Acknowledgment

I would like to use this opportunity to express my deepest gratitude to many people without whom this enriching experience at Lake Diamond wouldn't have been possible.

We have set light, an ever-burning ame of gratitude and deep sense of obligation to our honorable supervisor Ph.Doctor Rebai Ahmed, for his valuable guidance, constructive criticism and inspiring attitude during our ShineOn project. This opportunity made us realize the value of the experiences in any different working environment with its challenges.

I put a head my thanks to Dr. GALLO Pascal, Lake Diamond's CEO and the leader on Diamond producing, due to whom this incredible experience took path in excellent conditions and for his believe in me to make this incredible internship opportunity happens. Dr.Gallo, with his trust in me and motivation, brought out passion to our work environment and lead with LakeDiamond to be the leader of diamond innovation territory, in Switzerland.

I will never end thanking Dr.Pascal GALLO and my tutor, Mr.Nicolas MALPIECE,for their faith in me and effort to get me to join LakeDiamond and offer me the chance to work on such an innovative project I was passionned about during 6 months of career journey, which turned to be the changing point of my future path.

I'm not only obliged to thank MR. MALPIECE in regards for the internship supervision and guidance, but also for his wise conversations that helped me grow as a person in a professional life. His daily support and advice allowed me to exceed my limits and acquire much thorough hard and soft skills. I have gained a lot of knowledge to build a career of an engineer thanks to him.

I address a special thanks to Mr.Theophile MOUNIER, who spent time from his valuable schedule to deal with the tedious administrative tasks to allow me to work on Swiss soil. As he was the one who was always available to answer my questions on how to get me integrated into a swiss life model of living.

Moreover, the expression of my deepest gratitude goes to Mr.Eli Kapon Head of photonics without whom the projects of photonics wouldn't have happened the way he lead to.
I'm also grateful for Dr.Jelena Vukajlovic and Dr.Riad Berrazouane for warmly welcoming me into the nanotechnology laboratories.

I would express how proud I was to work with Dr.Mehdi NAAMOUN, diamond production supervisor, for being the example with his team work skills.
Mr CALIMAN Andreï, Diamond Laser Project Leader, and Dr. MEREUTA Alexandru, researchers at EPFL laboratory, also participated in the proceedings of my internship.

I won't forget to take time and make a note for the work of the previous intern Ilyass ZAKDALI, that contributed to keep a good light-hearted work for ShineOn project.
I mention Margot Ohnimus, the new intern, she created a really good atmosphere within the internship, allowing me to do my job in a warm atmosphere. As by helping each other I could improve myself on working as a team.

Great appreciation goes to the rest of LakeDiamond group whom contributed to helped us during the project and social integration with the life changes. The whole program brought us together to appreciate the true value of respect.

This written trace of my internship would not be complete without a grateful notion of Moez Maamer, the CTO of Sirop startup in Geneva, whom participated to the good running of my internship at its difficult tasks.

Finally, I would like to extend my acknowledgment to the support of my beloved family throughout this internship. As to my university, ESPRIT for bringing me such knowledges, memories and giving me the chance to become a qualified engineer inside its institution.

Contents

List of Figures

List of Tables

List of acronyms

EPFL : Federal Institute of Technology in Lausanne.

ESPRIT : Private Higher School of Engineering and Technologies.

P.O.C : Proof of concept.

GANTT : Generalized Activity Normalization Time Table.

2-TUP, Y : Two Track Unified Process, agile methodology.

A.P.I : Application program interface.

SSH : Secure shell protocol.

R.G.B : Red, Green, Blue.

1 General introduction

During my ESPRIT Private Higher University of Engineering and Technology course, I leaned towards software engineering, administration network and computer science with a deep fascination in regard to computer vision and artificial intelligence in general.
When the opportunity has arisen to apply my technical skills, I got engaged for LakeDiamond Power beaming project for wireless drone recharge with diamond laser. Lake Diamond aims to have a worldwide impact, combining innovative projects, and the use of new technologies.

As the last step of our training as engineers we have to do an internship of 6 to 8 months. Which is the chance to apply our knowledge to design solutions and also to show our worth as future engineers in a professional environment.this internship holds a bigger stake in our future professional career.

My sight was naturally set towards computer vision based internships. Indeed, I managed to do my internship abroad at LakeDiamond, in Switzerland. The focus of this internship is directly aligned with what I was looking as skills to obtain during this experience.
As an intern at Lake Diamond in collaboration with EPFL laboratory, my mission was to work on a new project called ShineOn and after thoroughly studying the hard context of our subject: Visual real tracker for wireless recharge of drones, we have decided to start by

implementing a project for drones detection which aims to manipulate the laser platform to the position of the drone.
In terms of the quality of our diamond laser and essential necessities for a better detection of information that could be offered by our system for drone detection, as with the manifold services that will be proposed from our Jetson TX2 card implemented on the laser platform.Our project will be divided into different parts:
Each part has its own technology : Software part for drone detection algorithm, embedded system part for API algorithm to manipulate the laser platform, implementation for Client / Server socket on secured ssh protocol for a remote control.

To achieve it, I had to start by awakening my knowledge on computer vision basics while testing the already existed eight algorithms techniques, to compare and implement the best adaptable algorithm.
This is how I could figure out the limits in each algorithm and move to developing an adaptable algorithm to our drone detection system after understanding the basics of object detection algorithms.

After finding an adaptable mathematics solution for detecting objects through a simple computer vision iterations, we developed a simple prototype using python version 3, to make fast comparing the results precision at the end.

In further steps, as we will be integrating our algorithm with an API of Drotek written in C++ on Jetson TX2 system card for manipulating the laser platform. We had to implement our proposed solution of the prototype to c++.

At the end to communicate through ssh protocol, we integrated a c++ linux socket , server and client architecture between the real-time drone detection client and the server on the Jetson tX2 card API for laser platform manipulation. To reach the objectives in an efficient way, my internship was divided into 3 parts with a last iterations which was to integrate the 3 following algorithms:
-Algorithm for detecting drone,

-API for manipulating the platform,
and a socket to connect the client code and the server of Jetson tx2 card , on a secured ssh protocol.
Our integrated project of these algorithms is to maintain on an already deployed Jetson TX2 card in our system.
To implement our proof of concept, we followed the simple agile methodology: Cascade to make it fast and simple, as it could be modified after the end of this implementation.

The principle of this methodology is to develop a software upgraded Incrementally, maintaining a fully transparent list of requests for changes or corrections to develop.
The method is based on iterative development at a constant rate for a period of steps, following a GANTT diagram that defines tasks and deadlines for our progress.

This internship brought me a new knowledge and an enlightening professional and personal experience. And, more specifically, taught me how and where to research efficiently the needed materials and how to choose from a pool of solutions the one that corresponds the best with our needs, while keeping an eye out for newly designed algorithms.

ShineOn project was about implementing a visual tracking system for drones.
In fact, the success of any study depends on the quality of its departure. Therefore, throughout this chapter, which represents the preparatory phase of the project we concentrated on presenting an overview of our scientific research implementation.
In this project, the target application is wireless recharge for standard commercial drones.
This report resume my entire researches, experiments, results and activities by merging technologies for ShineOn project, a drone project based on a new replenishment technique for power beaming recharge, at LakeDiamond, Switzerland.
This work took 6 months of hard work and scientific discipline and it must be kept CONFIDENTIAL.

2

Functional branch, Specification and Analyses

Introduction

On the aim to present you an over view on the project, we went through a global analyse which will be presented in the next chapter.

2.1 Customer specifications

2.1.1 Customer overview

LakeDiamond SA, Lab-grown diamonds for high-tech industrial applications, especially power beaming applications. A Swiss start-up led by our CEO Pascal Gallo, is leading the way for lab-grown ultra-pure diamonds. Our synthetic diamonds have numerous potential technological applications, beyond jewellery. A custom-developed Micro-Wave Chemical Vapor Deposition reactors grow the highest quality diamond on earth, mono-crystals up to the centimetre in scale.

Figure 2.1: LakeDiamond SA Logo

The Swiss company LakeDiamond, founded in 2015, produces and markets monocrystalline diamonds through an innovative growth technology. LakeDiamond is also developing diamond-based components in partnership with EPFL laboratories. Diamond surpasses all other materials in terms of transparency, hardness and thermal conductivity. These extraordinary properties are exploited in opto-electronic components, including lasers, with multiple applications. LakeDiamond is at the search for motivated students to participate in the development of systems integrating these lasers, which hold a record of power in the middle infrared. Incorporated in 2007 in Switzerland, and builded on years of research at EPFL, LakeDiamond Micro-Wave Chemical Vapor Deposition reactors are capable of growing the highest quality of diamond on earth thanks to a perfectly controlled and precise technology and cutting edge diamond growth expertise.

Figure 2.2: LakeDiamond SA position

LakeDiamond is a global leader in the production and transformation of ultra-pure lab-grown diamonds for high-tech industrial applications.

Table 2.1: LakeDiamond presentation

Website	http://www.lakediamond.ch
Industry	Nanotechnology
Company size	11-50 employees
Headquarters	Lausanne, Vaud
Type	Privately Held
Founded	2015
Specialties	Synthetic diamond growth, Synthetic diamond characterization, Synthetic diamond transformation, Man-made diamonds, CVD, Diamond platelets, Diamond etching, Laser-power beaming, Blockchain, Swiss watches, Transistors, Photonics, EPFL, Quantum computing, Ethereum, Smart contract, cryptocurrency, token, and coin

2.1.2 Customer story

The qualities attained make the company's ultra-pure lab-grown diamonds plates suitable for the most demanding applications. The company sells diamond plates and explore new exciting high-tech industrial applications beyond the high-end jewellery market, such as diamond micro-mechanical parts, Diamond laser power beaming, diamond high power transistors and diamond high-precision magnetometers and this is how its story have started:

2007
Pascal Gallo earns his PhD on crystal growth in collaboration with Albert Fert, Nobel Prize 2007.

2010
He joins Eli Kapon's group at EPFL, develops diamond-based lasers and breaks a world record for laser energy transmission. Disappointed about not finding good supplies of ultra-pure diamonds he decides to grow his own.

2011
Pascal Gallo meets Theophile Mounier, Christophe Provent and David Rats. They develop a CVD reactor to grow ultra-pure diamonds and create a business plan.

2015
LakeDiamond SA is incoporated.

2017
Alex Kummerman joins the team with the plan to leapfrog from start-up to industrial scale production capacity. He organizes the strategy to launch an ICO and brings along Romain Braud in the process.

2018
World premiere: Swissquote, a Swiss bank supports LakeDiamond ICO.

2019
Launch of a public ICO with a hard cap of CHF 60.5 million.

2.1.3 Customer requirements

2.1.3.1 Real-Time application for detecting and tracking drones

LakeDiamond combine all the competencies and infrastructure needed to grow, transform and sell diamonds. We are growing through different research application on the aim to apply a wireless recharge for detected and tracked drones using computer vision and I.A applications.

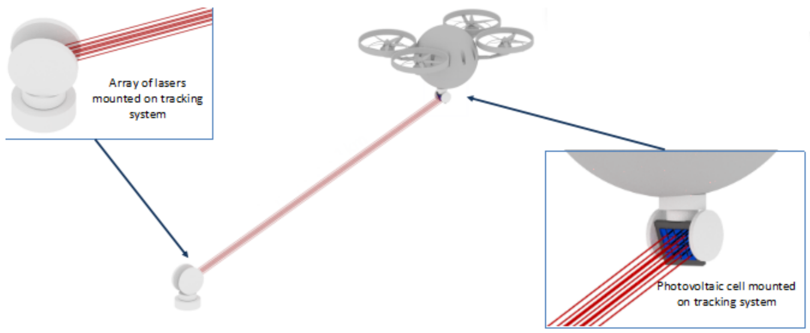

Figure 2.3: System integration

Drones are being used for a growing number of purposes. Their designs are ever more efficient, and techniques for flying them are being further refined all the time. But drones still have the same weak point: their battery. This is particularly true of propeller drones, which are popular for information-gathering purposes in dangerous or hard-to-reach regions. These drones can fly for only around 15 minutes at a time because their engines quickly burn through their batteries. One way of addressing this limitation without weighing the drones down would be to recharge them while aloft using a power beaming system: an energy-rich laser beam that is guided by a tracking system and shines directly on photo-voltaic cells on the drones' exterior.

For this reason, we aim to develop our own optimised algorithm to detect the flying object position with its highest precision and get to manipulate our laser platform by distance in a way that we could insure a real-time tracking while recharging.
On the following section, we'll be describing the most important parts of the project.

Figure 2.4: Application general structure composition

2.1.3.2 Python P.O.C

After going through an analysis that compares the 8 existing algorithms and getting out with different limits and conclusions, we found ourselves ought to implement our own way and algorithm using computer visions strategies to find how we could detect our drones without the need of any datasets.

We'll be describing the principle of our strategy in the following chapter

2.1.3.3 Socket deployment

To communicate between our client and Drotek A.P.I deployed on the server of Jetson TX2 card, we had to implement a socket for a secured communication on an ssh protocol between both client and server as a middle ware.

We describe its general schema in the following diagram:

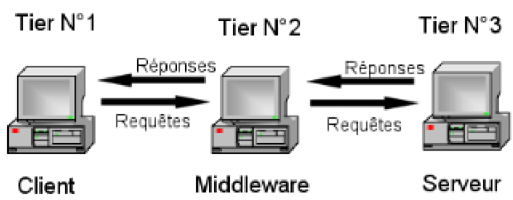

Figure 2.5: Socket middle-ware general schema presentation

2.1.4 Methodology design

During this project we'll be following the 2-TUP (Two Track Unified Process.

2-TUP offers a development cycle in Y, which dissociates the technical aspects from the functional aspects. It starts with a preliminary study that essentially consists in identifying the actors who will interact with the system to be built, the messages that the actors and the system are exchanging, to produce the specifications and to model the context (the system is a box black, the actors surround him and are connected to him, on the axis which links an actor to the system one puts the messages that the two exchange with the direction). The process is then articulated around three essential phases:

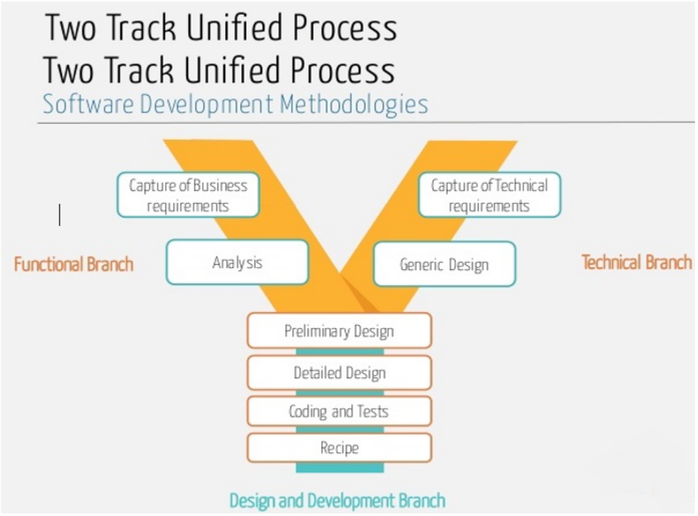

Figure 2.6: 2-TUP approach

2.1.4.1 Choice justification

The Unified Process method is an iterative and incremental method just like the life cycle adopted in our tasks iterations trough the whole project.
The 2-TUP separates the technical aspects from the functional aspects and revolves around 3 different branches: the technical branch, the functional branch and the implementation phase.
As our R.N.D project went throughout all these 3 steps, starting by the research phase moving on to the development phase and ending by testing our prototype life cycle by the end we found that we ought to follow this methodology till the delivery of the project.
Following we describe the details of our agile choice: - The functional branch capitalizes on the knowledge of the business of the company. This branch captures business needs, resulting in a model focused on the end-user business.

- The technical branch capitalizes on technical know-how and / or technical constraints.

The techniques developed for the system are independent of the functions to be performed.

- The implementation phase consists of bringing the two branches together, allowing for an application design and finally the delivery of a solution adapted to the needs.
Scrum methodology, doesn't discuss a lot of details. But that's OK because it's a process framework. Also, this RND project have been done through years of continuous work of different individuals but never been done in groups of people, otherwise scrum methodology impose a special structure of groups and team of works like a product Owner, development Team, scrum Master.
This was another reason to adopt this methodology, as we need that the prototype would be flexible enough to be modified and changed through years of work experiences.
After taking a look at SCRUM agile methodology, it doesn't seem to lead a project since it doesn't take in consideration the technical side of the project but it focus on the events Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) and artifacts (Product Backlog, Sprint, and Increment), this is how I concluded that the best is to plan a project : Divide the project iteration and organise it through a Gantt while using the Y method.

2.1.5 Customer specifications

2.1.5.1 Problematic

Over 1.1 billion people in the world do not have access to electricity. In Africa, it represents most of the half of the population. Almost all of those people live in rural or remote areas where the cost of new infrastructure is prohibitive. Power Beaming is part of the solution, it can be used as a sustainable and economical way to transport energy to remote locations. However, that technological feat cannot be accomplished without the unique properties of diamond. LakeDiamond works on both developments for lab-grown diamonds and for power beaming. Our prototype to lunch this experience for wireless recharge of civilian drones was the first on a worldwide ladder, a sustainable solution for energy and its technology deployment.

Currently drones have a flight autonomy of about fifteen minutes. With our power-beaming technology, it is possible to increase this flight time, as well as to reduce the size of the on-board batteries. With the use of diamond laser, the heat emitted by the laser beam.

Even as drone technology advances, power constraints limit the amount of equipment unmanned systems can carry as well as the time they can stay in the air.
« Drops the batteries - Diamonds and Lasers could power your drone »

2.1.5.2 Project description

LakeDiamond is developing a technology for its ShineOn project to send energy, wireless, to autonomous vehicles. Currently, drones have a flight range of approximately 15 min; thanks to ShineOn technology, it is possible to increase this flight time, as well as to decrease the size of the batteries on board. In the long term, the drones will have unlimited autonomy, and will be able to heavy loads. The ShineOn project implements a technology of "Power Beaming", that is to say, a technique of wireless charging. This technique consists of a transmitter producing a light beam of power and a receiver that converts the received light energy into electrical energy. When the drone is in flight, the transmitter targets the receiver that is embedded on the drone to transmit it light energy. The transmitter, meanwhile, remains on the ground. Thus, the drone does not need to return to the ground to recharge. In addition, a tracking system allows the transmitter to synchronize with the drone. In order to simplify the alignment of the beam on the receiver, a communication link between the transmitter and the drone is established.

2.1.5.3 Functional requirements

In the course of the internship, the optimization of the demonstrator will have to be realized by:
- Develop a computer vision algorithm to detect flying objects.
- Deployment of drone detection algorithm.
- Deployment of tracking drone algorithm.
- Optimization of video tracking.

- Piloting a mobile platform to guide the transmitter.
- Synchronization between the movements of the drone and the transmitter.
- Tracking performance measurements (accuracy, response time ..)
- Test for the best technology performance.

2.1.5.4 GANTT project

See annex 1.

2.1.5.5 Analysis of the existence

Currently on the market, we charge drones only with a cable. Unless, As for detecting the drones we have already 8 implemented algorithms but after getting through the tests it seems to have a large limits that we conclude in the table that follows.
Study and tests of 8 existing algorithms for detection of drone, mainly:

1. On-line boosting.
2. Multiple Instance Learning(MIL), using machine learning.
3. Median Flow.
4. Tracking-Learning-Detection (TLD), using deep learning.
5. Kernelized Correlation Filter(KCF).

2.1.5.6 Limits of existing algorithms

In the following pictures we tried to make some tests on the existing algorithms such as KFC, Boosting, Mosse, TLD and conclude it's limits, so one of it's limits is that in a bad quality of pictures it lose the detection of the object trying to detect another areas.
We show the results of the algorithm detection in green rectangle and in blue the real position of the drone.

Other algorithms just like: CSRT, MIL and MedianFlow are performance and precise enough but really slow at time of tracking and detection.

While the deep learning and machine learning algorithms are limited in a way that we

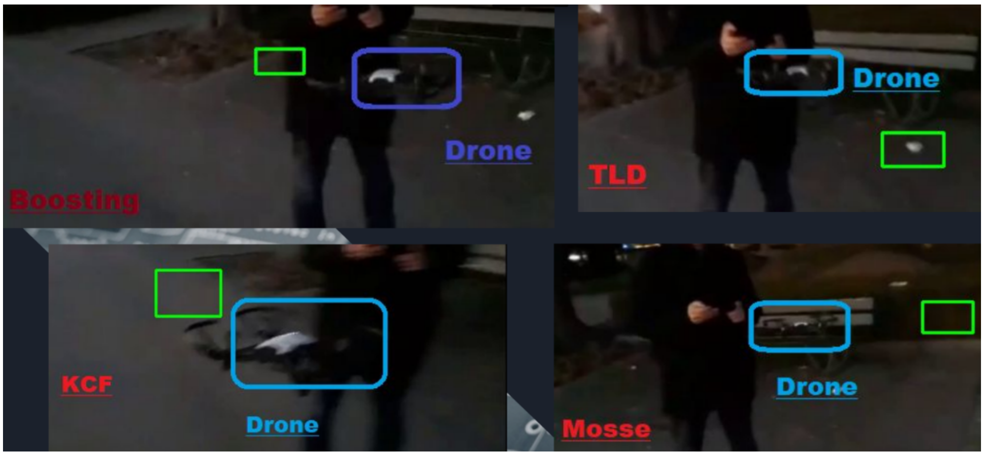

Figure 2.7: Limits of the existing algorithms

Figure 2.8: Limits of the existing algorithms

can't control its complexity, its precision neither its time of response we found ourselves ought to implement our own algorithm.

2.1.5.7 Algorithms criteria

In a way of comparing the simplicity, rapidity and precision of the best performing algorithms we came out with the following table of analysis:

Table 2.2: Algorithms criteria

After concluding the limits of the best performing algorithms from the eight existent, we

	Simplicity	Rapidity	Precision
MIL	Based on deep learning and need time to train the algorithm.	The best in performance and response time due to the deep learning algorithms.	
MedianFlow	The best real-time tracker	After an FPS test, MedianFlow seems to be the fastes in time of treating the frequent of video trams.	The advantage of this algorithm is that it keeps tracking the object no matter how bad quality of the background in the pictures.
CSRT	It can't assure the track in real-time.	The slowest algorithm of tracking.	Precise under any condition of test.

went out with a complexity of deploying and integrating it with an existing API or code. Also, we came out with the necessity of having a datasets of drones that till nowadays this does not exist and needs a training for the algorithm.All these issues lead to slow the time of response of our drone tracking in real-time for the fact that it depends on OpenCV library speed without having the hand to modify any of the existent algorithms.

2.1.5.8 Proposed Solution

To analyse our proposed solution we had to start with the few next steps starting by the image processing chain, then the video trams comparative treatment, and after applying the filters to detect our objects on the image by iterating all the video tram and applying our personalised algorithm to detect the flying objects without the need for datasets and with a total control on the precision and the speed of the algorithm even thought its complexity which will be explained in the next structures.

2.1.5.9 Image processing chain

To understand the behavior of our algorithm, we needed to pass by analysing all the image processing chain, as it follows: Our video is based on a sequence of pictures trams, each picture is composed from a defined number of pixels that could be represented as a matrix. To apply our image analysis, we need to launch a pre-treatment algorithms that could be resumed by implementing filters of the image pixels to improve or modify its resolution. Moving on to the segmentation that could be manifested in a personalised algorithm to apply the mathematical solution that aims to recognise the position of the object in a close

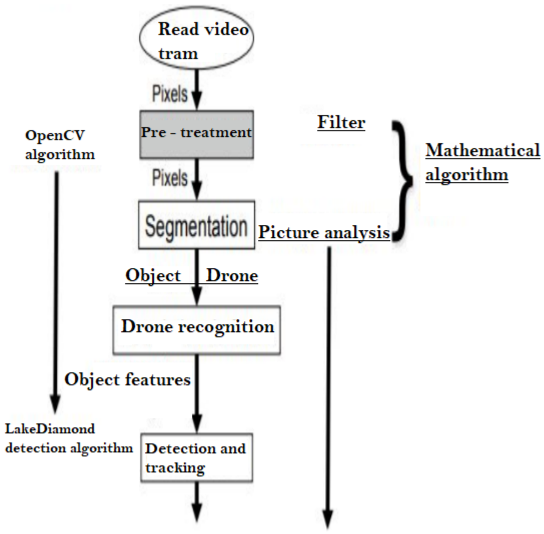

Figure 2.9: Image processing chain

environment with a white simple background.

2.1.5.10 System processing chain

Starting by the process of our system to detect the object:

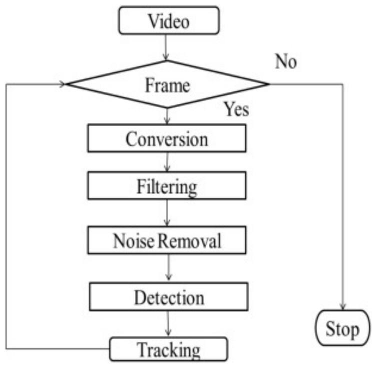

Figure 2.10: System processing chain

The first step will be reading the files using the Opencv library we move on to detect and draw the frame that will show the tracking and the detection of the flying object. The conversion of the colors tenses from R.G.B to the gray scale then the binary tenses so thaw we make the application of the mathematical algorithm easier. By applying the filter we'll be removing all the noises from our picture ending up only with the flying object on the picture and this is how we could detect our object by comparing the different values of our pixels then drawing the rectangle which will be tracking the objects throw the sequence of the multiple pictures in our video and by applying the subtraction operation on the pixels till we can follow the variation of the drone during the timeline of real-time tracking.

2.1.5.11 Object detection algorithm principle

Getting into more details about our algorithm presentation, we resume its principle as it shows:

The subtraction of the picture pixels, shows only the differences between the pixels values

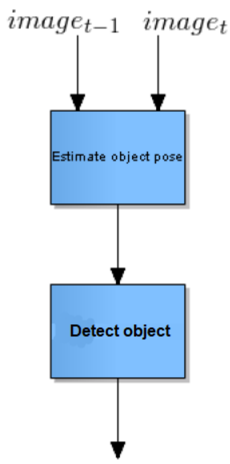

Figure 2.11: Object detection algorithm principle on picture

that have been changed and this is how it will only left the pixels of the flying object through the tram of the picture sequences.

2.1.5.12 Object positioning algorithm principle

The advantage of our system is that we could at the end detect the position of our object in a high value of precision that could be detected by centi-degrees as it's presented in our diagram:

After applying our mathematical theory to detect the object, we had to present its position in a rectangle through that we could manage to calculate the gravity center of our bounding box and find the right position of the drone in the picture. After that we had to convert the position into a radian angle to find the PAN and TILT position of the drone that we could send the command to the server implemented in the Jetson card to make our laser platform moves in real-time following the drone after converting it into centi-degrees.

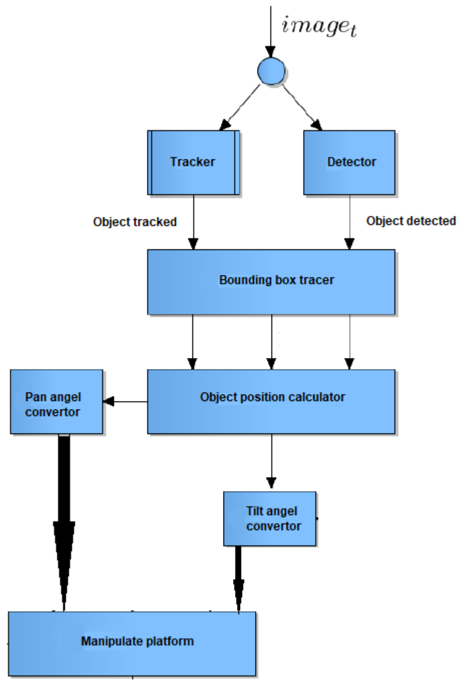

Figure 2.12: Object detection algorithm principle on picture

2.2 Internal specifications

Any project's requirements need to be well thought out , balanced and clearly understood by all involved , but perhaps of most importance is that they are not dropped or compromised halfway through the project.
For that reason, in this part we'd be criticising our mathematical solution as it follows:

2.2.1 Client mathematical solution

To explain our principle algorithm, we'll start out of an example that could be resumed in a few steps:

In the first step of our pixel ordering, we start by taking the lowest value of the matrix and stock it in a declared variable. The scale of our pixel values would be between: 0 and 255.

50	30	30
60	125	20
50	130	168

Figure 2.13: Mathematical solution : First step

30	10	10
40	108	0
30	110	148

Figure 2.14: Mathematical solution : Second step

For the second step, we subtract the values of all the pixels with the minimum previous number we've found in the first step. After that we take the highest value of all the pixels and stock it as the maximum variable.

45	15	45
60	163	0
45	189	255

Figure 2.15: Mathematical solution : Third step

In a third step, we divide our pixels values with the highest previous variable in the second step. Then we do multiply all the pixels numbers with the value 255 as it's the threshold number.

At the forth step, we subtract the last matrix we had with the first initiative and declared matrix before applying any changes. Here we start to see the position of our object by finding a negative numbers that will be neglected and a 0 values which represent the neutral color.

With the fifth step, we apply a mask to put the negative numbers at 0. Here we end up with only the pixels of the object.

After drawing a bounding rectangle on the object pixels, we could calculate its width

-5	-15	-5
0	37	20
-5	59	75

Figure 2.16: Mathematical solution : Forth step

0	0	0
0	37	20
0	59	75

Figure 2.17: Mathematical solution : Fifth step

and height which are the variables to calculate the gravity center of our object.

2.2.2 Server mathematical solution

For the calibration process of the platform, we came to explain the pan and tilt angles for the platform as it follows:

In this part, we had to deduct the angle where to move our platform as it's explained in this schema:

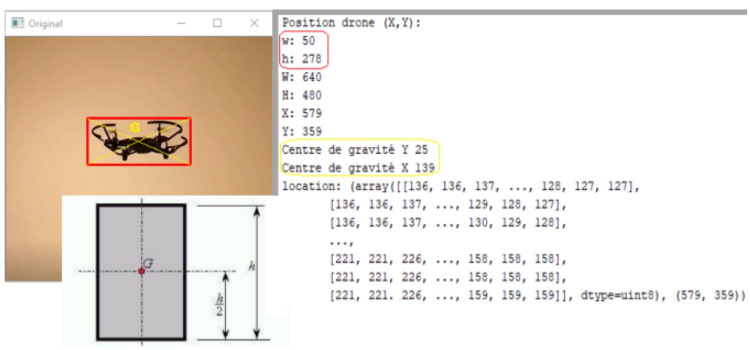

Figure 2.18: Mathematical solution : Drone position

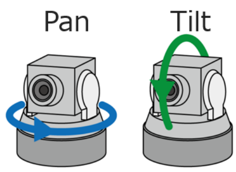

Figure 2.19: Mathematical solution : Pan and tilt angle

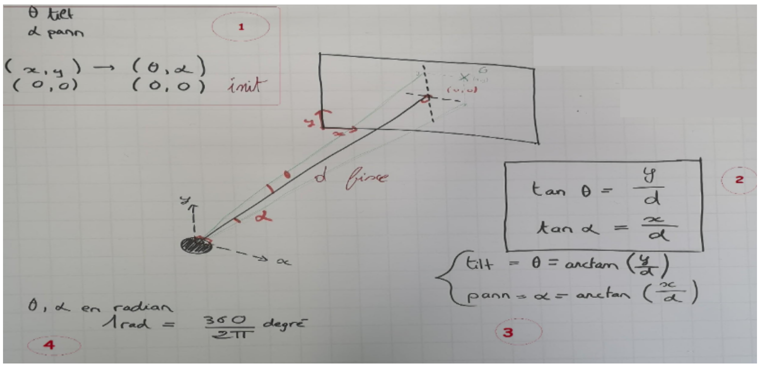

Figure 2.20: Mathematical solution : Drone angle

In the first part, we had to initialise two principle angle at 0. the we calculate the tangent angle for the both axis X and Y. Moving on to deducting the Pan, tilt angles using arctangent function, then we convert the angle value from radian to centi-degrees.

2.2.3 Functional requirement

In this section, we present the exigency this system has to cover. The definition of a functional requirement is Any requirement which specifies what the system should offer.
In other words , a functional requirements will describe a particular behavior of function of our system when certain conditions are met . For example: make advanced research, adding a certain precision in our algorithm.

Table 2.3: Basic functional requirements

As we'll be working on maintaining a monitoring sub-module between the client and the server, while generating datasets on the background process.

Module	Sub-module	Functionality Code	Functionality
Client	Drone tracking and detection	Python 0	Real-time video reading
		Python 1	Image processing
		Python 2	Implement the tracking algorithm
		Python 3	Detect drone position
	Server connection	Python 4	Connect with the jetson card
Server	Manipulating platform	C++ 0	Find the angle
		C++ 1	Convert the angle
		C++ 2	Send the cmd
		C++ 3	Update the angle
	Client connection	C++ 4	Connect on ssh protocol
		C++ 5	Recieve data variables
		C++ 6	Implement Drotek API

2.2.4 Non-Functional requirement

Any requirement which specifies how the system perform a certain function. In other words, a non-functional requirement will describe how our system should behave and what limits there are on its functionality.

Non-functional requirements generally specify the system's quality attributes or its characteristics, for example: "Updatd position data in the server should be updated while accessing within 2 seconds."

To resume,This application must satisfy several non-functional requirements such as:

The application may offer a security level to ensure the integrity from accidental or malicious damage by connecting on an ssh protocol.

Reliability: The application should, under normal conditions, perform the required functions successfully.

Usability: The user should not have to spend more than 10 second to load the video on the interface.

Availability: Users can edit or send information at any time.

2.2.4.1 Security

The application has to offer full security for the user to manage any information which means a secured accessibility and usability using the ssh protocol .

2.2.4.2 Performance

This one shouldn't come as a surprise. Quality software has to be fast in response or at least feel to be it. "It's not fast enough" is a battle we never want to get into. When we think about an application being performing, we must think about specifying the following:
Response times:
How long should my app take to load? What about screen refresh times or choreography?
Processing times:
How long is acceptable to perform key functions or export / import and read data?
Query and Reporting times:
This could be covered off with general reporting times, in case of providing an API we should probably consider acceptable query times too.
Capacity and Scalability:
How much do we need to cope with now, and how much do you think we'll need to cope with in the future. Unless you are truly blessed with an incredibly forgiving audience (or a great illustrator), you won't be able to get away with your own fail whale.
Availability:
Our application must be available, users can edit or send information at any time.
Hours of operation:
When does our application need to be available? If we need to do an update, an upgrade in term of versions or a system backup, can we take the system offline while we do it?
Locations of operation:
A few things to think about here: Geographic location, connection requirements and the restrictions of a local network prevail.
Reliability:
Our application has to be 100% reliable, clients should rely on its social networking services.
Usability:
Our application has to be easy to use, with standard looking interfaces and with clear and

beautiful visual style.

We end by the maintainability:

Our application must be quickly maintainable and flexiable with changes.

Conclusion

This chapter is a brief presentation of this report scope, it contains a small description of the key elements for our project.

3

Global analysis

Introduction

On the aim to present you an over view on the project,we went through a global analyze.Which will be presented in the next chapter.

3.1 Global architecture

The software design stage comes after the requirements analysis and specification, it is a process of problem solving and planning for a software solution and it is a key step in the software life cycle.

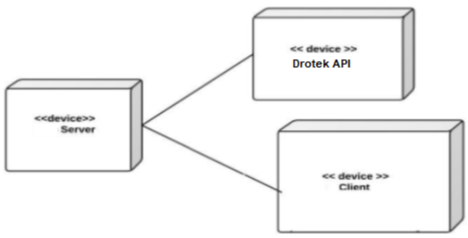

Figure 3.1: Global architecture

3.2 Use case

The Use cases gives a clear view about the sequence of functions of our system. We did not give all the use cases specifications as many of them are either obviously similar or easily comprehensive. Now we have to concentrate upon the design of our system using UML (Unified Modelling Language)

3.2.1 Actor identification

Our platform is used by different kind of actors,this is where we define them:

Table 3.1: System identification and description

Client	Detect and track the drone
Server	Manipulate platform

Use Case Name	Add a document
Participating Actors	Server
Entry Conditions	The server must be connected on an ssh protocol and listening for a client.
Flow Of Events	1- Recieve the angle position 2- Treat data 3- Manipulate platform
Alternative Flows	Error message: Missing mandatory information
Exit Conditions	Message of confirmation: The action was accepted and created.

Use Case Name	Consult a document
Participating Actors	Client
Entry Conditions	Client socket must be connected on an ssh protocol
Flow Of Events	1- Detect the drone 2- Track the drone 3- find position 4- Update position
Alternative Flows	Error message: SERVER CLOSED!
Exit Conditions	Confirmation message: Socket closed.

In this section, we analyze one of our use cases and give detailed description. After this part, we present our global use case diagram with the most important functionalists.

3.2.2 Use case diagram

In order to manage the commands transaction on the ssh protocol of communication the socket of the client and server must be connected.

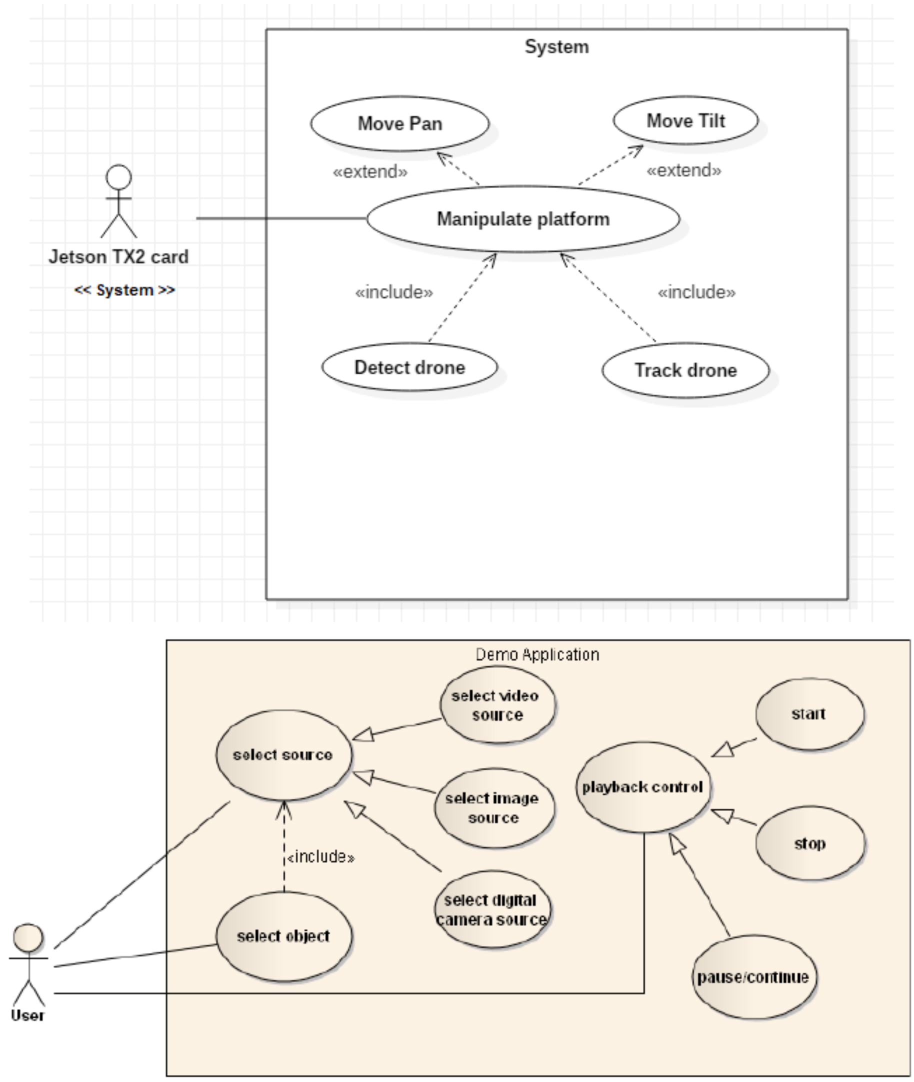

Figure 3.2: Use Case diagram

As it shows in this diagram, we have the user who will manage the video for the real time image processing treatment.After we select the source for reading the videos from the client side and we move on to the server side where we'll need to communicate the position of our object to the Jetson card where we had implemented the drotek API manipulate the platform.

3.3 Class diagram

We used the class diagram to describe the structure of the system by showing the system's classes, their attributes, methods, and the relationships among objects.

3.3.1 Global class diagram

Find the detailed global class diagram attached in annex 3, which will be resumed into the following diagram:

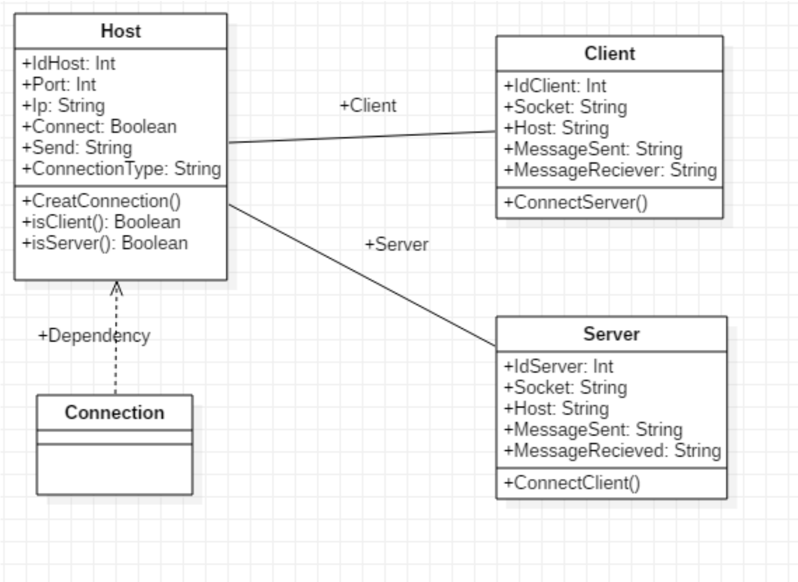

Figure 3.3: Global class diagram

As we work on a protocol, we needed to communicate between two different sides, a client and a server which could be detailed in the following diagrams:

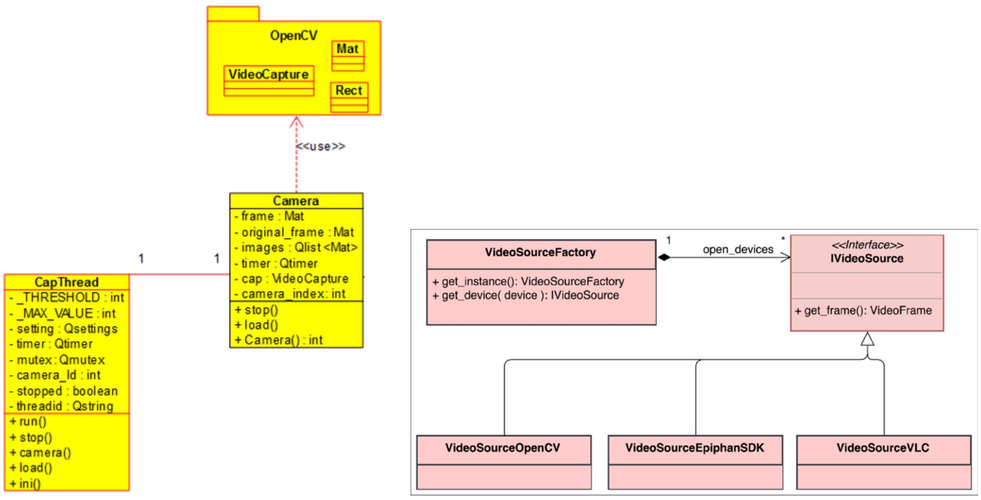

Figure 3.4: Client class diagram

These diagrams could detail the way our client communicate with the library of OpenCV to read and treat the processing of the files.

3.4 Functional architecture

UML modeling helped us to plan our program before the programming takes place. It's used as a standard, a widely understood and well known. This made it easy for a new programmer to step into our project and be productive from day one.Now after explaining the aspects of our application.

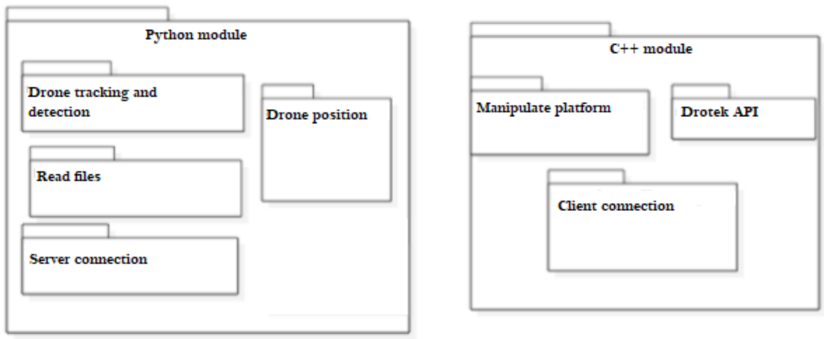

Figure 3.5: Global functional Architecture

3.4.1 Logical architecture

We define in this step the software logical architecture which is the 2-tiers architecture.

3.4.2 The 2-Tiers architecture

The 2-tiers architecture (distributed architecture, or multi-tier architecture) is a client-server architecture in which presentation, application processing, hardware and Drotek API management layers are separated.

3.4.2.1 Presentation layer

It puts out the results to the browser/client tiers and all other tiers in the network. (In simple terms it is a layer which users can directly access in a secured way, or an operating system that could remotely controlled) In our application, we'll be using Python and C++ for the socket presentation layer.

3.4.2.2 Application layer

It contains treatments representing the business rules and it controls the application's functionality by performing detailed processing. We'll be developing our services with a socket architecture to maintain a lifetime code by consuming our already developed algorithm.

3.4.2.3 Communication layer

This layer is responsible for mapping logic objects and data components. It's the only layer that is linked with our Drotek API using a socket on an ssh and TCP protocol.

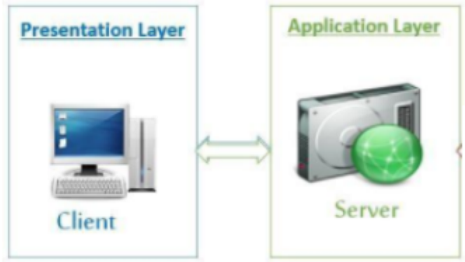

Figure 3.6: Global functional Architecture

3.4.3 Physical architecture

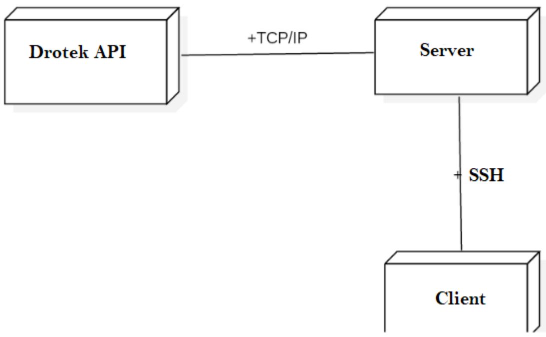

Figure 3.7: Deployment diagram Architecture

We had to think of an architecture that can be communicated data between these platforms. For this, we used the solution that is of use to communicate all the platforms with Drotek API which is presented on the server.

3.4.4 Road map

Our milestone of 2-TUP branches during the hole process:

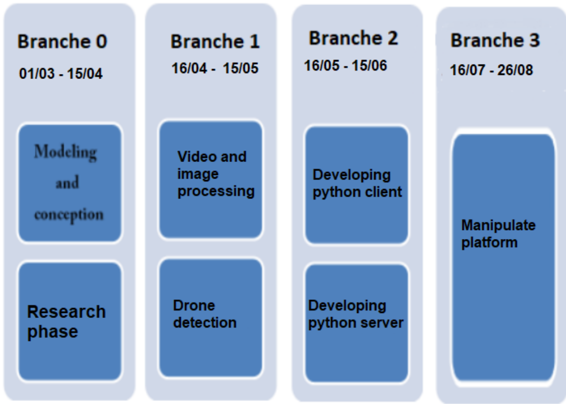

Figure 3.8: Road map

3.5 Conclusion

With this project we are going to develop a platform for I.R.M.C and its prime objective is not only to help us master the new technologies such as Python , C++ , Drotek API and ssh protocol consuming the services but more importantly to sensitize the people about a new lifetime advanced research engine technology.

4

Technical branch

Introduction

On the aim to implement an interactive and easy to use platform. We had to collaborate the development tools in order to make that happen. We needed to set an efficient project architecture that will help us build a trade application.

4.1 Software tools

In this part, we will move to the process of the project implementation by shedding lights on the various mastered tools, libraries and platform.

4.1.1 Client platform

4.1.1.1 Programming language Python

Python 3.7.4 is a set of coordinated specifications and practices that enable solutions for the development, deployment, and management of multi-tiers centralized applications on a server.
What we should know about it is: Powerful,fast; plays well with other standards; runs eon

any IDE; Easy to learn; is open source.
Python can be easy to pick up whether you're a first time programmer or you're experienced with other languages. Python is developed under an OSI-approved open source license, making it freely usable and distributively. The Python Package Index (PyPI) hosts thousands of third-party modules for Python. Both Python's standard library and the community-contributed modules allow for endless possibilities.
One of the most important packages we used with python for the image processing were the OpenCV 4.1.1, which will be detailed as a programming tool.

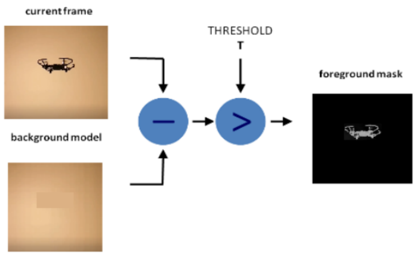

Figure 4.1: Imports Python 3.7.4

For the architecture, we adopted MVC 2 : A software architecture model(allows the use of multiple controllers) for both the client and the server side just like for the C++ as well.

Figure 4.2: Python 3.7.4

4.1.1.2 Programming language C++

C++ is a cross-platformed language that can be used to create sophisticated high-performance applications. It gives programmers a high level of control over system resources and memory as it's the most recommended for embedded systems. An object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs. As we have our API for manipulating the platform written in C++ we had to integrate our socket server to receive the cmd in C++ code.

Table 4.1: Implementation specifics

Specifics	Used technology	Version	Multiplatform
Programming language	C++	compiler version : g++-4.4.3	YES
Image processing library	OpenCV	2.1	YES

4.1.1.3 Pycharm

Cross platform development frameworks, PyCharm offers great framework-specific support for modern web development frameworks such as Django, Flask, Google App Engine, Pyramid, and web2py. PyCharm integrates with IPython Notebook, has an interactive Python console, and supports Anaconda as well as multiple scientific packages including matplotlib and NumPy. Run, debug, test, and deploy applications on remote hosts or virtual machines, with remote interpreters, an integrated ssh terminal.

Figure 4.3: Pycharm Version: 2018.1.6

4.1.2 Programming tools

4.1.2.1 OpenCV 2.1

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. All of the new developments and algorithms in OpenCV are now developed in the C++ interface but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

Figure 4.4: OpenCV library 2.1

4.1.2.2 Drotek API

A complete end-to-end solution with multiple options at every level. IP-friendly permissive BSD license. Neutral and transparent governance. The Dronecode platform contains everything needed for a complete UAV solution: flight-controller hardware, autopilot software, ground control station, and developer APIs for enhanced/advanced use cases. Using a platform ensures that everything needed for a complete drone solution is delivered in product that is a well integrated, well tested, easily modifiable and and consistently licensed.

Figure 4.5: See annex 4 Drotek

4.1.3 Application compiler

4.1.3.1 Microsoft sisual studio

Figure 4.6: Microsoft visual studio logo

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps.Its advantages are:
IntelliSense Performance Enhancements for C ++ File.
Local development with many common emulators.
Simplifying access to tests in Solution Explorer
Git management and repository creation in the IDE

4.1.4 Platform implementation

4.1.4.1 Socket

Our system is connected through a socket between a python client and a c++ server as it's showed in the component diagram: The port we're using is fixed at the 22, TCP/IP

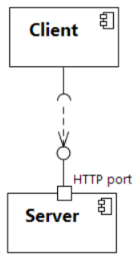

Figure 4.7: General socket component diagram

protocole . Our server address is the Jetson card IP: 192.168.1.240 . We can contribute to connect on an ssh protocol for data transfer from the client side using Putty:

Figure 4.8: Putty ssh client

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

4.1.5 Application design and analysis

4.1.5.1 StartUML

For the UML modeling open source software. Via this platform, it is possible to design our diagrams such as creating class diagram, use case,package, activity or sequences.

Figure 4.9: StartUML application designer

4.1.6 Collaborative development environment

4.1.6.1 GitLab

GitLab is a Web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

Figure 4.10: Collaborative development environment

4.2 Logical architecture

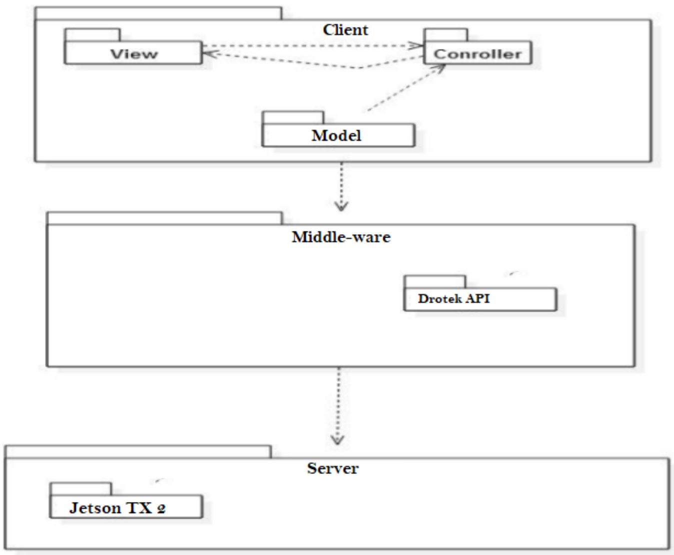

Figure 4.11: System package diagram

This diagram represents the Logical architecture of our application which is composed with 3 elements, our Client which communicates with the server and the middle ware that assure the standardisation protocol of communication between both peers.

4.3 Physical architecture

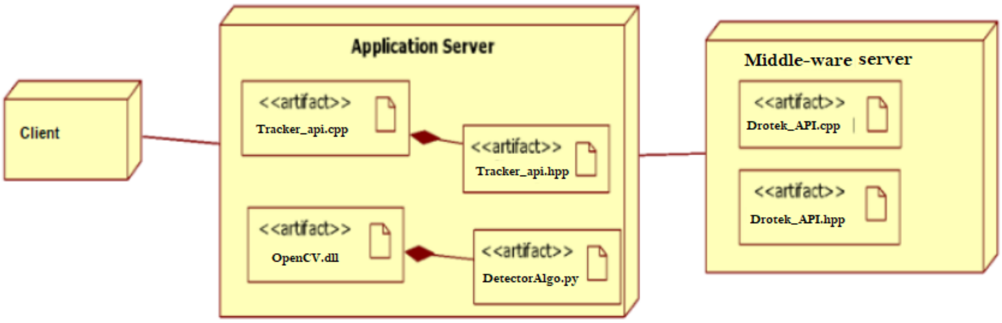

Figure 4.12: Global system deployment diagram

This diagram represents the physical architecture of our application composed with our c++ server implemented in the Jetson card and the middle-ware container of the platform API code.

4.4 Preliminary and detailed conception

4.4.1 General system package diagram

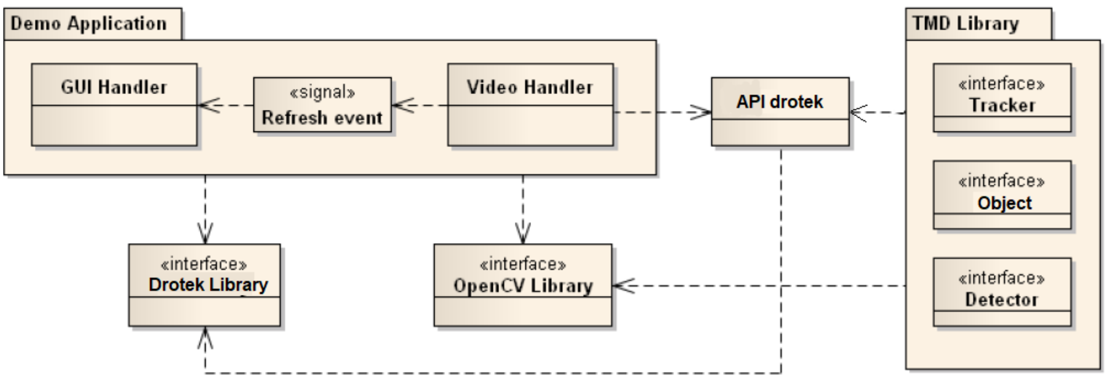

Figure 4.13: Client package diagram

This diagram shows a global view on our packages and implemented interfaces.

4.4.2 Detailed system package diagram

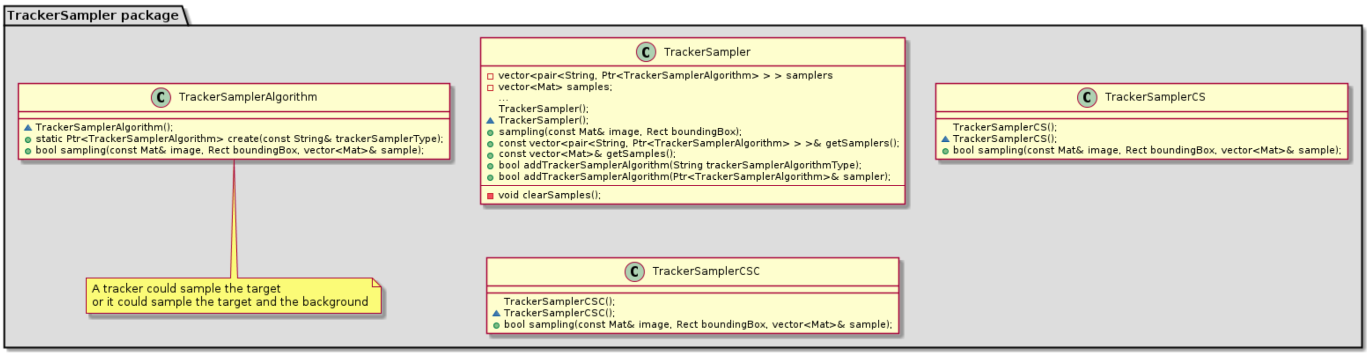

Figure 4.14: Server package diagram

Our diagram shows a detailed view on the tracker API part implemented in the server side.

4.4.3 General sequence diagram

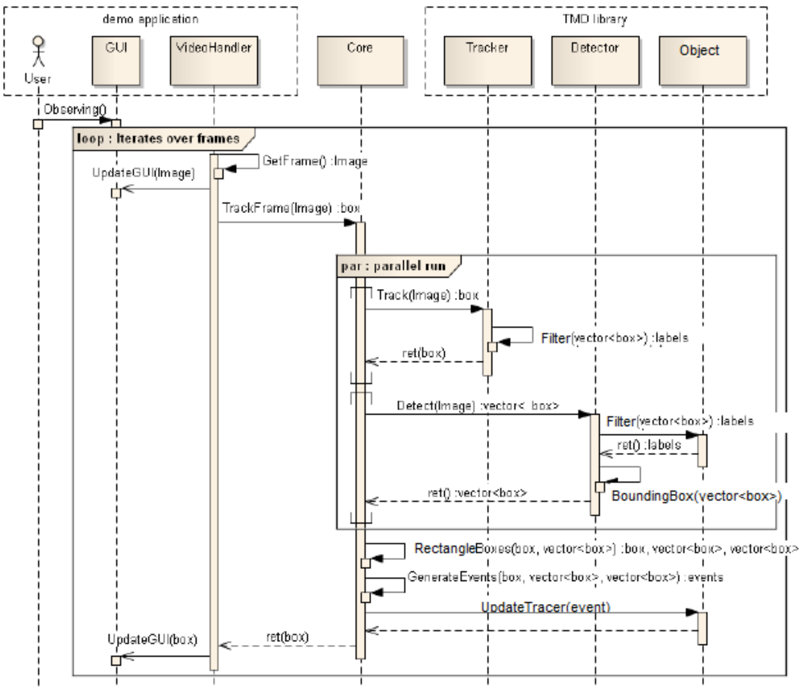

Figure 4.15: General sequence diagram

In this side, we sum-up the different iterations of our system. It details our algorithm in term of loops, functionalities and conditions.

4.4.4 Client side sequence diagram

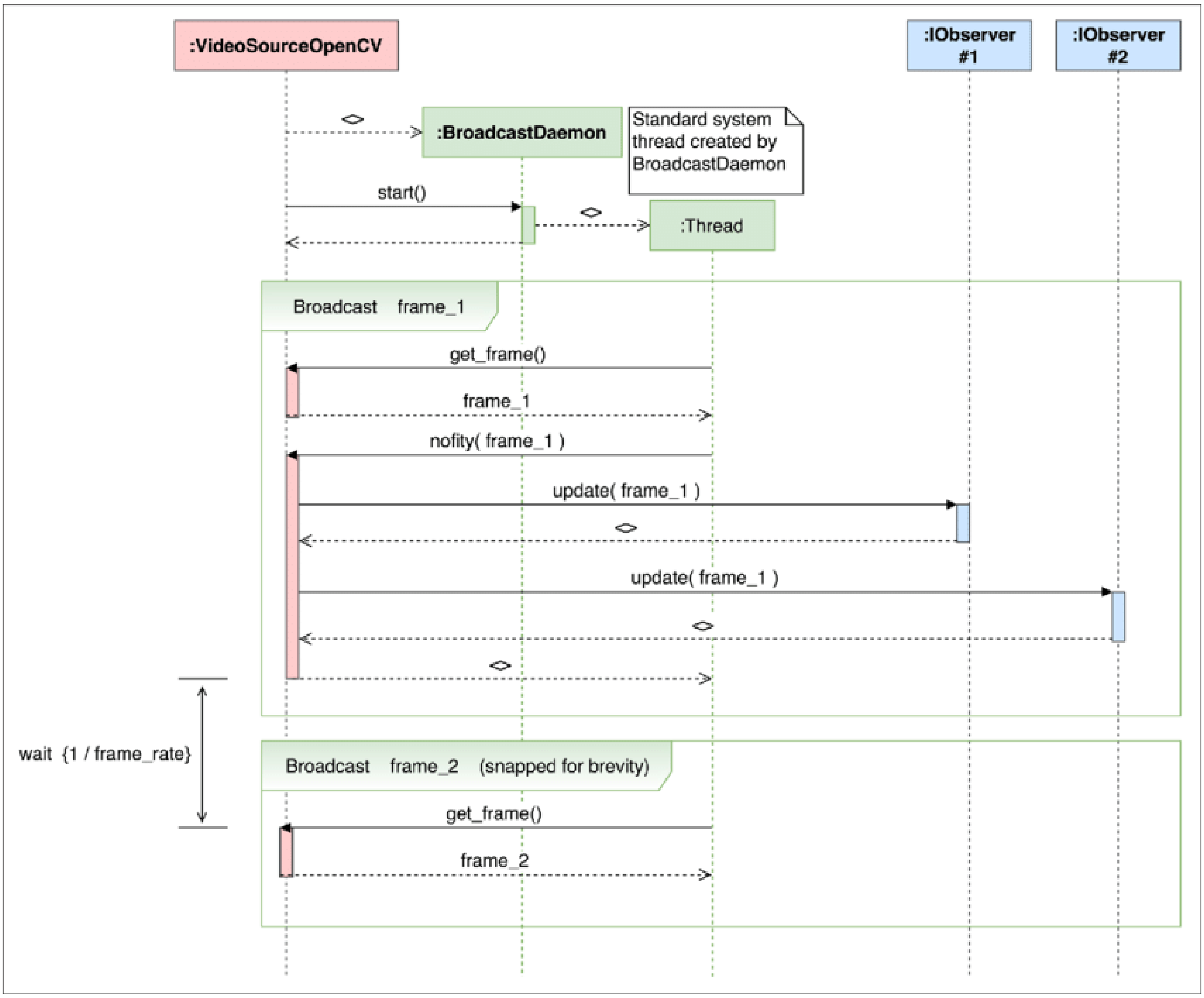

Figure 4.16: Detailed client side sequence diagram

This diagram details only the part of reading and treating videos in real-time broadcast demo using the OpenCV library for iterating different frames.

4.4.5 Design class diagram

This diagram shows the different states of the socket connections and the the different layers of our adopted architecture.

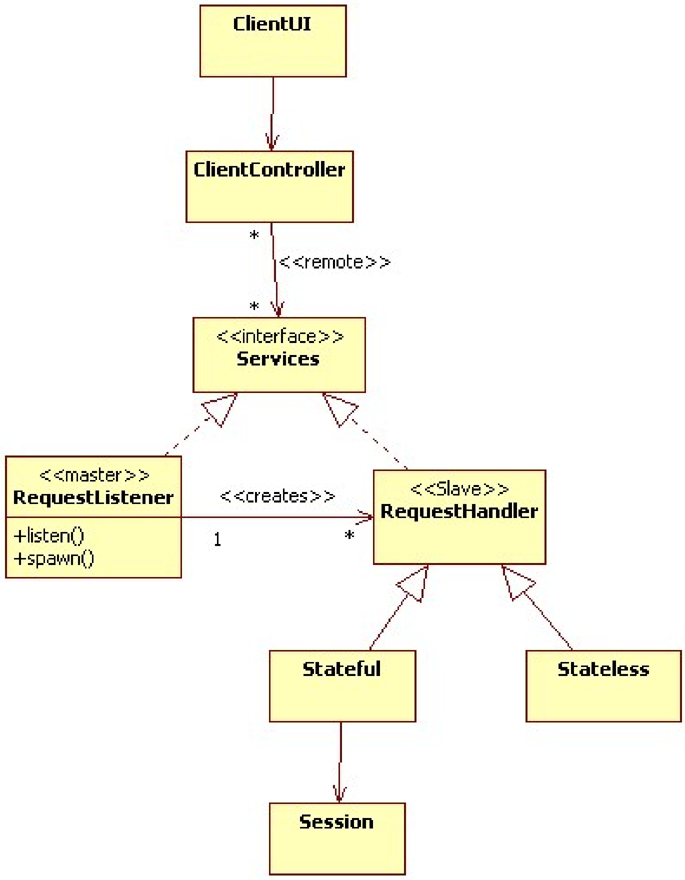

Figure 4.17: Design class diagram

4.4.6 Component diagram

This is the components diagrams which contains all the components that defines our system, starting from the general to its details. Our system bellow shows the core main component of image processing system and its relation with the other different parts of the library mentioned after.

As for the second schema we'll get into detailing the component of the image processing iterations and transaction.

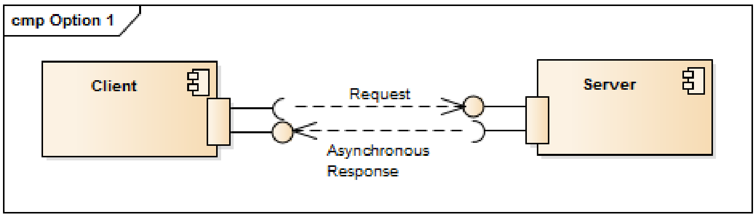

Figure 4.18: Global component diagram

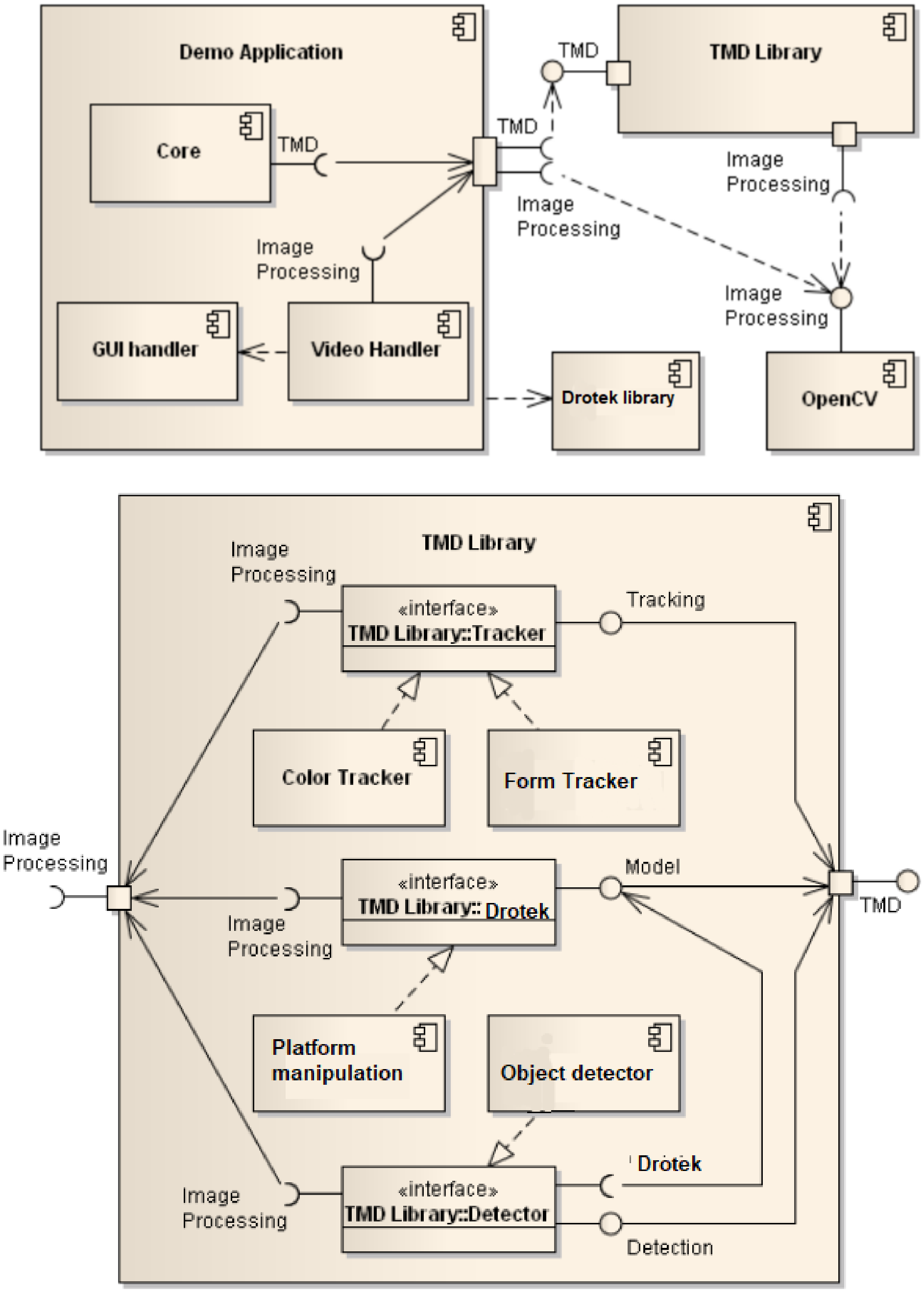

Figure 4.19: Detailed component diagram

4.4.6.1 Server component diagram

To show the components of the server side that manipulates the platform we thought about the following proposed conception:

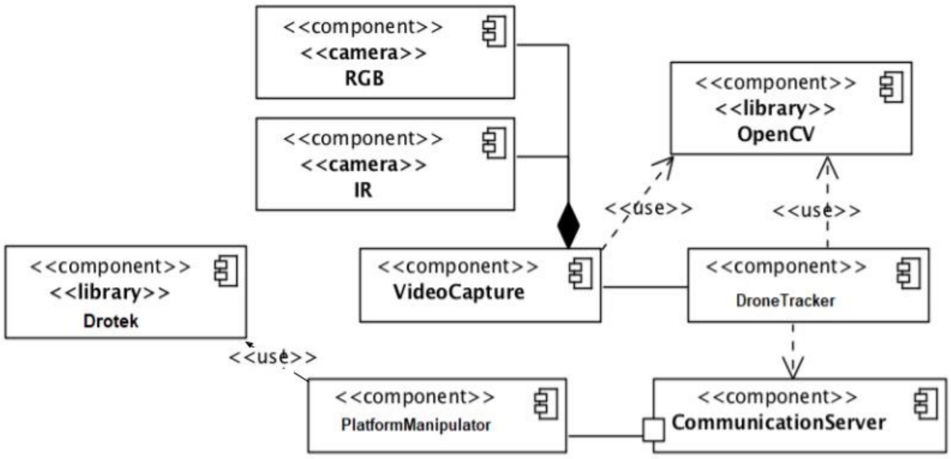

Figure 4.20: General component diagram view

4.4.7 Deployment diagram

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. Deployment diagrams are used for describing the hardware components, where software components are deployed.

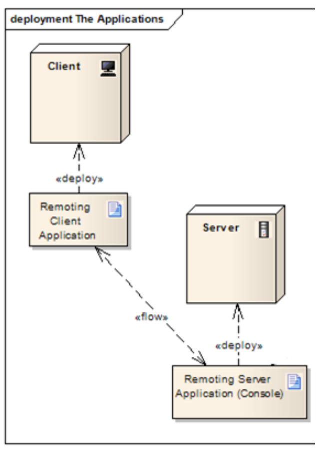

Figure 4.21: Deployment diagram

4.4.8 Bloc diagram

To put in lights the communication in the socket layer between the different activities that will be detailed later we used the bloc diagram that resumes the collaboration between the classes and the header files.

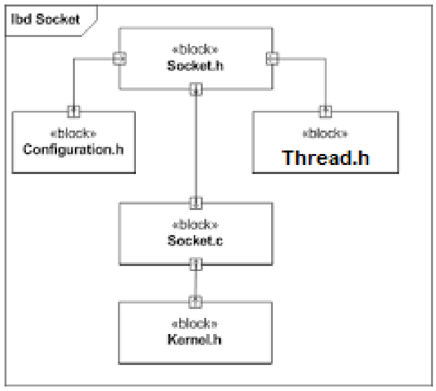

Figure 4.22: Bloc diagram

4.4.8.1 Detailed bloc diagram

This conception details the management of the connection queue manager and the CMD of Pan and Tilt management.

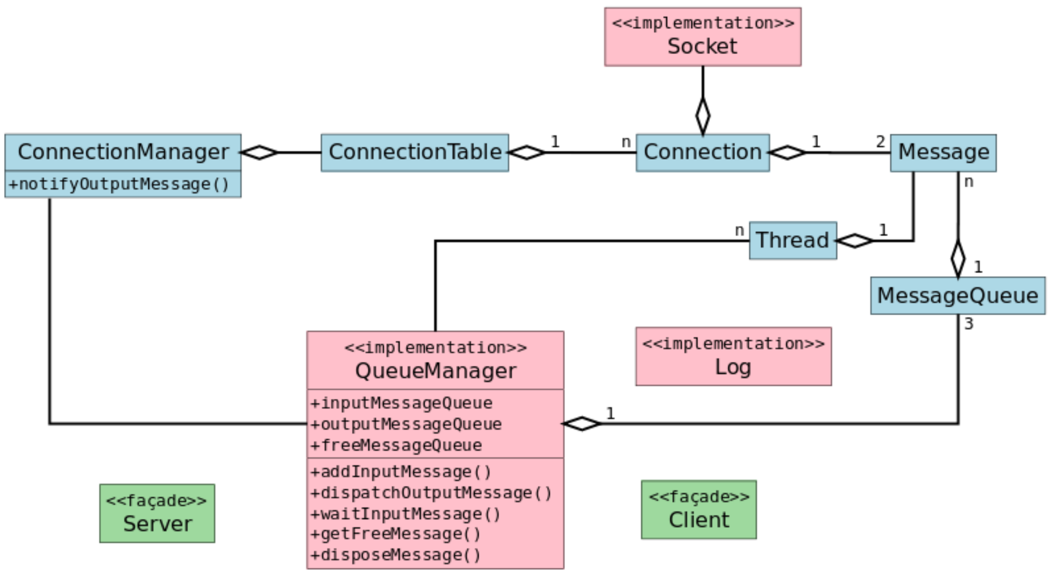

Figure 4.23: Detailed bloc diagram

4.4.9 Activity diagram

The basic purposes of activity diagrams is similar to the previous diagrams. It captures the dynamic behavior of the system.

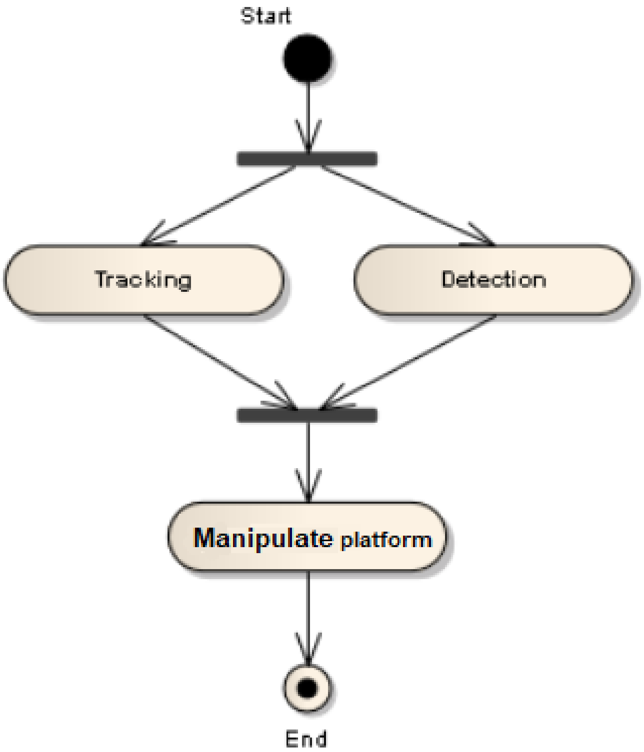

Figure 4.24: General activity diagram

.

4.4.10 Detailed activity diagram

The following activity diagram shows the message flow from one object to another, as it's used to show the messages flow from one activity to another. Activity Diagrams can be used to describe internal processing as well as action-object flow. This gives three benefits: a smooth transition from business processes to use cases, an abstract specification of complex object interactions and a succinct description of system functions affecting several objects.

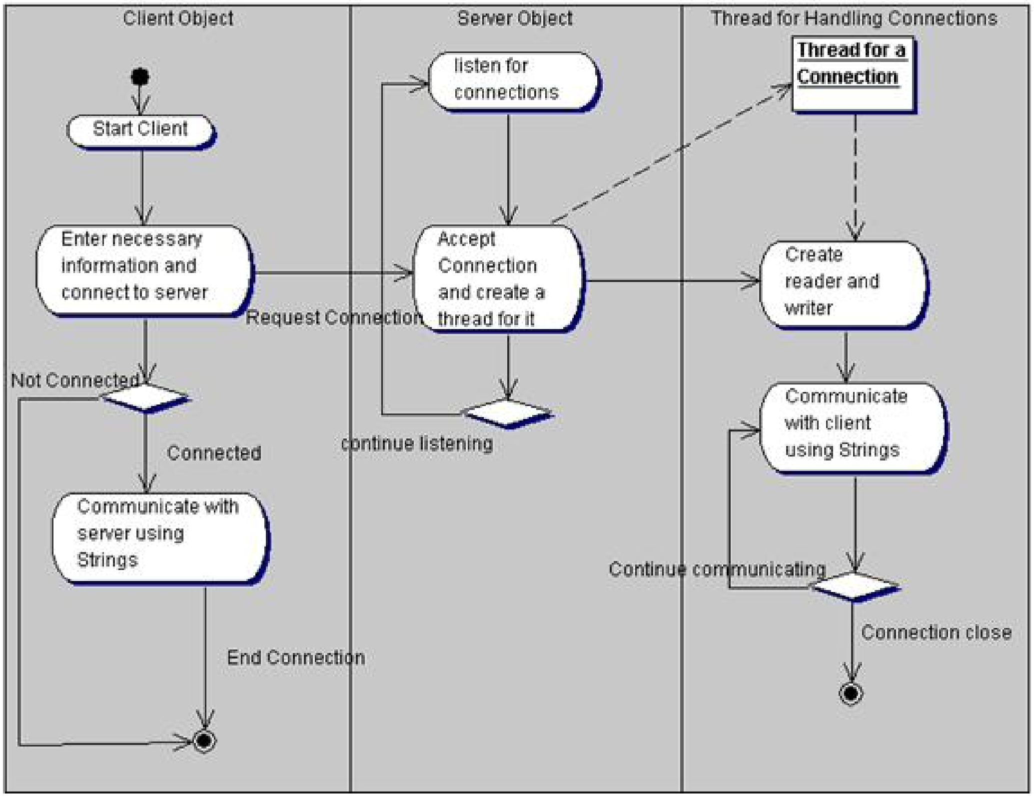

Figure 4.25: Detailed activity diagram

4.5 Implementation and test

4.5.1 Application interfaces

4.5.1.1 Image processing

We started by the basics and had to subtract the two different picture, then analyse where's the differences finding out the contour of the drone.
Our environment was close, in a white background without any noise.
We aimed to detect any kind of drone without requiring datasets or a data base full of drone's pictures that could never be a reference.

Another detection method have been applied is a detection by color to filter the noises of the drone colors that could appear:

The final result was as the following. We could detect the gravity center of the drone after drawing the rectangle: The next pictures shows the socket cmd transactions.

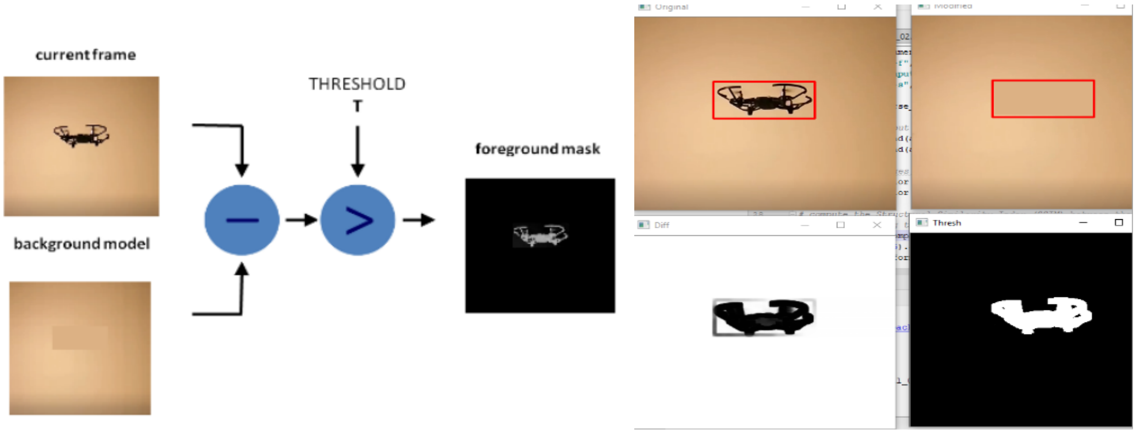

Figure 4.26: Frames subtraction

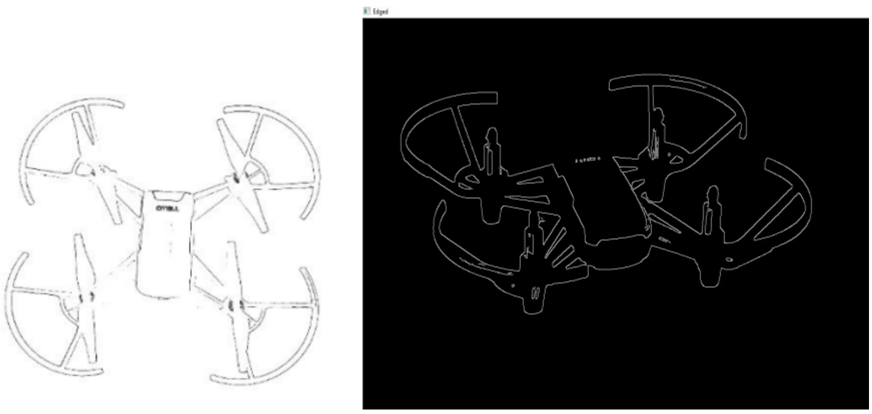

Figure 4.27: Contour detection

4.6 Implementation tips and integration

For the implementation, we choose python for the client side design phase and we started by deploying the algorithm principles on the same server to manage the integration of our work .

For the integration, we used Gitlab as a technology to work in group, which is the most widely used modern version control system on a the worldwide.

Git has the functionality, performance, security and flexibility that most teams and individual developers needed.

At the end we deployed the project on GIT ,so that everybody could start the development to ease the integration of the different project parts.

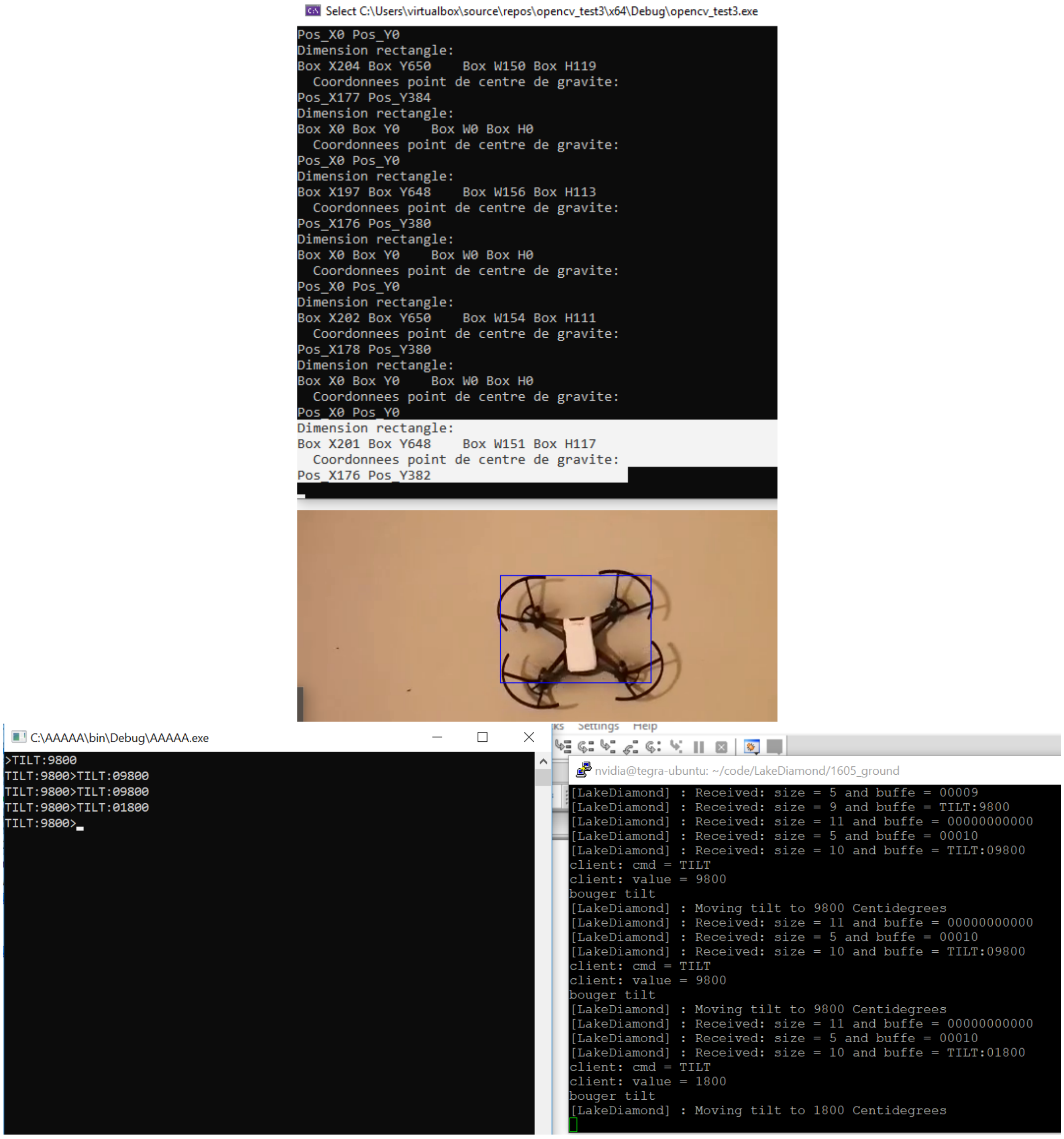

Figure 4.28: Socket communication

4.7 Application test

We used several types to check our work, Integration tests to validate the integration of the different modules between them and unit tests to validate the quality of the code and the performance.

For the test of the application, we didn't find any important constraints. We started by creating a mathematical solution then we deployed our algorithm and after that we had to apply the image processing filters . Now we can detect any kind of drones. Moreover, we made tasks with a deadline and affected to a responsible, this last will validate his tasks before the deadline.

The final step was the test of sending cmd between the client and the server and verify the precision of the platform manipulation.

4.8 Conclusion

After we had defined and explained our software development process. In this module, we designed the conceptual diagrams which helped us understand how our application is supposed to work. Knowing how the packages interacts with each other, we were able to develop the python part of our project management tools and its services. As a matter of fact, we are now able to interact with our Jetson card by adding, deleting, updating or displaying records through the interfaces. We are now ready to start working on training a dataset.

5

Conclusions and perspectives

After examined up-close the gears of a steadily growing project. After exploring and comparing the different localization techniques, I managed to implement a high accuracy visual detector for the drone. It is a real-time tracker as it runs around 12 fps. I did the preliminary work to implement a personalized algorithm and creates the sets to improving it with deep learning . Because drone datasets are scarce, if not inexistent, I had to create my own conditions in the algorithm behavior so that I can filter the drone contours from the noisy vectors. Even if the algorithm may be very advanced, if it is trained with a bad dataset, the accuracy will drop. A good training dataset has to have diversified samples that still represent well the object of interest. This enables the network to be more robust, to have a good generalization and to avoid over-fitting. The tracking system can be further improved, but I believe I helped LakeDiamond to acquire the basics of computer vision and offered them a state of the art in the field. I provided an advanced sketch of the drone tracking architecture that they can enhance to have more accuracy and faster processing time. Our framework was successfully implemented in C++ and encapsulated to a library. This re-implementation is not just a copy of opencv implementation, but rather an independent realization of detection method . By loosening from just rewriting OpenCV library code a new piece of knowledge about the detection algorithms functionality emerge and help with designing experiments that explore interesting parts. This independent realization carry downsides as well. Matlab implementation was polised for couple of years, but re-implementation with testing had to be done in four months. Despite of great effort the c++ implementation in general doesn't have balanced results as matlab implementation. In some testing sequence c++ implementation outperform matlab implementation and in other performer it's even worse. Simple demo that illustrate our algorithms functionality was implemented with user interface that have been described in the last chapter. Development run on UNIX platform, on the Jetson card TX2 therefore application currently operate only under this platform. As mentioned, porting application with OpenCV library to other platforms is not a difficult task, rather than time demanding to resolve all environment incompatibility. TMD library and demo application used external libraries for image processing (OpenCV) . Demo application is written in multi threaded fashion, by using cross-platform threads implementation. Threads are used for boosting detector speed by distributing patch evaluation to number of parallel computation. This acceleration allow to process multiple times more patches, depending on number of used threads.

Long-term tracking is deep issue and there is a lot of space for improvement. Method for object shape estimation is one of possible future improvement. By exploiting object shape and his connected components, we can refine detector to target specific area of object or connected components. Moreover, we can used shape estimation to selectively track only pose invariable parts of object or track additional components that are connected to object, therefore make object tracking more robust to shape changes. Further work should explore possibilities of online model reconstruction based on obtained knowledge about object appearance. Introduce fitness function for ferns evaluation on level of features and method for malfunction features replacing with reasonable time consuming.

6
Bibliography

Petar. «Night-time Detection of UAVs using Thermal Infrared Camera.» International Conference on Air Transport. 2017.

Langley, Richard B. «RTK GPS.» University of New Brunswick: www.gps-world.com, s.d.

Sun, Qian. «Pursuing Precise Vehicle Movement Trajectory in Urban Residential Area Using Multi-GNSS RTK Tracking.» World Conference on Transport Research. 2016.

Malik, Asad Abdul. «Object Detection and Tracking using Background Subtraction and Connected Component Labeling.» International Journal of Computer Applications, 2013.

Iyer, Anusha. «How to track objects with stationary background?» http://layer0.authentise.com. 2016.

Jianxin. Convolutional neural networks. LAMDA Group. 2018.

Girshick, Ross. «Rich feature hierarchies for accurate object detection and semantic segmentation.» Computer Science: Computer Vision and Pattern Recognition (cs.CV). 2014.

Girchick, Ross. «Fast R-CNN.» Computer Science: Computer Vision and Pattern Recognition (cs.CV). 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.» Computer Science: Computer Vision and Pattern Recognition (cs.CV). 2016.

Triggs, Navneet Dalal and Bill. Histograms of Oriented Gradients for Human Detection. Rhone-Alps: INRIA, 2005.

Rozantsev, Artem. Detection of Aircrafts on a Collision. Lausanne: School of Computer and Communication, Sciences Swiss Federal Institute of Technology (EPFL), 2013.

Reid, Victor Adrian Prisacariu and Ian. fastHOG - a real-time GPU implementation of HOG. University of Oxford, Department of Engineering Science, Parks Road, Oxford, UK, 2009.

NVIDIA. CUDA C/C++ Basics, Supercomputing 2011 Tutorial. 2011.

Redmon, Joseph. «You Only Look Once: Unified, Real-Time Object Detection.» 2015.

Farhadi, Joseph Redmon and Ali. YOLOv3: An Incremental Improvement. University of Washington, 2018.

CASTELLANO, Brandon. :movie$_{c}amera : APython/OpenCV-basedscenedetectionprogram, usin$
$Breakthrough/PySceneDetect[enligne].Python.2019.[Consultle4avril2019].Disponiblel'adresse :$
$https : //github.com/Breakthrough/PySceneDetect$

(PDF) Evaluation of Visual Tracking Algorithms for Embedded Devices. ResearchGate [en ligne]. [Consulté le 5 mars 2019]. Disponible à l'adresse : https://www.researchgate.net/publication /317803149$_{E}valuation_{o}f_{V}isual_{T}racking_{A}lgorithms_{f}or_{E}mbedded_{D}evces$

YUMPU.COM. Filtrage gaussien K =. yumpu.com [en ligne]. [Consulté le 11 mars 2019]. Disponible à l'adresse : https://www.yumpu.com/fr/document/read/17503289/introduction-au-traitementdimages- michele-gouiffes/35 Filtrage gaussien K = Filtrage médian : exemples Bruit gaussien Médian 3x3 Bruit impulsionnel

LEBEDA, Karel. 2D and 3D Tracking and Modelling. . pp. 222.

7.1.2. Mini-tutoriel de traitement d'images — Python scientifique - ENS Paris. [en ligne]. [Consulté le 7 mars 2019]. Disponible à l'adresse : http://python-prepa.github.io/ateliers/image$_{t}uto.htm$

11$_{t}$*racking.pdf* [*enligne*].[*Consultle*5*mars*2019].*Disponiblel'adresse* : *https* : //*files.ifi.uzh.ch/rpg*

841-Choix-Eclairage-Microscan.pdf [en ligne]. [Consulté le 26 mars 2019]. Disponible à l'adresse : http://www.mesures.com/pdf/old/841-Choix-Eclairage-Microscan.pdf

ROSEBROCK, Adrian. A gentle guide to deep learning object detection. PyImageSearch [en ligne]. 14 mai 2018. [Consulté le 1 mars 2019]. Disponible à l'adresse : https://www.pyimagesearch.co /2018/05/14/a-gentle-guide-to-deep-learning-object-detection/ Deep learning object detection can be challenging to understand, but in this tutorial I'll break it down and give you a strong deep learning object detection foundation.

Aaronia$_{A}$*ARTOS*$_{D}$*rone*$_{D}$*etection*$_{S}$*ystem.pdf* [*enligne*].[*Consultle*26*mars*2019].*Disponiblel'adresse* *https* : //*downloads.aaronia.com/datasheets/solutions/drone*$_{d}$*etection/Aaronia*$_{A}$*ARTOS*$_{D}$*rone*$_{D}$*etec*

About us – AARONIA AG Germany • AARTOS Drone Detection. [en ligne]. [Consulté le 26 mars 2019]. Disponible à l'adresse : https://drone-detection-system.com/about-us/

BOLON, Philippe, CHASSERY, Jean-Marc, COCQUEREZ, Jean-Pierre, DEMIGNY, Didier, GRAFFIGNE, Christine, MONTANVERT, Annick, PHILIPP, Sylvie, ZÉBOUDJ, Rachid, ZERUBIA, Josiane et MAÎTRE, Henri. Analyse d'images : Filtrage et segmentation [en ligne]. MASSON, 1995. [Consulté le 13 mars 2019]. Enseignement de la physique. Disponible à l'adresse : https://hal.archivesouvertes. fr/hal-00706168

LOUVAT, Benoit. Analyse de séquences d'images à cadence vidéo pour l'asservissement d'une caméra embarquée sur un drone. . pp. 76.

LELORE, Thibault. Application aux documents anciens. . pp. 46.

SWINNEN, Gérard. Apprendre à programmer avec Python 3. Developpez.com [en ligne]. [Consulté le 4 avril 2019]. Disponible à l'adresse : http://python.developpez.com/cours/apprendrepytho

Apprendre les bases du traitement d'image - Chapitre 1 : introduction, un cours de Patrick Bonnin. Developpez.com [en ligne]. [Consulté le 12 mars 2019]. Disponible à l'adresse : http://applications.developpez.com/actu/98157/Apprendre-les-bases-du-traitement-d-image-moins- Chapitre-1-introduction-un-cours-de-Patrick-Bonnin/

Bachelor, Master and Doctoral School Projects – CVLAB. [en ligne]. [Consulté le 6 mars 2019]. Disponible à l'adresse : https://cvlab.epfl.ch/projects/

Barycentres. [en ligne]. [Consulté le 4 avril 2019]. Disponible à l'adresse : http://gilles.dubois10.free.f

CONNECTEDREAMS.COM. Beginner's Guide to Computer Vision. Connectedreams Blog [en ligne]. 18 novembre 2016. [Consulté le 4 mars 2019]. Disponible à l'adresse : https://medium.com /readers-writers-digest/beginners-guide-to-computer-vision-23606224b720

Compare video frames - OpenCV QA Forum. [en ligne]. [Consulté le 4 avril 2019]. Disponible à l'adresse : http://answers.opencv.org/question/194295/compare-video-frames/

CVLAB – Computer Vision Laboratory. [en ligne]. [Consulté le 26 mars 2019]. Disponible à l'adresse : https://cvlab.epfl.ch/

CVLAB – Computer Vision Laboratory. [en ligne]. [Consulté le 4 mars 2019]. Disponible à l'adresse : https://cvlab.epfl.ch/

Course: Computer Vision. [en ligne]. [Consulté le 5 mars 2019]. Disponible à l'adresse : https://moodle.epfl.ch/course/view.php?id=472