a square window. There may be good reasons for defining a spatial polynomial of different order in x and y, where notational simplicity is would be one such good reason. For the particular application considered here, however, it is considered the special case of a P'th order spatial polynomial defined as below:

$$f(x,y) = \sum_{K=0}^{P} \sum_{I=0}^{P-k} \Theta_{k,l} x^k y^l = \Theta^T Z(x,y)^T \qquad (2.1)$$

With the following vector notation:

$$\Theta^{T} = [\Theta_{k,I}, k = 0, \dots, P \quad I = 0, \dots, P - k]$$

$$Z(x,y)^{T} = [x^{k}y^{l}, k = 0, \dots, P \quad I = 0, \dots, P - k] \qquad \text{(2.2 and 2.3)}$$

The Least-Squares estimator structure is fairly similar to the one found in the 1D case:

$$\hat{\Theta} = \left(\sum_{I=-K}^{k} \sum_{j=-K}^{K} Z(i \cdot \Delta_x, j \cdot \Delta_y)\, Z(i \cdot \Delta_x, j \cdot \Delta_y)^{T} \right)^{-1} \left(\sum_{j=-K}^{K} {}^{g}(i \cdot \Delta_x, j \cdot \Delta_y)\, Z(i \cdot \Delta_x, j \cdot \Delta_y) \right) \qquad (2.4)$$

Where the inverse coefficient matrix can be found as

$$B = \left(\sum_{j=-K}^{K} Z(i \cdot \Delta_x, j \cdot \Delta_y)\, Z(i \cdot \Delta_x, j \cdot \Delta_y)^{T} \right)^{-1} = \begin{bmatrix} b_{00} & \cdots & b_{0P} \\ \vdots & \ddots & \vdots \\ b_{P1} & \cdots & b_{PP} \end{bmatrix} \qquad (2.5)$$

The coefficients can also be interpreted as a convolution mask with support on *[-K;K]* x *[-K;K]* where the mask features the following form:

$$\Phi = \begin{bmatrix} \phi_{-K,-K} & \cdots & \phi_{-K,K} \\ \cdots & \cdots & \cdots \\ \phi_{K,-K} & \cdots & \phi_{K,K} \end{bmatrix} \qquad (2.6)$$

The convolution sum is computed as

$$G_{\Phi}(x,y) = \Phi \otimes g(x,y) = \sum_{i=-K}^{K} \sum_{j=-K}^{K} g(x + i \cdot \Delta_x, y + j \cdot \Delta_y) \cdot \phi_{i,j} \qquad (2.7)$$

As well as in the 1D case, each coefficient in the approximation polynomial can be interpreted as a convolution mask. The individual filter weights is computed as

$$\Theta_{m,n}(k,I) = \left(\frac{1}{\Delta_x}\right)^{m} \left(\frac{1}{\Delta_y}\right)^{n} \sum_{i=-0}^{P} \sum_{j=0}^{P-i} B_{\frac{(2P+3-m)m+2n}{2} \cdot \frac{(2P+3-i)i+2j}{2}} (k)^{i} (I)^{j}$$

$$= \left(\frac{1}{\Delta_x}\right)^{m} \left(\frac{1}{\Delta_y}\right)^{n} B_{\frac{(2P+3-m)m+2n}{2}} \cdot Z(k,I) \qquad (2.8)$$

Where indices range as below:

m=0,…,P k=-K,…,K and n=0,…,P-m l=-K,…,K. (2.9)

The indices structure associated with summation over the inverse coefficient matrix owes to the order in which the individual polynomial orders are introduced into the equation, as seen in (2.1). The value of a given coefficient is now computed as

$$\theta_{i,j}(x,y) = \Theta_{ij} \otimes g(x,y) = \sum_{k=-K}^{K} \sum_{l=-K}^{K} \Theta_{ij}(k,L) \cdot g(x-k, y-l) \qquad (2.10)$$

The convolution mask equations corresponding to each of the coefficients in the polynomial are seen to be quite similar to those obtained for the 1D case. Furthermore, as in the 1D case, it is possible to apply this local image approximation for extracting information about the local image properties. The i,j^{th} order derivative is obtained using

$$\frac{\partial y^{ij}}{\partial^i x \ \partial^j y} f(x,y) = w(i) \cdot w(j) \ \theta_{i,j}(x,y) \qquad (2.11)$$

where $$w(i) = \begin{cases} i \ for & i > 0 \\ 1 \ for & i = 0 \end{cases}$$ □

4.3.3 Gradient Information

For many applications it is of importance to have knowledge about both the local gradient magnitude and gradient orientation; the gradient approximations are dependent upon the order of the approximation polynomial as well as the size of the window in which the image function in approximated. For an increasing value of the polynomial orders it is possible to notice that the filter is still more responsive to high frequency content in the image and thereby also to noise (see Fig 4.4). On the other hand, the local image properties are also modeled far better by a higher-order polynomial. The choice of the polynomial order is a trade-off, and must be chosen according to the specific application. The result of applying the gradient convolution masks to a retinal fundus image is shown below.

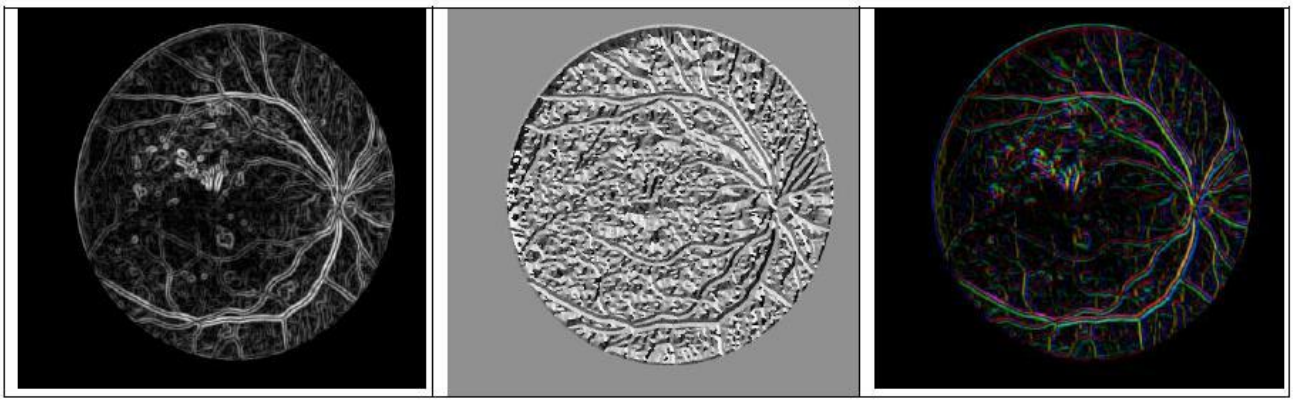


Figure 4.4 - This figure illustrates the result of applying the gradient convolution masks to a fundus image. The leftmost image shows the magnitude, the middle the orientation. The rightmost image shows a magnitude/orientation image, where a circular color-coding has been chosen.

4.3.4 Local Image Curvature

Quite often also the Laplacian is used as measure off local image curvature, but it is mainly used in order to obtain the Hessian matrix, which the following Eigen values as obtained from:

$$\begin{pmatrix}\lambda_1(x,y)\\ \lambda_2(x,y)\end{pmatrix} = \frac{1}{2}\begin{pmatrix}\theta_{20}(x,y)+\theta_2(x,y)+\sqrt{(\theta_{20}(x,y)-\theta_2(x,y))^2+4\theta_{11}(x,y)^2}\\ \theta_{20}(x,y)+\theta_2(x,y)-\sqrt{(\theta_{20}(x,y)-\theta_2(x,y))^2+4\theta_{11}(x,y)^2}\end{pmatrix} \tag{2.12}$$

By using of the Eigen values, it is possible to use a non-linear combination of the directional derivates of the Laplacian as

$$\lambda_1(x,y)+\lambda_2(x,y)=\theta_{20}(x,y)+\theta_{02}(x,y)$$

A simple interpretation of the first *eigen vector* is the maximum local curvature, whereas the second *eigen vector* is the curvature in a direction perpendicular to the first *eigen vector*.

It is possible to notice (Fig. 4.4) that the first eigen filter enhances the vascular structure, whereas the second eigen filter has a tendency towards enhancing the (dark) lesions in the image.

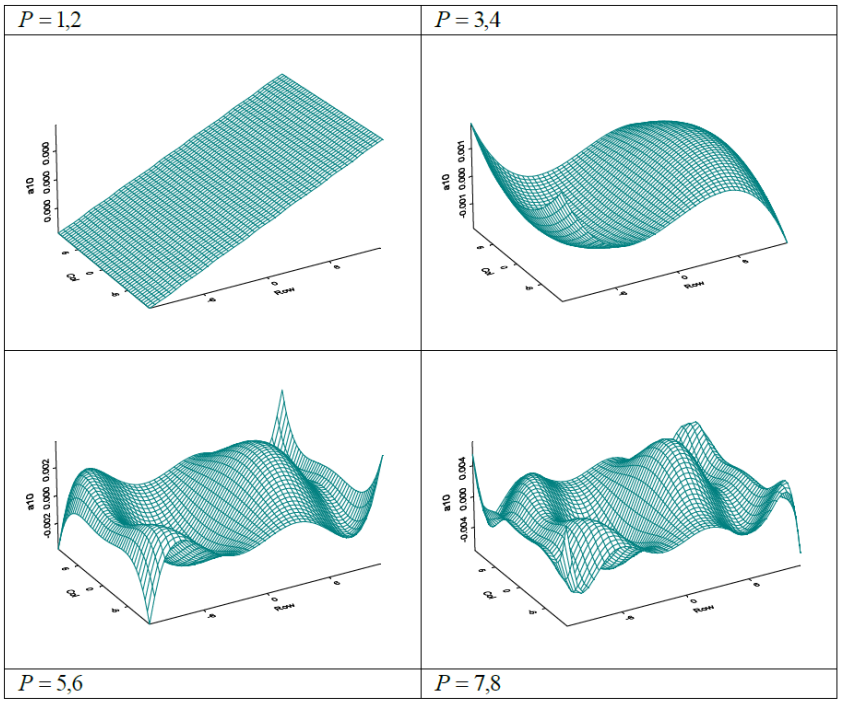


Figure 4. 5 - The convolution masks corresponding to with i=0 and j=1 or (j=1 and i=0, these masks only differ a 90 degrees rotation). The masks shown here are computed for polynomial orders of $P = 1,\ldots,8$.

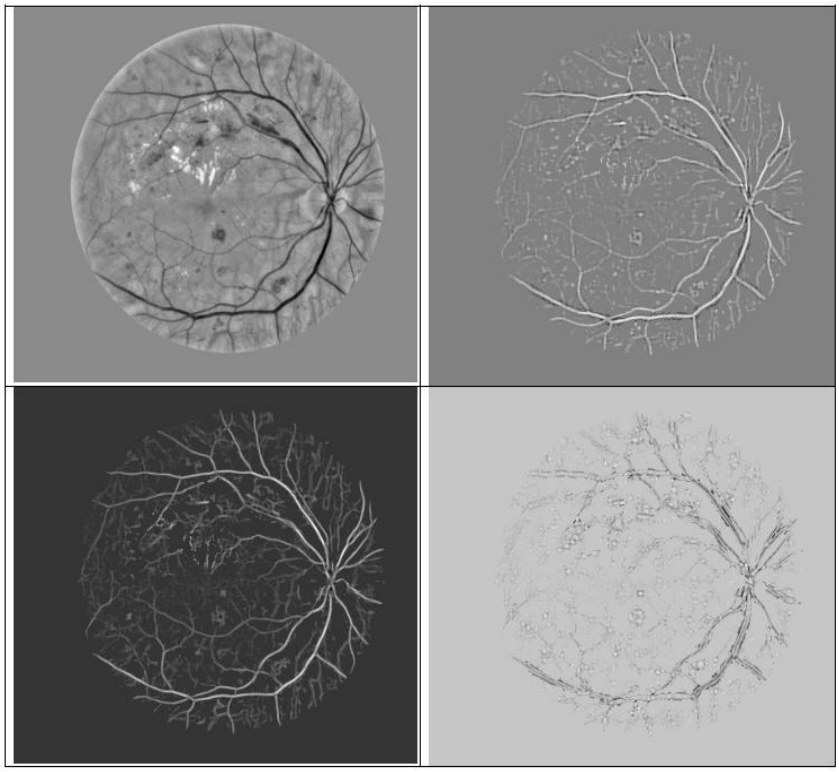


Figure 4.6 - The green band of a color fundus image (upper left) is filtered using the standard Laplacian and the two eigen filters described in Equation (2.12). The lower left image is the first eigen filter, whereas the lower right is the second eigen filter.

4.4 Filtering methods used in this project

In order to locate and extract the three desired components from the original image, the filtering methods with the following parameters values have been chosen. The reason for using these filters in particular are due to their previously described capabilities.

Image components to be extracted	Feature's shapes properties - OI -	Filtering method	Filter's parameter values
Neurite Branches	Linear Structures (variable thickness and continuity)	Polynomial Local Linear Filter (PLLF)	Kernel Size = 15 Filter Order = 3
Neuron Cell Bodies	Circular Structures (Area ≥ 80 pixels)	Median Filter (MF)	Horizontal Width = 7 Vertical Width = 7
Small Beads	Circular Structures (Area < 80 pixels)	(MF) and (PLLF)	See values above.

Table 4.1 - Filters applied on the image in order to extract the listed components

For instance, by inputting the following test images (see Fig.4.7a) , it is possible to obtain the following outputted results from each of the used filters (see Fig. 4.7b, 4.7c)

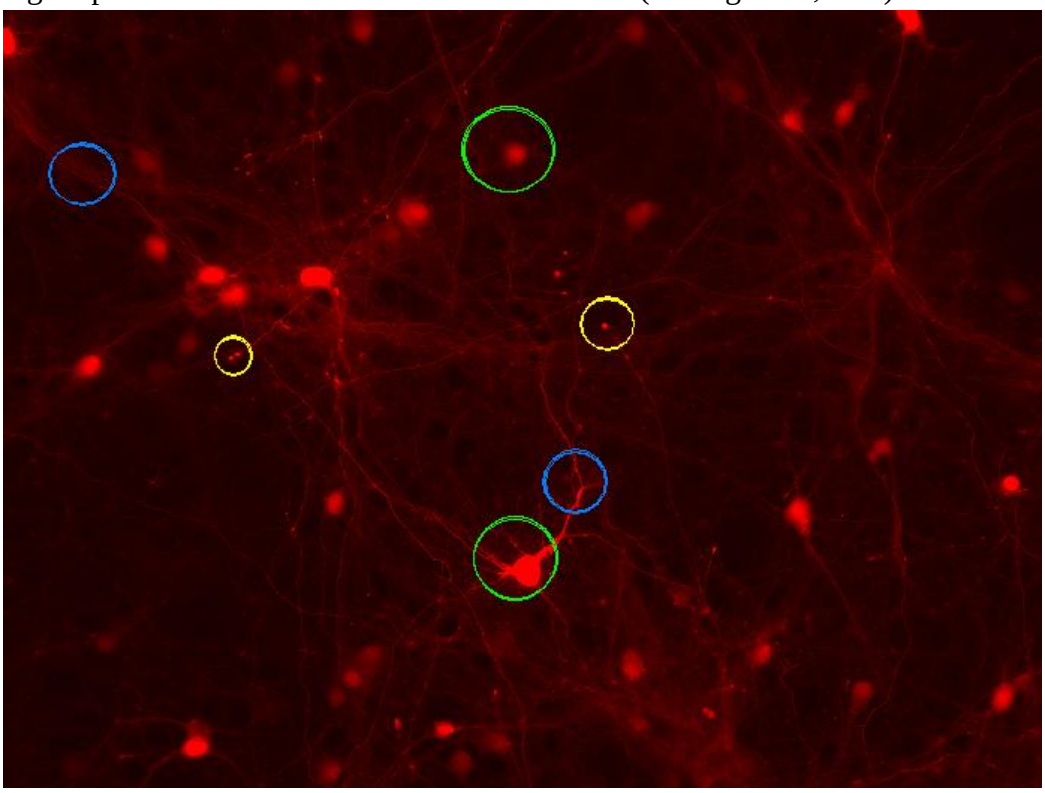


Figure 4.7a – Original Image, which the filters must to work on

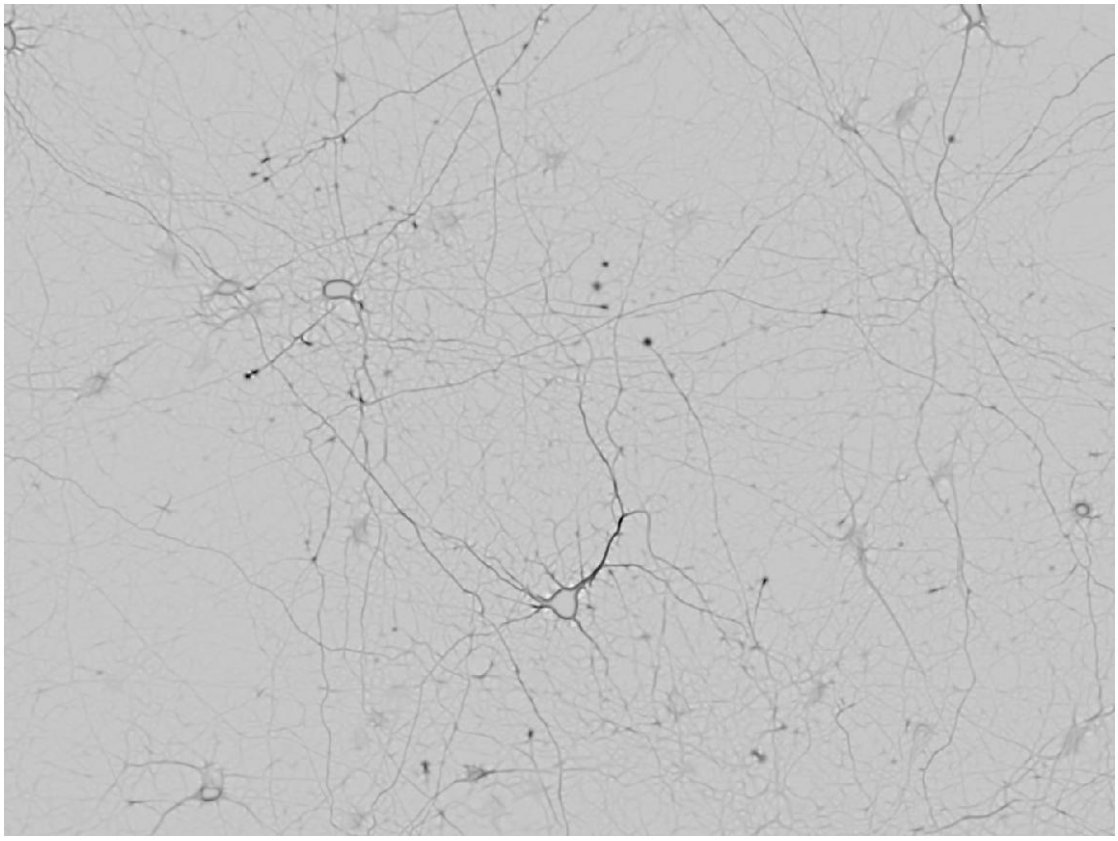


Figure 4.7b – PLLF filtered Image: linear structures' pixels are darker than the non-linear ones

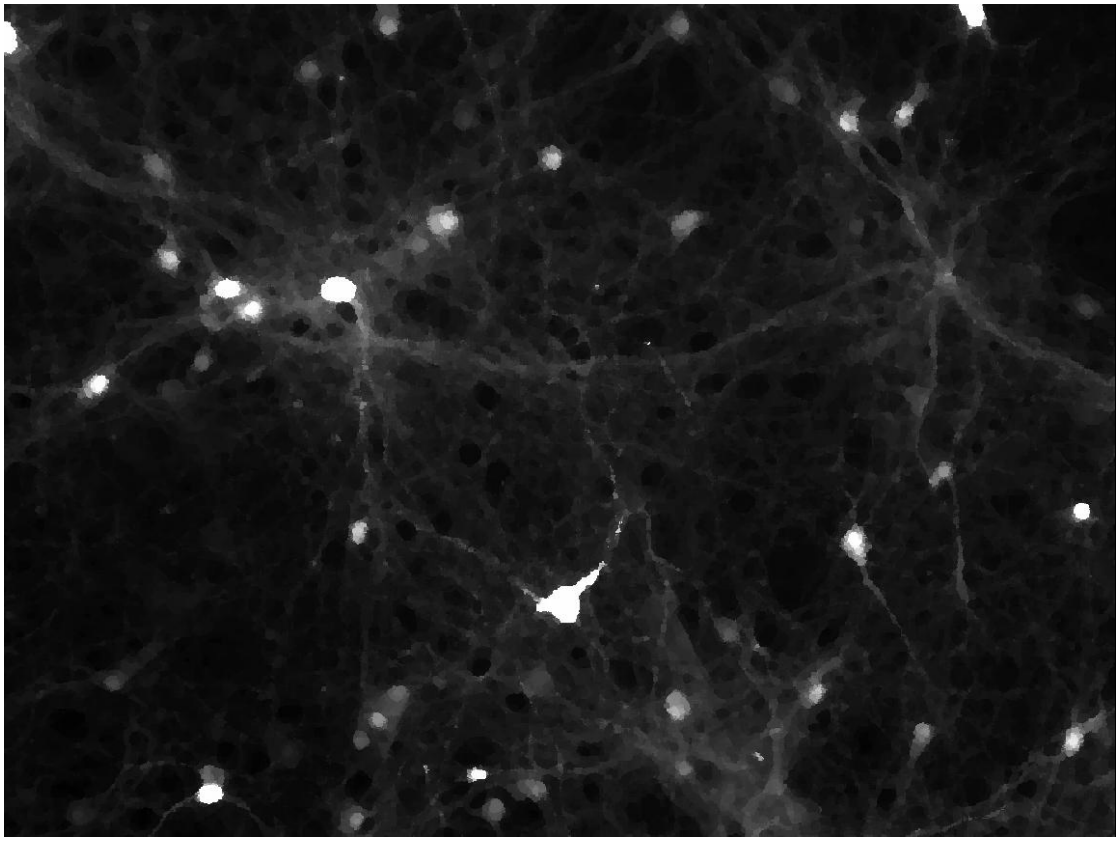


Figure 4.7c – MF filtered Image: areas featuring a large amount of bright pixels (as in the case of neuron cell bodies, in the original image) take the highest values (in the filtered image)

Referring to Fig. 4.7a, the filtering operations aims to:

- locate and extract the linear structures (see the green circled objects) both the thickest ones (the centermost one, connected to a neuron cell body) and the thinnest ones (leftmost and right most green circled neurites) which brightness is weaker.

- Locate the neuron cell bodies (see the blue circled objects) both the brightest ones (see centermost one, connected to the thick neurite) and the less bright ones (uppermost)

- Locate the small beads; some of them are detected from the Median Filter (yellow circled ones), whereas almost all of them (e.g. the white circled ones) are detected from the Polynomial Local Linear Filter, too.

The application of each filter to the input images produces as output a 1 band image, which enhancing the desired image's features and structures and whose pixel intensity values range as showed in the following table:

Filtered Image	**Image's pixel intensity (P. I.)**
PLLF	$-20 \leq$ **P.I.** ≤ 20 (average values)
MF	$0 \leq$ **P.I.** ≤ 255

Table 4.2 - Pixel intensity values of the image outputted from the used filters

It is possible to notice that in the PLLF filtered image, the features that seem more linear structures, consist of darker pixels. The purpose of the PLLF is to locate linear structures but it is also possible to notice that it regards any edge of an object (pixels of the borders) as a kind of linear structure. Therefore in every area of the original image where a "fast" pixel intensity variation is present - which involves an important gradient variation, for a detailed description of the mathematical content of the topic, see (par. 4.3.3) - it is possible to get a dark structure outputted from the filtered image. Thus most of the beads and the neuron cell bodies' outer border have been regarded as linear components and their structure on the filtered image is also fairly dark. Moreover it is possible to notice that the MF filtered image does feature white colored areas where wide areas of high intensity pixel values are present, in the original image. Thus the MF outputted structures of either neuron cell bodies, neurites and the biggest beads are fairly white and the lower the filter's horizontal and vertical width parameter values is (which define the window width used by the filter to compute the value on the outputted image), the more the filtered images looks like the original one. On the contrary, high values of those parameters will make the smallest components to be regarded as noise and they no longer show up on the outputted image. The pre-processing task of the Image analysis protocol is now carried out, and the extracted features composing the filtered images are ready to be classified and enhanced.

Chapter 5

Image Classification

Image classification - or segmentation - is the second step in this image analysis protocol. Classification partitions an image into non overlapping regions and allows the pattern recognition.

5.1 Classification – a brief overview

With regard to a digital image, a region is defined as a homogeneous group of connected pixels which respect a chosen, common homogeneity which can either be color, gray levels, motion, texture, etc. According to [8] "*the image segmentation problem is basically one of the psychophysical perception, and therefore not susceptible to a purely analytical solution*". Basically, classification methods consist of decision-theoretic approaches to perform an identification of images or parts of images. The Image classification purpose is to analyze the numerical properties of various image features and to organize data into classes. Each classification algorithm is based on two general assumptions:

- the image in question shows one or more different features (*e.g.*, particular geometric structures or spectral regions)
- each of these features belongs to one of several distinct and exclusive classes.

Classification methods are then subdivides into two main types:

2) ***Supervised classification****:* the classes may be specified *a priori* by the analyst;

3) ***Unsupervised classification*** **(Clustering):** features are automatically clustered into sets of prototype classes, where the analyst only has to specify the number of desired categories.

Classification and segmentation objectives are closely related, as the former is another form of component labeling that can result in segmentation of various features in a scene.
Classification algorithms typically employ two stages of processing:

1)Training: it is the initial stage where characteristic properties of typical image features are isolated and, in function of these, a unique description of each class (*i.e. training class*) is created; in this stage, the algorithm "learn" how to classify images.

2) T*esting*: the classes features that have been created in the training stage are used to classify image features

In supervised classification, *statistical* processes (*which are* based on an *a priori* knowledge of the probability distribution functions) or *distribution-free* processes can be used to identify the class descriptors. The Unsupervised classification instead, depends on *clustering* algorithms to automatically segment the training data into prototype classes. In either case, the motivating criteria to create the training classes are three. Training classes must be

- *Independent of each other:* if there is a change in the description of a training class, there should not be changes the value of the other one;

- *Discriminatory*: different image features must have significant different descriptions;

- *Reliable:* all the image features within a training group must share the same unequivocal and final descriptions of that class.

5.2 The K-means classification algorithm

The k-means classification algorithm is a clustering algorithm that is widely used in statistics and machine learning to partition n objects into *K* clusters, where $K < N$. In images analysis, the k-means classification algorithm is an iterative technique used to partition image's components into *K* clusters. The difference is typically based on pixel color, intensity or texture and location, or a weighted combination of those factors present in the image, thus the feature selection is based on the number, color and shape of objects. *K* can be selected manually, by a heuristic or randomly. The k-means classification algorithm for partitioning (based on the mean value of the objects in the cluster) is based on the following properties:

- Input: number of clusters *K* and a database containing *N* objects.

- Output: a set of *K* clusters that minimizes the squared-error criterion.

- Working principle:
  1. Arbitrary choice of *K* objects as the initial cluster centers;
  2. Repeat: 2.a (re)assignment of each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
     2.b Updating of the cluster means, (i.e. calculate the mean value of the object for each cluster);
  3. Until: convergence is attained (e.g. no pixels change clusters or cluster average value has varied less than a given value).

This algorithm is guaranteed to converge and it is suitable to cluster large amounts of data. The quality of the solution depends on the initial set of clusters and the value of *K* and it may be, in practice, much poorer than the global optimum. Selection of distance measure is an important step in clustering as well. Distance measure determines the similarity of two elements. It greatly influences the shape of the clusters, as some elements may be close to one another according to one distance and further away according to another. An inappropriate choice of *K* may yield poor results. The algorithm also assumes that the variance is an appropriate measure of cluster scatter.

5.3 Bayesian Classification

The Bayesian classification is a statistically based classification method based on the probabilities that a given set of measurement come from objects belonging to certain class. The decision making are based on probability theory and the principle of choosing the most probable option [9]. The practical use of Bayes' theorem is to turn probabilities that can be estimated from a training set into probabilities required for classification. Assume that there is a classification task to classify feature vectors to different classes. A feature vector is denoted as $x[x_1, x_2, \ldots, x_D]^T$ where D is the dimension of a vector. The probability that a feature vector belongs to a class is $P(w_i|x)$ and it is often referred to as a posterior probability. Bayes' theorem allows the calculation of the posterior probability and it is stated formally as:

$$p(w_i\,|x) = \frac{p(x|\,w_i\,)\;P(w_i\,)}{p(x)}$$

where

$$p(x)\;=\;\sum_{i=1}^{n} p(x|\,w_i\,)p(w_i)\,.$$

The posterior probability $p(w_i|x)$ is the one needed for classification. In terms of pixel intensity value, it is the probability of a given pixel belonging to the class after measuring the feature vector *x*. The prior probability $p(w_i)$ is the probability of the pixel being component of , given no information about its feature values, i.e. before a measurement is made and $P(x|w_i)$ is the probability density function of class w_i in the feature space. Assume that there is a classification task to classify an image into *foreground* w_f, and the background w_b, classes, the segmentation of image can be considered as the classification process of each pixel into 'background' or 'foreground' classes based on pixel's color value. Bayes theorem gave the probability for pixel to belong to the foreground on its color basis as follows:

$$p(w_f\,|x) = \frac{p(x|\,w_f\,)\;p(w_f\,)}{p(x)}$$

with x being the (H, S, L) color component feature vector; $p(w_f|x)$ the posterior probability for the foreground or the probability for a pixel having the color x to belong to foreground, $p(x|w_f)$ is the probability density function or probability for a pixel belonging to class 'foreground' to have the color x, $p(w_f)$ the foreground color class a prior probability; $p(x)$ the probability to observe the color x (the two classes blended). The complementary probability is given by:

$$p(w_b|x) = \frac{p(x|\,w_b)\,p(w_b\,)}{p(x)}$$

where $p(w_b|x)$ is the posterior probability for the background or the probability for a pixel having the color x to belong to a background and $p(w_b)$ is the background prior probability. The classification of an image can be done according to posterior probability, where, for each pixel, foreground posterior probability and background posterior probability are calculated, and that the pixel belongs to class which has the higher posterior probability. A posterior is a between-class measure for a single observation, but probability density function values can be used as a within-class measurement to select the best representative of class. Instead of estimating posterior probability, because $p(x)$ has the same value for both classes and by considering equal prior probability, classification can be done based on probability density function.

5.4 Thresholding Classification

Thresholding is the simplest method to perform image segmentation. From a grayscale image, thresholding and slicing can generate a binary image by setting all pixel values in a given range, $[T_{min}; T_{max}]$, to 1 and all other pixel values to 0. T_{min} and T_{max} are called threshold values.

$$v_{out} = \begin{cases} 0 \quad \text{if} \quad v_{in} < T_{min} \quad \text{or} \quad v_{in} > T_{max} \\ 1 \text{ otherwise} \end{cases}$$

When only one of the threshold values in used this operation is called thresholding. With both thresholding values used we talk about slicing (or double thresholding) [10].

Adaptative Thresholding is when different threshold values are used for different regions in the image.

Multiple Thresholding is when pixel values of the classified image can be set to one of multiple grey level values.

5.5 Classification methods used in this project

In order to classify the three desired features in the original image (neuron cell bodies, neurites and small beads), the Crossed Multiple Thresholding classification method was used, based on the pixel intensity of both original image's main band and filtered image, and for each of the different features classification processes. Threshold values were set by a visual analysis of samples of neurites, neuron cell bodies and small beads on both images, determining which values were the extremes (lower bound, upper bound) to separate them into classes. The following classification methods have been tried, as well, with the listed variants:

1) Bayesian Classification (Supervised, based on stored parameters for features labeling)
    (a) quadratic method
    (b) linear method

2) K-means classification
    (a) Supervised (based on stored parameters for features labeling)
    (b) Unsupervised (only the desired final number of classes was specified)

The thresholding method was chosen as best-result-giving method because:

- regarded as most suitable for frequent parameter values variations due to different kinds of input images,
- it worked in Un-supervised mode (label values parameters do not have to be specified on the image, unlike Bayesian method or supervised K-means method)
- K-means unsupervised method did not give better result in linear structures classification

An example containing resulting images, and a brief description of that choice are given in sub-chapter 5.5.3

5.5.1 Classification of the linear structures

With regard to the classification of the linear structures, the classification algorithm is applied to the PLLF filtered image. It was agreed with Dr. Steiner to divide the linear features into three main classes, depending on properties of their shape and pixel intensity. In the PLLF filtered image each pixel has been labeled with one of the three main labels, which were chosen according to the following criterion:

1) On the original image:

Label Color	Pixel intensity (PI)	Structure Continuity	Structure Shape
RED	P.I. ≥ 43	Yes (always)	Robust, thick; Good view
BLUE	21 ≤ P.I. ≤ 220	Yes (mostly)	Thin; Blurry view
GREEN	P.I. ≥ 21	No	Parts of neurites

2) On the PLLF filtered image:

Label Color	Pixel intensity (PI)	Structure Continuity
RED	P.I. ≤ - 1.20	Yes (always)
BLUE	-3.90 ≤ P.I. ≤ - 0.51	Yes (mostly)
GREEN	-0.51 ≤ P.I. ≤ - 0.12	No

Table 5.1A, 5.1B – Threshold values used to classify the linear structures on both original and filtered image

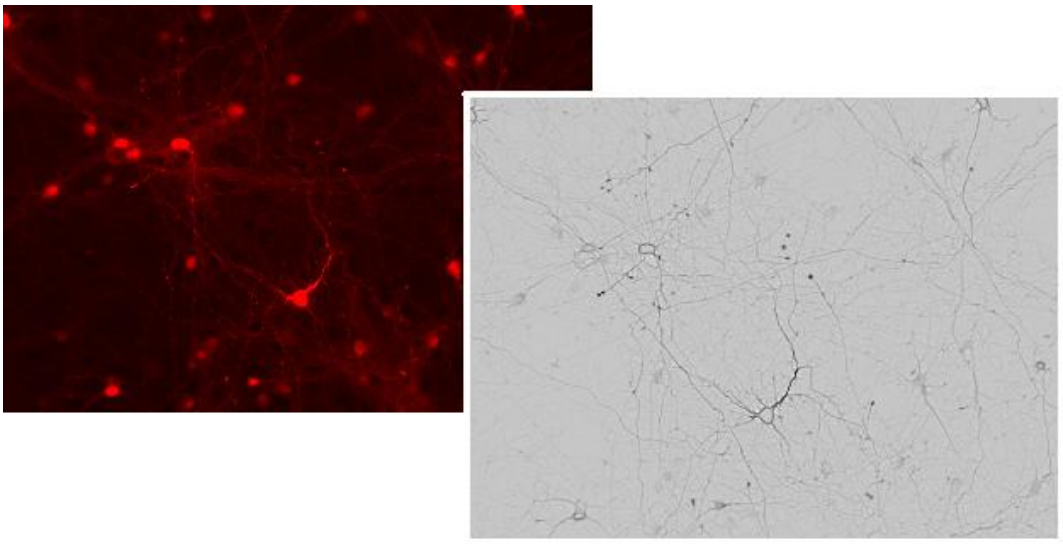


Figure 5.1 – The original image has been filtered in order to locate the linear structures; the filtered image is the input image in which the features are going to be classified according to their pixel intensity

In the following, the feature belonging to these sub-classes - into which linear structures were divided - will be re-named as:

Label Color	Sub-Class name
RED	Class 1 neurites
BLUE	Class 2 neurites
GREEN	Class 3 neurites

Table 5.2A – Sub-division of neurites in three sub-classes

The threshold values of Table 5.1 have been chosen by means of a scrupulous analysis of pixel intensities of the features in both images. Thus most of the linear structures that were regarded as belonging to a given class, were actually classified in the right class. Moreover, the boundaries are not mutually exclusive (e.g. some of the values of pixel intensity which would get a pixel the red label, are also included in the blue labeling range of values). Despite that, the conditions which rule the classification of a pixel in an inferior class - e.g. class 2 - (as it will be explained in a more detailed way in Chapter 8) does not allow a pixel that has just been labeled with another label – e.g. class 1- to be labeled again, within the classification process. Summarizing, if a pixel meets the conditions for multiple classes it always gets assigned to the class with the lowest number and hence gets the corresponding label. Furthermore, it is not possible that, for instance, a pixel's value in the OI is in class 1 (and not class 2) and its FI value is in class 2 (and not class 1), because the boundaries have been designed so that this situation cannot occur.

Giving as input such obtained filtered image to the classification algorithm, the following classified image is obtained as output (see Fig. 5.2).

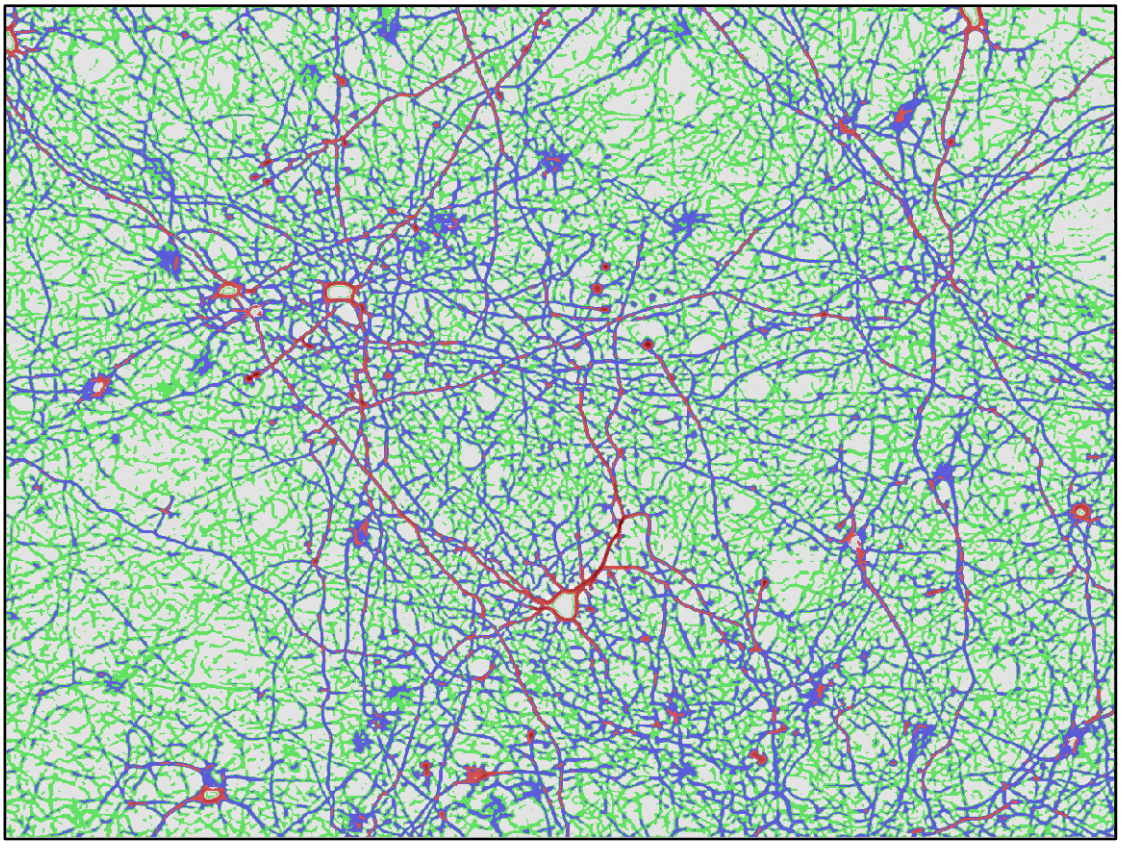


Figure 5.2 – The Classified image is a 4 color labeled image; background pixel have been labeled with white

Here follows a brief comment of the obtained image, showing which good result were achieved (√), what represent unexpected results or misclassifications (!) and what should be improved by the Post-Processing step to get a better final classification result (⇒)

1) √ Neurites which were supposed to be regarded as Class 1 neurites, (the thickest ones) were actually classified in the correct class, many Class 2 neurites, which structure was thinner

! their structure looks thicker than their original structure (on the original image)

! their structure is not always completely connected (see black-circled neurite, Fig. 5.3)

⇒ Neurite shapes will have to be morphologically modified, in order for them to look more congruent with the OI original neurite structures.¨

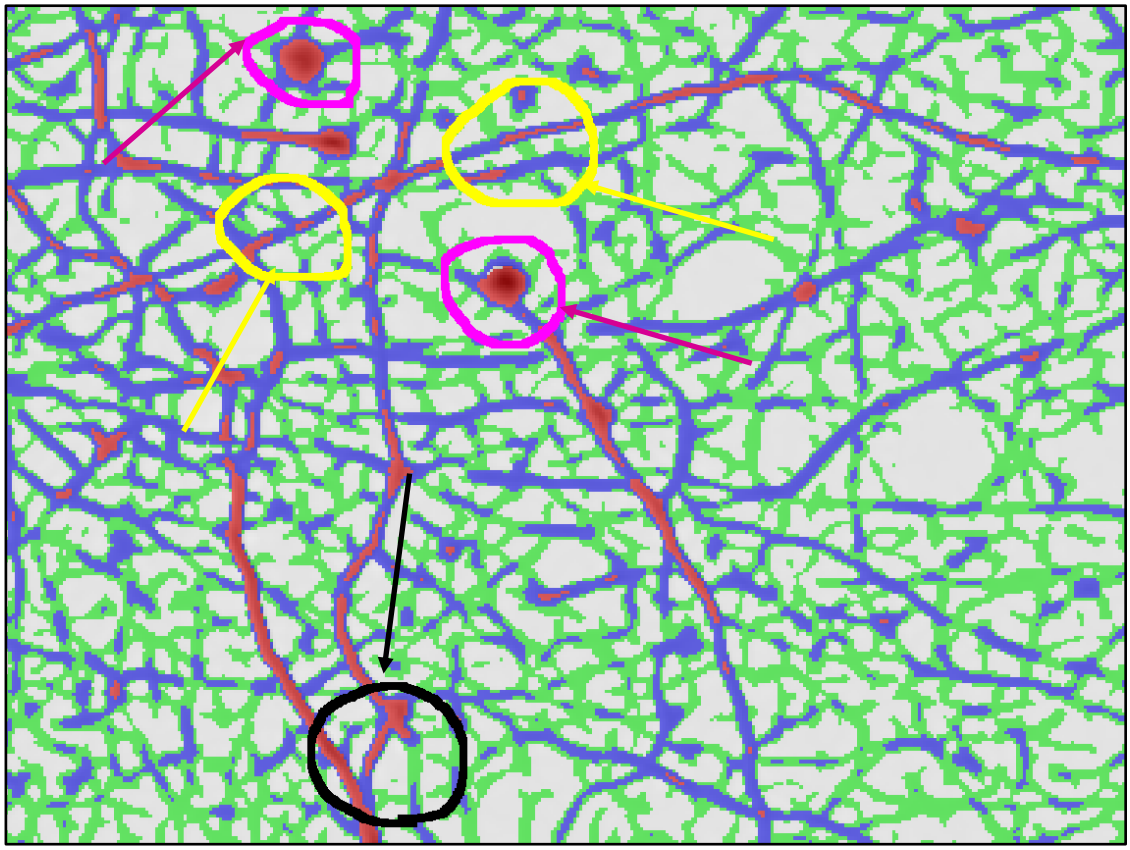


Figure 5.3 – A magnified view of the classified image and instances of misclassified areas

2) √ Many neurites which were supposed to be regarded as Class 2 neurites were rightly classified even thought they were fairly long

! Different parts of the same neurite structure may be classified with different labels (see the yellow-circled neurites, Fig. 5.3) because of different values of linearity (on the FI) and pixel intensity (on the OI).

⇒ Neurites labeled with more than one label will have to be labeled with only one label

3) ! A lot of features were labeled as Class 3 neurite, but by meaning of a comparison with the original image, they should were labeled as Class 2 neurites

⇒ morphological operations will have to be performed on their shapes, in order to connect parts that are supposed to belong the same neurite

4) ! Those objects which were supposed to be a small beads were red-labeled and regarded as Class 1 neurites. (see purple-circled neurites, Fig. 5.3)

⇒ a re-classification of those objects will have to be carried out, taking advantage of the fact that their shape has different features from the real-Class 1 objects: smaller area, smaller perimeter and smaller coefficient of shape irregularity (which is defined as the ratio Area/Perimeter).

5.5.2 Classification of the round objects

With regard to the classification of the round objects (mainly Neuron Cell Bodies and Small Beads) the classification algorithm is applied to the MF filtered image. In the MF filtered image, each pixel has been labeled either with light-blue or brown labels, which were chosen according to the following criterion on both images:

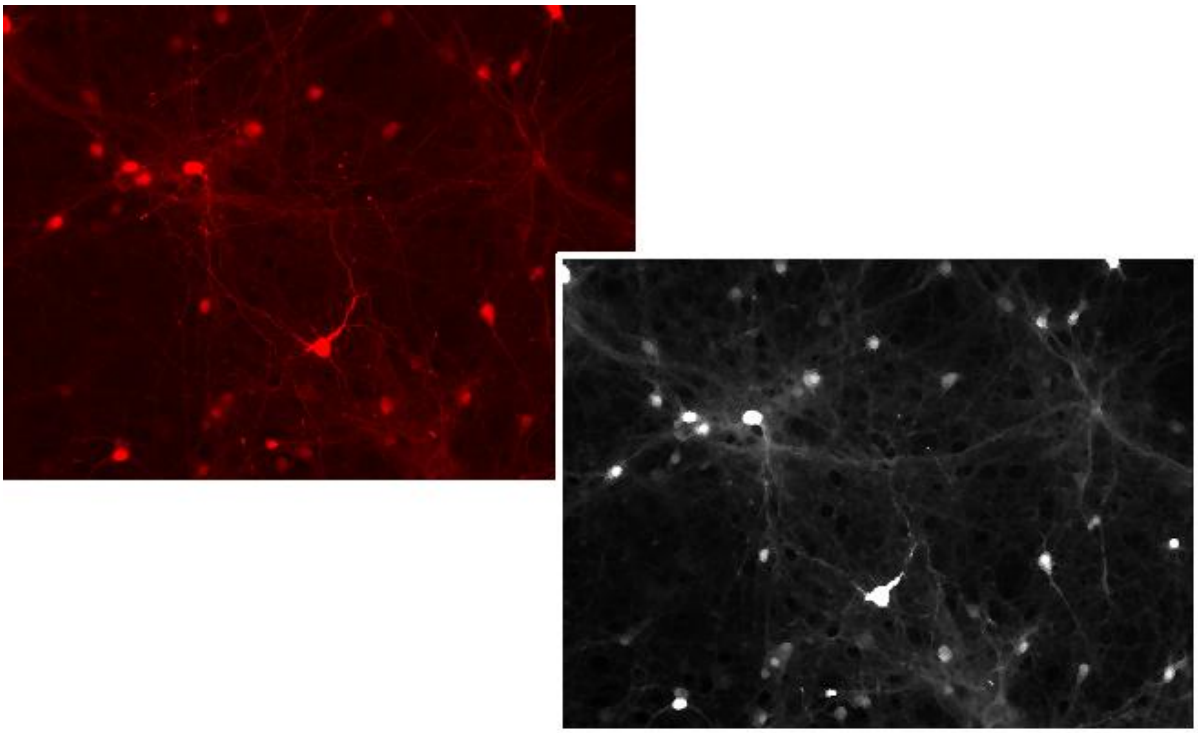


Figure 5.4 – The median filter localizes areas composed of large amounts of bright pixels, as round objects

1) On the original image:

Label Color	Pixel intensity (PI)
BROWN	P.I. ≥ 130
LIGHT-BLUE	90 ≤ P.I. ≤ 245

Table 5.3a – Threshold values used to classify the linear structures on original image

2) On the PLLF filtered image:

Label Color	Pixel intensity (PI)
BROWN	P.I. ≥ 240
LIGHT-BLUE	P.I. ≥ 84

Table 5.3b – Threshold values used to classify the linear structures on filtered image

Giving as input such obtained filtered image to the classification algorithm, the following classified image is obtained as output (see Fig. 5.5).

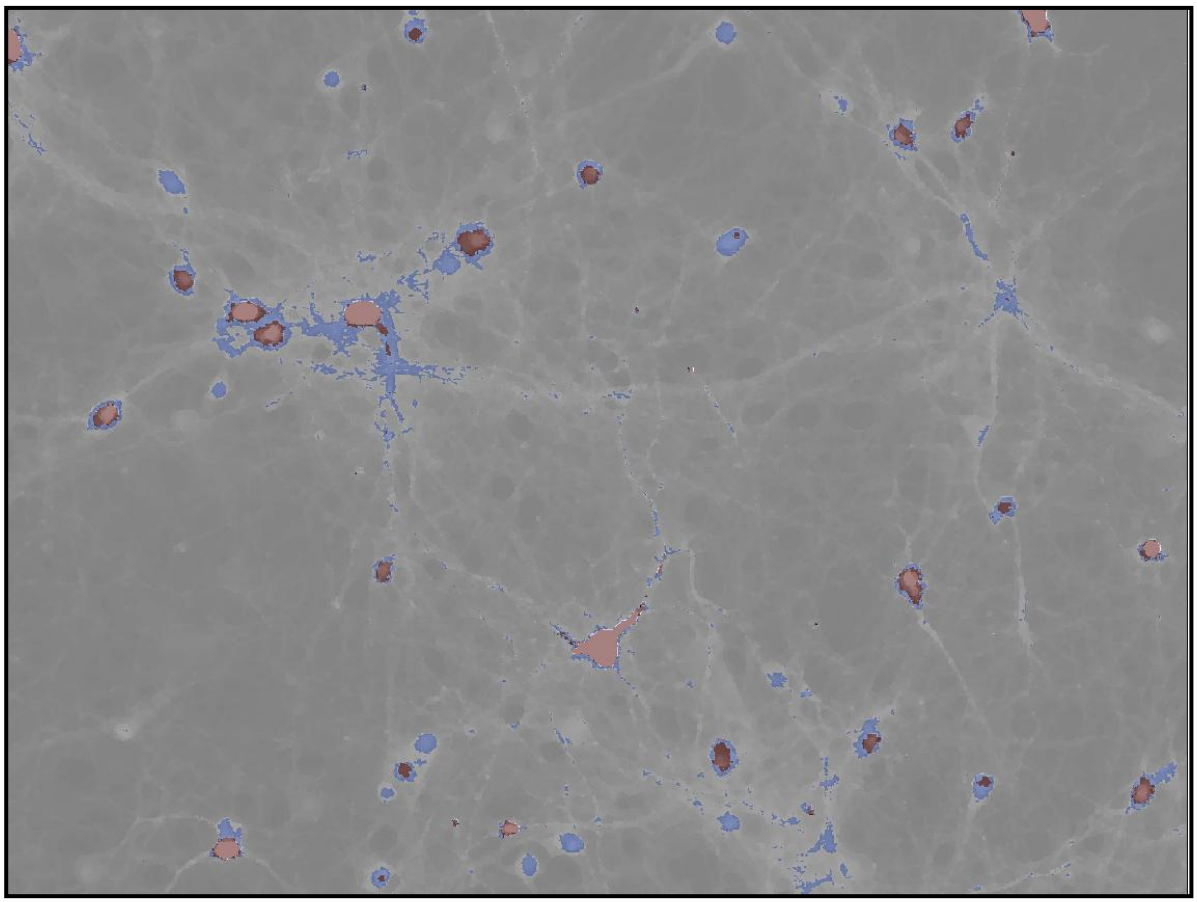


Figure 5.5 – The Classified image is a 3 color labeled image; background pixel have been labeled with white (background here looks grey because of the overlapping with a fairly dark image)

Here follows a brief comment of the obtained image, that shows which good result were achieved (√), what represent unexpected results or misclassifications (!) and what should be improved by the Post-Processing step to get a better final classification result (⇒)

√ Most of bright neuron cell bodies were right away classified in the main class, especially the ones which are linked with a lot of neurites or very thick neurites and it can also be said that their shape is already congruent with the original image's one.

! in some cases, the part of the neurites which were either connected with the neuron cell body or next to the neuron cell body has been labeled with brown (see red cicle, Fig. 5.6)

⇒ morphological operations will have to be performed on the neuron cell body shapes, in order to get the brown labeled neurites structure to be regarded as background whilst the neuron cell body shapes to be as much as possible congruent with their own original shapes.

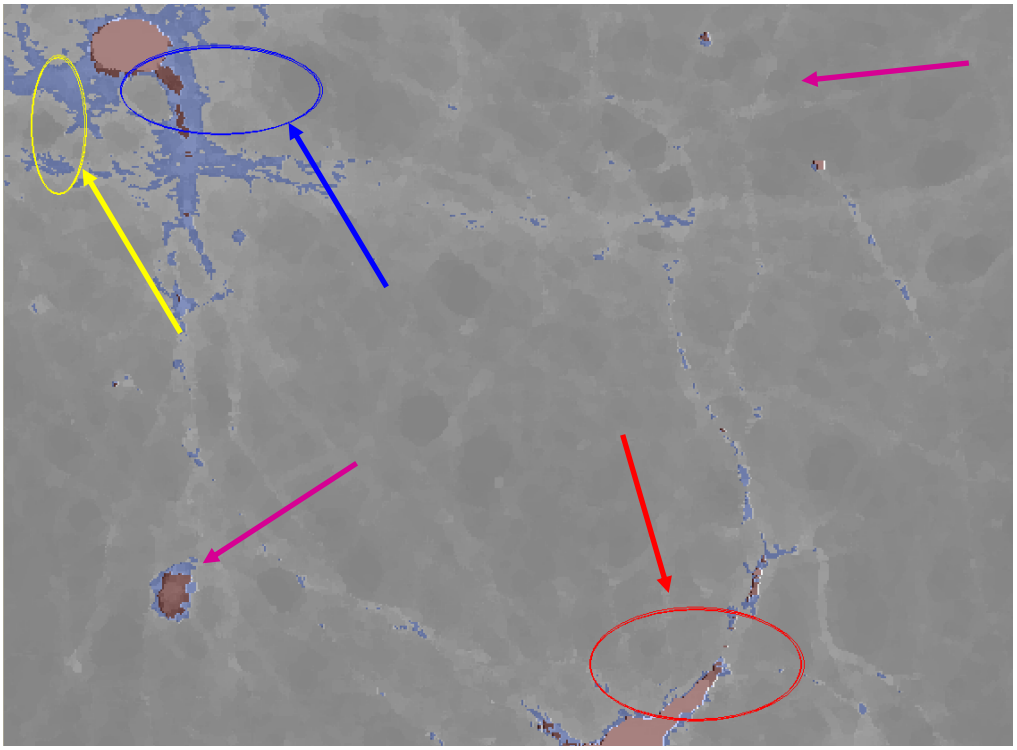


Figure 5.6 – A magnified view of the classified image and instances of misclassified areas

! All of the detected small beads and less-bright neuron cell bodies have been classified in the light-blue class, but those two different classes of objects will have to be re-classified in different ways, being two of the three main features that have to be measured as output of the IAP (see purple arrows, Fig. 5.6)

⇒ Because those two different features have different value of areas as well as perimeter and irregularity coefficient, morphological operations based on the just mentioned important differences will have to be performed on those different structures, aimed to separate them into two different classes and to enhance their shape, as well.

! Some of the bright area nearby neuron cell bodies (especially around the brown labeled ones) was classified in the light-blue class (see blue and yellow circled areas, Fig. 5.6) but such a structures do not correspond to any feature that has to be classified

⇒ Because those misclassified areas are supposed to feature a high value of irregularity (whereas neuron cell bodies are fairly round structures featuring a low value of that size), morphological operations based on the irregularity should help to get rid of such structures.

5.5.3 Classification methods not used in this project

The following classified image was obtained by means of a function performing the Bayesian supervised classification on the PLLF filtered image (where the value of the label were given to the image' feature by a visual assignation and then stored for potential further application of the method in a batch process)

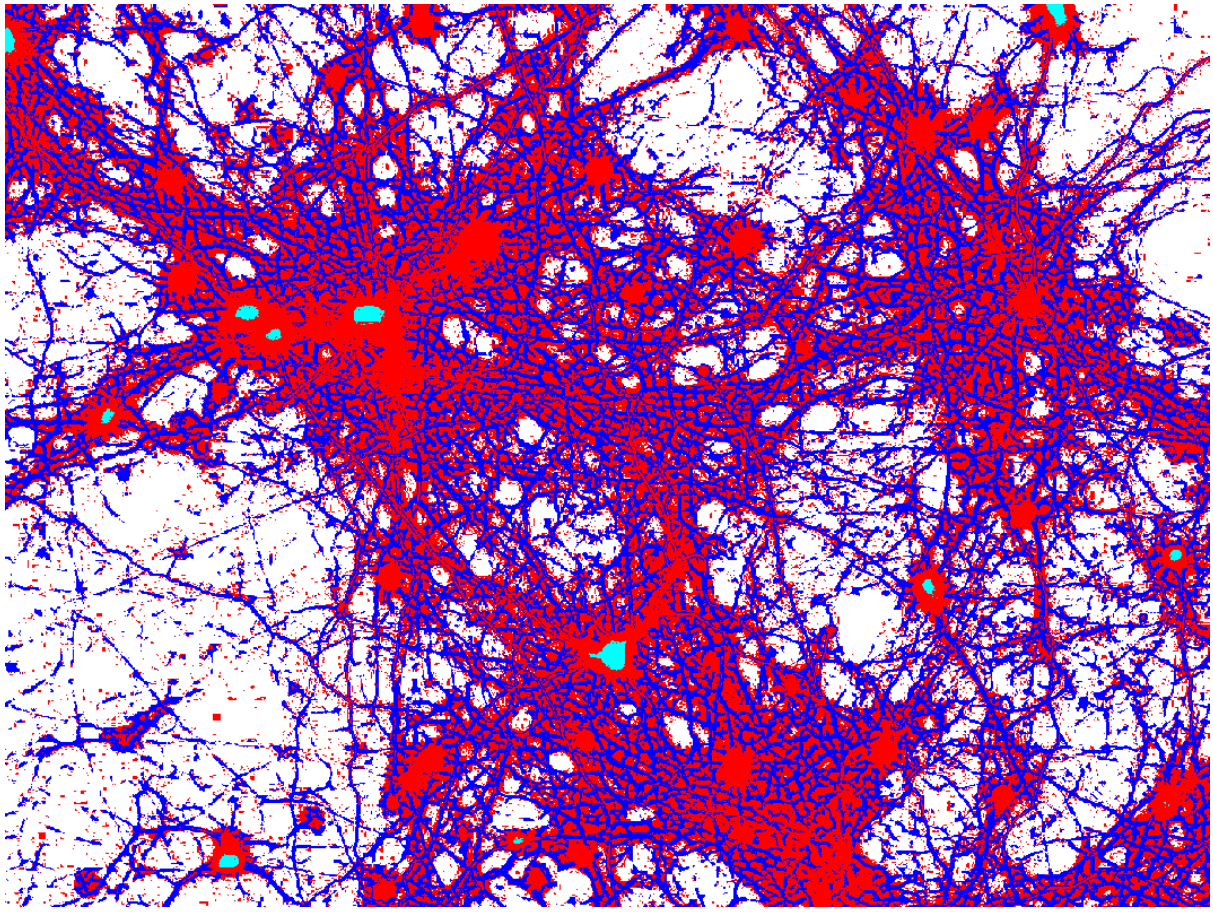


Figure 5.7 – The Bayesian method classified image is a 4 color labeled image;

Only a few neuron cell bodies are clearly detectable, whereas most of them where classified as neurites. About the linear structures, many of them where classified as class 2 neurites, but it is not possible to detect any linear structure in the red labeled area. Furthermore it is not possible to locate any class 3 neurite, whose structures have been regarded as background. The not good result is also attributable to a wrong initial label assigning of the features by the user. This is not a good starting point from which to perform post-processing methods.

Here follows the image obtained be means of the Unsupervised K-means classification. The function receives as input only the desired number of output classes, into which the image's features will be classified; therefore possible classification errors are no attributable to a wrong initial labeling of the features by the user. The resulting labeling process is better than the one obtained by means of the Bayesian classification because several linear structures are better distinguished among each other and the most of the neuron cell bodies were detected Several linear structures were not detected (they were classified as background, instead) and in most of the cases, blue and red labeled areas correspond to bundles of neurites on the original image, whose single components can not be identified. Once again, that is not a good result from which to perform post-processing methods.

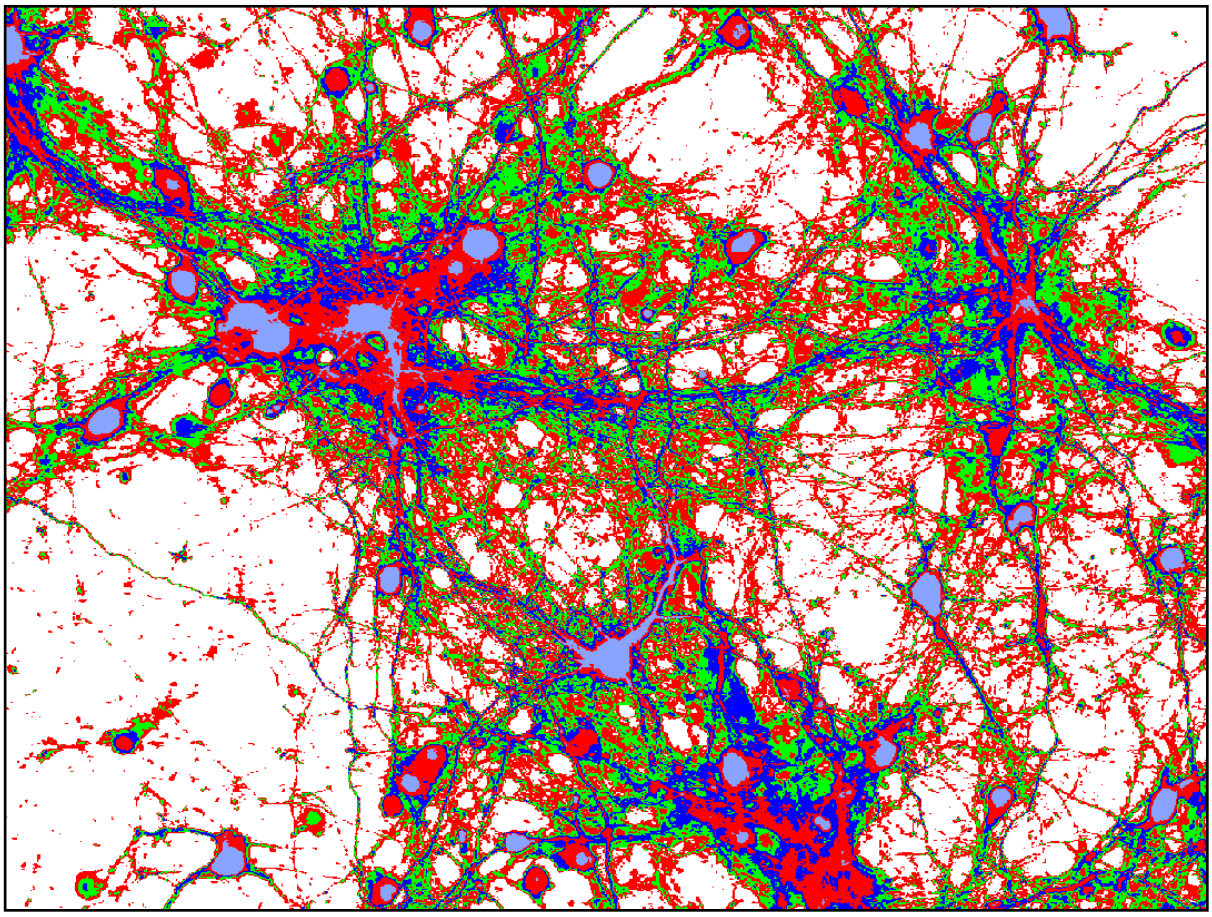


Figure 5.8 – The K-means method classified image is a 4 color labeled image;

Chapter 6

Morphological Post-Processing

The IAP´s morphological post-processing operations allow, in general, to improve the shapes and the structure of the classified features and to get the desired features to be ready to be measured. As well as in the former chapters, such step has been carried out on the TestImage outputted from the former analysis stage, as in the classification step previously described.

6.1 Erosion and Dilation

Mathematical morphology is an approach to image analysis based on shapes. The purpose of Morphological operations is to produce contrasting results when a structuring element (e.g. a label) is applied to either grayscale or binary images. The value of each pixel in the output image is based on a comparison between the corresponding pixel in the input image and its neighbors. After the choice of size and shape of the neighborhood, it is possible to construct a morphological operation that is sensitive to specific shapes in the input image. Erosion and dilation are the most basic morphological operations. Erosion shrinks image objects by removing pixels on object boundaries while dilation expands them by adding pixels to the boundaries of shapes in an image.

6.1.1 How the erosion and dilation algorithms work - general description

The number of pixels added (or removed) from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the following rules are used to process every pixel:

Operation	Rule
Dilation	The output pixel assumes the *maximum* value of all the pixels in the input pixel's neighborhood.
Erosion	The output pixel assumes the *minimum* value of all the pixels in the input pixel's neighborhood.

Table 6.1 – Rules to process pixels in erosion and dilation

6.1.2 Dilation - fundamental concept

Definition The dilation of the set X with the structuring element B is the set $X \oplus B = \bigcup$

We obtain that $X \oplus B = \{z \in S \mid \exists\, b \in B \;\; \exists\, x \in X : z = x + b\}$
$= \{x + b \in S \text{ for some } x \in X$
and for some $b \in B\} = \bigcup B_x$

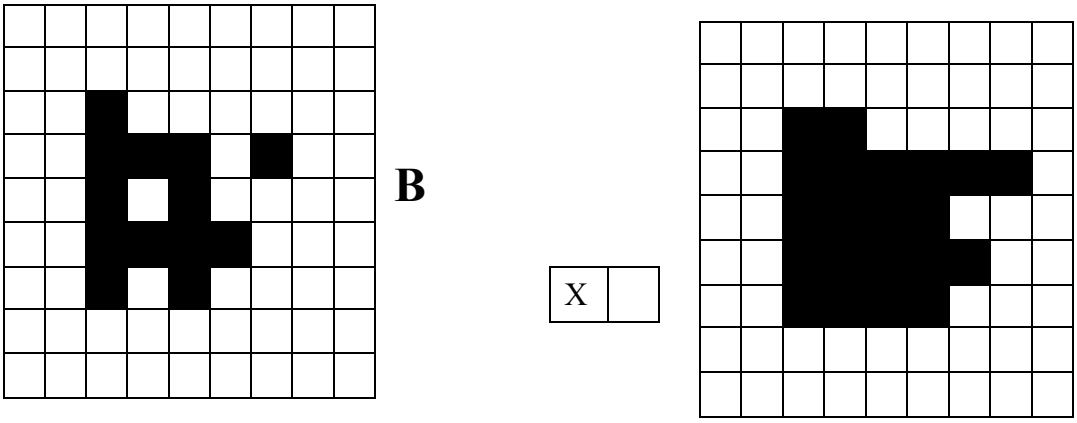


Figure 6.1 - An example illustrating the definition of the dilation with structuring element B

6.1.3 Erosion - fundamental concept

Definition The erosion of the set X with the structuring element B is the set $X \ominus B = \bigcap X_{-b}$

We get the realtion $X \ominus B = \{z \mid \forall\, b \in B \;\; \exists\, x \in X : z = x - b\}$
$= \{ z \mid \forall\, b \in B : z + b \in X\}$
$= \{z \mid B_z = X\}$

Erosion consists of those points z for which the translated structuring element is completely contained in X. [10]

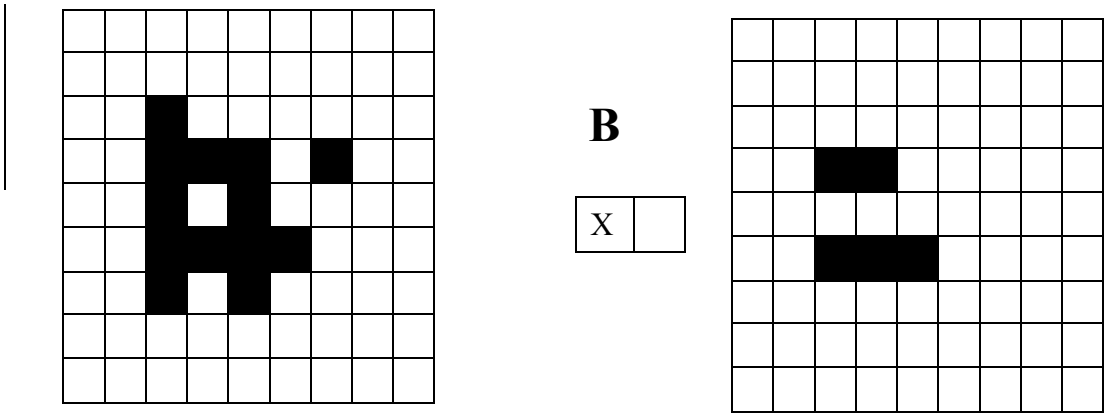


Figure 6.2 - An example illustrating the definition of the erosion with structuring element

6.1.4 Final Result on shapes after Erosion and Dilation application

- Erosion tends to decrease the size of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element.

- In grayscale images, erosion reduces the brightness (and therefore the size) of bright objects on a dark background by taking the neighborhood minimum when the structuring element over the image is passed.
- In binary images, erosion completely removes objects smaller than the structuring element and removes perimeter pixels from larger image objects.

- Dilation generally increases the sizes of objects, filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element.

- In grayscale images, dilation increases the brightness of objects by taking the neighborhood maximum when passing the structuring element over the image.

- In binary images, dilation connects areas that are separated by spaces smaller than the structuring element and adds pixels to the perimeter of each image object.

6.2 Skeletonization

Skeletonization is a shape analysis method to reduce the foreground regions of a binary image into a skeletal remnant. It is useful when it is not important to keep the size of the pattern but rather the relative position of strokes in the pattern [11].

Its purposes are either

- preserving the topological properties of the original region (mainly the connectivity);
- throwing away most of the original foreground pixels.

The skeleton is a representation of the shape it has been obtained from, and it is also definable as an abstraction of the shape length, direction and width.

6.2.1 How the Skeletonization algorithm works

A skeleton can be created in two different methods:

- By using some kind of morphological thinning that successively delete pixels from the boundaries (while preserving the end points of line segments) until no more thinning are possible. At that point what is left approximates the skeleton.
- By the previous calculation of the distance transform of the image. The skeleton then lies along the *singularities* (*i.e.* creases or curvature discontinuities) in the distance transform.

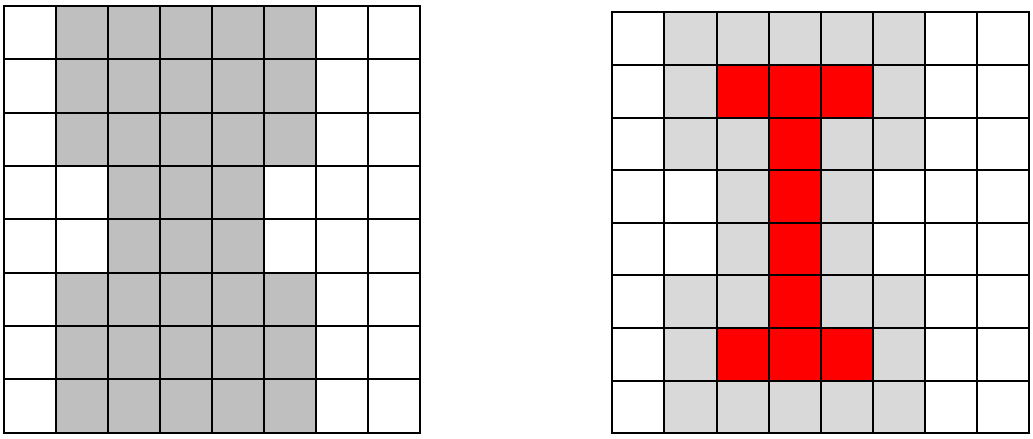


Figure 6.3 – A pattern before and after Skeletonization: red pixels are the skeleton of the grey pattern

The function used in this project performed a skeletonization based on the first approach, by implementing the Hilditch's algorithm. Once again, the basic idea is that after pixels have been peeled off, the pattern should still be recognized. Hence the skeleton obtained must have the following properties:

- as thin as possible
- connected

- centered.

When these properties are satisfied, the algorithm must stop. Before starting to explain the algorithm functioning, the following notions need to be defined. In order to decide whether to peel off the pixel p1 or keep it as part of the final resulting skeleton (see Fig. 6.4), the 8 neighbors of p1 must be arranged in a clock-wise order and further two main functions must be defined (see an example in Fig. 6.4a and 6.4b).

P9	**P2**	**P3**
P8	**P1**	**P4**
P7	**P6**	**P5**

B (p1) = 2, A (p1) = 1

P9	**P2**	**P3**
P8	**P1**	**P4**
P7	**P6**	**P5**

B (p1) = 2, A (p1) = 2

Figure 6.4.a,b – Different patterns featuring different values of the two main functions

Functions to be defined:

- **B(p1)** = number of non-zero (colored) neighbors of p1;
- **A(p1)** = number of 0-1 patterns in the sequence p2, p3, p4, p5, p6, p7, p8, p9, p2.

The Hilditch's algorithm works either by using a 4X4 or a 3X3 window; in the following the latter one is explained. The algorithm performs multiple passes on the pattern. On each pass it decides to change a pixel from yellow to white if the following four conditions are satisfied:

1) 2 < = B(p1) < = 6
2) A(p1)=1
3) p2.p4.p8 = 0 or A (p2) != 1
4) p2.p4.p6 = 0 or A (p4) != 1

⇨

IF:

- Conditions are satisfied: YELLOW PIXEL is turned into WHITE
- Nothing changes: ALGORITHM STOPS (and no more pixel can be turned into WHITE)

Each of the conditions above ensures a particular property of the skeleton.

- **Condition 1 -** 2 < = B(p1) < = 6: This condition combines two sub-conditions: The first condition ensures that no end-point pixel and no isolated pixel be deleted (any pixel with 1 yellow neighbor is an end-point pixel); the second condition ensures that

the pixel is a boundary pixel.

- **Condition 2** - (A(p1)=1):This is a connectivity test. For instance (see Fig. 6.4b where A(p1)>1) by changing p1 to 0 the pattern will become disconnected.

- **Condition 3** - p2.p4.p8 = 0 **or** A(p2)!=1:This condition ensures that 2-pixel wide vertical lines do not get completely eroded by the algorithm.

- **Condition 4** - p2.p4.p6 = 0 **or** A(p4)!=1:This condition ensures that 2-pixel wide horizontal lines do not get completely eroded by the algorithm.

6.2.2 Properties of the Hilditch's Algorithm

The Hilditch's Algorithm is a parallel-sequential algorithm.

- It is parallel because at one pass all pixels are checked at the same time and decisions are made whether to remove each of the checked pixels;

- It is sequential because this step just mentioned is repeated several times (until no more changes are done).

However, Hilditch's algorithm turned out to be not the perfect algorithm for skeletonization because it does not work on all patterns. In fact, there are patterns that can be completely erased by the algorithm.

6.3 Morphological Post-processing methods in this project

The morphological post-processing (MPP) part of the image analysis protocol receives as input a classified image (where it is already possible to recognize the three main classes in which the three main features have been divided) and improves the shape of the objects by means of operations of

- erosion and dilation (E&D)

- area, irregularity and perimeter conditions (AC, IC, PC)

- change of label for particular conditions of pixel surrounding labels (SC)

6.3.1 Applications on the PLLF filtered image

Here are listed the main properties of algorithms performing morphological post-processing on classified PLLF-filtered images:

INPUT IMAGE: The classified image is a three main labels image (the 4th label (W), refers to the background pixels) in which the three sub-classes of linear structures have been divided

POST-PROCESSING ALGORITHM GOAL: to fortify the shapes of the linear structures and to connect components which are supposed to be part of the same neurite (where possible)

ULTIMATE PURPOSE: to create a more linked and robust neurite labeling layer, in order to get a better starting point on which the measurement algorithms of the post-processing step can work

- OUPUT IMAGE: A four main labels image (the 5th label (W) refers to the background pixels) in which the three main sub-classes of neurites and most of the small beads have been re-classified

Example of output of morphological post-processing algorithms for linear structure enhancement:

Receiving as input a classified PLLF filtered image (like the one in Figure 6.5) the algorithm must modify the label layer in order to:

(1.a) get the **CLASS 1** neurite shapes to be enhanced and the "false" class 1 neurites (with structure that is too weak or fragmented to be regarded as class 1) to be regarded as **CLASS 2** neurites;

(2.a) get most **CLASS 2** neurites to be connected to each other and their shapes to be enhanced and get the "false" **CLASS 2** neurites (with structure that is too weak or fragmented to be regarded as a linear structure) to be regarded as **CLASS 3** neurites.

(3.a) get most **CLASS 3** neurites to be connected to each other and their shapes to be enhanced and get the "false" **CLASS 3** neurites (with structure that is too weak or fragmented to be regarded as any main image object) to be regarded as background.

(4.a) classify the Small Beads that will be labeled as linear structures, in their own class.

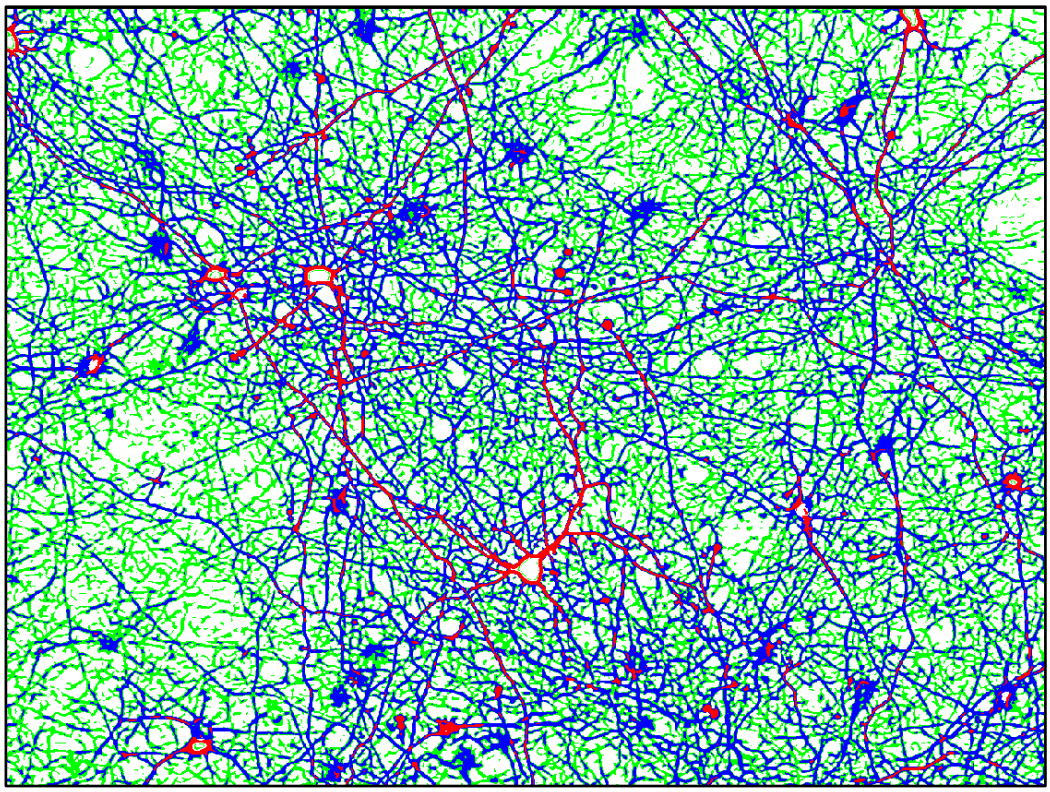


Figure 6.5 – The Classified image before the morphological post-processing

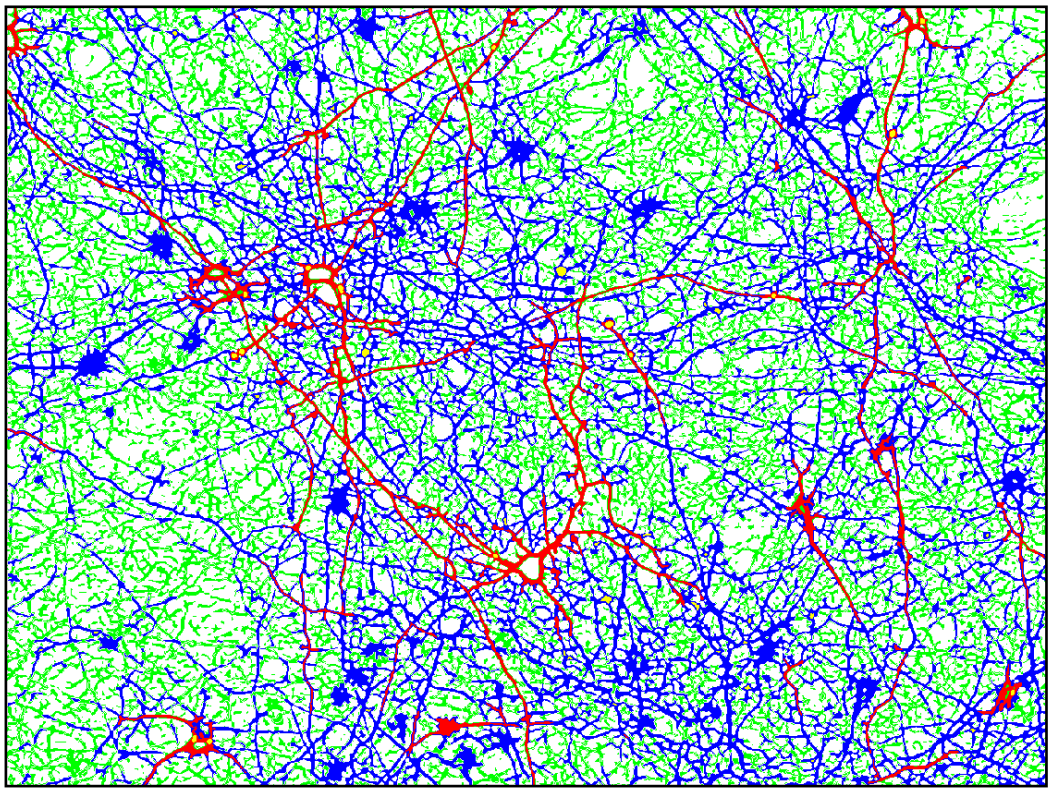


Figure 6.6 – The Morphological-Modified, where most of the shapes of the objects have been improved

The desired improvements of the shape of the components of the classified images (using the number listed above) have been achieved, by performing an ordered sequence of the methods previously listed. Here follows a comparison of patterns belonging to the same area of the Test Image, respectively in the classified image (leftmost) and in the Morphological-Post-processing-modified image (FI1).

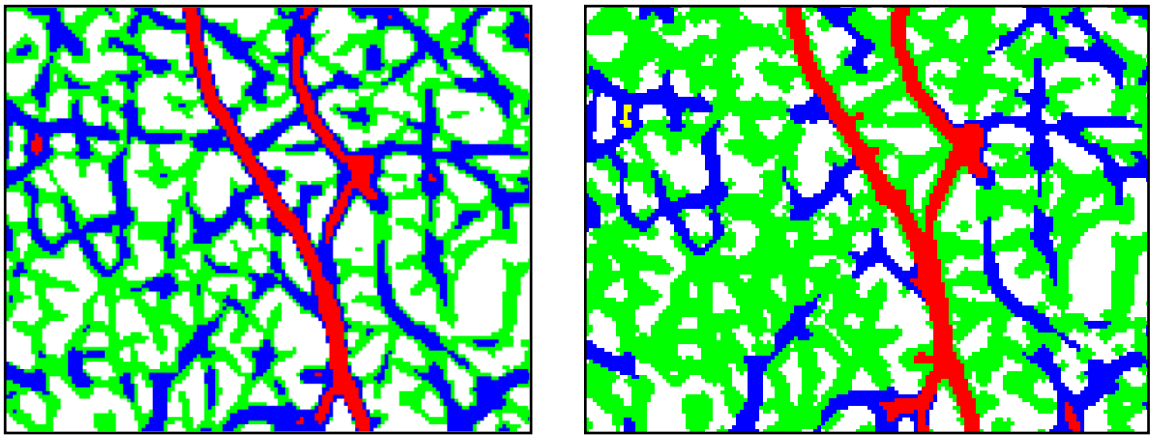


Figure 6.7 – Connection of two parts of the same class 1 neurite

(1.a) CLASS 1 neurites - shape enhancement: neurite structures have been fortified, connecting parts of near non-linked class 1 neurites (see Fig. 6.7). Furthermore, all smallest parts of class 1 neurites that it was not possible to gather into bigger structures were re-classified as CLASS 2 neurites (see black arrows in Fig. 6.8);

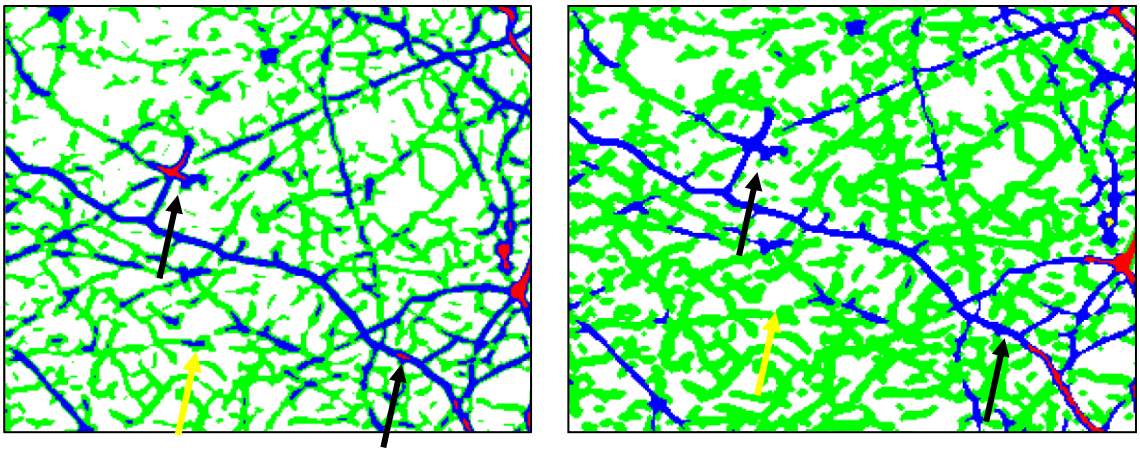


Figure 6.8 – Smaller parts of class 1 neurites have been re-classified (and re-labeled) as class 2 neurites

(2.a) **CLASS 2** neurites - shape enhancement: neurite structures have been fortified, connecting each other parts of near non-linked class 2 neurites. Furthermore, smallest parts of class 2 neurites that it was not possible to gather into bigger structures were re-classified as **CLASS 3** neurites (see yellow arrows, Fig. 6.8);

(3.a) **CLASS 3** neurites - shape enhancement: parts of class 3 neurites have been connected each other.

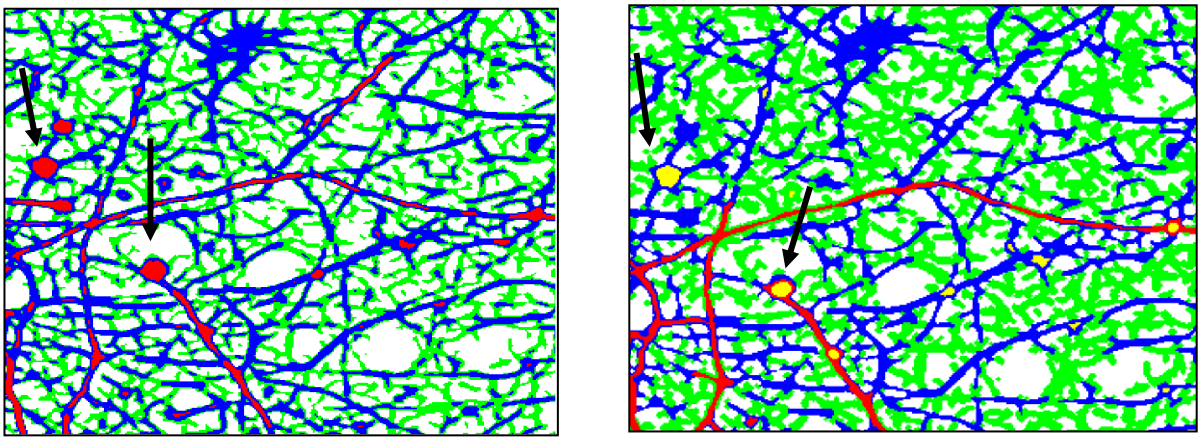


Figure 6.9 – Some of the beads have been detected and labeled with yellow

(4.a) BEADS detection and labeling: a kind of undesired effect of the linear structures classification was that the shape of the so-called smaller beads were originally recognized as linear structures and, in most of the cases, regarded as CLASS 1 linear structures. A dedicated part of algorithms performing the linear structures morphological post-processing has been implemented, hence most of these round objects have been re-classified into a the main feature class of the Small Beads (see black arrows in Fig. 6.9);

A detailed description of the sequence of methods applied to the image is displayed in Chapter8 "Implementation and Running of the IAP's methods", along with images visualizing the improvement achieved by applying of method.

6.3.2 Applications on the MF filtered image

Here are listed the main properties of the algorithms performing morphological post-processing on classified MF-filtered images:

INPUT IMAGE: A three label image (the 3rd label (W) refers to the background pixels) which represents a labeling of the Neuron Cell Bodies (NCB) and some of the Small Beads (SB) indistinctly

POST-PROCESSING ALGORITHM GOAL: to locate the real NCB, re-classify the smallest round structures as SB and classify what has been misclassified in either NCB or SB as background

ULTIMATE PURPOSE: to create a more robust and consistent NCB labeling, in order to get a better starting point on which the measurement algorithms of the post-processing step can work

OUPUT IMAGE: A three label image (the 3rd label (W), refers to the background pixels) into which both the NCB and some of the SB have been re-classified.

Example of output of the morphological post-processing algorithms for the enhancement of round objects:

Receiving as input a classified MF-filtered image (see Figure 6.11a) the algorithm modifies the label layer in order to:

(1.b) perform shape enhancement of previously detected NCB shapes in order to get their shape in the classified image to fit as much as possible to the one of the respective objects in the original image.

(2.b) perform a detection of the SB, followed by shape enhancement and finally by re-classification.

(3.b) label as background the pixels that compose the misclassified areas or objects, like thickest parts of neurites, which were part of the brightest objects outputted from the MF filtered image (nearby the NCB, mainly).

The NCB shape enhancement has been carried out in two successive steps:

- by modifying and enhancing the shape of the brown labeled neuron cell bodies
- by modifying and enhancing the shape of the light-blue labeled neuron cell bodies

In the resulting image, the classified round objects have finally been labeled with brown.

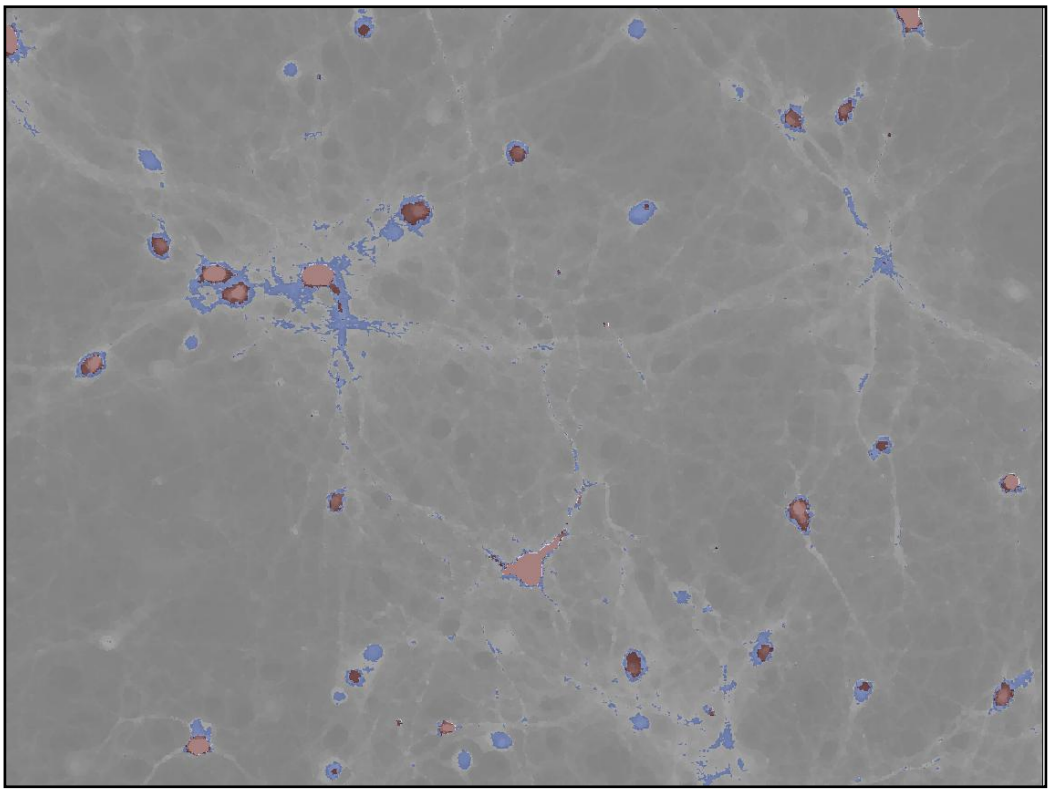


Figure 6.10a – In the Classified image the round structures (both NCB and SB) have been indistinctly labeled

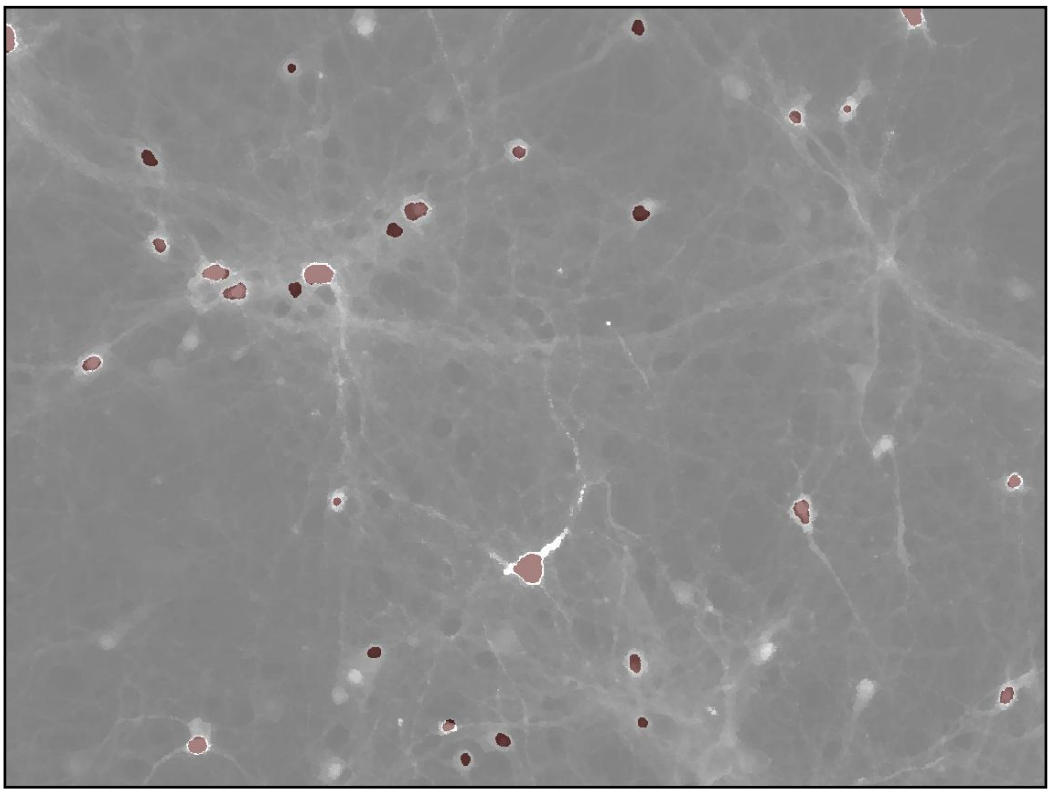


Figure 6.10b – After the morphological post-processing, the classified image is a 1 color labeled image in which only the neuron cell bodies have been correctly classified. It was not possible to classify the small beads

The desired improvements of the shape of the components of the classified images (using the number indicated above) have been achieved, by performing an ordered sequence of the methods previously listed. Here follows a comparison of patterns belonging to the same area of the Test Image, in the classified image (leftmost) and in the Morphological-Post-processing-modified image (FI2) respectively.

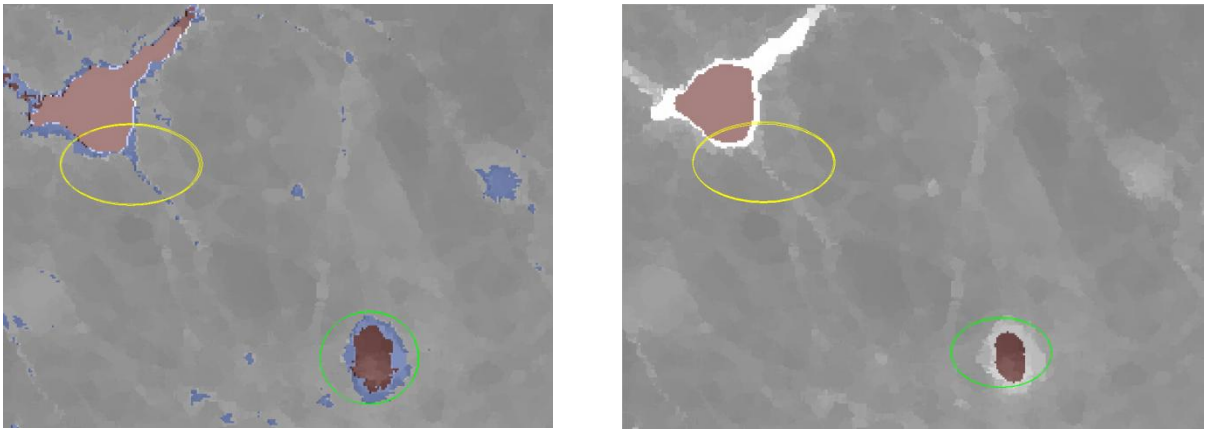


Figure 6.11 – Effects of the morphological post-processing on the MF classified image

(1.b) Neuron Cell Body - shape enhancement: shape of previously detected NCB have been improved to fit as much as possible to the shape of the respective objects in the original image (e.g. green circled structures of Fig. 6.11)

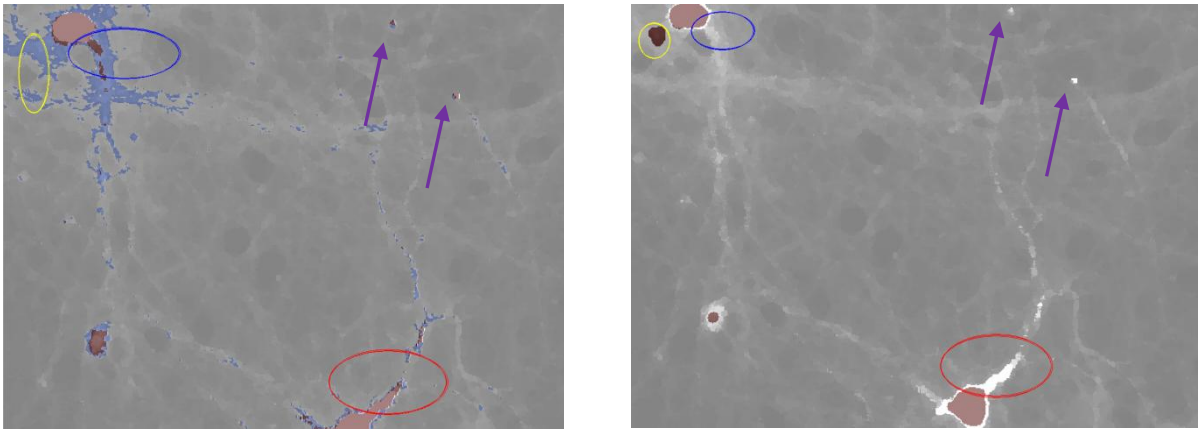


Figure 6.12a , 6.12b – Effects of the morphological post-processing on the MF classified image

(2.b) Small Bead - shape enhancement: in all of the images (as well as in the Test Image) it was not possible to preserve the previously detected SB. No sequence or singular application of the Post-processing methods (Erosion, Dilation, Condition of Irregularity and Area, and Smoothing Function) were capable of differentiating them from other objects which were labeled with the same label and which should not be part of the final result, being neither SB nor NCB. Therefore such objects were excluded from the final result of the round objects morphological post-processing step (compare brown labeled object on right-topmost corner

of Fig. 6.12a which are no longer present in Fig. 6.12b, purple arrow). A more detailed explanation of such a outcome can be found in Chapter 8 “Implementation and Running of the IAP’s methods”.

(3.b) Misclassified object - re-classification: most of the wrongly classified objects (which were neither NCB nor SB) were labeled as background (compare red and blue circled structures, Fig. 6.12a and 6.12b). In some images (as well as in the Test Image) it was not possible to get rid of them, therefore they are instances of misclassified objects which are going to be part of the final classification result (compare yellow circled objects, Fig. 6.12a and 6.12b)

6.3.3 Final Classified Image and consideration about the post-processing results

The post processing part of the image analysis protocol that has just been carried out, outputs the final version of the classified image (FIC). Such image is given by the overlapping of the label layers of the two post-processed versions of classified images. The label layer of the image where the circular structures have just been post-processed is overlapped to the label layer of the image where linear structure have been post-processed, so that the final label color of the image’s features that have been classified as NCB is the brown (from FI2), and the final label about both neurites and small beads is their own label by FI1 (see Fig. 6.13). Some important considerations must now be made. These considerations were drawn up after the comparison of the post-processed version of all the classified images of the dataset with the respective original images, and therefore they have a general validity.

1) Positive results were obtained in terms of:

1.1) class1, class2 neurites and neuron cell bodies classification: Most of these structures were correctly classified in their own class.

1.2) several beads have been detected and rightly classified

1.3) despite some of the neuron cell bodies were not completely inside the borderline of the image, they were correctly identified in their own class

For each of these image features the results were fairly good, although the widths or areas in the final classified image were not perfectly matched in all cases.

2) Aspects that should be improved:

2.1) some neurons have been misclassified as beads, and vice-versa. The number of this kind of misclassification is low, compared to the actual number of beads and neuron cell bodies that are present in every image

2.2) Although neuron cell bodies were correctly identified, it was not certain that their measured areas corresponded to their actual areas. This was due to the fact that not all cell bodies were completely arranged inside the borderline of the image.

2.3) some of the class 2 neurites were downgraded as class 3 even though their structure was barely identifiable as linear and even thought it can be said that they belong to a neurite. Despite that, as it was not possible to connect them to other parts of class two neurites and the selection by area or by irregularity performed afterward get them to be regarded as class 3 neurites. That selection was made because a structure that should be regarded as neurite was supposed to be continuous and to have a minimum length. Anyway the minimum area value used was 100 pixels - as it is described in chapter 8 – is fairly low value since from a comparison with the original image it was noticed that by using greater values, several features which were actually part of a neurite would have been downgraded. This excessive skepticism means that, statistically speaking, a lot of False Negative (type II error) would have been part of the final classification. On the other hand, that led to introduce a number of False Positive in the classified image

2.4) most of the green labeled objects were not actually identifiable as neurites, even thought the pixel intensity values in such areas were slightly higher than the background values, in the original image.

➡ **IMPORTANT DECISION**: According to 2.4) it was decided not to consider the class 3 neurites as part and parcel of the classified neurites structures on which perform the measurement of the main output "Neurite Branches Total Length".

The total length of class 3 neurites will be calculated anyway in order to obtain as much more information as possible about the content of the images.

3) The choice of each post-processing method's parameter, as well as the particular sequence of application of methods, was made, as usual, as a combination of these three factors:

- by an accurate visual inspection of original images
- according to the values of their filtered versions
- by evaluating the initial features´ labeling, which is the image outputted from the classification step

in order to obtain what was regarded as the best final results in terms of right classified features.

The so-called best result is not to be regarded as a perfect classification of the image's features. The main problem of this step is that every morphological action performed in order to improve the shape features in a particular area of the image, also changes feature's shapes on the remaining area of the image. This means that when a result was achieved in terms of shape improvement of a feature (or of an area), the sequence of method applied might have worsen (even substantially) the shape of many other features in other areas of the image. That is why it was not easy to achieve good results in terms of classification, and the final classified image must be interpreted as a compromise between what the real image depicts and what it is possible to achieve by means of the classification, in order to obtain a good tradeoff between features that have been correctly classified and feature that have been not.

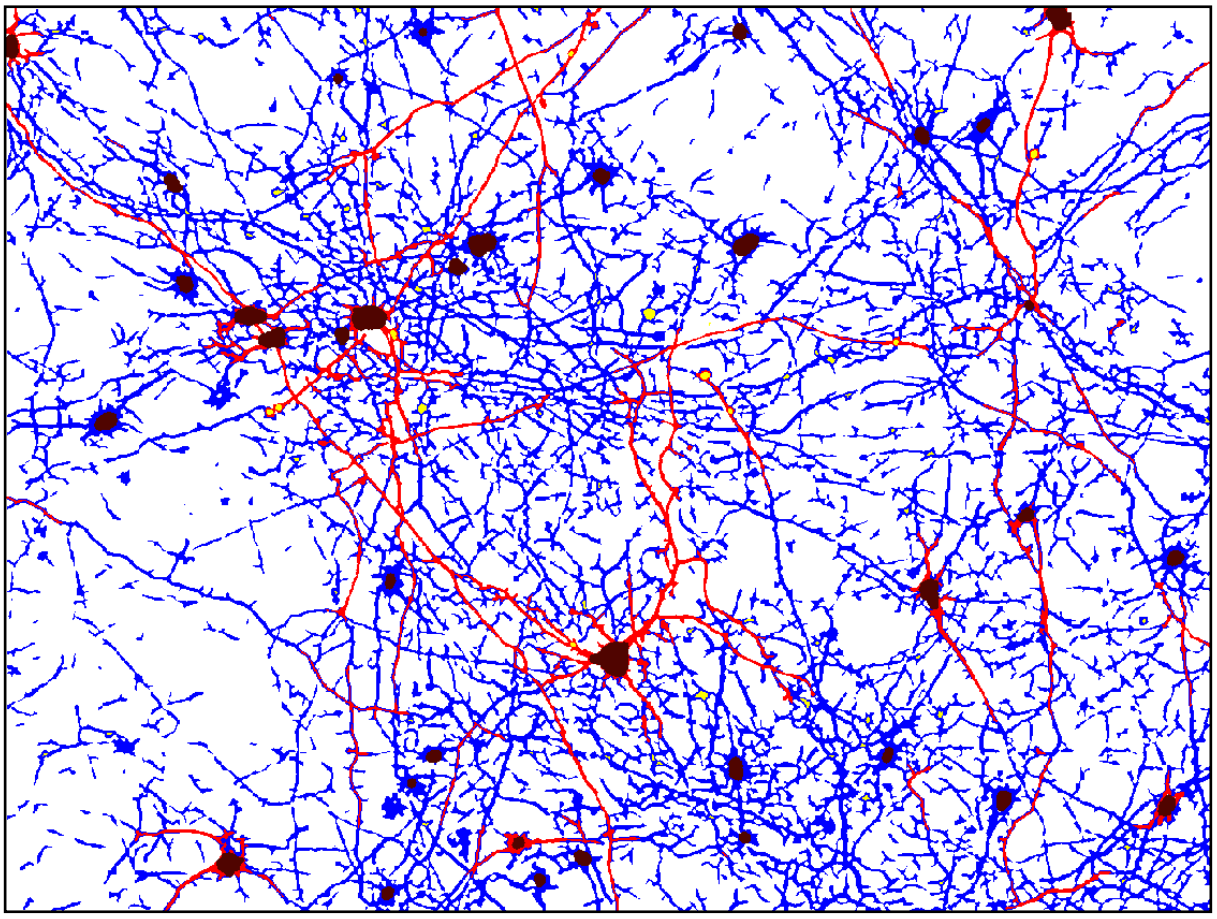


Figure 6.13 – The final classified image, where the label layer of FI2 is overlapped to the label layer of FI3. The green labeled features are no longer part of the final neurites classification. The measurement of the desired outputs will be performed on this label layer, on the non-white labeled pixels.

Chapter 7

Measurement Post-Processing

The final purpose of the IAP´s measurement post-processing stage is to calculate the desired values of image features. As well as in the former chapters, such step has been carried out on the TestImage outputted from the former analysis stage (the post-processing step previously described). The desired values calculated as output of this step are:

- Neurite Total Length
- Neuron Cell Bodies Total Area
- Neuron Cell Bodies Total Number
- Small Beads Total Area
- Small Beads Total Number

A description of particular functions and algorithms used to calculate such output values is provided.

7.1 Neurite total lengths measurement

Neurite total length is computed as the sum of lengths of each CLASS and CLASS 2 neurites which compose the image. Both these two values were an output of the program, as well as the total length of the CLASS 3 neurites. In order to calculate the length of a structure which is (in most of the cases) wider than one pixel, it was chosen to perform an operation of skeletonization on each neurite involved in the measurement process, in order to obtain a 1-pixel-width structure representing the skeleton of such neurite. In particular, the skeletonization function works with the label values of the image obtained as output of the post-processing of the classification of the PLLF filtered image (FI1 and FI2) and:

- receives as input the color of the label with which the neurites were labeled (respectively red, blue and green)
- creates as output the skeletonized version of neurite structure, by labeling the pixels composing the skeleton with the same label color of the neurite from which the skeleton has been created from
- sets as background (white) each other pixel that was part of the original neurite structure

The calculation of the total length of each class of NB is computed by means of a function that runs through pixels of the image resulting from the skeletonization of FI1, looks for the pixels which label color *C* equals one of the three colors (red, blue and green) with which these features were labeled, and, whenever such equality is verified, increases by one the total area value of that sub-classes of NB. The final value of the NB total area is given by the sum of the values of area of the blue and red skeletonized NB.

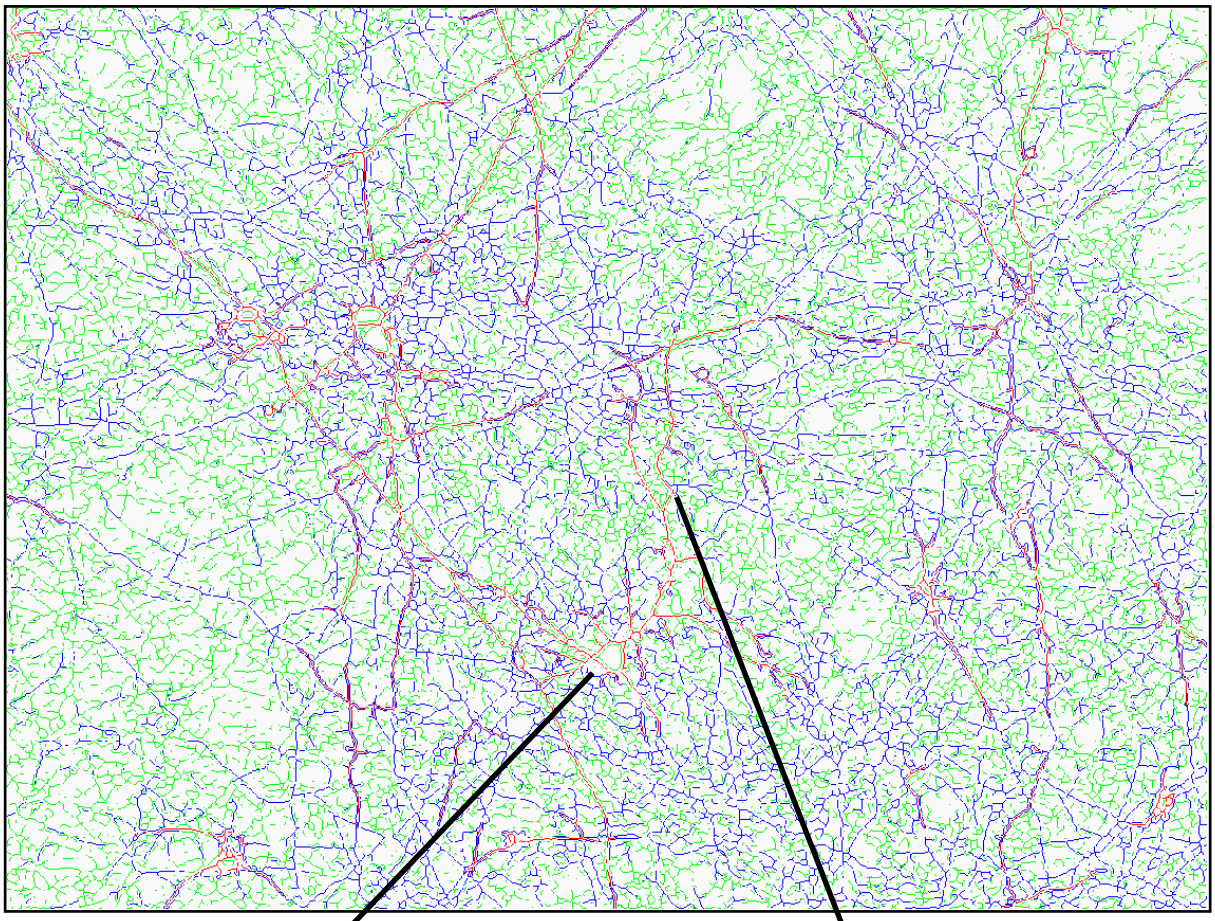


Figure 7.1 – The skeletonized image

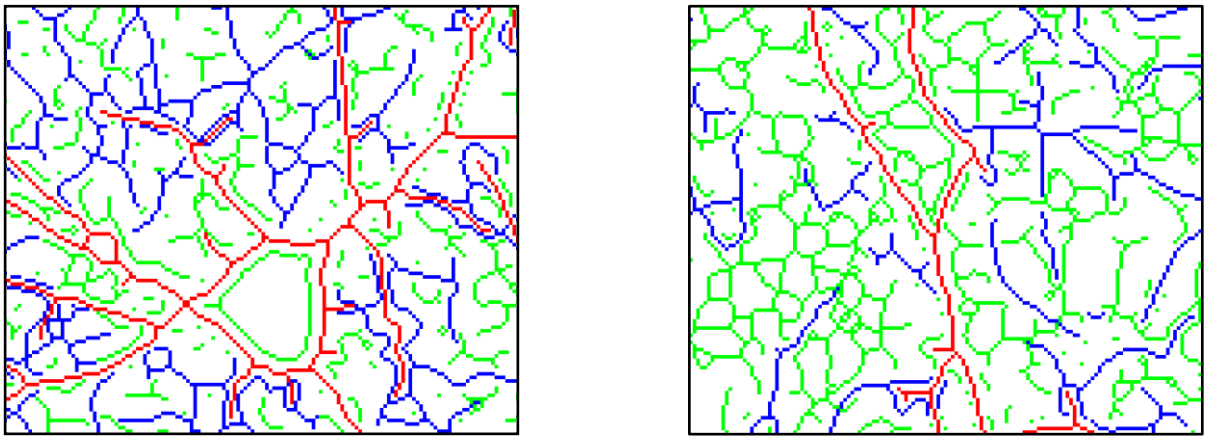


Figure 7.2a, 7.2b – Magnified areas of the skeletonized image

7.2 Neuron cell bodies and Small Beads total area

Neuron cell bodies (NCB) and Small Beads (SB) total area is computed as the sum of pixels in which the label is respectively brown for NCB and yellow for SB. The calculation of the total length of both NCB and SB is computed by means of a function that runs through the pixels of the image resulting from the morphological post-processing of FI2, looks for the pixels which label color *C* equals the color with which either NCB or SB were labeled, and, whenever such equality is verified, increases by one the total area value of that object.

7.3 Neuron Cell Bodies and Small Beads total number

Neuron cell bodies (NCB) and Small Beads (SB) total number is computed as the sum of objects whose structure was labeled respectively with brown (for NCB) and yellow (for SB). A function runs through the pixels of the image resulting from the morphological post-processing of FI2, detects individually singular objects labeled with those label colors (areas labeled with the given label completely surrounded by area labeled with the background label) and, whenever such a objects are found, increases the respective total amount value by one.

More detailed descriptions of the functioning of

- skeletonization function
- calculation of the total area of the desired features;
- calculation of the total amount of NCB and SB

are given in Chapter 8 “Implementation and Running of the IAP’s methods.

PART III

Methods Implementation

Chapter 8

Implementation and Running of the IAP methods

This section provides a further explanation of the implementations of the Image Analysis Protocol (IAP) methods chosen, by a description of the practical functioning of algorithms and functions which perform the pre-analysis and analysis steps previously introduced. The IAP methods have been implemented in C++ functions, which uses methods and functions of the *CImage Class* (see Chapter 1). Therefore, as TestImage is an object of the CImage Class, it can perform each function of the VIS' *Imaging Utilities* library, which are mainly image handling, basic manipulation and displaying. In this chapter, the algorithmic steps of most of the functions are explained by using pseudo code, and no source code is included in this chapter. A brief explanation of the meaning of the main parameters is provided as well as an explanation of the reason why particular values of these parameters were chosen. A description of resulting images from each application step is also given, showing which function allows which particular outcome to be achieved, as well as the final measured values of the features.

8.1 Pre-Analysis algorithms

The purpose of the pre-analysis stage is to obtain important information about the image format and content. Here follows a description of algorithms that carry out the computation and the extraction of such values.

8.1.1 Main Color Band

Main color band (MCB) – that for images used in this project was the red or the green one - is a variable name denoting which color band is the main in the image. It is an integer value that has been calculated by performing a comparison of the pixel intensity values on the three main color band (RGB). A function runs through the image's pixels, and for each of them it compares the three values of the three color bands. For each pixel, the band whose value is the highest is considered the main band and the value of the respective counter variable gets increased by one. This simple method was chosen after a visual inspection on images by using VIS. Since it was noticed that most of the content of images was concentrated in one main band and the remaining two bands did not have any useful information to describe the desired features, it was decided to not use their content. The lack of information on the non-main bands is relied to the fact that fluorescent image capturing results in information in one band, which color depends on the kind of fluorescent substance used to stain the cultures (es

red for tdTomatoe, green for GFP). Hence the only band that gets used is the main color band.

8.1.2 Magnification Value

The magnification value is passed as input value at the beginning of the execution of the program. It is a variable that can assume the following values: 10, 20, 40. This variable is then passed to the classification function and to the morphological post-processing function, which load the correct set of parameter to be used to obtain the desired results for that particular kind of magnified image.

8.2 Pre-Processing algorithms

Pre-processing algorithms consist of filtering functions which perform the two kind of filtering operations explained in subchapter 4.4.

8.2.1 Polynomial Local Linear Filter function

In order to localize and extract the linear structures - which were neurite branches (NB) and most of Small Beads (SB) - a function named *PolyFilter* (from Imaging Utilities) has been run on the original image.
The main parameters it receives as input are:

- the original image (an object of CImage Class)
- the main color band of the image (an integer value)
- the filter's Kernel size and Order (integer values)

The main color band is an integer value which ranges from 0 to 2 (0 stands for the red band, 1 for green band and 2 for blue band) which indicates the band whose content is filtered by a PLLF filter which main input parameters are the given values of Kernel Size and Order.
By inputting the Test Image as original image, the value "1" as main color band and Kernel size and Order of respectively 15 and 3, it is possible to obtain the best result in terms of linear shapes detection and view (see Fig. 4.7b).
The more the Kernel size parameter value get increased, the wider the squared area of pixels around the central pixel. All the values of the pixel contained in this area are used to compute the result of filtering transformation (see chapter 4) and that result is the value given to the central pixel of the window. In general, the wider is this area, the more blurry the resulting

image becomes. The output image is an object of the CImage class, as well as the input image which it has been generated from. The outputted image has:

- Only one band(with intensity values ranging from about -20 up to 20)
- no overlaid label layer

8.2.2 Median Filter function

In order to localize and extract the round objects - which were neuron cell bodies (NCB) and some of the Small Beads (SB) - a function named *ModusFilter* (from Imaging Utilities has been run on the original image. The main parameters it receives as input are:

- the original image (an object of CImage Class)
- the main color band of the image (an integer value)
- the filter's width (horizontal size) and height (vertical size) (both integer values)

By inputting the Test Image as original image, the value "1" as main color band and width and height value both of 7, it is possible to obtain the best result in terms of linear shapes detection and view (see Fig. 4.7c).
As in the general the wider the window utilized to compute the filtering operation is, the sharper the resulting image becomes.. The lower such values are, the more similar to the input image the resulting image will look. The output image is again an object of CImage class, as well as the input image which it has been generated from. The outputtet image has:

- Only one band (with intensity values ranging from 0 up to 255)
- no overlaid label layer

8.3 Classification algorithms

Classification algorithms consist of functions performing a multiple thresholding on both original image and filtered image, in order to carry out the two kinds of classification operations explained in subchapter 5.5.

8.3.1 Function for the Classification of linear structures

The purpose of this function is to classify the linear structures into the three main classes based on the image outputted from the PLLF function. The thresholding function runs through each pixel of both the original (OI) and the filtered image (FI) and, in function of the relative values of each pixel, it set the value of the label of that pixel to one of the four label values, using the upper and lower boundary values given in Table 5.1a and 5.1b. The algorithm works as follows:

(1) FOR each pixel of both images

- IF (2.a) AND (2.b) – (using the boundary values for class1 neurites)

  where:
  (2.a): the OI's pixel intensity value is included between the upper and lower boundaries which characterizes a particular class, relative to the OI

  (2.b): the FI's pixel intensity value is included between the upper and lower boundaries which characterizes a particular class, relative to the FI
  THEN

  DO: LABEL that pixel with the class 1 label value

(3) ELSE IF (2.a) AND (2.b) – (using the boundary values for class 2 neurites)

(same as before, but for class2 neurites)

(4) ELSE

DO: LABEL that pixel with label value 0 (background).

These labels are identified with integer numbers within VIS' Image Utilities libraries, the label used for this particular classification were identified with the following numbers:

3 - **red**- for **class1** NB; 2 - **blue**- for **class 2** NB; **1** - **green**- for **class 3** NB

This function:

- receives as input the Original Image and the PLLF filtered image, which is an objects of the CImage class with one band and no labels

- outputs another CImage object, the classified image, which is a copy of the PLLF filtered image and has:

    - one band (with pixel intensity values ranging from about -20 up to 20)

    - the overlaid layer of the labels, given by four colors:
        - the three different class colors, used to label the neurite branches;
        - the white, used to label pixels belonging to the background.

The classes' upper and lower boundaries are contained in variable of type variable.

8.3.2 Function for the Classification of round objects

This function performs a classification of round objects in an image and works similar to the function performing a classification of linear structures described above. The differences are that it receives as input the MF filtered image and, by using upper and lower boundary values given in Table 5.2a and 5.2b, it outputs an image which features are labeled with two main different labels which are:

5 - brown and **51 – light blue** both for NCB, with different pixel intensities.

The temporary label **51** is useful in order to perform particular morphological operations (see 8.4.2).

This function:

2) receives as input the Original Image and the MF filtered image, which is an object of the CImage class with one band and no labels

2. outputs another CImage object, the classified image, which is a copy of the MF filtered image and is composed of:

    - one band (with intensity values ranging from 0 up to 255)

    - the overlaid layer of the labels, given by three colors:

        - the two different class colors, used to label objects with different pixel intensity

        - white, used to label pixels belonging to the background.

8.4 Morphological Post-processing algorithms

The following morphological post-processing algorithms carry out a shape enhancement of the features which have been classified during the Classification step. The latter consist of the application of an ordered sequence of functions which work on the label layer of the classified image, outputted by the classification function, as explained in sub-chapter 6.3. The set of functions utilized consists of the following functions:

- erosion and dilation
- area and irregularity conditions
- change of label for objects surrounded from other objects,for a given percentage

Erosion and dilation

Erosion and dilation are implemented in the same function (*DrawObjectBorder*, from Imaging Utilities), whose main parameters are:

- the label layer of the classified image (passed as a pointer to a pointer of a byte variable effectively pointing to a byte double array, therefore the function modifies the image's label without working on the pixel intensity of the image)
- the value of the label that must be eroded/dilated (an integer value)
- the value of the label that must erode/dilate (an integer value)
- the value of the quantity of pixels that will be erode from the structuring element, from the eroding label pixels (an integer value)

Area and Irregularity conditions

This Function (edited version of *FilterObjects*, from Imaging Utilities) performs a selection of objects dependent on their area and irregularity . The main parameters for the function are:

- the classified image's label layer (passed as a pointer to a pointer of a byte variable effectively pointing to a byte double array, therefore the function modifies the image's label without working on the pixel intensity of the image)
- the value of the label with which the objects of interest are labeled
- the value of the label with which the objects of interest will be labeled if they do not meet the condition

- the minimum value of area/irregularity used as condition (a double value)

- the value of the method selector variable, which allows the function to work on the area or the irregularity values of the objects (an integer value)

Change of label for objects surrounded from other objects, for a given percentage

This function (*ChangeSurroundedObject*, from Imaging Utilities) works by:

1. Detecting objects that are labeled with a given label *A* (with label value given as input, as an integer variable) which are surrounded by pixels labeled with label B by a minimum percentage. This area coverage value can range from 0, that means that there is no connection between the two kinds of labels for the object, to 1 which means that label A is completely surrounded by label B, and is also passed as a double variable.
2. Turning into B the label value of the object labeled with A.

8.4.1 Functions of post-processing of the classification of the PLLF filtered image

The post-processing function working on the classified PLLF-filtered image (I1) consists of the ordered sequence of functions listed below, by using the labels in Figure 1 and the structuring element of figure 2.

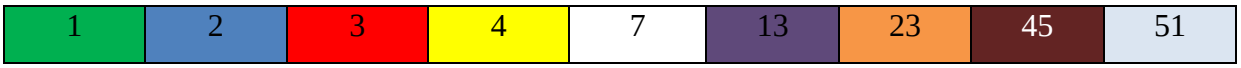
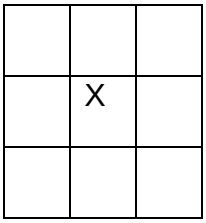


Figure 8.1 – List of labels used to perform operations of erosion and dilation



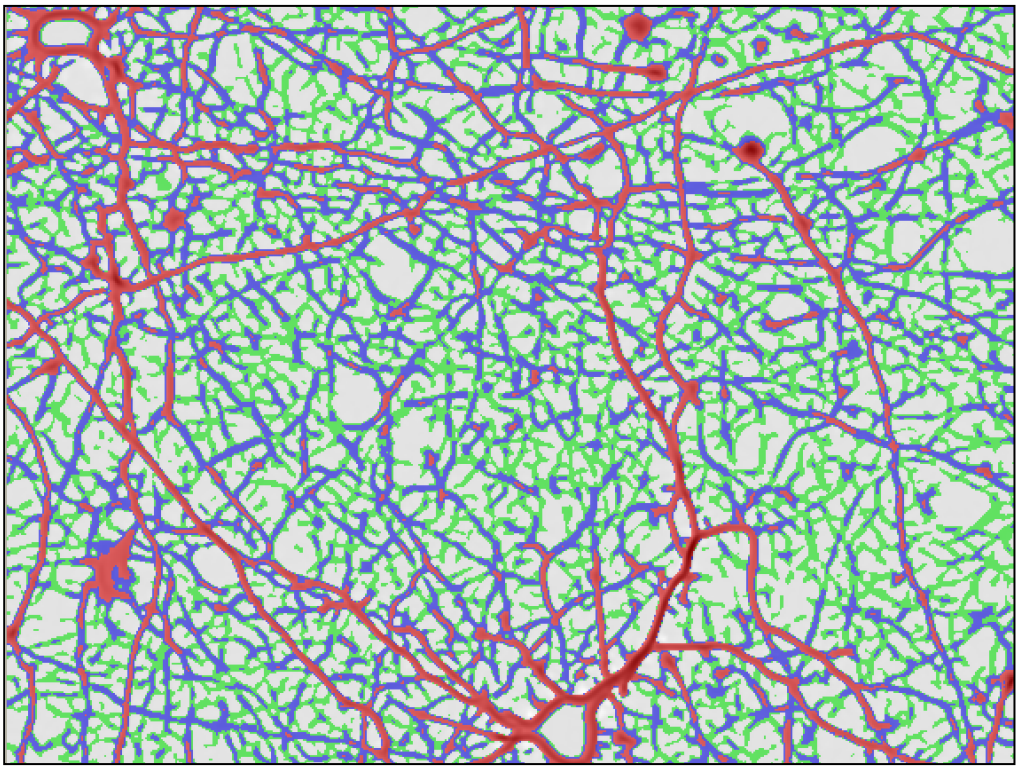
Structuring element used: the central pixel C is the starting point from which operations of erosion and dilation are performed, in each directions for all of the pixels surrounding pixel X

Figure 8.2 – Structuring element used to perform operations of erosion and dilation

1) The goal of the first stage is to label class 1 neurites (label 3) and the biggest round structures (label 23) hat will be reclassified as beads later.

1.1) Dilation of objects of label 2 with label 2 by 1 pixel; (growth of blue labeled area, by turning into blue every non-blue pixel connected with a blue pixel).

1.2) Dilation of objects of label 3 with label 3 by 1 pixel; (growth of red labeled area, by turning into red every non-red pixel connected with a red pixel).

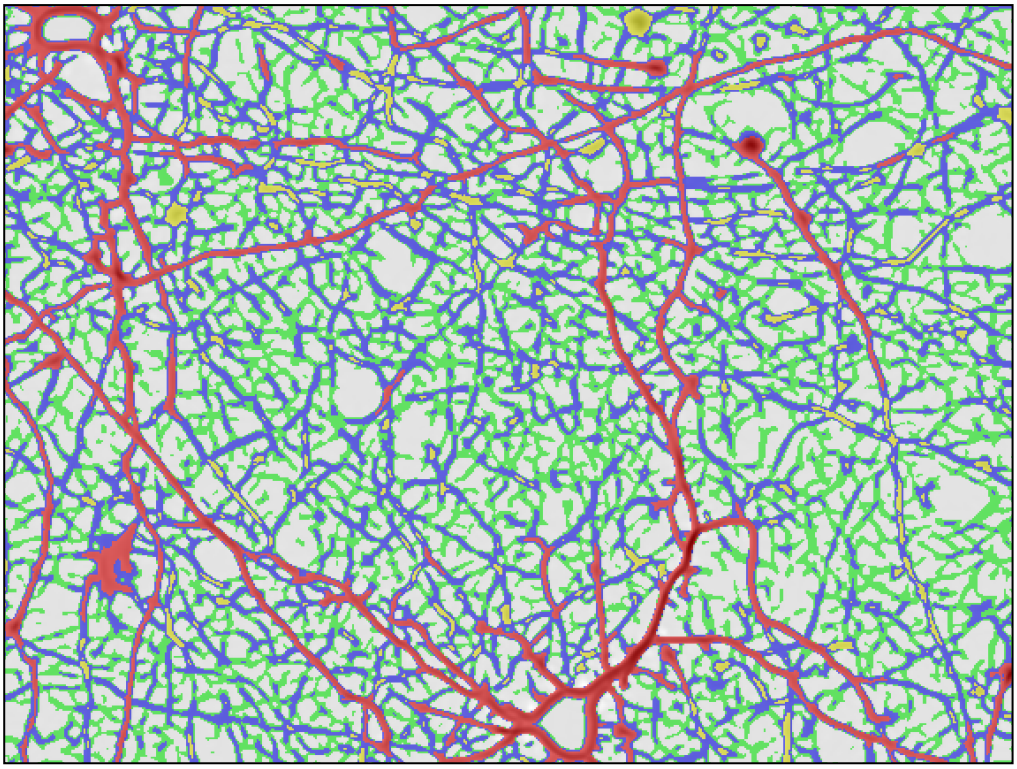


➔ **Figure 8.3 -** It is possible to observe that a both red and blue labeled structure are now more connected each other (red with red, blue with blue)

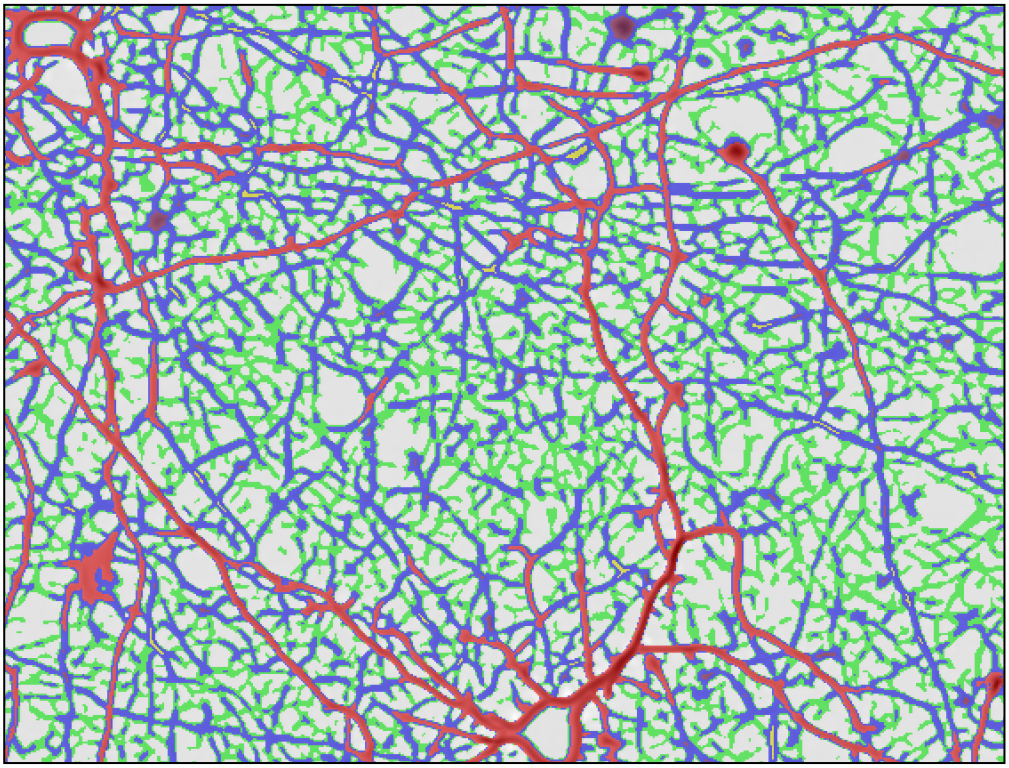
1.3) Minimum Area condition (area > 210 pixel) on the 3 labeled objects; all objects not meeting this condition are assigned label 4 (red labeled objects whose area is smaller than 210 pixel will be labeled with yellow, see Fig. 8.4).

1.4) Erosion of objects of label 4 with label 2 by 1 pixel; (yellow areas are eroded by blue).

1.5) Maximum Irregularity condition (irregularity < 3.1) on the 4 labeled objects; (yellow labeled objects whose irregularity coefficient is greater than 3.1 will be labeled with purple, see Fig. 8.5).



➔ **Figure 8.4 -** By a comparison with the former image it is possible to notice that some of the former red neurite is now labeled with yellow



➔ **Figure 8.5 -** After these two operations, most of the yellow labeled objects which structure is fairly regular (most of the round objects) are labeled with purple

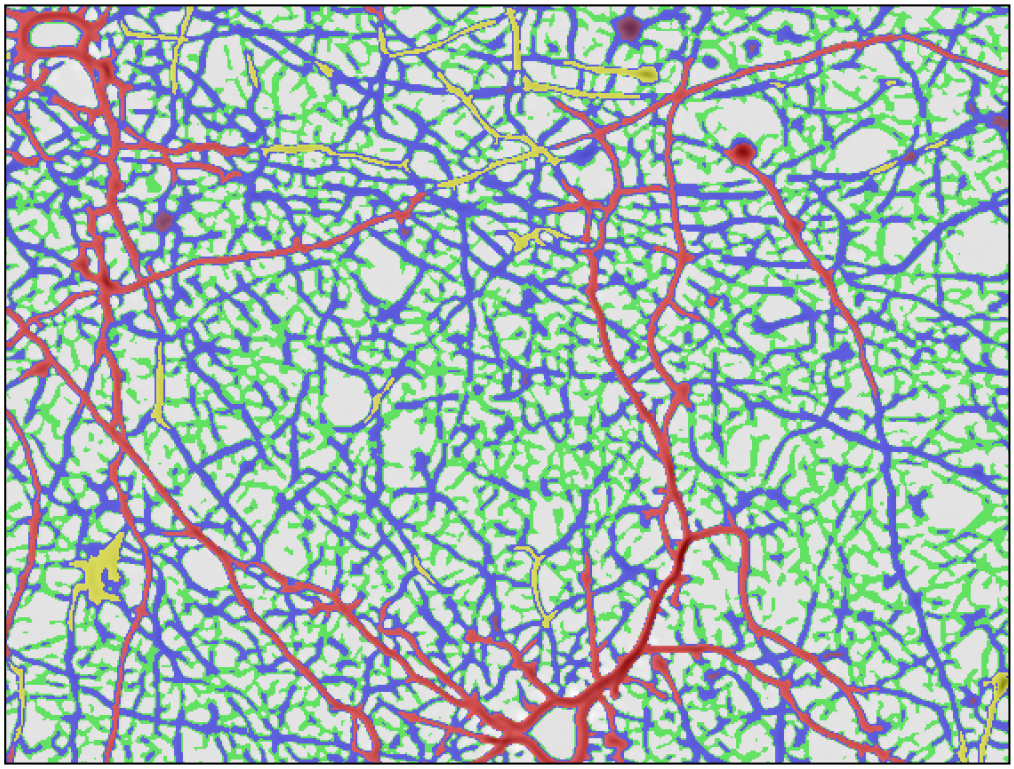
1.6) Minimum Area condition (area > 9 pixels) on the 13 labeled objects; (purple labeled objects whose area is smaller than 9 pixels will be labeled with blue).

1.7) Change of label (from 4 to 2): yellow labeled objects will be labeled with blue.

1.8) Dilation of object of label 2 with label 2 by 1 pixel; (growth of blue labeled area, by turning into blue every non-blue pixel connected with a blue pixel).

- ➔ The purple round objects smaller than 9 pixels are labeled with blue.
- ➔ Most of the yellow labeled object are now non-round objects, or vey irregular objects: they are labeled with blue

1.9) Minimum Area condition (area> 1200 pixel) on the 3 labeled objects; (red labeled objects whose area is smaller than 1200 pixels will be labeled with yellow).



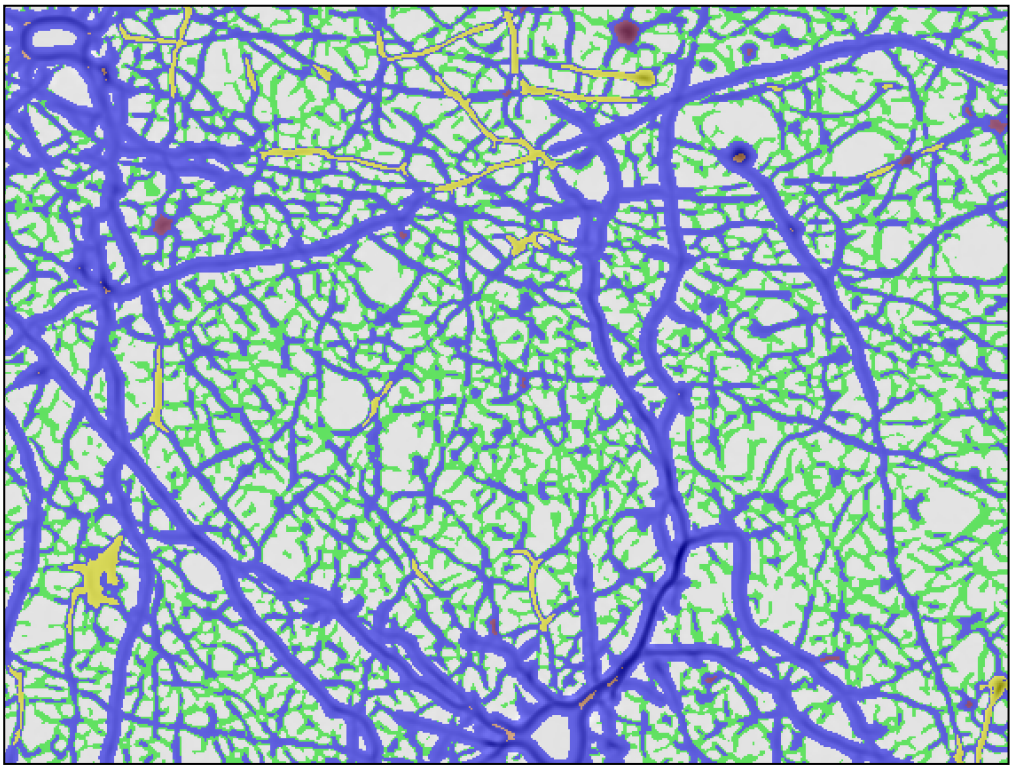
➔ **Figure 8.6 –** Smallest red labeled areas have been labeled with yellow

2) The goal of this stage is to locate Small Beads and to label them with yellow

2.1) Creation of I2: a copy of I1 and its label layer, to work on the shape re-modeling of the Small Beads, SB.

2.2) Change of label (from 3 to 23): red labeled objects will be labeled with orange.

2.3) Erosion of objects of label 23 with label 2 by 6 pixels; (orange areas are eroded by blue, by turning into blue the outer pixels of every orange labeled object, and repeating this sequence on the obtained label layer for 5 times).



➔ **Figure 8.7** - Most of the orange labeled object are eroded completely by blue. It is possible to notice that, among the remaining orange-labeled objects, there are some objects that should be labeled as beads

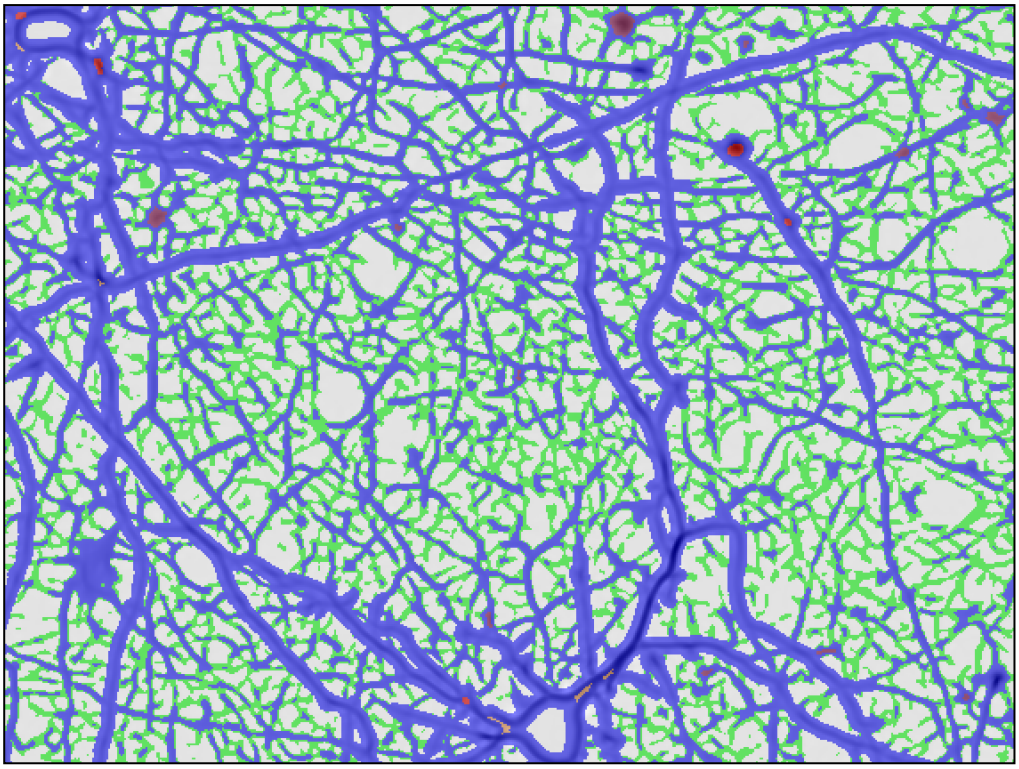
2.4) Maximum Irregularity condition (irregularity < 3) on the 23 labeled objects; (orange labeled objects whose irregularity coefficient is smaller than 3 will be labeled with red).

➔ Most of the orange labeled object are now non-round objects, or vey irregular objects

2.5) Change of label (from 4 to 2): yellow labeled objects will be labeled with blue.

2.6) Dilation of objects of label 3 with label 3 by 3 pixel; (growth of red labeled area, by turning into red every non-red pixel connected with a red pixel, and repeating this sequence on the obtained label layer for 2 times).

2.7) Minimum Area condition (area > 13 pixel) on the 3 labeled objects; (red labeled objects whose area is smaller than 13 pixel are labeled with blue).



➔ **Figure 8.8 -** After an area growth, smaller regular objects labeled with red are labeled with blue

2.8) Change of label (from 3 to 4): red labeled objects will be labeled with yellow.

2.9) Change of label (from 23 to 2): orange labeled objects will be labeled with blue.

➔ The remaining red labeled objects are regarded as Small Beads, therefore are labeled with their own class color, that is yellow

➔ The remaining orange labeled objects are labeled with blue

2.10) Dilation of objects of label 4 with label 4 by 2 pixels; (growth of yellow labeled area, by turning into yellow every non-yellow pixel connected with a yellow pixel, and repeating this sequence on the obtained label layer once).

➔ Increasing of the Area of the objects classified as Small Beads.

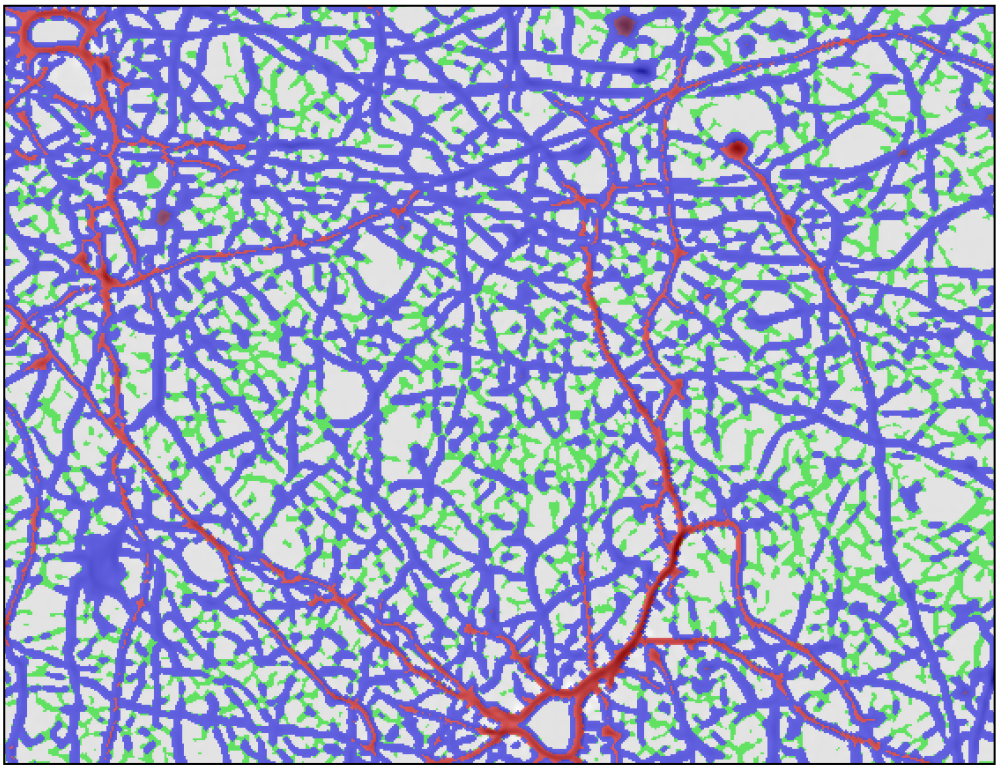
3) The goal of this stage is to enhance the shape of the blue and green labeled neurites

3.1) Creation of I3: a copy of I1, to work on the shape re-modeling of the class 2 and class 3

Neurite Branches.

3.2) Change of label (from 4 to 2).

3.3) Dilation of objects of label 2 with label 2 by 3 pixel.



➔ **Figure 8.9** – Growth of the blue labeled area

3.4) Erosion of label 2 with label 7 by 1 pixel: (blue neurites areas are eroded by background, see Figure 8.10).

3.5) Minimum Area condition (area> 100 pixel) on the 2 labeled objects; (blue labeled objects whose areas are smaller than 100 pixel will be labeled with green, see Fig. 8.10)

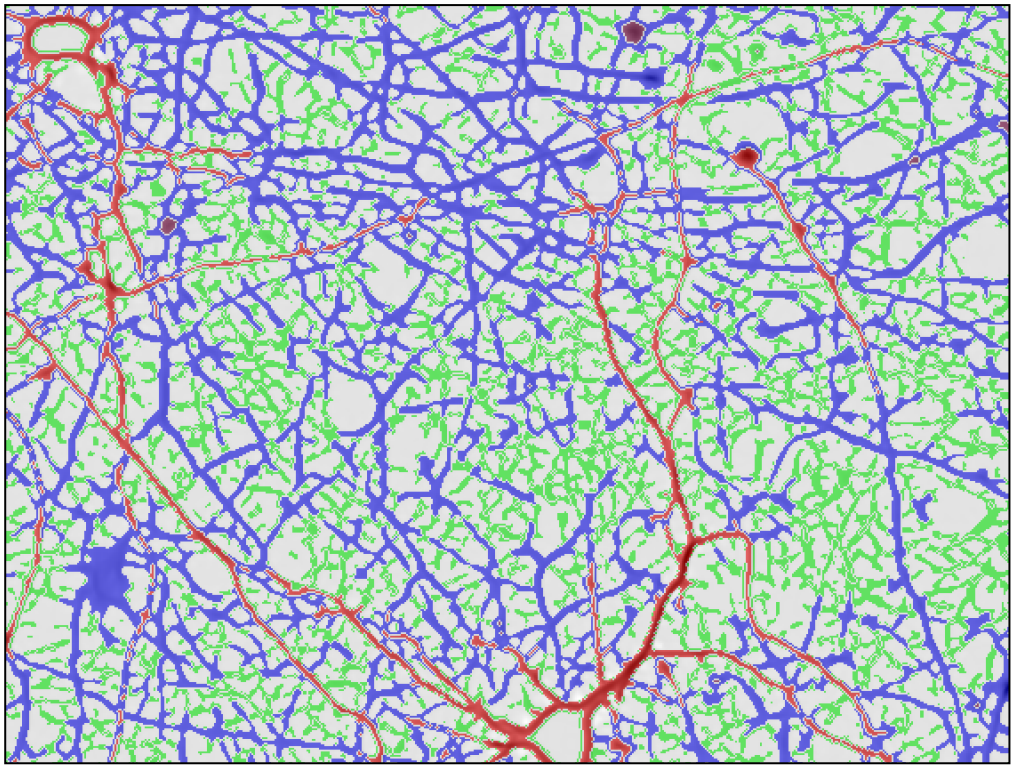
3.6) Dilation of objects of label 1 with label 1 by 1 pixel; (growth of green labeled area, by turning into green every non-green pixel connected with a green pixel, and repeating this sequence on the obtained label layer once).

4) The Final Image FI1 is created using the following pseudo code:

4.1)

(1) FOR each pixel

(2) IF label value of I2 equals 4



➔ **Figure 8.10 -** Smallest blue neurites are downgraded as class 3 neurites

THEN

DO: LABEL that pixel with the label value 4 (yellow);

(3) ELSE IF label value of I1 equals 3 or equals 13

THEN

DO: LABEL that pixel with label value 3 OR 13

(4) ELSE DO: LABEL that pixel with the label value of the corresponding pixel in I3.

4.2) Minimum Area condition (area > 11 pixel) on the 13 labeled objects; (purple labeled objects whose area is smaller than 11 pixel will now be labeled with blue).

4.3) Change of label (from 13 to 4): purple labeled objects will be labeled with yellow.

- ➔ Labeling of new objects as Small Beads.
- ➔ Performing these steps the final morphological-post-processed classified PLFF filtered image (FI1) is obtained.

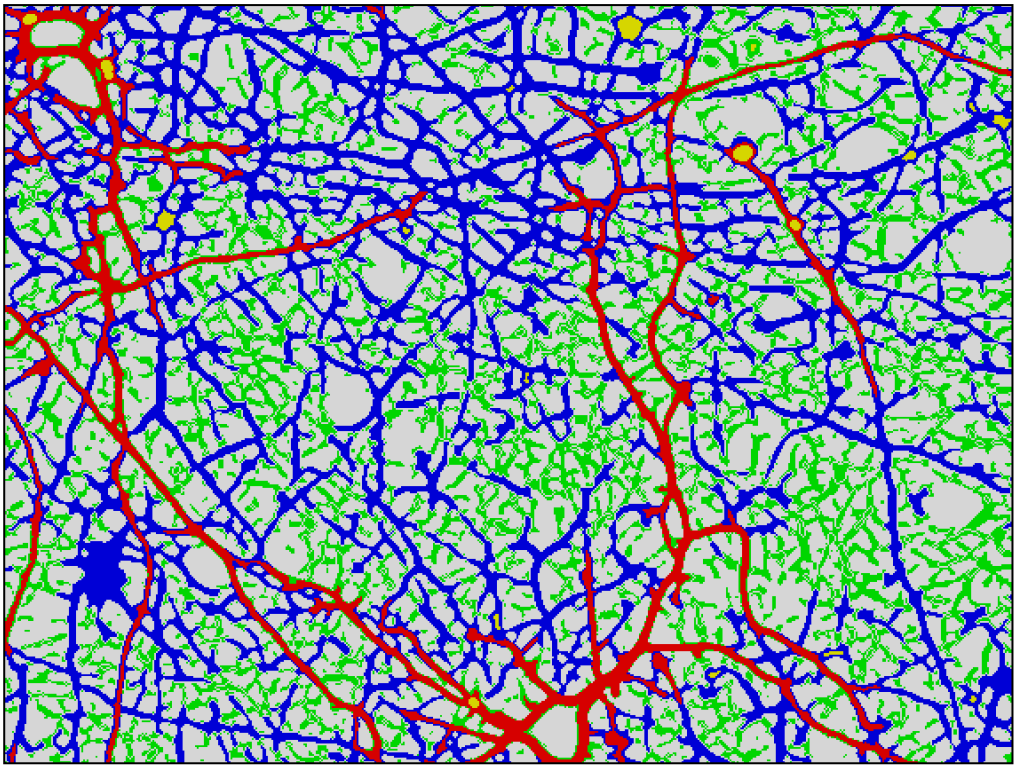


Figure 8.11 – Final version of the FI1 image: the former images look less bright than this because they were overlapped to the filtered image (which is fairly dark), in order to evaluate whether the morphological modification of the labels matched with the actual shapes of the linear structures below, or not.

8.4.2 Functions of post-processing of the classification of the MF filtered image

The post-processing function working on the classified MF-filtered image consists of sequence of functions listed in the follow.

5) The goal of this stage is to label the biggest neuron cell bodies (label 8)

Working on a copy of the classified MF filtered image, where the brown label are present:

5.1) Erosion of objects of label 45 with label 7 by 10 pixel; (brown areas are eroded by background).

5.2) Dilation of objects of label 45 with label 45 by 14 pixels; (growth of brown labeled area, by turning into brown every non-brown pixel connected with a brown pixel).

- ➔ At the end of this step, the remaining brown labeled object will have a shape that is more round than their initial one; parts of neurites that were labeled with brown have been eroded by white. The labeled shapes of the Neuron cell body labeled are also quit similar to their original shapes.

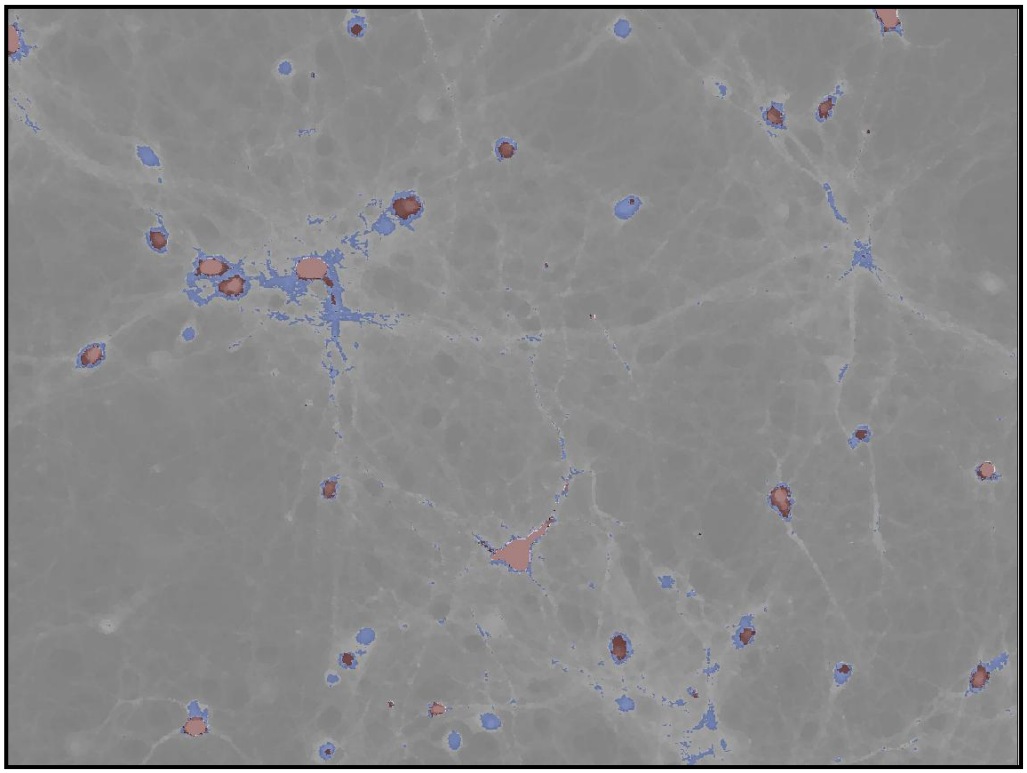
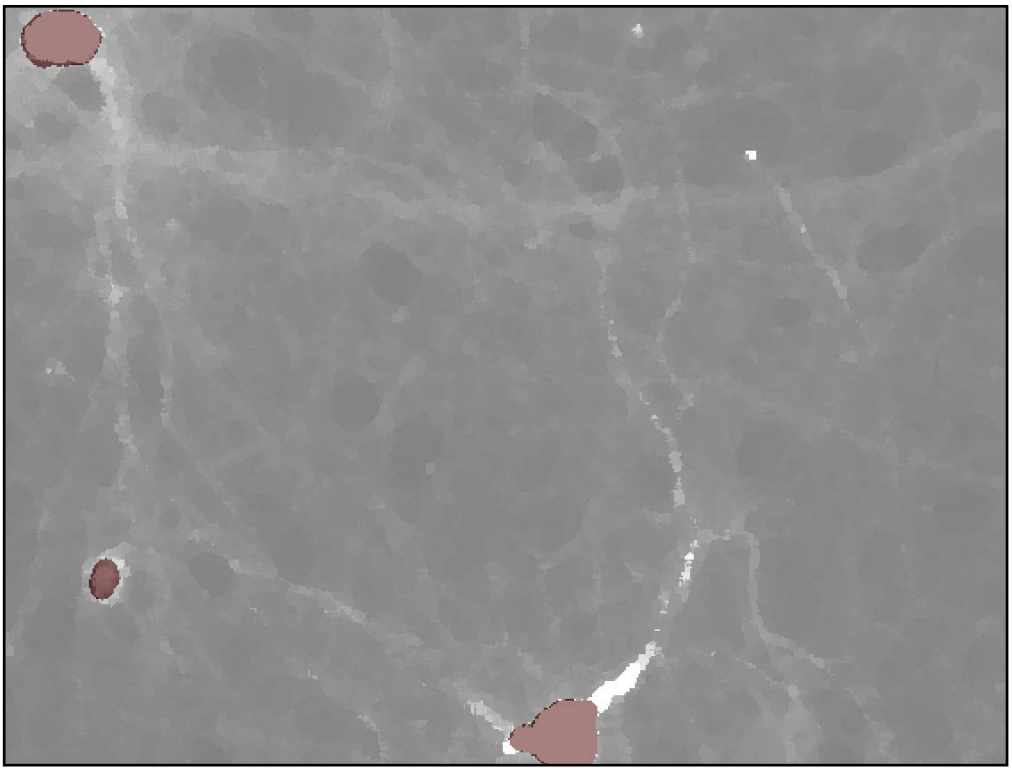


Figure 8. 12 – The classified MF-filtered image; the next stage will work only on the brown labeled object.



➔ **Figure 8.13** – At the end of this step, parts of the remaining brown labeled object will have a shape that is more round than their initial one; parts of neurites that were labeled with brown have been eroded by white. The shapes of the labeled neuron cell body are also quit similar to their original shapes.

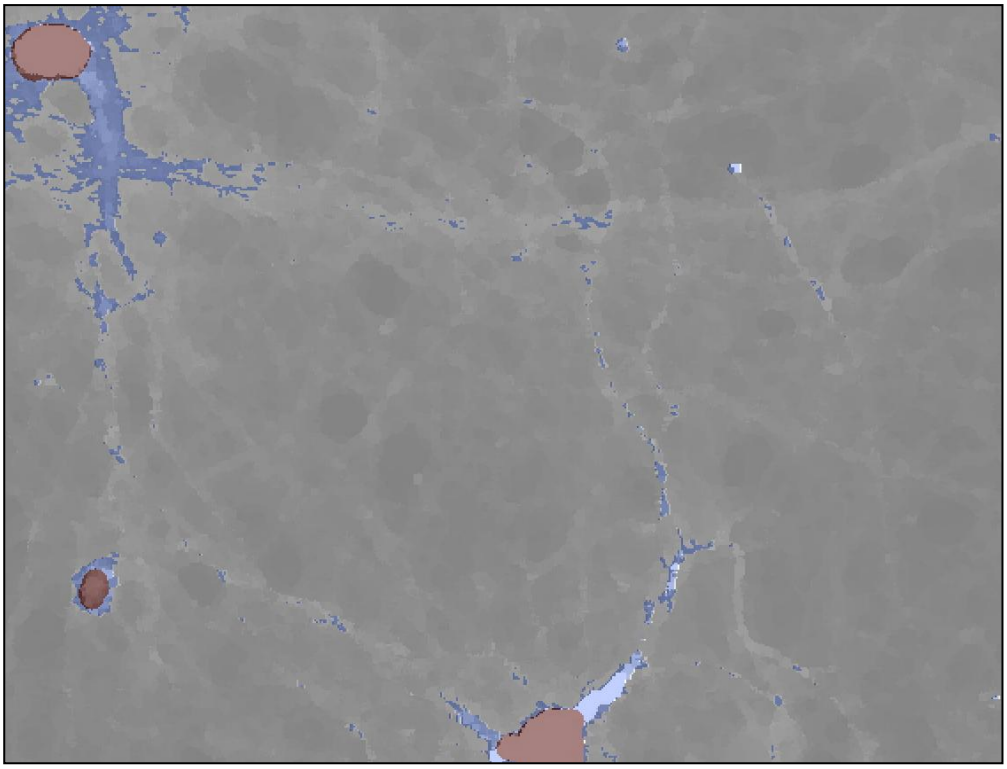
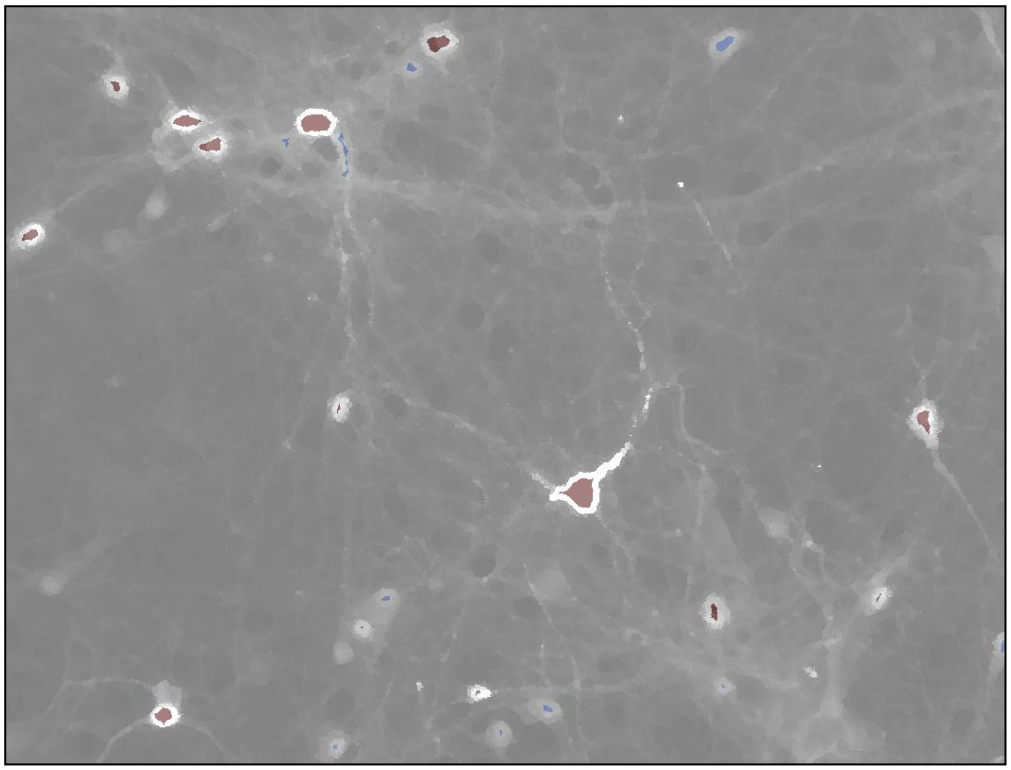


Figure 8. 14 – A reclassification of the non-brown labeled object is performed.

5.3) Erosion of label 51 with label 7 by 14 pixels,



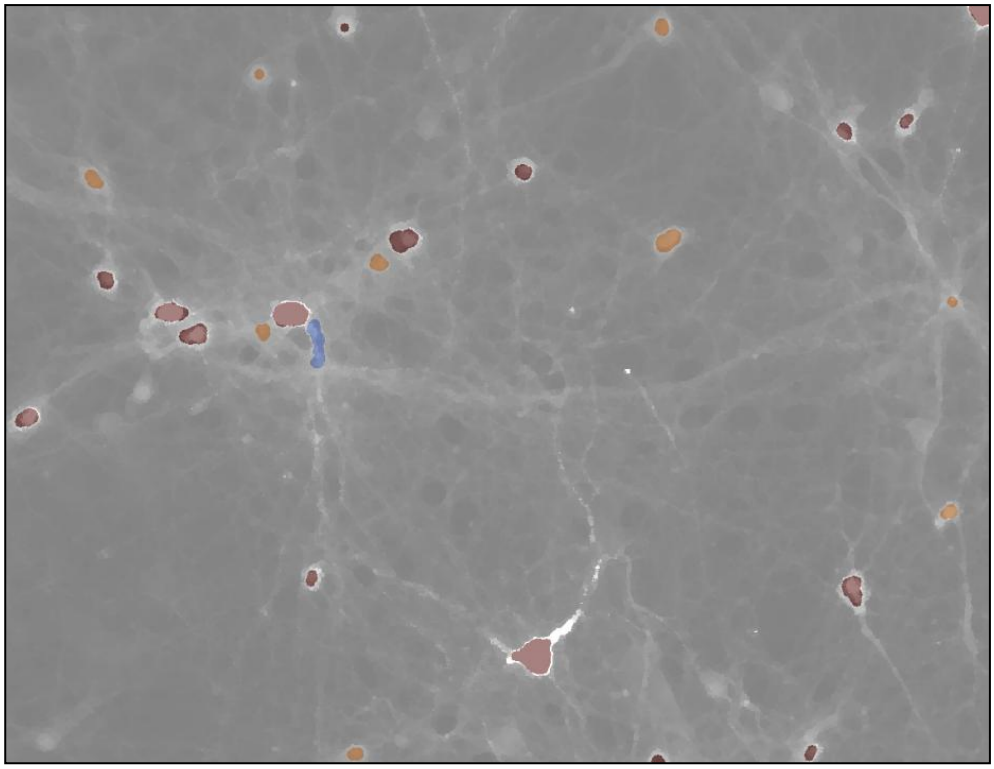
➔ **Figure 8. 15** – Some other features have been labeled with light blue

5.4) Dilation of label 51 with label 51 by 11 pixels.

5.5) Dilation of label 45 with label 45 by 10 pixels.

5.6) Change Surrounded Object: areas labeled with 7 which are completely surrounded by areas labeled with 51, will be labeled with 45:(white labeled areas completely surrounded by light blue labeled areas will be labeled brown).

5.7) Maximum Irregularity condition (irregularity < 2.5) on the 51 labeled pixels (light blue labeled objects whose irregularity coefficient is greater than 2.5 will be labeled with orange).



➔ **Figure 8. 16** – Light blue labeled objects which shape was also the most regular, have been labeled with orange. It can be noticed that most of the neuron cell bodies are now labeled either with brown on orange. Orange labeled objects will be labeled with the neuron cell bodies own class color, that is brown.

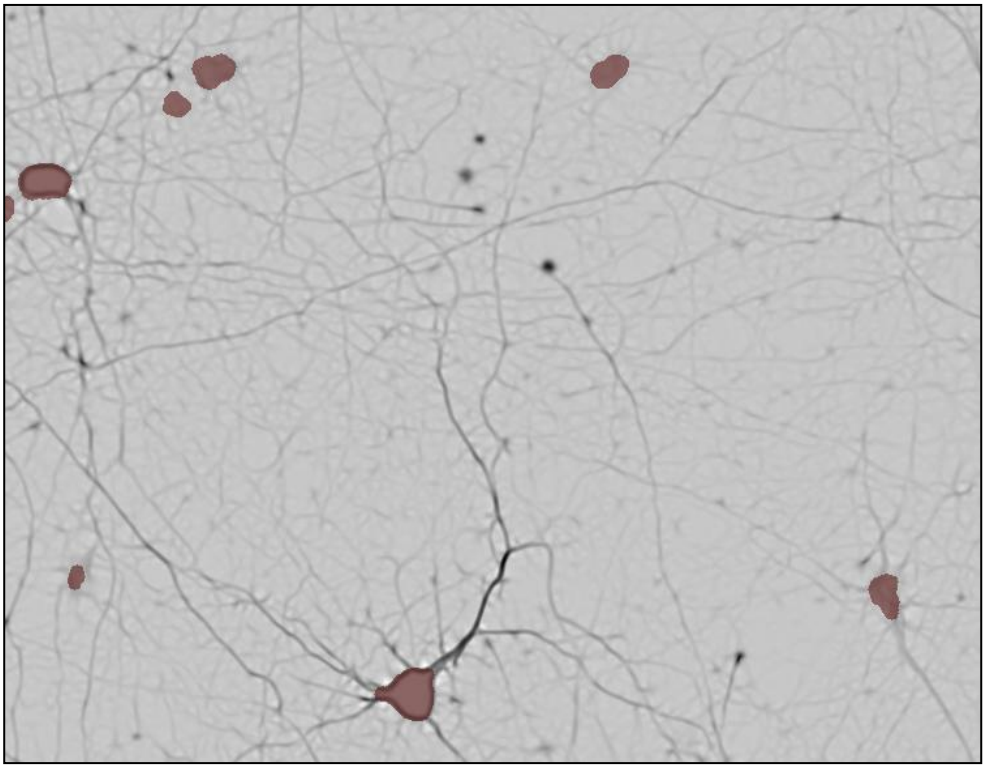
5.8) Change of label (from 23 to 45).

Performing these steps, the final morphological-post-processed classified MF-filtered image (FI2) is obtained (see Fig. 8.16).

8.4.3 Function for creating the final classified image

The function for creating the final classified image, receives as main inputs the following images:

- The final morphological-post-processed classified MF-filtered image (FI2)
- The final morphological-post-processed classified PLLF filtered image (FI1)



➔ **Figure 8.17 -** After these steps, the final morphological-post-processed classified MF-filtered image (FI2) is obtained.

For every pixel of the Final Classification Image (FCI) the desired label value, belonging either to FI2 or FI1 is selected as follows:

(1) FOR each pixel

(2) IF label value of I2 equals 45 (brown, with which the NCB have been classified)

THEN

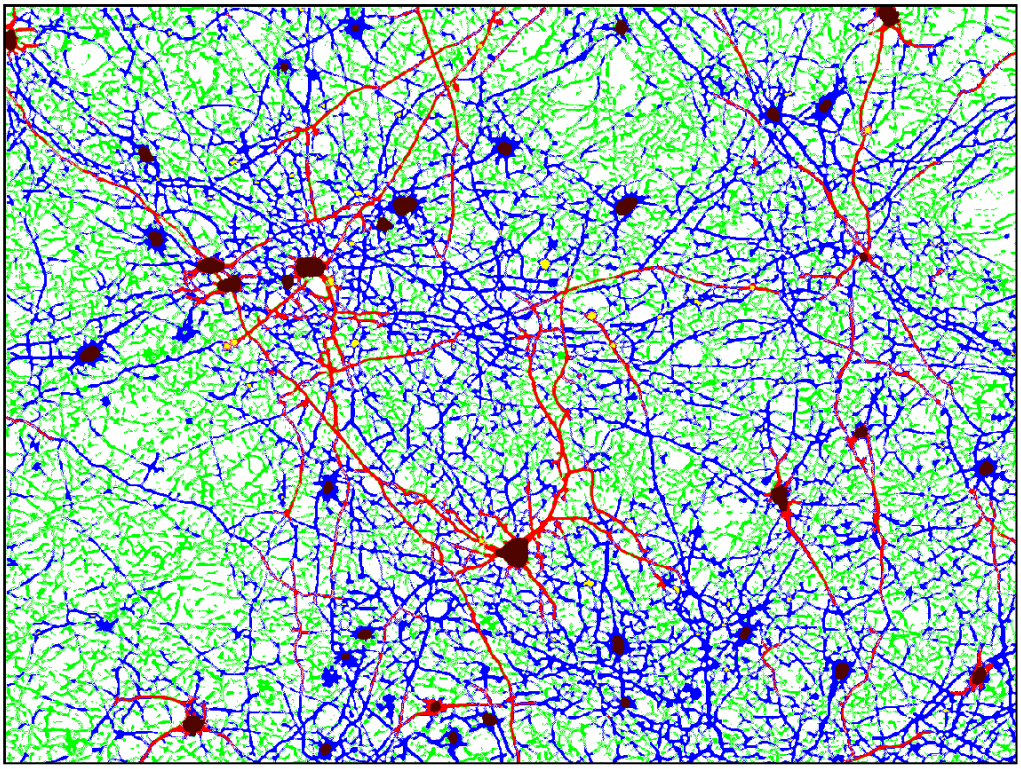
DO: LABEL that pixel with the label value 45

(3) ELSE

DO: LABEL that pixel with label value of the corresponding pixel in I1

Finally, the following morphological modification is performed on the FCI:

Change Surrounded Object: areas labeled with 4 which are in contact (even if only by one pixel) with areas labeled with 45, will be labeled with 45 (SB in contact with NCB will be labeled with the color of the NCB).



➔ **Figure 8. 18** - Performing these steps the following final classification resulting image (FCI) is obtained.

8.5 Measurement Post-processing algorithms

Measurement post-processing algorithm performs the skeletonization of the structures that compose the NB's, calculates the total area of NCB and SB and the number of such objects, as well. The new image which is an object of the CImage class is created, with:

- the skeleton of each NB
- the NCB (complete structure)
- the SB (complete structure)

The skeletonize function performs another morphological modification of the label layer on which it is run. The main parameters of the skeletonize function are:

- The label layer if the morphological post-processed classified image (passed as a pointer to a pointer of a byte variable effectively pointing to a byte double array, therefore the function modifies the image's label without working on the pixel intensity of the image)

- The value of the label with which the skeleton of the label of interest must be labeled.
- The value of the label with which the remaining part of the label of interest must be labeled (the non-skeleton part of the neurites will become this label)

This function modifies the morphological post-processed classified image's label layer, by outputting a new label layer, where the skeleton of the neurites is labeled with the same color which with was labeled the neurite it comes from, whereas the remaining part of the neurite has been labeled as background.

8.5.1 Output generation - NB total length and NCB area and SB area

The total length and area value is quickly computable by a simple function which runs through the pixels of the image and performs the following operations:

COMPARE the label value *V* of the current pixel (an integer value) with the following values:

- 1 (label value with which the skeleton of the Class 3 NB has been labeled)
- 2 (label value with which the skeleton of the Class 2 NB has been labeled)
- 3 (label value with which the skeleton of the Class 1 NB has been labeled)
- 4 (label value with which the SB have been labeled)
- 45 (label value with which the NCB have been labeled)

(2) IF *V* is one of the previously listed values;
(2.a) DO: INCREASE the counter variable concerning the area value (integer) of objects labeled with label of value *V* by one

ELSE

DO: MOVE ON next pixel.

8.5.2 Output generation - NCB total number and SB total number

The total number of neuron cell bodies and beads is computable by means of a function which performs a detection of objects labeled with a given label value *V* (which is either 45-light blue or 4-yellow, respectively for NCB and SB). This function runs through all pixels of the FCI and performs the following operations:

(1) COMPARE the label value *V* of the current pixel (an integer value) with the following values:

- 4 (label value with which the SB have been labeled)
- 45 (label value with which the NCB have been labeled)

(2) IF *V* is one of the previously listed values;
THEN

(2.a) DO: FIND all the pixels featuring the same label value which is connected (*) with the former pixel or a pixel connected with the former pixel; at the end of this process, an object which pixels are all connected to each other is found, then the algorithm will avoid identifying this object again when it meets another pixel which is part of that object)

(2.b) INCREASE the counter variable (integer) concerning the total amount value of objects labeled with label of value *V* by one.

(*) both 4 neighbor an 8 neighbor are available [10]

PART IV

Results and Evaluation

Chapter 9

Summary of Results and Conclusion

The C++ program gathering the implementations of the IAP methods, were run on all of the image in the database (including Training set and Test set). This chapter summarizes the results obtained by the application of the IAP methods to the three image groups in the Test set. It also explains the importance of the parameter calculated as ratio of the total length of neurite branches divided by the total area of the neuron cell bodies. This parameter, along with the values of the three main outputs, was used in order to evaluate the efficiency of the therapeutic agents administered to the lesioned cultures of neuronal tissue.

9.1 Results on the Test Set

The three following graphs represent the scattering of the values of the three main outputs computed by the program (total length of the neurite branches, total area of the neuron cell bodies and total number of beads) based on the 35 images in the test set (there are 5 images in each of the seven groups). In each graph, values on the ***x axis*** refer to the numbers of the groups and values on the ***y axis*** refer to the measured output, each measurement corresponds to an image. In table 9.1 it is possible to see the name, the number of toxin and the treatment which the seven groups of cultures were submitted to.

GROUP NUMBER	GROUP NAME	ADMINISTRES SUBSTANCES	QUANTITY
1	Control Cultures:	No toxins nor treatments	-
2	2mM 3NP	3NP Toxin	2 mM
3	2mM 3NP + 100nM P	3NP Toxin Paroxetine	2 mM 100 nM
4	2mM 3NP + 1uM P	3NP Toxin Paroxetine	2mM 1 uM
5	5mM 3NP	3NP Toxin	5 mM
6	5mM 3NP + 100nM P	3NP Toxin Paroxetine	5 mM 100 nM
7	5mM 3NP + 1uM P	3NP Toxin Paroxetine	2mM 1 uM

Table 9.1 – The seven groups of images composing the test set

For each group of images it is also possible to evaluate the values of all the other outputs of the protocol in graphs and tables, this can be found in sub-chapter 9.4. For both graphs and tables, the numbers of the groups are listed in Table 9.1. The group name is also the folder name that can be found in the folder named "Image Database" in the attached CD.

9.1.1 Total Length of the Neurite Branches

The total value of the length of the neurite branches was calculated as the sum of the lengths of class 1 and class 2 neurites in the final classification image, as explained in chapter 6. The graph of Fig. 9.1 shows total length values calculated based on the final classification images of the images in the seven groups. The Total Length values about the cultures treated with toxins only were, in average, the smallest, whereas the toxin plus paroxetine cultures images' total length values are more similar to the control cultures' ones.

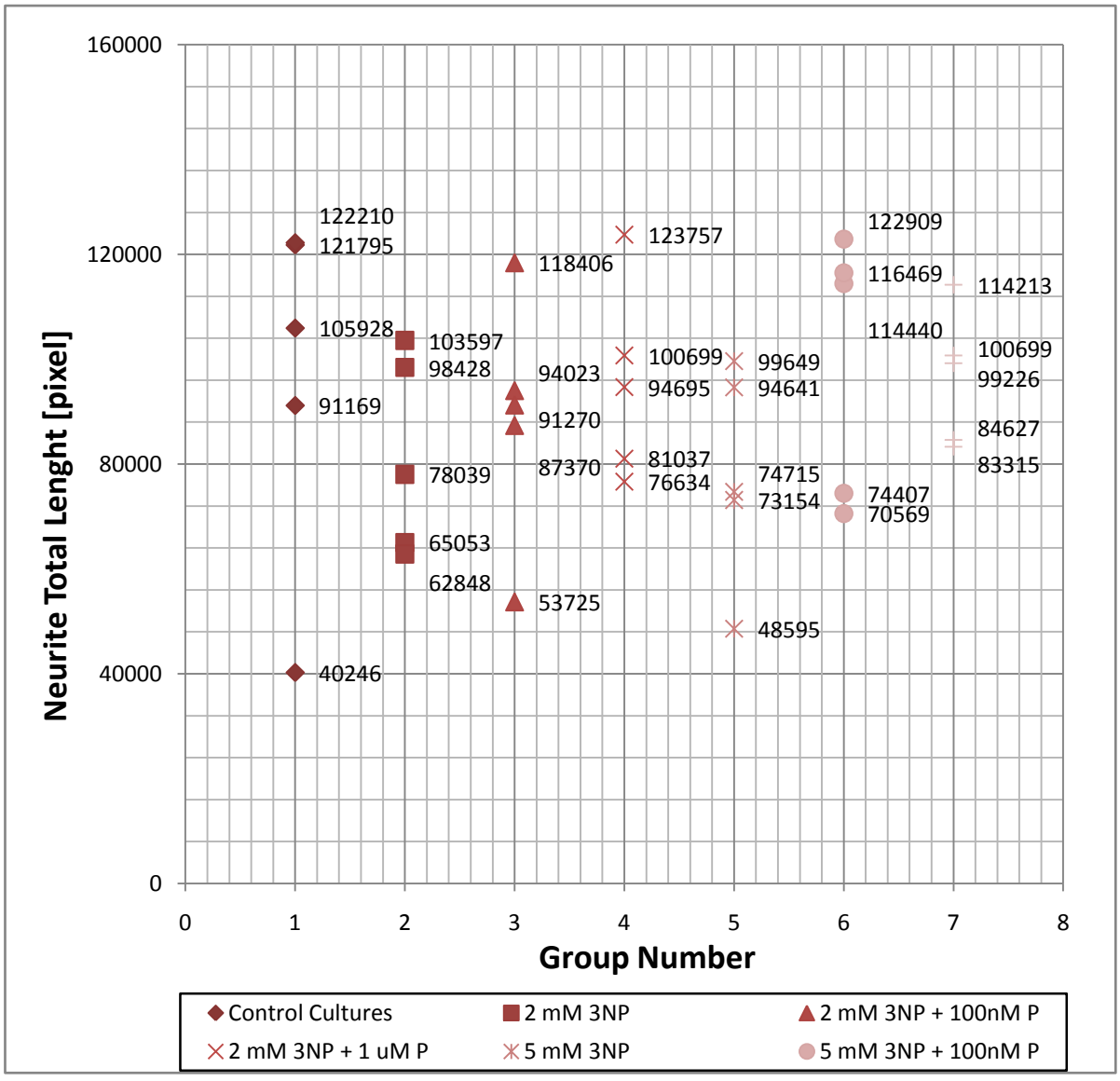


Figure 9.1 – Scatter graph of the total length of the Neurite Branches measured in five images per the seven groups of the Test set

9.1.2 Total Area of the Neuron Cell Bodies

The graph of Fig. 9.2 shows the total NCB area values calculated on the final classification images of the 35 images in the seven groups. As expected, the values for the cultures treated with toxins only, have (in mean) the smallest values of total NCB area, whereas the toxins plus paroxetine cultures' total area values are greater (in mean) than the ones measured on control cultures.

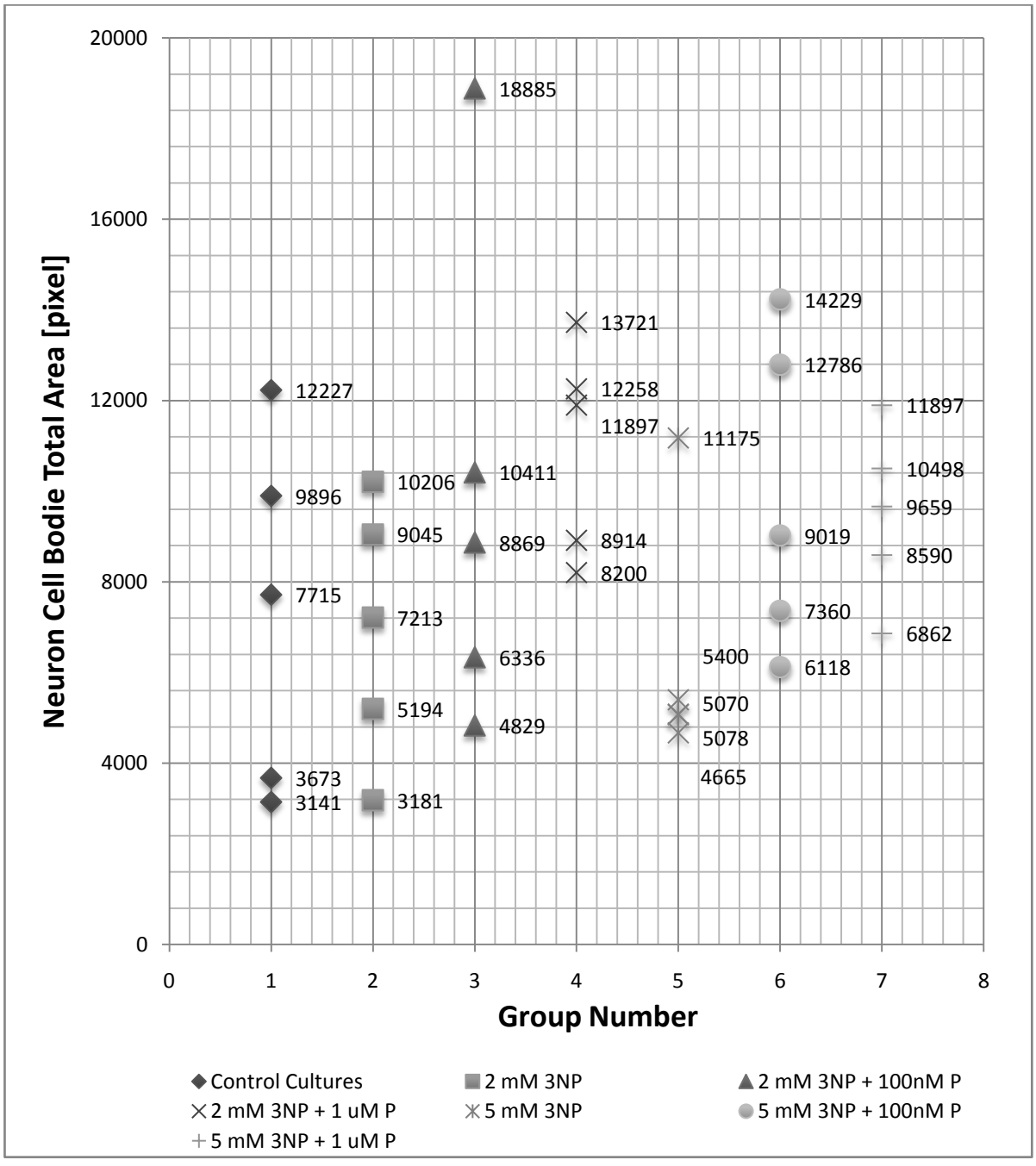


Figure 9.2 – Scatter graph of the total area of the Neuron Cell Bodies measured on the seven groups of the Test set

9.1.3 Total Number of Small Beads

The graph Fig. 9.3 shows the values of the total number of SB per image, calculated on the final classification images of the 35 images of the seven groups. The values calculated on the cultures treated with toxins only are high in the case of group 2 but it is fairly low in the case of group 5. Generally the total number of beads in the toxin-only treated groups is not greater than the one measured in the other groups of cultures that were submitted to a treatment. The values of the parameter in the control cultures are smaller than they are in the remaining six groups.

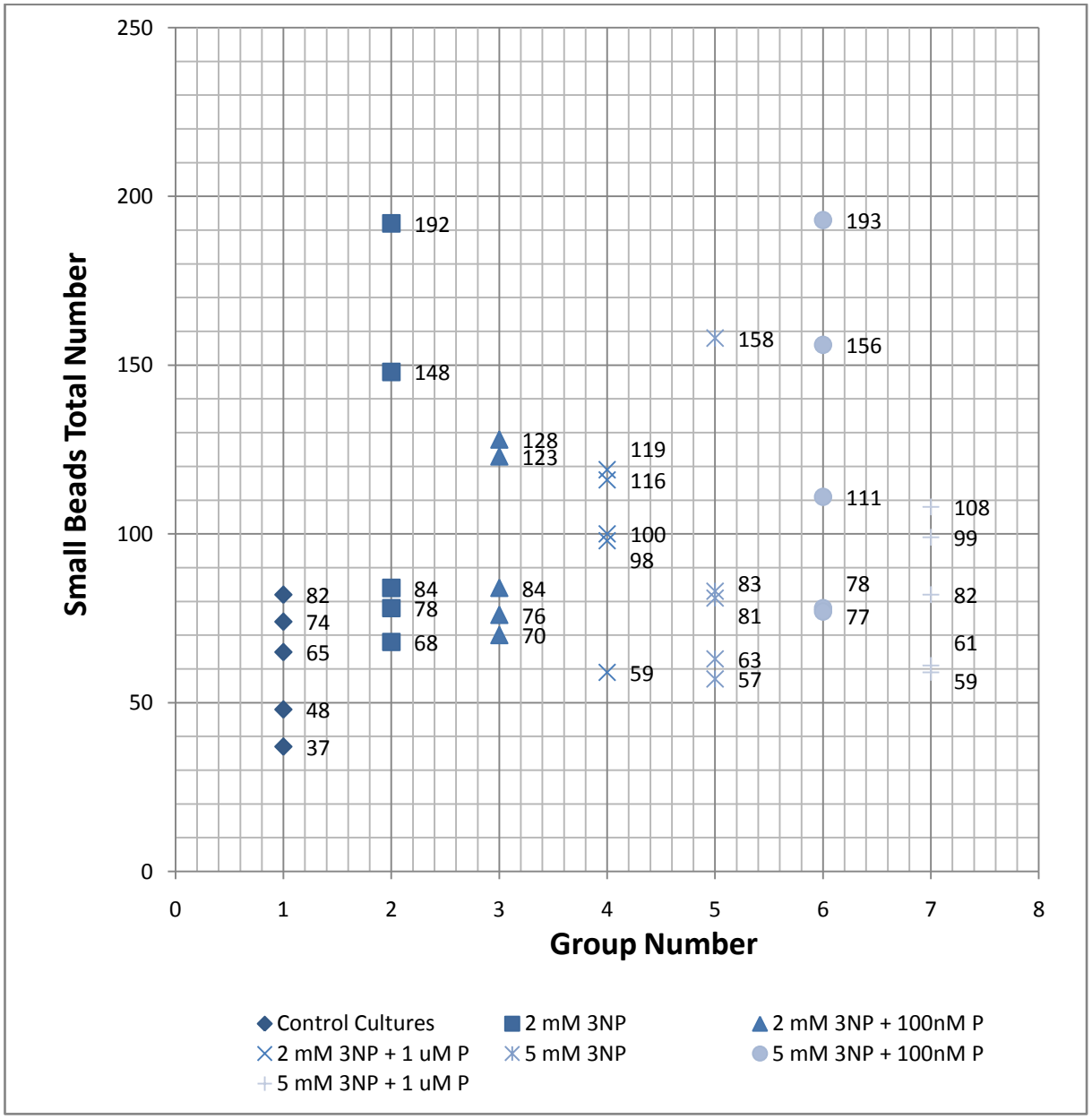


Figure 9.3 – Scatter graph of the total number of the Small Beads measured on the 35 images of the seven groups of the Test set

9.1.4 An important index of evaluation: the ratio Total NB Length / Total NCB Area

According to Dr. Joseph Steiner, the three main output values whose variations are shown in the former graphs, represent an absolute index for the evaluation of the health status of the neuron tissue. However, as known from the statistical analysis of biological samples of neurons, neuron cell bodies from the same samples of tissue should be linked to the approximate same number of neurites. Thus to establish that the obtained results show anatomical consistence, it is important to verify that the total number of neurites measured, are statistically related to the total number of cell bodies present in the same sample. This can be done by calculating the ratio between NB Total Length and NCB Total area:

$$r_1 = \frac{Neurite\ Branches\ Total\ Lenght}{Neuron\ Cell\ Bodies\ \mathbf{Total\ Area}}$$

As well as calculating the ratio between the NB Total Length and NCB Total Number

$$r_2 = \frac{Neurite\ Branches\ Total\ Lenght}{Neuron\ Cell\ Bodies\ \mathbf{Total\ Number}}$$

The result from this analysis should unveil that cultures submitted to a treatment of both 3NP toxin and Paroxetine have a value of this parameter more similar to the one calculated for the only-toxin treated cultures.
Furthermore, Total Area and Total number of the neuron cell bodies are supposed to be proportional to each other, therefore the corresponding IAP measured values are expected to be proportional each other, as well.

From a comparison of the values of the control group (group n. 1) on the trend of the graph and on the AVG values on the tables, the following observations can be made. In the second case (***r2***), it is not possible to say that there is a difference between the 3NP plus Paroxetine cultures (groups 3,4,6,7) and the 3NP Cultures (groups 2 and 5), even though the values calculated on the latter ones are closer to the one calculated on the control group than the ones calculated on the other groups. In the first case, instead, the ***r1*** value of the Control Group is more similar to the one calculated on 3NP plus Paroxetine cultures than the one calculated on the only toxin cultures.

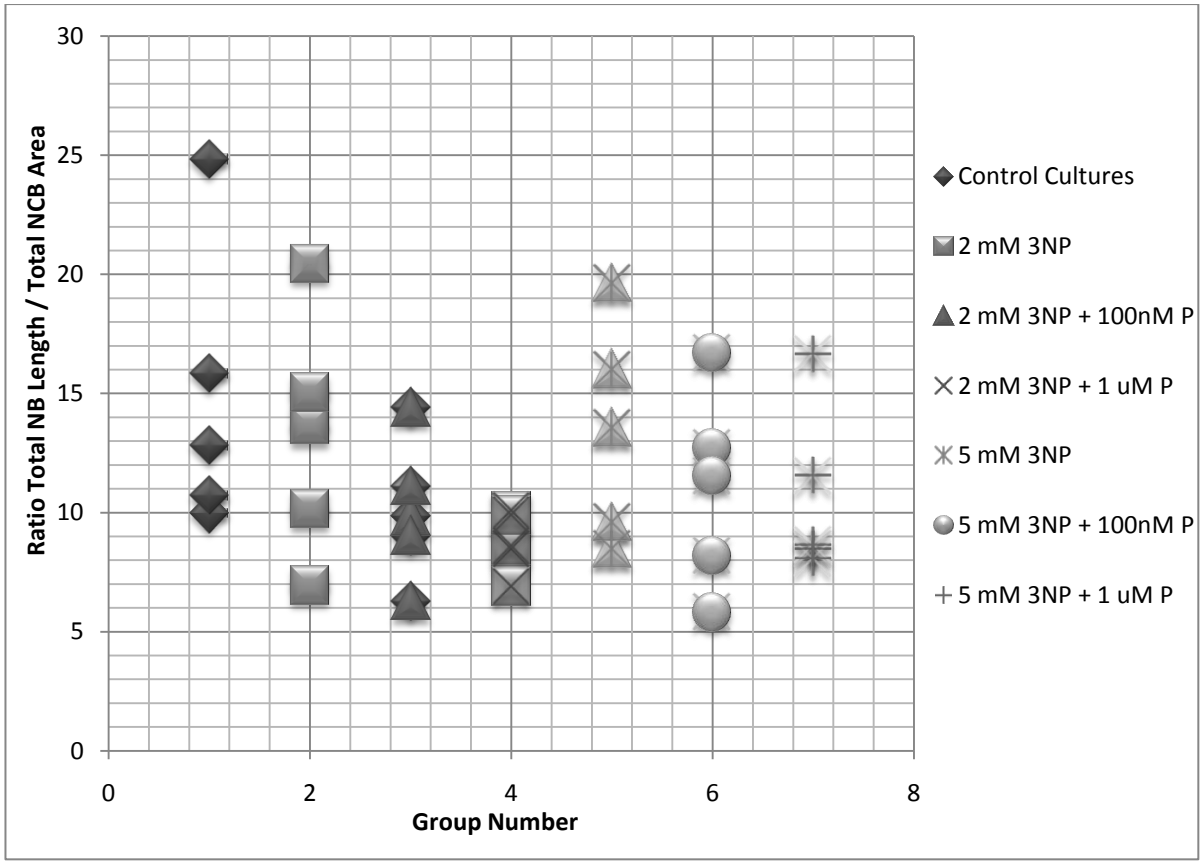


Figure 9.4 – Scatter graph of the value of the ratio: Length of NB/Area of NCB on the 35 images of the seven groups of the Test set

The values of the scatter graph are reported in Table 9.2, where the mean value of r1 is calculated.

Group #	r1 = NB Total Length / NB Total Area					AVG
1	9,961152	10,70412	15,84057	12,81312	24,8214	**14,82807**
2	20,45049	13,64592	15,02484	6,948369	10,1506	**13,24404**
3	9,851167	11,12549	6,269844	9,031121	14,40499	**10,13652**
4	8,464235	10,09602	8,597038	6,901465	9,882561	**8,788264**
5	8,468993	9,584813	19,62367	16,01608	13,54704	**13,44812**
6	12,68877	11,53465	5,819412	8,185326	16,69959	**10,98555**
7	8,464235	8,06125	8,625634	11,55134	16,64427	**10,66935**

Table 9.2 – Values of the ratio **r1**= Length (NB) / Area (NCB) on the 35 images of the seven group of the Test set. The averaged value of the ratio, for every group, is reported on the rightmost column, and the yellow highlighted value refers to the control group.

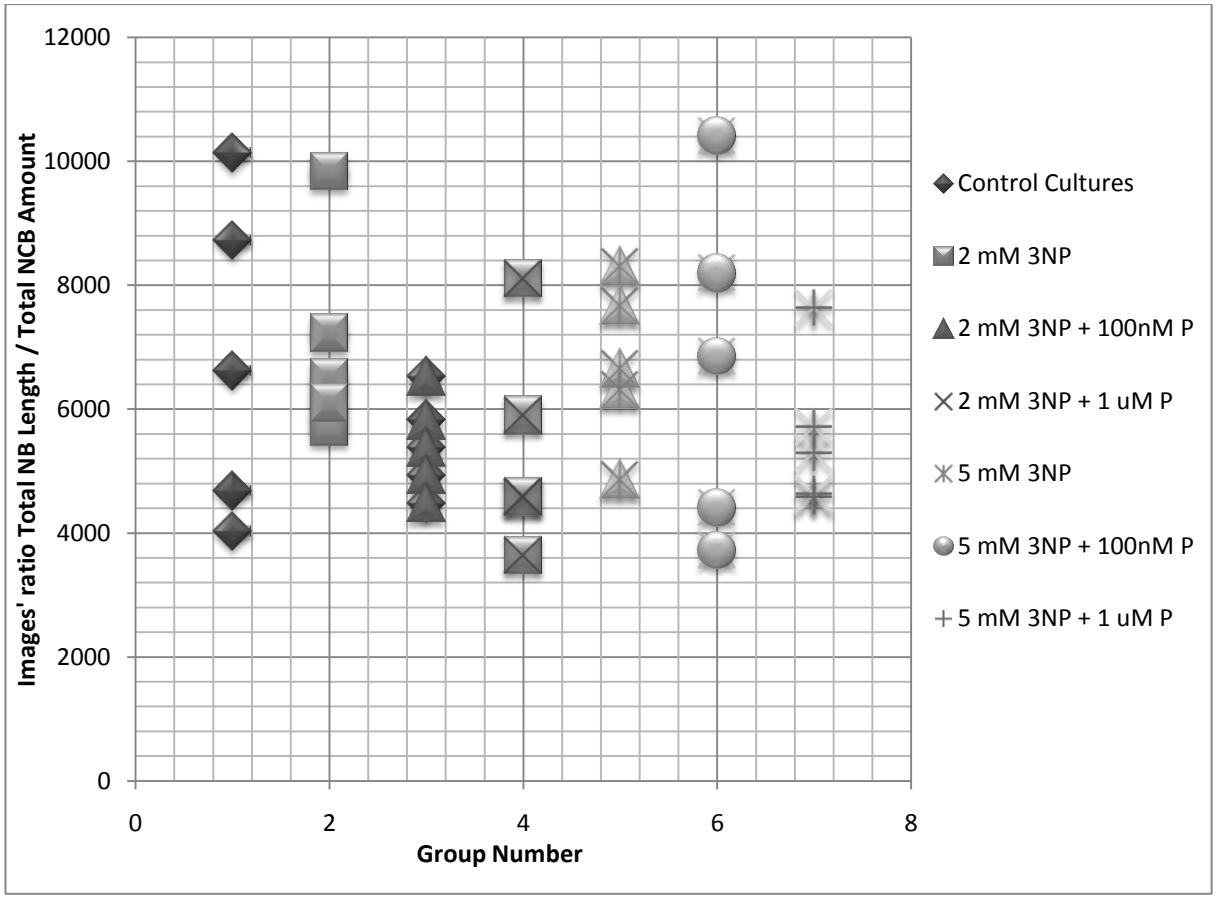


Figure 9.5 – Scatter graph of the value of the ratio: Length of NB/Number ofNCB on the 35 images of the seven group of the Test set

The values of the scattering graph are reported in Table 9.3, where the mean value of r2 is calculated.

Group Number	r2 = NB Total Length / NB Total Number					AVG
1	4684,423	6620,5	8729,286	4024,6	10129,892	**6637,74**
2	7228,111	9842,8	6503,25	5713,455	6093,941	**7076,311**
3	5824,667	5372,5	4933,583	4477,286	6519,286	**5425,464**
4	4577,227	4583,593	5894,923	3642,115	8103,7	**5360,312**
5	6309,4	4859,5	7665,308	8301,667	6650,364	**6757,248**
6	10403,64	4410,563	3720,35	6851,118	8193,933	**6715,92**
7	4577,227	5289,188	4628,611	7632,769	5710,65	**5567,689**

Table 9.3 – Values of the ratio **r2** = Length (NB) / Number (NCB) on the seven group of the Test set. The averaged value of the ratio, for every group, is reported on the rightmost column, and the yellow highlighted value refers to the control group.

9.2 Statistical validation of the results

In order to statistically validate the results, it is possible to apply the two sample unpaired T-test (using a two tailed distribution and assuming the samples are independent). The T-test is applied to the series of three main outputs calculated on the seven groups of images composing the test set. It compares the actual difference between the means of two sets of samples in relation to the variation in the data and it gives a probability value referring to the hypothesis that the two populations which the set of samples come from can be regarded as the same population. The goal of the application of the T-test on the groups of images of this project is to perform a statistical comparison between the Control group and the groups of cultures which were submitted to different treatments. The probability values for each comparison with the output value of the Control group, is an indicator of similarity between the compared groups.

Output values compared	Group number					
	2	3	4	5	6	7
NB Length	0,423	0,702	0,959	0,340	0,858	0,993
NCB Area	0,875	0,425	0,119	0,634	0,305	0,310
Number of SB	0,091	0,055	0,027	0,22	0,050	0,267
r1	0,75	0,37	0,370	0,929	0,961	0,437
r2	0,595	0,096	0,04	0,62	0,18	0,14

Table 9.4- Values of probability outputted from the T-test. Each group (columns) has been compared with the Control group by a statistical analysis of the variation of the values of the three main outputs (rows) and the ratio values r1 and r2.

It is possible to assert that the comparison by NB length gives results in line with expectations, as the cultures treated with toxin only have (in mean) the lowest value of probability of being regarded as belonging to the same population of the Control Set). The values of probability which refer to the comparison with the groups on cultures treated with toxin plus paroxetine are fairly high, especially for the groups (4 and 7). The values of probability computed by using the NCB total area as means of comparison are generally quite low, and no conclusions can be made, as well as for the number of beads, for which the values of probability are very low and for the ratio values r1 and r2.

9.3 Evaluation of the methods and of the results and conclusion

Taking in account all the results of the final classification images, the comments made in chapter 6. and the comments about the output values made in 9.1 and 9.2, the following consideration can be made. The program developed was able to perform the methods of the Image Analysis Protocol for each kind of input image that was made available, by performing a pre-analysis step that allowed the program to recognize the main properties of the image

given as input, for the program finally to load the sequence of methods and their correct parameter values. The overall result of the final image classifications is positive, as most of the main features of the images were classified correctly, even though there is definitely room for improvement in terms of shapes of the classified features (for both neurites, beads and neuron cell bodies) and of the number of misclassified objects. The minimum acceptable value of area may cause misclassification of beads as neuron cell bodies. The choice of avoiding to consider the class 3 neurites as part of the final features could be called into question, as those areas were not actually part of the background, as they have been downgraded at the end. Nevertheless, as the goal was to detect and classify the actual neurite branches, it was decided that it was not opportune to include them in the final classification result. Finally, from the standpoint of the evaluation of the treatments efficiency, by a comparison of the values of the non-treated cultures (that is the control group) it is possible to assert that:

1) By the comparison of the Total Length of NB, the cultures treated with toxins plus paroxetine are more similar (and in some cases also comparable), to the control group ones, whereas cultures treated with toxin only shows low values for this parameter

2) By the comparison of SB Total number , both cultures treated with toxins only and cultures treated with toxins plus paroxetine can be regarded as different from the control group, (the latter has smaller values for this parameter)

3) By the comparison of the total area of neuron cell body it is not possible to draw up any conclusion about the similarity of the groups of cultures.

4) The values of r_1 do not give any information to distinguish the cultures from each other.

5) The values of r_2 do not give any information to distinguish the cultures from each other. This value was also supposed to be proportional to r_1, but this cannot be confirmed. This fact can be due to three main causes:

   5.1) by a visual inspection of the images it is noticeable that the value of the area varied greatly, even in the same image (see Fig. 9.6a)

   5.2) according to what asserted in the former point, as some of the neuron might have been misclassified as beads, the real value of the total area of the neurons in the images do not correspond to the actual area

   5.3) some of the neuron cell bodies were not completely inside the border of the image, so that the value of the total number increases, but the area of the neuron measured is not that neuron's actual area (see Fig. 9.7a)

6) The application of the T-test confirms what has already been asserted in points 1, 2 and 3.

The obtained results might have been different if the amount of images of the database was greater. Based on an impression of the results graphs and a general evaluation of the values in the tables it seems there is a lot of variation in the images and hence in the output variables from the different groups. If a greater amount of images were provided, with more data in each group, it would have also been easier (and possible) to exclude images which results could be regarded as outliers, and it would have been possible to get a better impression of the differences between the groups.

9.4 Examples of original and classified images

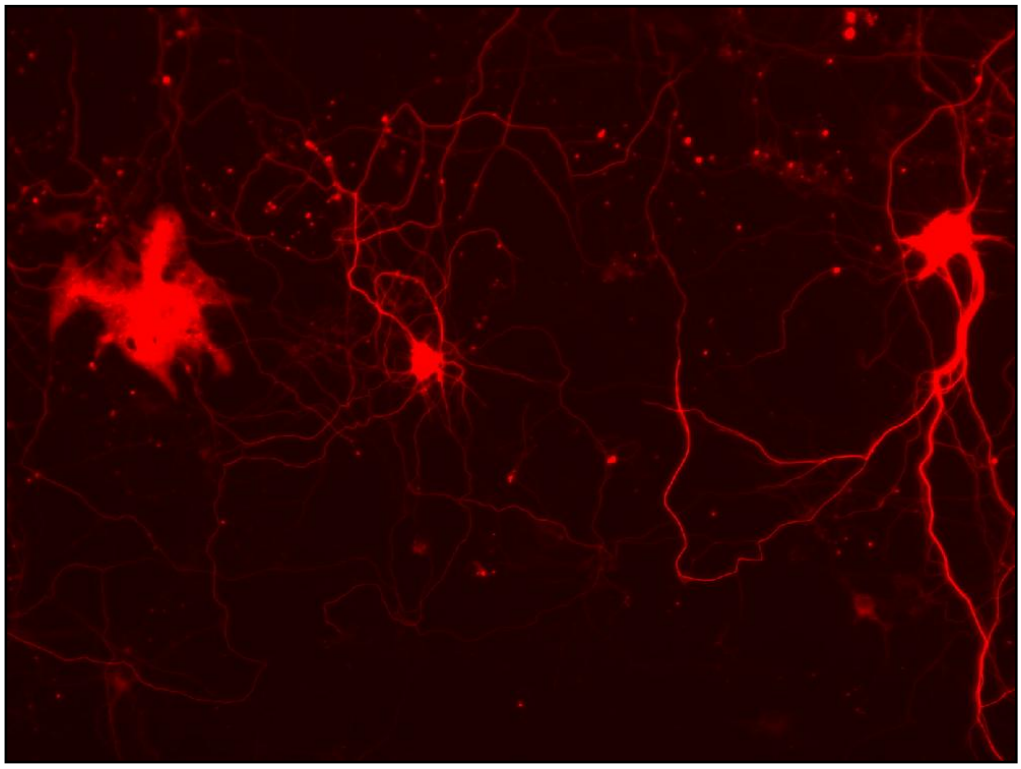


Figure 9.6a – Original image of a culture treated with toxins only. Neuron Cell Body area can vary greatly

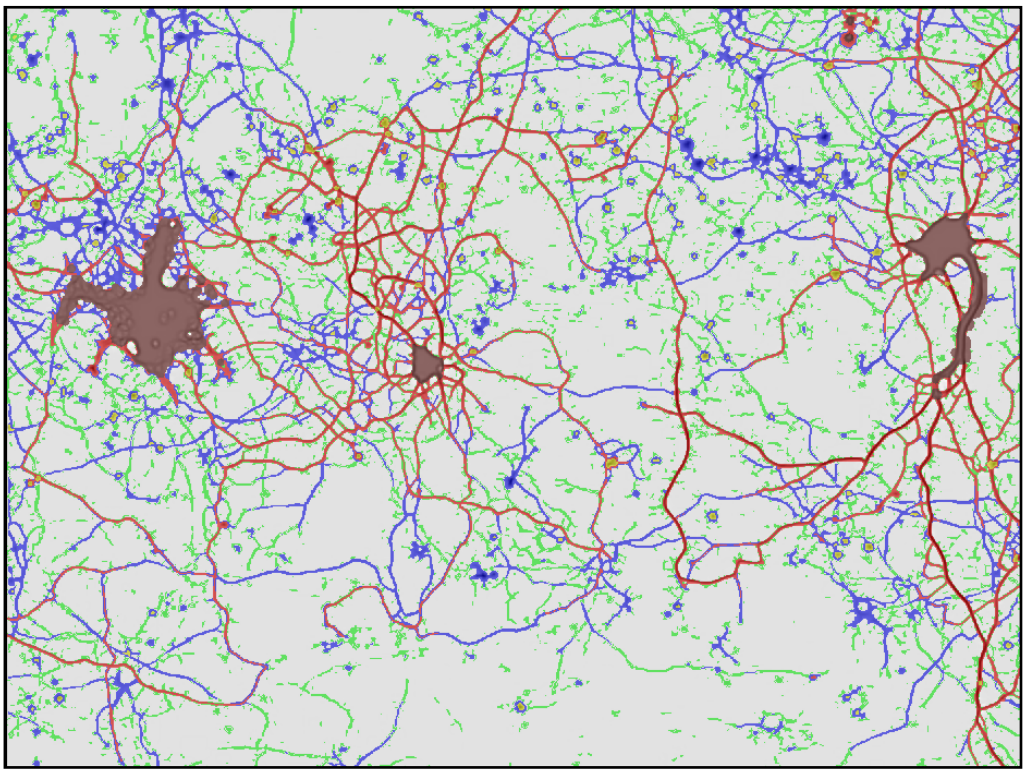


Figure 9.6b – Final classified version of image 9.6a.

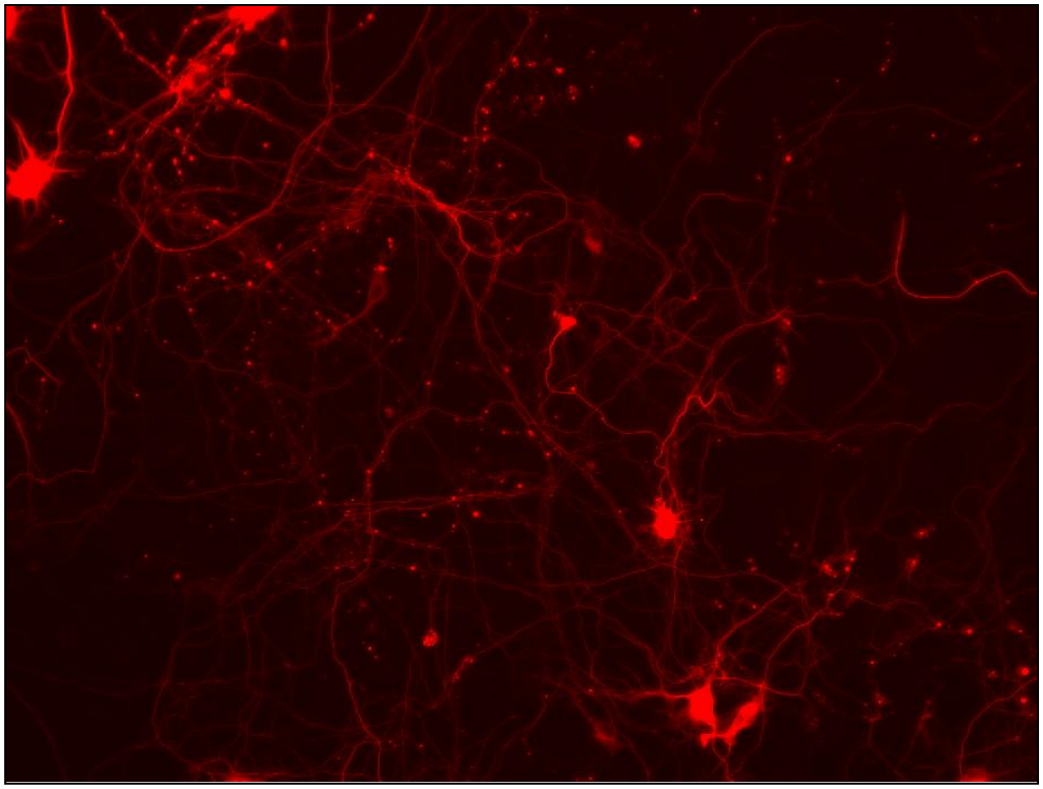


Figure 9.7a – Original image of a culture treated with toxins only. Some of the neuron cell bodies are not completely inside the border of the image (upper leftmost corner)

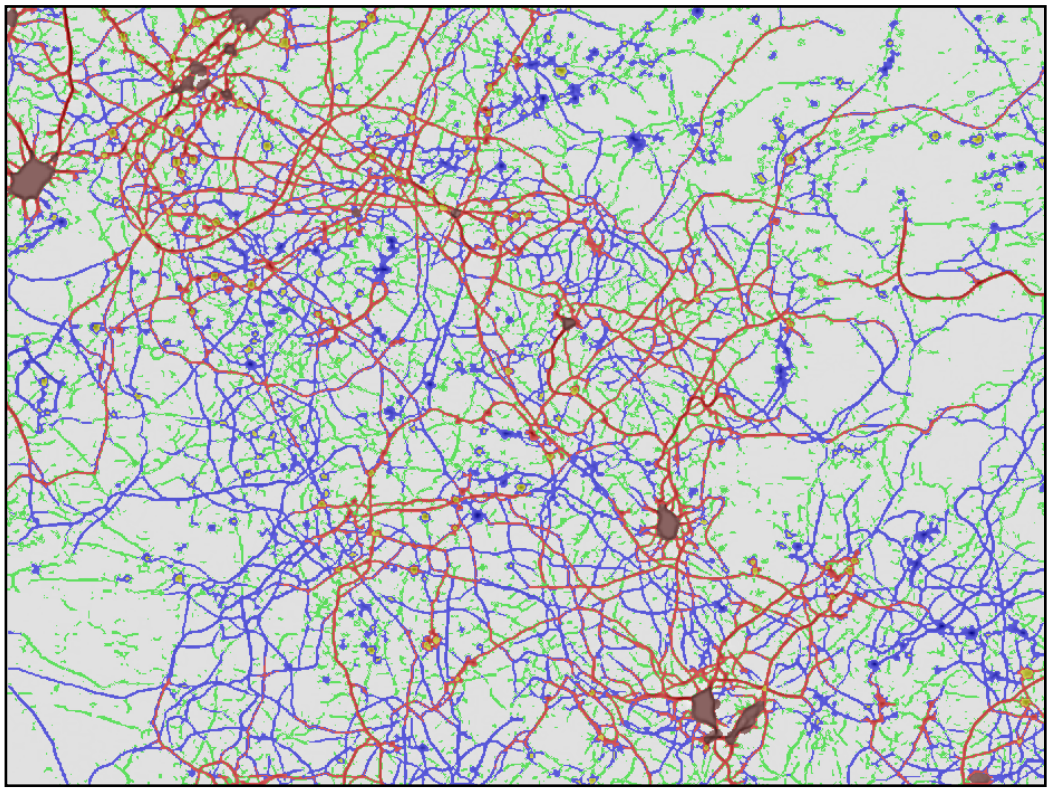


Figure 9.7b - Final classified version of image 9.7a

In the following final classified images only the red and blue labeled neurites are displyed. These are examples of images on which the skeletonization function was performed, followed by the measurement of the total length of the neurites.

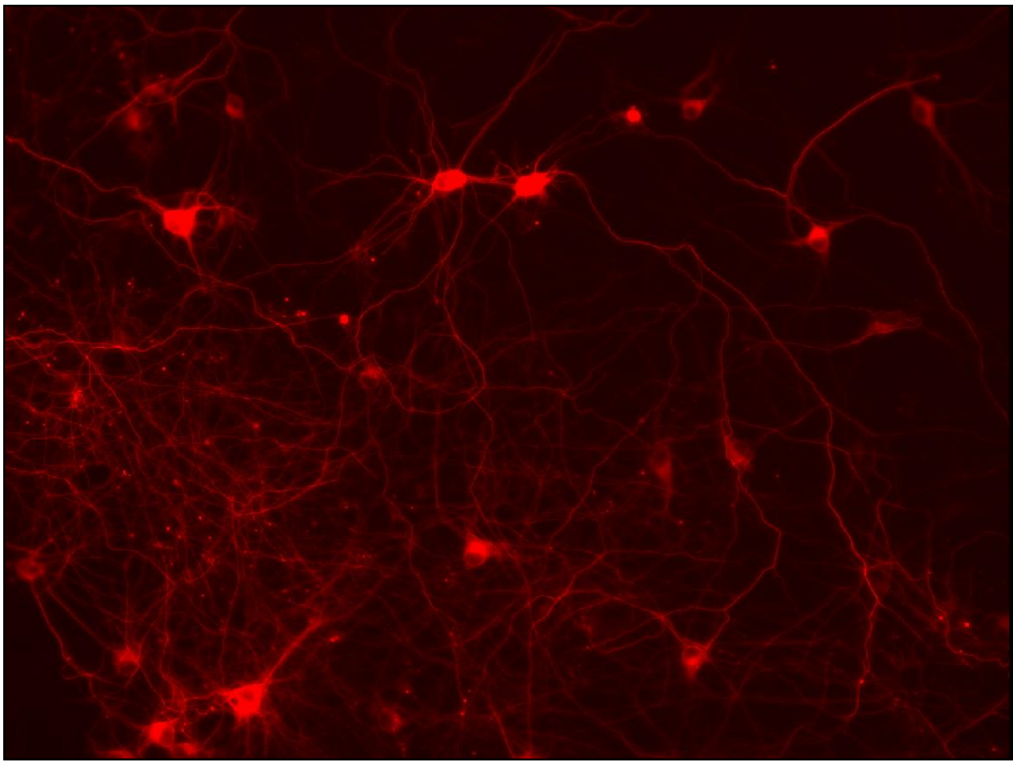


Figure 9.8a – Original image of a culture of the control group

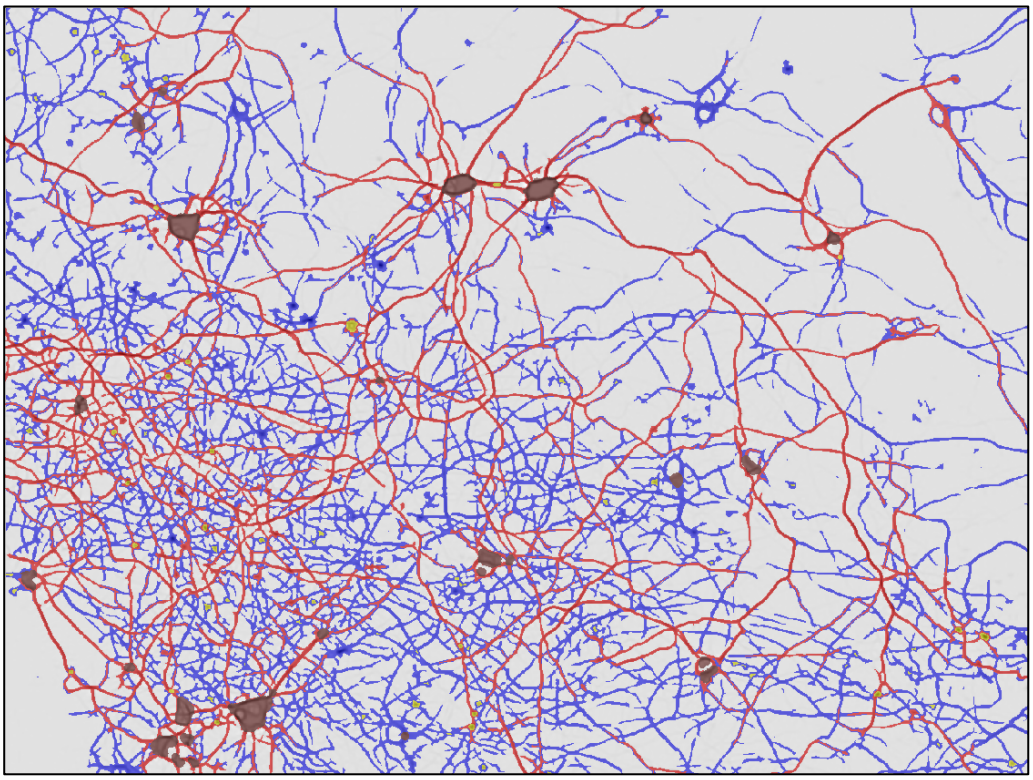


Figure 9.8b - Final classified version of image 9.8a

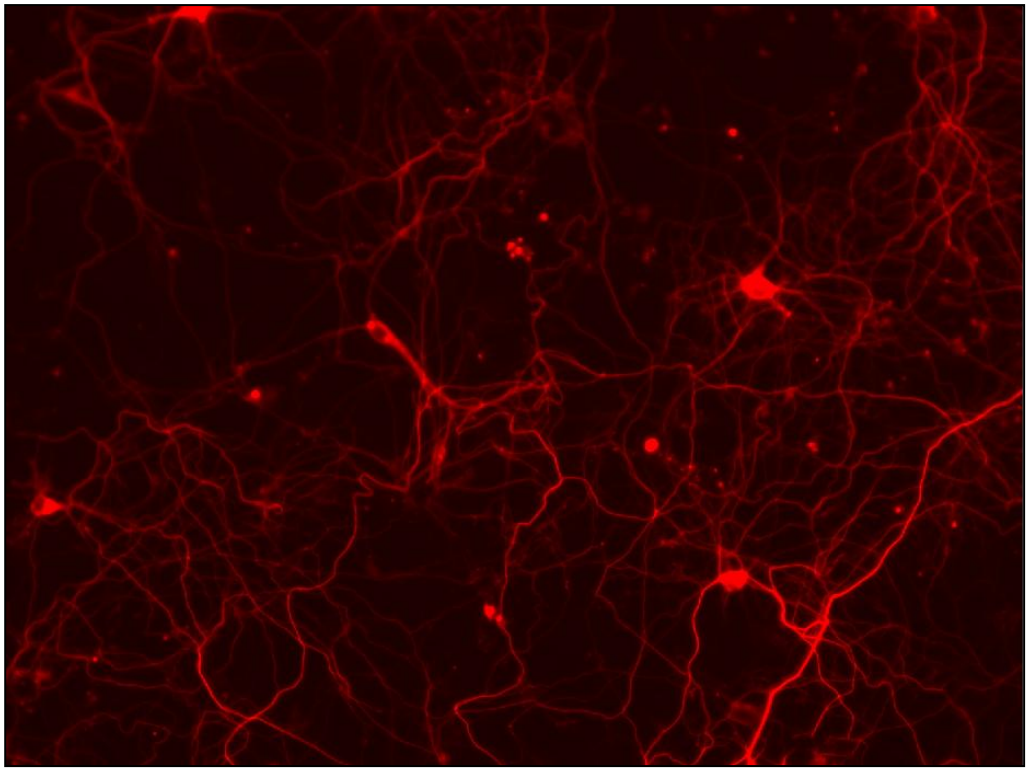


Figure 9.9a – Original image of a culture treated with toxin (5mM 3NP) plus therapeutic agent (1uM Paroxetine)

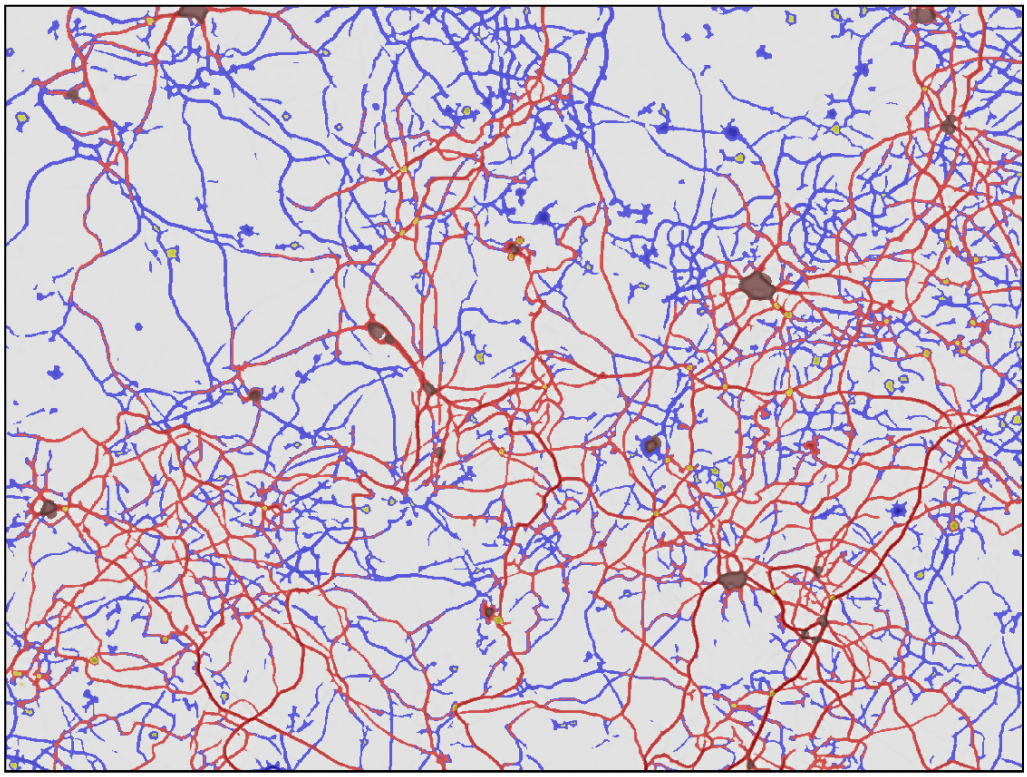


Figure 9.9b - Final classified version of image 9.9a

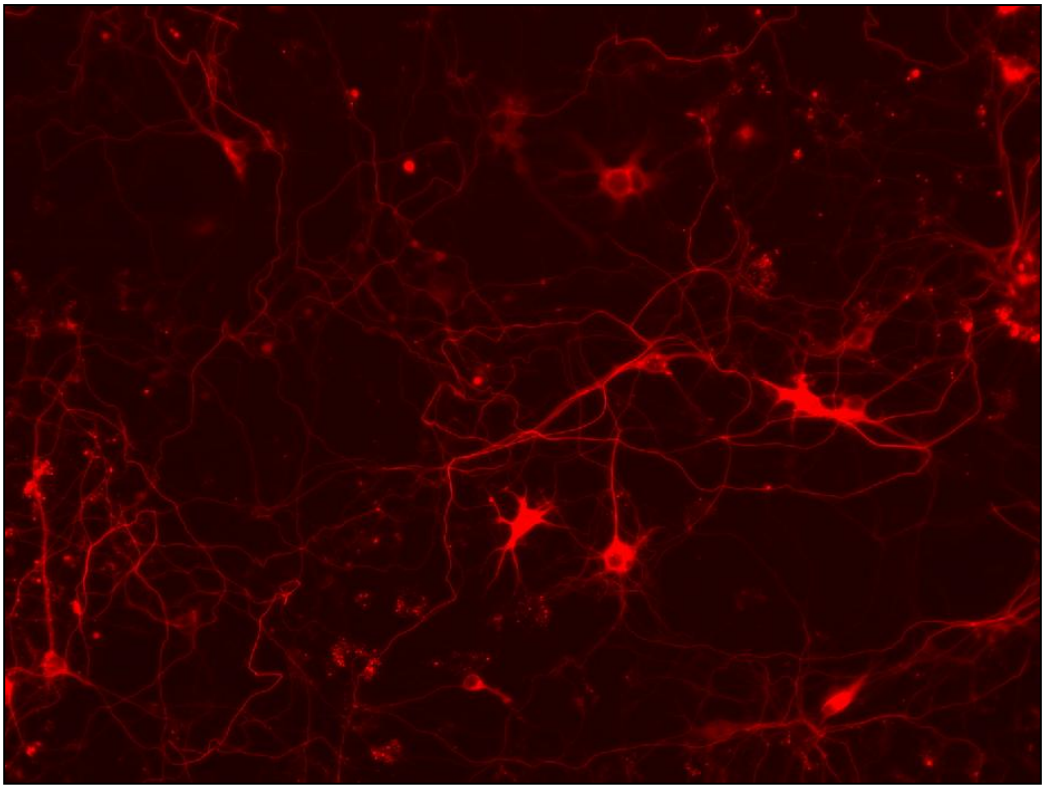


Figure 9.10a – Original image of a culture treated with toxin (2mM 3NP) plus therapeutic agent (100nM Paroxetine)

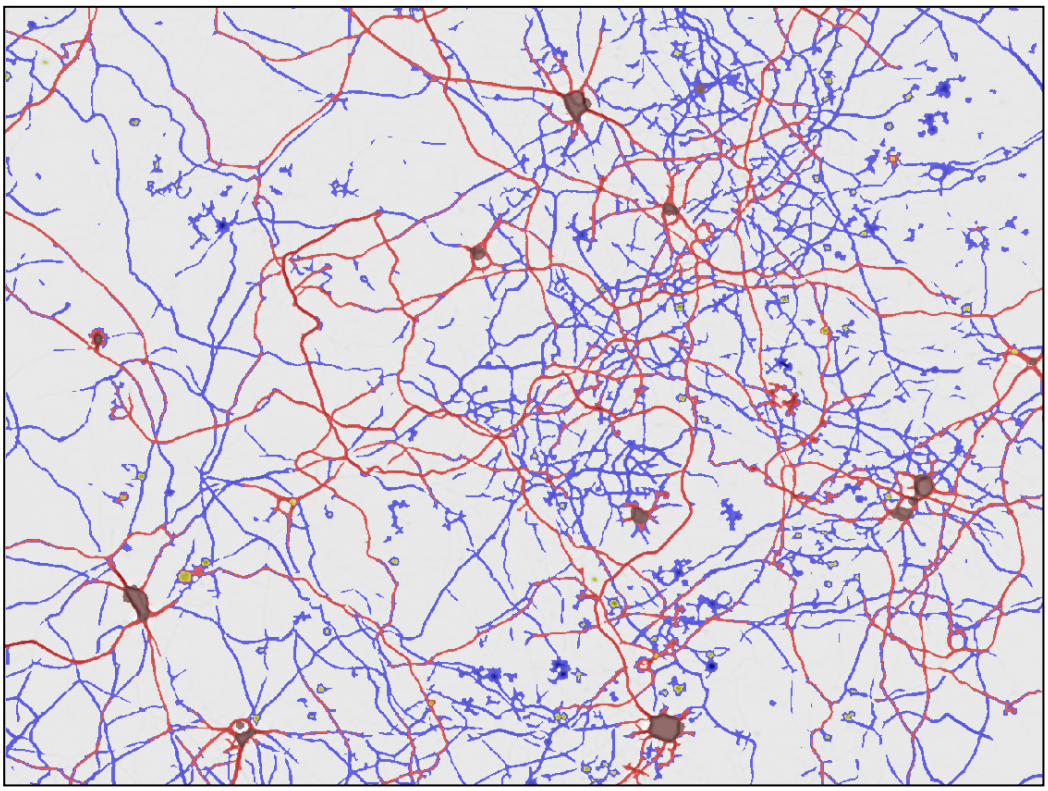


Figure 9.10b - Final classified version of image 9.10a

9.5 Tables of all the outputs computed by the Image Analysis Protocol

Here follow the seven tables, displaying all the feature's measurements performed by the algorithm implementing the steps of the Image Analysis Protocol.

Control Cultures	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	38154	42966	37351	8034	22862	29873
BLUE NEURITES LENGTH	83641	62962	84859	32212	68307	66396
NEURITES TOTAL LENGHT	121795	105928	122210	40246	91169	96269
GREEN NEURITES LENGTH	82801	83170	75184	51502	78871	74306
TOTAL AREA - NCB	12227	9896	7715	3141	3673	7330
TOTAL AREA - SB	2580	2323	2807	1198	1611	2104
TOTAL NUMBER - NCB	26	16	14	10	9	15
TOTAL NUMBER - SB	82	65	74	37	48	61

Table 9.5 – Outputs measured on the images of group 1

2 mM 3NP	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	14295	30078	18658	24130	35670	24566
BLUE NEURITES LENGTH	50758	68350	59381	38718	67927	57027
NEURITES TOTAL LENGHT	65053	98428	78039	62848	103597	81593
GREEN NEURITES LENGTH	62106	79216	74165	69085	76423	72199
TOTAL AREA - NCB	3181	7213	5194	9045	10206	6968
TOTAL AREA - SB	2336	3321	2957	6529	7357	4500
TOTAL NUMBER - NCB	9	10	12	11	17	12
TOTAL NUMBER - SB	68	84	78	148	192	114

Table 9.6 – Outputs measured on the images of group 2

2 mM 3NP + 100nM P	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	34058	14140	53508	38035	26677	33284
BLUE NEURITES LENGTH	53312	39585	64898	55988	64593	55675
NEURITES TOTAL LENGHT	87370	53725	118406	94023	91270	88958
GREEN NEURITES LENGTH	74738	60402	91234	83974	82908	78651
TOTAL AREA - NCB	8869	4829	18885	10411	6336	9866
TOTAL AREA - SB	3095	2455	5084	4820	2693	3629
TOTAL NUMBER - NCB	15	10	24	21	14	17
TOTAL NUMBER - SB	76	70	123	128	84	96

Table 9.7 – Outputs measured on the images of group 3

2 mM 3NP + 1 uM P	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	26376	49047	26179	46647	30830	35816
BLUE NEURITES LENGTH	74323	74710	50455	48048	50207	59549
NEURITES TOTAL LENGHT	1000699	123757	76634	94695	81037	95364
GREEN NEURITES LENGTH	83133	91232	78621	81064	88179	84446
TOTAL AREA - NCB	11897	12258	8914	13721	8200	10998
TOTAL AREA - SB	1774	3141	3803	4509	2552	3156
TOTAL NUMBER - NCB	22	27	13	26	10	20
TOTAL NUMBER - SB	59	100	119	116	98	98

Table 9.8 – Outputs measured on the images of group 4

5 mM 3NP	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	31076	10571	26673	21336	24591	22849
BLUE NEURITES LENGTH	63565	38024	72976	53379	48563	55301
NEURITES TOTAL LENGHT	94641	48595	99649	74715	73154	78150
GREEN NEURITES LENGTH	76252	57763	75102	64972	66360	68090
TOTAL AREA - NCB	11175	5070	5078	4665	5400	6278
TOTAL AREA - SB	2777	1923	5690	3390	1931	3142
TOTAL NUMBER - NCB	15	10	13	9	11	12
TOTAL NUMBER - SB	81	63	158	83	57	88

Table 9.9 – Outputs measured on the images of group 5

5 mM 3NP + 100mM P	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	38991	13019	27091	46361	25450	30182
BLUE NEURITES LENGTH	75449	57550	47316	70108	97459	69576
NEURITES TOTAL LENGHT	114440	70569	74407	116469	122909	99758
GREEN NEURITES LENGTH	81248	70642	73560	87432	71770	76930
TOTAL AREA - NCB	9019	118	12786	14229	7360	8702
TOTAL AREA - SB	6429	2527	4599	8186	2784	4905
TOTAL NUMBER - NCB	11	16	20	17	9	15
TOTAL NUMBER - SB	156	78	111	193	77	123

Table 9.10 – Outputs measured on the images of group 6

5 mM 3NP + 1 uM P	1	2	3	4	5	VALUES AVG
RED NEURITES LENGTH	26376	30752	31849	40229	45305	34902
BLUE NEURITES LENGTH	74323	53875	51466	58997	68908	61514
NEURITES TOTAL LENGHT	100699	84627	83315	99226	114213	94416
GREEN NEURITES LENGTH	83133	83547	78430	79103	83160	81475
TOTAL AREA - NCB	11897	10498	9659	8590	6862	9501
TOTAL AREA - SB	1774	4202	2198	2975	4080	3046
TOTAL NUMBER - NCB	22	16	18	13	20	18
TOTAL NUMBER - SB	59	108	61	82	99	82

Table 9.11 – Outputs measured on the images of group 7

Chapter 10

Future Works

In this section some of the possible improvements of the methods will be described, in particular depending on the new kind of data format that will be provided. Some suggestions for future work will also be provided based on the detected weaknesses of the methods used, the observation made and the experience achieved during this project.

10.1 Adapting the method to the new image file format

A first significant change to the implementation will be given by the new kind of format of image data into which the imaged cultures will be stored, that is the *zvi*. Such kind of data format, as already mentioned in chapter 2 – "Data Material" (see Table 2,1) and as it is explained in detail in Appendix D, has been developed specifically to meet the requirements of modern fluorescence microscopy, hence images can be saved as uncompressed raw data or using loss-less or JPG compression. These images feature high values of resolution (up to 49618 DPI) and they also feature the DAPI layer (4 bands) and 16 bits per band. Most of the *zvi* files analyzed in this project were unfortunately not good, because of a problem with the PFA fixation prior to DAPI staining, so that a complete analysis of the *zvi* dataset was not carried out. Despite that, it was possible to work on some of them and to carry out the first two steps of the IAP. The two properties previously listed involve the pixel intensity values to range from 0 up to 65535 (2^{16}) for each band and a very high definition in terms of capability to distinguish the different image's features.

The first main advantage is that it is possible to obtain a very good quality filtered image, resulting in a good starting point for the classification process.

Furthermore, as the pixel intensity values of the features can range in a wider interval, the second advantage is that it is possible to obtain a better fine-tuning of the threshold values in order to perform a more accurate classification of the image's main components. The combination of these two steps does output a classified image in which most of the features are already rightly classified in their own class. This results in applying a smaller number of morphological post-processing methods to the image in order to get the features to be correctly classified. Finally, by reducing the number of morphological editing methods used, the number of undesired modification of areas and features is the image is also reduce and, therefore, a better starting point for the measurement post-processing step.

10.2 Improvements and advices on the method implementations

From the standpoint of the methods application, the first issue concerning the classified image might be improved by using the *zvi* file format, for the reasons that have just been

cited. A second issue concerns the morphological post-processing.
The purpose of the application of a certain number of morphological editing methods is to get a sufficient number of features to be classified correctly without their shape in the label layer being too different from their original shape. The main problem is that a sequence of morphological functions, while improving the shape features in a particular area of the image, often worsens the shape of many other features in other areas. An advice is to evaluate carefully the effect of the application of every method separately from the others, on all the area of the image and on a high number of different images from different groups. In general, it is also better to avoid the evaluation of more than one step at a time.
Finally, it is important to validate the obtained results from the anatomical-statistical standpoint, by verifying that the total number of neurites measured is statistically related to the total number of cell bodies present in the same sample. Since the method of the calculation of the ratio r1 and r2 was regarded as not being a good index of evaluation for this purpose, another method should be used. One way of doing that is by implementing a function working on the detected neuron cell bodies one at a time, performing a detection of the linked-neurites for each of the neurons.

Chapter 11

Appendix

11.1 Appendix A: HIV Virus infections

HIV infection is a disease caused by the human immunodeficiency virus (HIV) a lentivirus which is a member of the retrovirus family. The condition gradually destroys the immune system, which makes it harder for the body to fight infections. Infection with HIV occurs by the transfer of blood, semen, vaginal fluid, pre-ejaculate, or breast milk. Within these bodily fluids, HIV is present as both free virus particles and virus within infected immune cells[1].
AIDS (acquired immune deficiency syndrome) is the final and most serious stage of HIV disease, which causes severe damage to the immune system.

The HIV virus - Description

HIV virus is a retrovirus which genes are composed of RNA (Ribonucleic Acid). It is an RNA virus that is replicated in a host cell via the enzyme reverse transcriptase to produce DNA from its RNA genome. The DNA is then incorporated into the host's genome by an integrase enzyme. The virus thereafter replicates as part of the host cell's DNA.

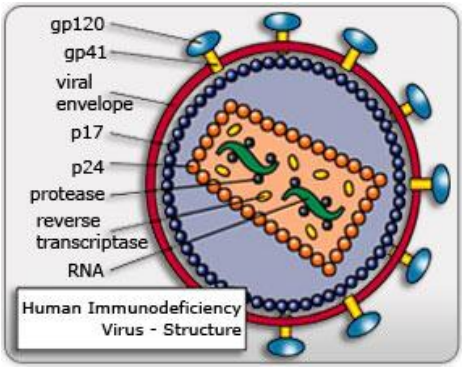


Figure A.1 - The proteins gp120 and gp41 together make up the spikes that project from HIV articles, while p17 forms the matrix and p24 forms the core.

Almost all organisms, including most viruses, store their genetic material on long strands of DNA. RNA has a very similar structure to DNA. However, small differences between the two molecules mean that HIV's replication process is a bit more complicated than that of most other viruses.Like all viruses, HIV cannot grow or reproduce on its own. In order to make new copies of itself it must infect the cells of a living organism. Outside of a human cell, HIV exists as roughly spherical particles (sometimes called virions). The surface of each particle is studded with lots of little spikes. An HIV particle is around 100-150 billionths of a metre in diameter. That's about the same as one seventieth of the diameter of a human CD4+ white blood cell.Unlike most bacteria, HIV particles are much too small to be seen through an ordinary microscope. However they can be seen clearly with an electron microscope.HIV particles surround themselves with a coat of fatty material known as the viral envelope (or membrane).

Projecting from this are around 72 little spikes, which are formed from the proteins gp120 and gp41. Just below the viral envelope is a layer called the matrix, which is made from the protein p17. [12] The viral core (or capsid) is usually bullet-shaped and is made from the protein p24. Inside the core are three enzymes required for HIV replication called reverse transcriptase, integrase and protease. Also held within the core is HIV's genetic material, which consists of two identical strands of RNA.

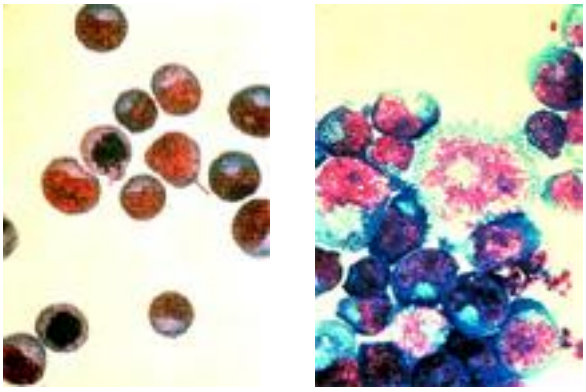


Figure A.2. - Normal T-cells and HIV-infected T-cells

The HIV Virus - How does it work

HIV infects primarily vital cells in the human immune system such as helper T cells (to be specific, $CD4^+$ T cells), macrophages, and dendritic cells. These cells are re white blood cells crucial to maintaining the function of the human immune system. As HIV attacks these cells, the person infected with the virus is less equipped to fight off infection and disease, ultimately resulting in the development of AIDS. HIV infection leads to low levels of $CD4^+$ T cells through three main mechanisms:

- direct viral killing of infected cells;
- increased rates of apoptosis in infected cells;
- killing of infected $CD4^+$ T cells by CD8 cytotoxic lymphocytes that recognize infected cells.

When $CD4^+$ T cell numbers decline below a critical level, cell-mediated immunity is lost, and the body becomes progressively more susceptible to opportunistic infections.

HIV destroys CD4+ T cells, which are crucial to the normal function of the human immune system. In fact, depletion of CD4+ T cells in HIV-infected individuals is an extremely powerful predictor of the development of AIDS. Studies of thousands of individuals have revealed that most HIV-infected people carry the virus for years before enough damage is done to the immune system for AIDS to develop; however, with time, a near-perfect correlation has been found between infection with HIV and the subsequent development of AIDS. Recently developed, sensitive tests have shown a strong correlation between the amount of HIV in the blood and the subsequent decline in CD4+ T cell numbers and development of AIDS. Furthermore, reducing the amount of virus in the body with anti-HIV drugs can slow this immune destruction.Antiretroviral medicines can help reduce the amount of virus in the body, preserve CD4+ T cells and dramatically slow the destruction of the immune system.People who are not infected with HIV and generally are in good health have

roughly 800 to 1,200 CD4+ T cells per cubic millimeter (mm^3) of blood. Some people who have been diagnosed with AIDS have fewer than 50 CD4+ T cells in their entire body.

How does HIV-type 1 virus induce neuronal death

The brain is a major target organ for HIV-type 1 (HIV-1) infection enters the central nervous system (CNS) in ≈80% of infected individuals early after infection and can cause a wide range of neurological disorders including cognitive motor impairment and HIV-associated dementia (HAD; also known as AIDS dementia complex). Although the incidence of HAD has markedly reduced with the introduction of highly active antiretroviral therapy (HAART), the overall prevalence of HAD is rising as the number of treated subjects with chronic HIV infection increases. Notably, HIV-1 infects a restricted number of cell types in the brain [2] [3].

Neurons are one of the few cell types in the human body that do not support HIV type-1 (HIV-1) replication. Although the lack of key receptors is a major obstacle to infection, studies suggest that additional functions inhibit virus replication to explain the exquisite resistance of neurons to HIV-1.

Human immunodeficiency virus type 1 (HIV-1) infection of the central nervous system may result in neuronal apoptosis in vulnerable brain regions, including cerebral cortex and basal ganglia. The mechanisms for neuronal loss are likely to be multifactorial and indirect, since HIV-1 productively infects brain-resident macrophages and microglia but does not cause cytolytic infection of neurons in the central nervous system. HIV-1 infection of macrophages and microglia leads to production and release of diffusible factors that result in neuronal cell death, including the HIV-1 regulatory protein Tat. Although brain tissue from patients with human immunodeficiency virus (HIV) and/or AIDS is consistently infected by HIV type 1 (HIV-1), only 20 to 30% of patients exhibit clinical or neuropathological evidence of brain injury. Extensive HIV-1 sequence diversity is present in the brain, which may account in part for the variability in the occurrence of HIV-induced brain disease.

Neurological injury caused by HIV-1 is mediated mainly by:

1) a direct way: through administration of neurotoxic viral proteins, most notably the Tat and gp120 containing the V1V3 or C2V3 envelope region from non-clade B, brain-derived HIV-1 sequences;

2) an indirect way: through excess production of host molecules by infected or activated glial cells.

11.2 Appendix B: Toxins

Patients infected with HIV-1 often exhibit cognitive deficits that are related to progressive neuronal degeneration and cell death. Treatments with agents that interupt this apoptotic cascade may prove beneficial in preventing neuronal degeneration and associated dementia in AIDS patients.

3-NP nitropropionic acid

The 3-NP or 3-nitropropionic acid is a mitochondrial electron transport chain blocker, active at inhibiting ETS complex 2. The 3-NP toxin is produced by certain plants and fungi and it is naturally present in leguminous plants, occasionally poisoning grazing livestock.The 3-NP is structurally similar to and isoelectronic with the substrate succinate and it is believed to be a suifigurecide inactivator of Complex II: it acts as a specific irreversible inhibitor of the enzyme complexes responsible for energy production in the mitochondrial respiratory complex, eventually causing oxidative stress via reactive oxygen species.The target of 3-NP is particularly the Complex II (Coles et al., 1979), which is both a member of the Krebs tricarboxylic acid cycle (oxidizing succinate to fumarate) and an entry point for electrons into the respiratory chain at the level of ubiqui nol.The intra- and extra-mitochondrial mode of action leads to the following clinical pictures in chronic intoxications. The symptoms of 3NP poisoning resembles Huntington's disease and is used in an animal model of that (Brouillet et al., 1999).Since the function of the nervous system requires lots of energy, the mitochondrial damage is probably reflected in the electrical activity of the brain. Beside damage in energy production, the 3-NP has also more direct effects in the nervous system, so its excessive presence should be detectable by electrophysiology. A 3-NP cultures treatment emulates the oxidative stress and neuronal cell death of the HIV virus, and axonal/dendritic pruning as well [13].

Protein gp-120

Gp120 is a glycoprotein which forms the spikes sticking out of a HIV virus particle, along with the gp41 that is another important component of the spikes. Its main function is to bind to CD4 in human cells. Its presence is essential for virus entry into cells as it plays a vital role in seeking out specific cell surface receptors for entry and in the ability of HIV-1 to enter $CD4^+$ cells, particularly.

Gp120 gene has a molecular weight of 120 kilodaltons. It is around 1500 nucleotides long - and since each amino acid in a protein is encoded by 3 nucleotides in DNA and the CD4 binding site of gp120 - the specific region of the molecule which attaches to CD4 via intermolecular attractions - has been found to include the amino acid residues numbered 400-430. The glycoprotein gp120 is anchored to the viral membrane, or envelope, via non-covalent bonds with the transmembrane glycoprotein, gp41. One half of the molecular weight of gp120 is due to the carbohydrate side chains (the "glyco-" in "glycoprotein"). These are sugar residues which form something almost like a sugar "dome" over the gp120 spikes. This dome prevents gp120 from being recognised by the human immune response. As the HIV virus and the human CD4 cell come together, the gp120 binding site "snaps open" at the last minute. The Human Immunodeficiency Virus (HIV) can mutate frequently to stay ahead of the immune system. There is however a highly conserved region in the virus genome near its

receptor binding site. It is involved in entry into cells by binding to CD4 receptors, particularly helper T-cells. Binding to CD4 is mainly electrostatic although there are van der Waals interactions and hydrogen bonds [14]. The HIV viral protein gp120 induces apoptosis of neuronal cells. Gp120 induces mitochondrial-death proteins like caspases which may influence the upregulation of the death receptor FAS leading to apoptosis of neuronal cells. It also induces oxidative stress in the neuronal cells, and it is also known to activate STAT1 and induce interleukins IL-6 and IL-8 secretion in neuronal cells.

Protein Tat

Tat is one of the nine genes contained in the HIV virus; it is a protein which is released from HIV-1-infected cells. Its name stands for "Trans-Activator of Transcription". Tat consists of between 86 and 101 amino acids, depending on the subtype.Tat protein plays different roles in the HIV disease process but its acts the main role of vastly increasing the level of transcription of the HIV dsRNA. Before Tat is present, a small number of RNA transcripts will be made, which allow the Tat protein to be produced. Tat then binds to cellular factors and mediates their phosphorylation, resulting in increased transcription of all HIV genes, providing a positive feedback cycle. This in turn allows HIV to have an explosive response once a threshold amount of Tat is produced, a useful tool for defeating the body's response. Tat also appears to play a more direct role in the HIV disease: the protein is released by infected cells in culture, and is found in the blood of HIV-1 infected patients.It can be absorbed by cells that are not infected with HIV, and can act directly as a toxin producing cell death via apoptosis in uninfected "bystander" T cells, assisting in progression toward AIDS. By interacting with the CXCR4 receptor, Tat also appears to encourage the reproduction of less virulent M-tropic strains of HIV early in the course of infection, allowing the more rapidly pathogenic T-tropic strains to emerge later. The protein Tat, was recently shown to be toxic toward cultured neurons. It induces apoptosis in cultured embryonic rat hippocampal neurons. Tat induced caspase activation, and the caspase inhibitor zVAD-fmk prevented Tat-induced neuronal death. Tat induced a progressive elevation of cytoplasmic-free calcium levels, which was followed by mitochondrial calcium uptake and generation of mitochondrial-reactive oxygen species (ROS). The intracellular calcium chelator BAPTA-AM and the inhibitor of mitochondrial calcium uptake ruthenium red protected neurons against Tat-induced apoptosis. zVAD-fmk suppressed Tat-induced increases of cytoplasmic calcium levels and mitochondrial ROS accumulation, indicating roles for caspases in the perturbed calcium homeostasis and oxidative stress induced by Tat. An inhibitor of nitric oxide synthase, and the peroxynitrite scavenger uric acid, protected neurons against Tat-induced apoptosis, indicating requirements for nitric oxide production and peroxynitrite formation in the cell death process. Finally, Tat caused a delayed and progressive mitochondrial membrane depolarization, and cyclosporin A prevented Tat-induced apoptosis, suggesting an important role for mitochondrial membrane permeability transition in Tat-induced apoptosis. Scientific studies demonstrate that Tat can induce neuronal apoptosis by a mechanism involving disruption of calcium homeostasis, caspase activation, and mitochondrial calcium uptake and ROS accumulation [15].

11.3 Appendix C: Paroxetine

Paroxetine is an antidepressant in a group of drugs called selective serotonin reuptake inhibitors (SSRIs). It is used to treat mainly depression, obsessive-compulsive disorder, anxiety disorders, post-traumatic stress disorder (PTSD), and premenstrual dysphoric disorder (PMDD).It works by restoring the balance of serotonin, a natural substance in the brain, which helps to improve certain mood problems. Marketing of the drug began in 1992 by the pharmaceutical company SmithKline Beecham, now GlaxoSmithKline.

Clinical Pharmacology: Pharmacodynamics

The efficacy of paroxetine in the treatment of major depressive disorder, panic disorder, social anxiety disorder and premenstrual dysphoric disorder (PMDD) is presumed to be linked to potentiation of serotonergic activity in the central nervous system resulting from inhibition of neuronal reuptake of serotonin (5-hydroxy-tryptamine, 5-HT). Studies at clinically relevant doses in humans have demonstrated that paroxetine blocks the uptake of serotonin into human platelets. In vitro studies in animals also suggest that paroxetine is a potent and highly selective inhibitor of neuronal serotonin reuptake and has only very weak effects on norepinephrine and dopamine neuronal reuptake. In vitro radioligand binding studies indicate that paroxetine has little affinity for muscarinic, alpha1-, alpha2-, beta-adrenergic-, dopamine (D2)-, 5-HT1-, 5-HT2- and histamine (H1)-receptors; antagonism of muscarinic, histaminergic and alpha1-adrenergic receptors has been associated with various anticholinergic, sedative and cardiovascular effects for other psychotropic drugs.Because the relative potencies of paroxetine's major metabolites are at most 1/50 of the parent compound, they are essentially inactive.

Chemical Structure Description

Paroxetine hydrochloride extended-release tablets are an orally administered psychotropic drug with a chemical structure unrelated to other selective serotonin reuptake inhibitors or to tricyclic, tetracyclic or other available antidepressant or antipanic agents. It is the hydrochloride salt of a phenylpiperidine compound identified chemically as (3S-trans)-3-[(1,3-benzodioxol-5-yloxy)methyl)]4-(4-fluorophenyl)-piperidine hydrochloride hemihydrate and has the molecular formula of C19H20FNO3 • HCl • ½ H2O. The molecular weight is 374.8 (329.4 as free base). The structural formula of paroxetine hydrochloride can be viewed in figure B.1. Paroxetine hydrochloride (hemihydrate), USP is an odorless, white to almost white crystalline powder, having a melting point range of 120° to 138°C and a solubility of 5.4 mg/mL in water. Each enteric film-coated, extended-release tablet contains paroxetine hydrochloride hemihydrate equivalent to 12.5 mg or 25 mg paroxetine. Inactive ingredients consist of colloidal silicon dioxide, hydroxypropyl cellulose, hypromellose, lactose monohydrate, magnesium stearate, methacrylic acid copolymer type C, microcrystalline cellulose, polydextrose, polyethylene glycol, polysorbate 80, sodium hydroxide, talc, titanium dioxide, triacetin and triethyl citrate. In addition, the 25 mg product contains the following coloring agents: D&C Red No. 30 Aluminum Lake, FD&C Blue No. 2 Aluminum Lake, FD&C Yellow No. 6 Aluminum Lake. In addition, paroxetine hydrochloride extended-release tablets may also contain imprinting ink consisting of either black pigment and natural

resin or black iron oxide and propylene glycol. Paroxetine hydrochloride complies with USP Chromatographic Purity Test 1.

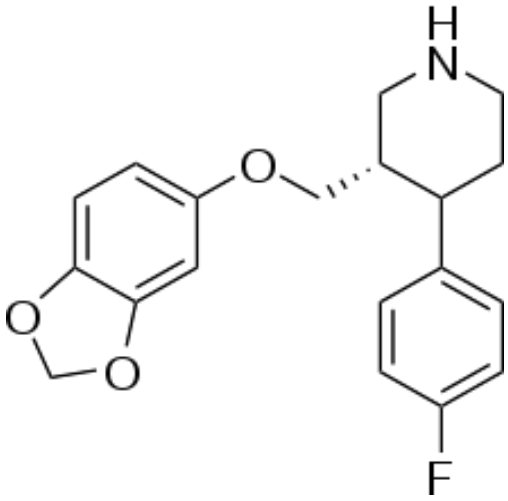


Figure C.1 – Structural formula of paroxetine

Pharmacokinetics

Paroxetine hydrochloride is completely absorbed after oral dosing of a solution of the hydrochloride salt. The elimination half-life is approximately 15 to 20 hours after a single dose of paroxetine hydrochloride extended-release tablets. Paroxetine is extensively metabolized and the metabolites are considered to be inactive. Nonlinearity in pharmacokinetics is observed with increasing doses. Paroxetine metabolism is mediated in part by CYP2D6 and the metabolites are primarily excreted in the urine and to some extent in the feces. Pharmacokinetic behavior of paroxetine has not been evaluated in subjects who are deficient in CYP2D6 (poor metabolizers).

Side Effects in

In adults, the efficacy of paroxetine for depression is comparable to that of older tricyclic antidepressants, with fewer side effects and lower toxicity.Differences with newer antidepressants are subtler and mostly confined to side effects. It shares the common side effects and contraindications of other SSRIs, with high rates of nausea, somnolence, and sexual side effects. Unlike two other popular SSRI antidepressants, fluoxetine and sertraline, paroxetine is associated with clinically significant weight gain and statistically significant increase in the risk of suicidality in adults. Pediatric trials of paroxetine for depression did not demonstrate efficacy and showed an increase in the risk of harmful outcomes, including episodes of self-harm and potentially suicidal behavior.Discontinuing paroxetine is associated with a high risk of discontinuation or withdrawal syndrome. Due to the increased risk of birth

defects, pregnant women or women planning to become pregnant are recommended to avoid or discontinue paroxetine use. [16]

11.4 Appendix D: ZVI Image Format

The ZVI format is a standard OLE (Object Linking and Embedding) compound file which has been developed by Carl Zeiss specifically to meet the requirements of modern microscopy and it is therefore widely used to store Microscope Images. The file extension is *.zvi*.It can be used to save images including multidimensional, time lapse or Z-stack images. The format groups together image data with all relevant information in a compact file:

- Acquisition date
- Microscope settings
- Exposure values
- Size and scale details
- Contrasting method used
- Fluorescent dyes used
- Stage position (x,y)
- Binning

All information is available at all times and images can be reproduced under identical conditions even years later. Microscope images can be saved as uncompressed raw data or using loss-less or JPG compression. Important concepts must now be introduced. **Loss-less** means there is no quality loss due to compression. Loss-less guarantees that it is always possible to read back exactly what it was previously thought to be saved, bit-for-bit identical, without data corruption. This is a critical factor for archiving master copies of important images. TIF, PNG, GIF, BMP and most other image file formats are loss-less. **Compression** works by recognizing repeated identical strings in the data, and replacing the many instances with one instance, in a way that allows unambiguous decoding without loss. This is fairly intensive work, and any compression method makes files slower to save or open. Most image compression formats are loss-less, with JPG files being the main exceptions. The ZVI file type is primarily associated with 'AxioVision' by Carl Zeiss AG., a software capable guiding the use to reproducible results with the aide of structured work-flows. When microscopic images are acquired and processed by using of AxioVision, valuable data about images are produced which should be saved as fully as possible alongside the images. Besides pixel-related image data, information about the acquisition conditions is also particularly important. The ZVI image format saves both types of data together in a single file, regardless of whether an experiment involves the acquisition of simple two-dimensional photomicrographs or 5D experiments with living cells. This can be a video or still image. The ZVI format contains a storage named "Image" for the container image and commonly used information. The image data (pixel array) items are contained in sub-storages named "Item(n)" where (n) is a value from 0 to Count-1. The ZVI format features the following structure:

[_ROOT_]
    [Image]
    <Thumbnail>
    <Tags>
    <\SummaryInformation>
    <\DocumentSummaryInformation>

The ZVI files used in this project are Unsigned 16-bit images, featuring the DAPI layer (4 bands, 16 bits per band) and the DAPI staining has been performed by using 2 different strainers.

TIFF Image Format

The TIFF format is the leading commercial and professional image standard. TIFF is the most universal and most widely supported format across all platforms, Mac, Windows, Unix. Data up to 48 bits is supported. The file extension is .TIF and it supports most color spaces, RGB, CMYK, YCbCr. It is a flexible format with many options. The data contains tags to declare what type of data follows. New types are easy to invent, and this versatility can cause incompatibly, but about any program anywhere will handle the standard TIFF types that might be encountered. TIFF can store data with bytes in either PC or Mac order (Intel or Motorola CPU chips differ in this way). This choice improves efficiency (speed), but all major programs today can read TIFF either way, and TIFF files can be exchanged without problem. Several compression formats are used with TIF. TIF with G3 compression is the universal standard for fax and multi-page line art documents.TIFF image files optionally use LZW loss-less compression. LZW is most effective when compressing solid indexed colors (graphics), and is less effective for 24 bit continuous photo images. Featureless areas compress better than detailed areas. LZW is more effective for grayscale images than color. It is often hardly effective at all for 48 bit images (VueScan 48 bit TIF LZW is an exception to this, using an efficient data type that not all others use). Image programs of any stature will provide LZW, but simple or free programs often do not pay LZW patent royalty to provide LZW, and then its absence can cause an incompatibility for compressed files. But TIF files for photo images are generally pretty large. Uncompressed TIFF files are about the same size in bytes as the image size in memory. Regardless of the novice view, this size is a plus, not a disadvantage. A 4 bit RGB image data is 3 bytes per pixel. That is simply how large the image data is, and TIF LZW stores it with recoverable full quality in a loss-less format. The TIFF files used in this project are RGB Unsigned 24-bit images (3 bands, 8 bits per band).

JPEG Image Format

The JPEG format (name stands for the Joint Photographic Experts Group which created the standard) is a widely used format for those photo images which must be very small files. The files extensions is *.jpg*. The JPG file is fairly small, often compressed to a low percentage of the size of the original data, which is a good thing for data exchange. However, this high compression efficiency comes with a high price. JPG files does not work the same way as Loss-less compressed files, JPG is a big exception: JPG compression is lossy that means "with losses" to image quality. JPG compression has very high efficiency (relatively tiny files) because it is intentionally designed to be lossy, designed to give very small files without the requirement for full recoverability. JPG modifies the image pixel data (color values) to be more convenient for its compression method.

Tiny detail that doesn't compress well (minor color changes) can be ignored (not retained). This allows high size reductions on the remainder, but when the file is opened and the data is expanded to access it again, it is no longer the same data as before. This lost data is like lost purity or integrity. It can vary in degree, it can be fairly good, but it is always unrecoverable

corruption of the data. This makes JPG be quite different from all the other usual file format choices. Even worse, more quality is lost every time the JPG file is compressed and saved again.JPG compression can be selected in many programs to be better quality in a larger file, or to be lesser quality in a smaller file. High Quality corresponds to Low Compression. Since each image varies a little, the file size is only a crude indicator of JPG quality, however it is a rough guide. For ordinary color images (24 bit RGB), the uncompressed image size when opened in memory is always 3 bytes per pixel. For example, an image size of 3000x2000 pixels is 6 megapixels, and therefore by definition, when uncompressed (when opened), this memory size is 3X that in bytes, or 18 MB. That is simply how large the 24 bit data is. The compressed JPG file size will be smaller (same pixels, but fewer bytes). Color compresses better than grayscale files, so grayscale doesn't decrease as much.JPG does not discard pixels. Instead it changes the color detail of some pixels in an abstract mathematical way. JPG is mathematically complex and requires considerable CPU processing power to decompress an image. JPG also allows several parameters, and programs don't all use the same JPG rules. Final image quality can depend on the image details, on the degree of compression, on the method used by the compressing JPG program, and on the method used by the viewing JPG program.

11.5 Appendix E: DAPI

DAPI is a chemical substance which name stands for 4'-6-Diamidino-2-phenylindole (2-(4-amidinophenyl)-1H -indole-6-carboxamidine). It is known to form blue-fluorescent complexes with natural double-stranded DNA (dsDNA), showing a fluorescence specificity for AT, AU and IC clusters. When DAPI binds to DNA, its fluorescence is strongly enhanced, what has been interpreted in terms of a highly energetic and intercalative type of interaction. There is also evidence that DAPI binds to the minor groove (where its fluorescence is approximately 20-fold greater than in the nonbound state). Here it is stabilized by hydrogen bonds between DAPI and acceptor groups of AT, AU and IC base pairs. Because of this property DAPI is a useful tool in various cytochemical investigations and it is used extensively in fluorescence microscopy. For fluorescence microscopy, DAPI is excited with ultraviolet light. When bound to double-stranded DNA its absorption maximum is at 358 nm and its emission maximum is at 461 nm. (This emission is fairly broad, and appears blue/cyan). DAPI will also bind to RNA, though it is not as strongly fluorescent. Its emission shifts to around 500 nm when bound to RNA. DAPI's blue emission is convenient for microscopists who wish to use multiple fluorescent stains in a single sample. There is fluorescence overlap between DAPI and green-fluorescent molecules like fluorescein and green fluorescent protein (GFP), or red-fluorescent stains like Texas Red. However, by using spectral unmixing or taking images sequentially can get around this. Apart from labeling cell nuclei, the most popular application of DAPI is in detection of mycoplasma or virus DNA in cell cultures. Since DAPI will pass through an intact cell membrane, it may be used to stain both live and fixed cells, though it passes through the membrane less efficiently in live cells and therefore doesn't stain as well. Furthermore the concentration of DAPI needed for live cell staining is generally much higher than for fixed cells. It is labeled non-toxic in its MSDS and although it has been shown not to have mutagenicity to *E. coli*, it is labeled as a known mutagen in manufacturer information. DAPI has greater photostability than Hoechst dyes, another common nuclear counterstain, when it is bound to double stranded DNA [17] [18].

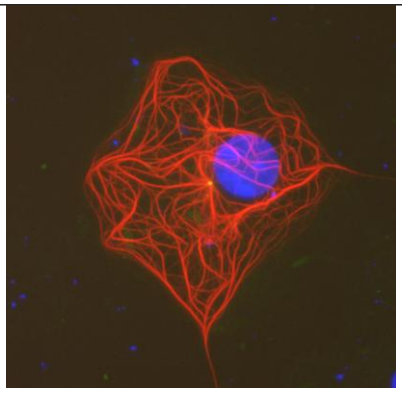


Figure E.1 - COS7 cells transfected with DCX DsRed and stained with gamma tubulin/FITC and DAPI.

DAPI staining procedures

Cells are grown in complete RPMI-1640. Harvested cells are washed once with PBS, and then resuspended in PBS containing 0.1 % Triton X (to induce holes in the cells' membrane = increase permeability) and incubated for 10 min on ice. Spin cells down and resuspend them at 5000 cells/µl in 4% PBS buffered paraformaldehyde solution containing 10 µg/ml 4'6-diamidino-2-phenylindole (DAPI, Sigma). 10 µl of this suspension are placed on a glasslide

and covered with a coverslip. The morphology of the cells' nuclei is observed using a fluorescence microscope (Olympus BH Series) at excitation wavelength 350 nm. Nuclei are considered to have the normal phenotype when glowing bright and homogenously. Apoptotic nuclei can be identified by the condensed chromatin gathering at the periphery of the nuclear membrane or a total fragmented morphology of nuclear bodies [19].

A blue nuclear counterstain for fluorescence microscopy

DAPI (diamidino-2-phenylindole) is a blue fluorescent probe that fluoresces brightly when it is selectively bound to the minor groove of double stranded DNA where its fluorescence is approximately 20-fold greater than in the nonbound state. This selectivity for DNA, along with cell permeability allows staining of nuclei with little background from the cytoplasm, making DAPI the classic nuclear counterstain for immunofluorescence microscopy. DAPI has greater photostability than Hoechst dyes, another common nuclear counterstain, when it is bound to double stranded DNA. DAPI has an excitation maximum at 358 nm and an emission maximum at 461 nm.DAPI is compatible with fluorescein and rhodamine dyes, as well as with DyLight and Alexa Fluors, for nuclear counter staining of DNA in fluorescence imaging. Stock solutions of DAPI are stable in solution at -20°C when stored in the dark.

Chemical and Physical Properties of DAPI Dye]

Molecular mass: 350.25

Excitation wavelength: 345 nm (near 360 nm when bound to dsDNA)

Emission wavelength: 455 nm (456-460 nm when bound to dsDNA)

Extinction coefficient: ~30,000/M cm at 347 nm in methanol

CAS #: 28718-90-3

Purity by HPLC: >95% (most lots >98%)

Solubility: Soluble in DMF, water and various non-phosphate aqueous buffers

Storage: Room temperature (RT), protected from light

Reactive groups: Non-covalent; binds to minor groove of double-stranded DNA

General Applications for DAPI stain

1) Assaying DNA in solution;
2) Diagnosing mycoplasmal infection of cell cultures;
3) Measuring nuclear content and sorting chromosomes in flow cytometry;
4) Assessing apoptosis;
5) Detecting nuclei and organellar DNA in immunofluorescent and in situ hybridization procedures;

6) Replacing ethidium bromide for staining DNA in agarose gels;
7) Counterstaining nuclei in histochemical methods when red-fluorescent antibodies have been used to detect specific targets;
8) Reports also indicate that DAPI will bind to polyphosphates and other polyanions, dextran sulfate and SDS. [20]

11.6 Appendix F: The fluorescence microscope

The fluorescence microscopy is an essential tool in biology and the biomedical and material sciences as well. It is based on the well-known physical phenomena of fluorescence and due to attributes that are not readily available in other contrast modes with traditional optical microscopy. it allows to achieve a level of sophistication that far exceeds that of simple observation by the human eye. The modern fluorescence microscope combines the power of high performance optical components with computerized control of the instrument and digital image acquisition. At present, optical image formation is the first step toward data analysis. The microscope accomplishes this first step in conjunction with electronic detectors, image processors, and display devices that can be viewed as extensions of the imaging system [21].

A brief introduction to the fluorescence microscopy

The fluorescence microscopy technique is based on the physical phenomena of fluorescence, that is the absorption and subsequent emission of light (or laser light) by organic and inorganic specimens. The light emission is through the fluorescence process is almost simultaneous with the absorption of the excitation light due to a relatively short time delay between photon absorption and emission, which usually lasts for less than a microsecond, in time duration. If emission persists longer after the excitation light has been extinguished, the phenomenon is referred to as phosphorescence.

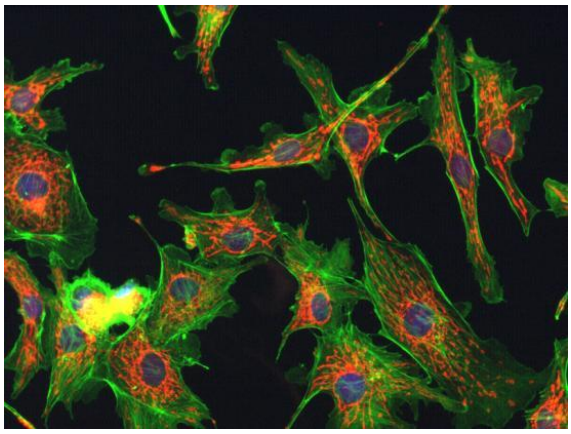


Figure F.1 : Image of cells acquired at the Integrated Microscopy Facility, University of California, Santa Barbara

The fluorescence phenomenon has been described firts in 1852, when the british scientist Sir George G. Stokes observed that the mineral fluorspar emitted red light when it was illuminated by ultraviolet excitation. He also noted that fluorescence emission always occurred at a longer wavelength than that of the excitation light. Early investigations in the 19th century showed that many specimens (including crude drugs, vitamins, and inorganic compounds) fluoresce when irradiated with ultraviolet light. The use of fluorophores was initiated in biological investigations to stain tissue components, bacteria, and other pathogens. Several of these stains were highly specific and stimulated the development of the fluorescence microscope. By using an array of fluorophores it is possible to identify cells and sub-microscopic cellular components with a high degree of specificity amid non-fluorescing

material. In fact, the fluorescence microscope is capable of revealing the presence of a single molecule. Through the use of multiple fluorescence labeling, different probes can simultaneously identify several target molecules simultaneously. Although the fluorescence microscope cannot provide spatial resolution below the diffraction limit of specific specimen features, the detection of fluorescing molecules below such limits is readily achieved.

The fluorescence microscope

The basic working principle of a fluorescence microscope is the following:

- 1) Irradiation of the specimen with a desired, wavelength-specific band light (or laser light).
- 2) Separation of the much weaker emitted fluorescence from the excitation light through the use of a spectral emission filter.
- 3) Composition of the multi-color images of several types of fluorophores by combining the several single-color images obtained at ther former step.

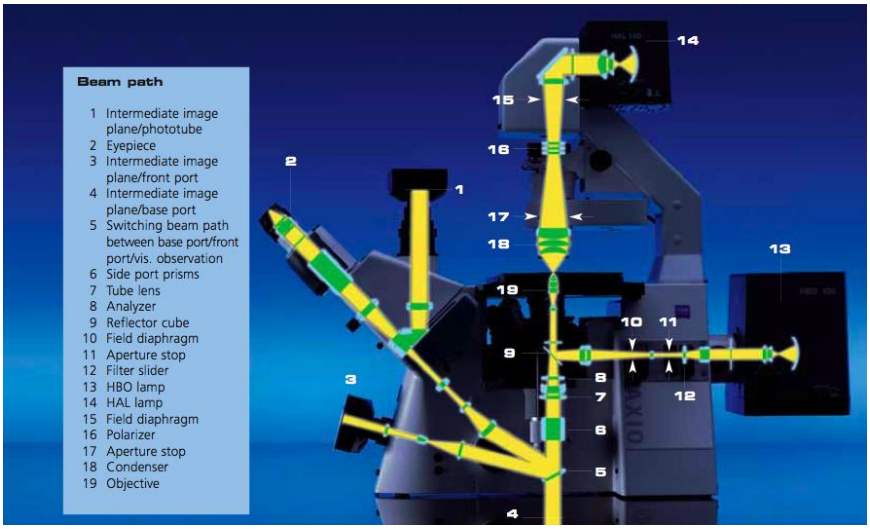


Figure F.2 - Internal view of a Zeiss AxioObserver Z1 and list of main components

About the specimens, in most cases a component of interest is labeled specifically with a fluorescent molecule called a fluorophore (such as green fluorescent protein (*GFP*), *fluorescein* or *DyLight 488*). The light is absorbed by the fluorophores, causing them to emit light of longer wavelengths (perhaps of a different color than the absorbed light). In the simplest form of fluorescence microscope, the laser operates continuously, and its wavelength is chosen to be in an absorbing spectral region of the fluorescent molecules, which can therefore be excited with single-photon absorption. Fluorophores lose their ability to fluoresce as they are illuminated in a process called photobleaching. Special care must be taken to prevent photobleaching through the use of more robust fluorophores, by minimizing illumination, or by introducing a scavenger system to reduce the rate of photobleaching.

Another typical components of a fluorescence microscope are the light source (xenon arc lamp or mercury-vapor lamp), the excitation filter, the dichroic mirror (or dichromatic beamsplitter), and the emission filter. The filters and the dichroic are chosen to match the spectral excitation and emission characteristics of the fluorophore used to label the specimen. In this manner, the distribution of a single fluorophore (color) is imaged at a time.
Most fluorescence microscopes in use are epifluorescence microscopes (i.e. excitation and observation of the fluorescence are from above the specimen). At the present, microscopy depends heavily on electronic imaging to rapidly acquire information at low light levels or at visually undetectable wavelengths. These technical improvements are not mere window dressing, but are essential components of the light microscope as a system. In a properly configured microscope, only the emission light should reach the eye or detector so that the resulting fluorescent structures are superimposed with high contrast against a very dark (or black) background. The limits of detection are generally governed by the darkness of the background, and the excitation light is typically several hundred thousand to a million times brighter than the emitted fluorescence. Computerized control of focus, stage position, optical components, shutters, filters, and detectors is in widespread use and enables experimental manipulations that were not humanly possible with mechanical microscopes. The increasing application of electro-optics in fluorescence microscopy has led to the development of optical tweezers capable of manipulating sub-cellular structures or particles, the imaging of single molecules, and a wide range of sophisticated spectroscopic applications. [21] [22]

11.7 – Appendix G: A Matlab-to-C++ Integration

The basic idea behind the realization of a C++ to Matlab code integrations (and Matlab-to-C++, as well) was that both programming languages can be used to achieve accurate results in various steps of the image analysis protocol. I was given a Matlab Integration Plan from my Visiopharm's supervisor (see Table G.1, below) containing tasks allowing to create a protocol of integration by using the Microsoft Visual Studio C++ 2005 and Matlab2009b. The final purpose was to obtain a complete data sharing between the two programming languages.

General purposes

The main advantages in using each programming language are listed below:

- By meaning of C++ it was possible to use directly the Visiopharm's SDK functions that have been used, optimized and edited with the correct parameters to perform particular operations on the given images. Here were included applications for image pre-processing, classification and post-processing.
  After a dedicated analysis of the database images has been carried out, that protocol can also be easily integrated and run in Visiopharm´s Software (VIS). Furthermore, by using of VIS it was possible:

  I. testing the efficiency of the images analysis protocol on the entire database by using of VIS' functionalities for batch processing;

  II. getting a quick and helpful support to test whether an algorithm (or a protocol) works well or not by mean of VIS functionalities (e.g. areas measurement or shapes detection).

- By meaning of Matlab it is quicker to implement new functions working with data structures like matrixes and arrays which the images will be converted into.
  It is furthermore possible to use several tools and libraries for images analysis.

The integration of generic functions from Matlab to C++ had been performed by using of the Matlab Runtime Compiler [5]; it was installed, tested and run with simple functions.
The programming softwares Matlab R2009b and Microsoft Visual Studio C++ 2005 were installed, as well.

The final result consists of two main stages:

1) A Matlab executable file, created by using the Matlab Runtime Compiler;

2) A C++ project capable of:

- Calling a Matlab function from within the C++ code itself.

- Passing a common data type (converted from a C++ typical data type, e.g. a matrix of integers) to the Matlab executable file, executing its embedded function. The data can be read, edited and processed by both programming languages.

- Getting the outputted result from the Matlab function in the common data type that has to be re-converted to a typical C++ data type.

Matlab Integration plan realization

The main goal to achieve is the creation of common data type that can be easily be handled and shared by both programming languages. The main data type class instance that had to be created was a mwArrays class objects that permit to pass input/output arguments to MATLAB Compiler generated C++ interface functions. This class consists of a thin wrapper around a MATLAB array. As explained in further detail in the MATLAB documentation, all data in MATLAB is represented by matrices (in other words, even a simple data structure should be declared as a 1-by-1 matrix). The *mwArray* class provides the necessary constructors, methods, and operators for array creation and initialization, as well as simple indexing. Since a *mwArray* object has been created, it can be processed as if being working in a .m file. Furthermore in the C++ code the pointers data type have been used. The library function calling the Matlab executable (mlf) requests an pointer to a mwArray input type, which can be a *mxArray´*s class object in C++ (in which a *mwArray* is quickly convertible). Hence the source code also allows to load, read and store data into *.mat* files from/to the working directory by using of *mxArray´*s class functions. After the C++ data type was converted into a mwArray and afterward into a mxArray, it is finally possible to call a M-function (invoked from within the C++ code) that

- receives a mxArray as input parameter and give another mxArray as output;

- executes a Matlab functions on the mxArray by calling a Matlab executable;

- returns as output another mxArray as result.

By a re-conversion into an mxArray, then mwArray and finally into a typical C++ data type, the Matlab-modified data it is processable by C++ again.

#	Task	Done / Date
1	**Matlab:** Create a small Matlab function and build a dll and ctf file from that. *Endpoint: Usable dll and ctf files outputted*	V (Feb. 24-25)
2	**C++:** Create a small Visual Studio Project and call functions from your Matlab dll. *Endpoint: Small program where the Matlab dll is successfully used*	V (Mar. 1-5)
3	**C++:** Create functions for converting a CImage (from Visiopharm's SDK) to a mwArray and back again. Furthermore create a small function that e.g. subtracts 100 from each pixel in your mwArray. *Endpoint: Small program that loads and displays an image, converts it from CImage to a mwArray, subtracts 100 from each pixel, reconverts the image to a CImage and displays the result.*	V (Mar. 12)
4	**Matlab:** Create a small function that e.g. subtracts 100 from each pixel in an image. Having tested that it works in Matlab, build a dll and ctf from it. *Endpoint: change your small program from #3 to do the subtraction of 100 using the Matlab dll instead.*	V (Mar. 17)
5	**C++ and Matlab:** Expand #3 and #4 to input and output labels and masks as well, including replacing overwriting the original image's labels and masks with your outputted ones. *Endpoint: A small program capable of setting all label pixels inside mask 1 to 1 if the corresponding feature image pixel is larger than 100.*	C++ V (Mar.19) M V
6	**C++:** Get a program from Michael that functions as a classification step in Visiomorph and generates a dll. Take the feature image, labels and masks that get inputted from VIS and use what you have developed previously to convert it to mwArrays call the Matlab function developed in #5, overwrite the image's labels and masks with your outputted labels and masks. *Endpoint: Compile the above described program so that you can use your own Matlab classification from Visiomorph just by pressing classify.*	C++ V (Mar.29) M V

Table G.1 – Matlab-to-C++ Integration Plan

11.8 Appendix H: References

[1] Goldman L, Ausiello D, eds. Cecil Medicine. 23rd ed. Philadelphia Pa: Saunders Elsevier; 2007: sect XXIV

[2] Juliane Haedicke, Craig Brown, and Mojgan H. Naghavi1 (School of Medicine and Medical Science, University College Dublin): The brain-specific factor FEZ1 is a determinant of neuronal susceptibility to HIV-1 infection

[3] American Society for Microbiology
Journal of Virology, June 2003, p. 6899-6912, Vol. 77, No. 12

[4] HealthyPlace.com Staff Writer
http://www.healthyplace.com/other-info/psychiatric-medications/paroxetine-paxil-full-prescribing-information/menu-id-72/ , jan 04, 2009.

[5] The Mathworks, Inc.
How do I create a C/C++ shared library with MATLAB Compiler that can be used in a Microsoft Visual C++ 2005 project using Windows Forms Application,
http://www.mathworks.com/support/solutions/en/data/1-2QTWCE/

[6] The kuleuven student Staff Writer
http://www.student.kuleuven.be/~m0216922/CG/filtering.html#Introduction

[7] Ruye Wang
http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node3.html

[8] K.S. Fu, J.K. Mui
A survey on image segmentation, Pattern Recognition 13 (1981) 3-16.

[9] Hamid Rahimizadeh, M.H Marhaban, R.M Kamil, N.B Ismail
Color Image Segmentation Based on Bayesian Theorem and Kernel Density Estimation
European Journal of Scientific Research ISSN 1450-216X Vol.26 No.3 (2009)

[10] J.M. Carstensen
Image Analysis, vision and computer graphics , 2nd edition, Tecnichal University of Denmark, 2002

[11] Danielle Azar
http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton
Pattern Recognition course given by Prof. Godfried Toussaint at McGill University (Montreal, 1997)

[12] Nielsen MH, Pedersen FS and Kjems J
Molecular strategies to inhibit HIV-1 replication", Retrovirology 2005; 2:10

[13] Szabolcs Takács, Sarolta Bankó, Andras Papp
Acute electrophysiological effects of a metal and a non-metal mitochondrial toxin

[14] Kwong PD, Wyatt R, Robinson J, Sweet RW, Sodroski J, Hendrickson WA
Structure of an HIV gp120 envelope glycoprotein in complex with the CD4 receptor and a neutralizing human antibody, WA. NATURE 393 (6686): 648-659 JUN 18 1998

[15] Jeang, K. T.
Human Retroviruses and AIDS: A Compilation and Analysis of Nucleic Acid and Amino Acid Sequences. Los Alamos National Laboratory (Ed.) pp. III-3–III-18, 1996

[16] Net Resources International Staff (a trading division of SPG Media Limited)
http://www.drugdevelopment-technology.com/projects/paxil/

[17] Molecular Probes, Inc.
Invitrogen DAPI Nucleic Acid Stain

[18] Scott Prahl
http://omlc.ogi.edu/spectra/PhotochemCAD/html/dapi(H2O).html

[19] CellDeath.de Staff Writer
http://www.celldeath.de/apometh/dapi.html

[20] The Pierce Protein Researcher staff
Counterstaining Reagent (DAPI Nuclear Stain),
http://www.piercenet.com/products/browse.cfm?fldID=01041204

[21] Kenneth R. Spring, Michael W. Davidson
http://www.microscopyu.com/articles/fluorescence/fluorescenceintro.html

[22] Zeiss web user guide for Zeiss AxioObserver Z1,
www.zeiss.de

[M] Michael Grunkin, Visiopharm internal document for development support