PEOPLES'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Blida 01 University

Institute of Aeronautics and Space Studies

Manuscript

In partial fulfillment of the requirements for

The Degree of Master in Aeronautics
Specialty: Avionics

Design and implementation of a forest fire detection and localization system using UAVs

---

A Thesis Submitted by :
Charef Wided
Bouzidi Farah

Under the Supervision of :
Dr. Choutri Kheireddine
Prof. Lagha Mohand

members of the jury :
Jury president: Dr. Dilmi Smain
Examiner: Dr. Bousaad Azmedroub

Blida 2022-2023

Acknowledgement

First and foremost, I thank God for everything that I have. As our parents have told us, one who does not express gratitude to the creatures, does not express it to the Creator,

I would like to thank everyone who has helped me accomplish this achievement. I will give my sincere gratitude to both of my advisors. Dr. Choutri kheireddine and Prof. Lagha Mohand who has provided me with these valuable chances and resources allowing me to pursue my master degree, and to successfully handle various challenges and difficulties in different critical phases of my thesis. Without their consistent support, encouragement, I cannot achieve such satisfactory research progress. Their great passion and diligence to research works, positive attitude to both work and life, organizational skills, and keen insight on solving challenging issues have been my relentless source of motivation and inspiration. It is really my great privilege being able to work under their insightful guidance.

In addition, it is my great honor to have some of my old teachers joining my examining committee and providing useful feedback and brilliant comments during my comprehensive exam and research proposal. I really appreciate their strong support and inspired advice. Dr. Lamri , thank you for all of your kindness and positivity.

I am also very grateful to my current colleague and my best friend Bouzidi Farah who have provided great help to our thesis, from the discussion of challenges in research and paper writing, through experimental platform development and algorithm validation, to the problems that I have met in my daily life. It is my great pleasure to conduct research works with her, share opinions with her, and learn from her.

Then, special thank goes to my mom whose prays solves a lot of my problems and to my dad for his mental support and encouragement words. Thank you , my dear, for your financial support throughout my studies. Thank you everyone in my family , my big brother walid , and the youngest: Aymen, Achref, Akram , Douaa and my lovely Grandmother for cheering me always and being by my side. Without their love, consistent support and understanding, I cannot make it.

Finally, I would like to thank a group member of graduated fiends who kept there for me , especially Sofia , to help me in different kind of need i showed .Thank you ITC members , and last but not least , Thank you CLOUD , that offers what is more than food.

I dedicate this work to my deceased Grandfather , God rest his soul.

Acknowledgement

First and formost, i must acknowledge my limitless thanks to ALLAH, the ever-magnificent, the Ever-Thankul, for His help and bless by giving me the opportunity, courage, patience and enough energy to carry out and complete the entire of this thesis.

My deepest gratitude should go to my supervisors, Dr.Choutri Kheireddine. This work would not have been possible without his expert. Who had provided me with his scientific guidance and corrections, suggestions and advice, and particularly for his patience,hard work and consistent support. Also i would like to thank profondly Prof.Lagha Mohand for his encouragement and help in doing a lot of Research and i came to know about so many new things I am really thankful to them.

My wormest thank goes to my best friend Wided, who have been closer to family than a friend to me.I felt blessed to have her in my life those 5 years was fully enjoyable because of her. I am grateful for what she offered me, her friendship, empathy, and great sense of humor.

My special thank goes to my Brother Salah who never give up on me, protected and took care of me, my mother Ratiba for the love, support and continues trust in my pursuits, morally and financially making it possible to go after my dreams, my father Ismail for his mental support and encouragement and my uncle Ali who's always by my side.

Finally i am grateful to have friends who helped me during all this year, especially Sofia

for being with me in the good and bad. Nesrine for being an amazing older sister and my special thanks goes to Issam for always pushing me forward, never give up in giving me support, information's and provided me with his help and cheers.

I dedicate this work to my brother SALAH who came from far away just to assist my graduation day.

Abstract

Owing to their rapid response capabilities and maneuverability, extended range and improved personnel safety, drones equipped with vision systems have great potential for forest fire monitoring and detection. Over the past decade, the demand for drone-based forest fire detection systems has increased, as forest fires have become a growing concern with climate change, as indicated by several environmental studies, and a reality in some parts of the world. Despite this, existing drone- based forest fire detection systems still present many practical problems for their use in operational conditions. In particular, successful detection of forest fires remains difficult, given the very complicated and unstructured forest environments, the movement of UAV-mounted cameras . These negative effects can seriously cause false alarms or faulty detection. In order to perform this mission, meet the corresponding performance criteria and overcome these increasing challenges, it is essential to investigate ways to increase the probability of successful detection and improve the adaptation capabilities to various circumstances in order to improve the accuracy of the forest fire detection system. Based on the above requirements, this master thesis focuses on the development of reliable and accurate forest fire detection algorithms applicable to drones. These algorithms provide a number of contributions, which include: (1) a learning-based forest fire detection approach is developed by considering the color characteristic of the fire; (2) a forest fire localization scheme is designed by combining both stereo vision and perspective projection; and (3) a design and control of a quadcopter using the PIXHAWK autopilot.

Keywords: UAV, Deep learning, Stereo vision, Artificial intelligence, Pixhawk, localization, Forest fire detection, Camera calibration.

ملخص

نظرًا لقدراتها على الاستجابة السريعة والقدرة على المناورة وتحسين سلامة الأفراد ، تتمتع الطائرات بدون طيار المجهزة بأنظمة الرؤية بإمكانيات كبيرة لرصد حرائق الغابات والكشف عنها. على مدى العقد الماضي ، ازداد الطلب على أنظمة الكشف عن حرائق الغابات القائمة على الطائرات بدون طيار ، حيث أصبحت حرائق الغابات مصدر قلق متزايد مع تغير المناخ كما أشارت العديد من الدراسات البيئية وحقيقة في بعض أجزاء العالم. على الرغم من ذلك ، لا تزال أنظمة الكشف عن حرائق الغابات القائمة على الطائرات بدون طيار تشكل العديد من المشكلات العملية لاستخدامها في ظروف التشغيل. على وجه الخصوص ، لا يزال الاكتشاف الناجح لحرائق الغابات يمثل تحديًا ، نظرًا لبيئات الغابات شديدة التعقيد وغير المهيكلة ، وحركة الكاميرات المركبة بدون طيار ، وأوجه التشابه في خصائص اللهب. يمكن أن تسبب هذه الآثار السلبية بشكل خطير إنذارات كاذبة أو اكتشافات خاطئة لإنجاز هذه المهمة ، لتلبية معايير الأداء المقابلة والتغلب على هذه التحديات المتزايدة ، من الضروري دراسة الوسائل لزيادة احتمالية الكشف الناجح وتحسين قدرات التكيف مع الظروف المختلفة

من أجل تحسين موثوقية ودقة نظام الكشف عن الحرائق. بناءً على المتطلبات المذكورة أعلاه ، تركز هذه الاطروحة على تطوير خوارزميات موثوقة ودقيقة للكشف عن حرائق الغابات تنطبق على الطائرات بدون طيار. توفر هذه الخوارزميات عددًا من المساهمات ، والتي تشمل: (١) تطوير نهج قائم على التعلم للكشف عن حرائق الغابات في باللون المميز للنار (٢) تم تصميم مخطط موقع حرائق الغابات من خلال الجمع بين رؤية ستيريو وإسقاط منظور ؛ و (٣) تصميم والتحكم في درون باستخدام الطيار الآلي.

كلمات مفتاحية

الطائرة بدون طيار، بيانات الحرائق، التعلم العميق، كاميرات بالرؤية المزدوجة، الذكاء الاصطناعي، الاحداثيات

Résume

En raison de leurs capacités de réponse rapide et de leur maniabilité, de leur portée étendue et de leur amélioration sécurité du personnel, les drones équipés de systèmes de vision ont un grand potentiel pour la surveillance et la détection des incendies de forêt. Au cours de la dernière décennie, la demande de systèmes de détection des incendies de forêt basés sur des drones a augmenté, car les incendies de forêt sont devenus une préoccupation croissante avec le changement climatique, comme l'indiquent plusieurs études environnementales, et une réalité dans certaines parties du monde. Malgré cela, les drones existants Les systèmes de détection des incendies de forêt à base posent encore de nombreux problèmes pratiques pour leur utilisation en conditions opérationnelles. En particulier, la détection réussie des incendies de forêt reste difficile, compte tenu de la très les environnements forestiers complexes et non structurés, le mouvement des caméras montées sur UAV et les similitudes dans les caractéristiques des flammes. Ces effets négatifs peuvent sérieusement provoquer de fausses alarmes ou des détections erronées. Afin d'accomplir cette mission, de répondre aux critères de performance correspondants et de surmonter ces défis croissants, il est essentiel d'étudier les moyens d'augmenter la probabilité de détection réussie et d'améliorer les capacités d'adaptation aux diverses circonstances afin d'améliorer la fiabilité et la précision de la forêt. système de détection d'incendie. Sur la base des exigences ci-dessus, cette thèse de master se concentre sur le développement d'algorithmes fiables et précis de détection des incendies de forêt applicables aux drones. Ces algorithmes fournissent un certain nombre de contributions, qui comprennent: (1) une approche de détection des incendies de forêt basée sur l'apprentissage est développée en couleur caractéristique du feu; (2) un schéma de localisation des incendies de forêt est conçu en combinant les deux vision stéréo et projection en perspective; et (3) une conception et un contrôle d'un quadcoptère utilisant le Pilote automatique PIXHAWK.

Mots clés: UAV, Apprentissage profond, Stéréo vision, l'intelligence artificielle, Pixhawk, localisation, Détection des incendies de forêt.

Contents

List of Figures

List of Tables

List of abbreviations

AP Average Accuracy

CNN Convolutional Neural Network

COCO Common Object in Context

Faster R-CNN Faster Region-Based Convolutional Neural Network

FN False Negative

FP False Positive

FPV First Person View

GNC Guidance , Naviation and control

GPS Global Positionning System

HSV Hue Saturation Value

IMU Inertial Measurement Units

IOR Intersection Over Union

MB Mediterranean Basin

PID Proportional Integral Derivative

PIL Processor In the Loop

PWM Pulse Width Modulation

PX4 PIXHAWK 4

R-CNN Region-Based Convolutional Neural Network

RGB RED,GREEN and BLUE

RPN Region Proposal Network

SSD Single-Shot Detector

TP True Positive

UAV Unmmaned Arial Vehicle

uORB u (Micro) Object Request Broker'

VM Virtual Machine

YOLO You Only Look Once

General introduction

Forests are the best gift of nature provided to mankind. It provides us with different essential services and fulfills our needs. It provides space for living to many of the living organisms as an example , about 1,492,000 ha of Algeria is forested(31) . in Algerian Kabylie region 13 000 km^2 (42). Apart from this, we are availing a lot of benefits from the forests.
Forests play numerous vital roles in nature. moderate climate, purify water and air, store carbon , In addition, forest products industry offers a vast number of jobs and contributes to a country's economic wealth.

Unfortunately, every year Wild forest fires occur frequently all over the world. millions of hectares of forests are destroyed by fires. These fires can have natural causes or can be human made forest fires usually have the characteristics of high risk and strong destructive potential, and pose a great harm to social and economic development, environmental protection and ecosystems.
Different from other fires, forest fires present specific damage modes due to their environment. In the open environment and with sufficient oxygen, fires are more likely to occur and spread in forests, causing serious safety risks. This danger is also a real threat for the people living inside or adjacent to the forest area. Each year thousands of people lose their homes due to wildfires, and hundreds of people die in these accidents; additionally tens of thousands of domestic animals perish.

Forest fires have become a severe we take Algeria as an example, Algeria's forests cover a vast area of land. It is the main fire hot spot of the Mediterranean Basin(MB). Fire is widespread in this country due to the presence of flammable fuels such as shrub-lands and forests, a climate favorable to ignition and propagation (10).

Unluckily, in 2021 . Fires raged in north and north-east of Algeria overnight on August .Started up in the Kabylian region and elsewhere. Multiple fires have burned Mediterranean trees, destroying olive trees . Many distant villages had very limited water. Some villagers fled, while others tried to hold back the flames themselves, using buckets, branches and rudimentary tools, due to the unavailability of fire fighting aircraft. This disaster is considered as one of the most catastrophic (1) .

A total of 610 km of electricity network and no less than 710 stations were destroyed by the fires that ravaged several municipalities in Tizi-Ouzou, according to a report released during an inspection visit of the CEO of the Sonelgaz group, Chaher Boulkhras. Some 5,193 hectares of fruit trees and 19,178 farm animals burned in the fires, according to a report from the Local Direction of Agricultural Services (DSA) and a total of 1,705 homes burned were appraised by engineers from the Technical Construction Control body (CTC)(1) .

According to a report made public in May 2022 by the Algerian Directorate General of Forests (DGF) (1), the total area of vegetation cover affected by fires during the summer of 2021, amounts to more than 100,000 hectares through 1,631 fire outbreaks recorded in 21 wilayas. A total of 260,135 hectares of forests (26% of the total area), 21,040 hectares of bushes (21.5%), 16,415 hectares of scrub (16.5%), 16,160 hectares of fruit trees (36%) and 352 hectares of Esparto (0.5%) were ravaged by the fires. The table above represent

Wilaya	Hectares
Tizi Ouzou	43,398
Bejaia	13,174
Khenchela	9,837
Guelma	5,927
El Tarf	5,090
Annaba	5,024

Table 0.1: Affected area by hectares

the affected areas 0.1

Fighting forest fires is thereby seen as one of the most important issues in the natural resources protection and preservation. In particular, because the fast convection propagation and long combustion period of forest fires, early detection of forest fires is considered to be a prominent way of minimizing the destruction that fires may cause.

Traditionally, the approaches to detect forest fires are: human patrol, smoke and thermal detectors, ground-based equipment, manned aircraft and satellite imagery. Each of these methods has its own drawback (2).

For example, smoke and thermal sensors require proximity to the fire and cannot provide information on the size or location of the fire. Ground-based equipment may have limited surveillance ranges. Human patrol is not practical in large and remote forests. Satellite images are not vivid enough to detect early stage fires and they lack the ability of continuously monitoring forests because they have less flexibility in their path planning. Manned aircraft operations are expensive and require skilled pilots. Moreover, this can potentially threaten the crews' lives because of hazardous environments and operators fatigue (44) . Over the last 20 years, At least 80 pilots have died during firefighting operations only in the USA.

With the new developments in UAV technology, cheaper commercial UAVs are available for numerous research projects. UAVs can access high-risk zones, provide over-the-hill view, perform night time mission with no risk of human lives.

The deployment of UAVs offers huge benefits:

- Cover wide areas, with some kinds of weather;
- Work at day time, night, with long duration;
- Easily recoverable and relatively cost-effective compared to other methods;
- In the case of electric UAV, is also a benefit to the environment;
- Carry large and different payloads for different missions even within one flight benefited from the space and weight saving comparing with manned vehicles since there is no need for pilot related life-guard equipment and devices;
- Be able to cover larger and specific target area efficiently.

In recent years, a huge amount of researches have been carried out in the field of forest fire monitoring and detection by UAVs. In forest fire detection operations, UAVs can play many roles. The initial usage of UAVs was to send them to forests and record a video and by watching the video later, an operator can define if a fire has happened.

Contributions

The main contributions of this thesis reside in the analysis of the problem of wildfire monitoring using UAVs , the design and implementation of the principal components of wildfire detection and localization system :

- Study the state of the art as an overview of the past and current trends of wildfire remote sensing from an autonomy perspective with a hint to future works.
- Propose and implement a simple training algorithm of wildfire detection , with most known object detection algorithms comparaison to produce estimations of the accuracy and precision.
- Formalize the problem of wildfire observation with UAVs and develop a localization algorithm that exploits stereo vision system and camera calibration .
- Design and control of a PIXHAWK based quadcopter and allow testing it in real time.

The work presented in this thesis has been carried out within the scope of a personal final year project in collaboration with researchers of the University of Saad Dahleb Blida 1 , Insitute of Aeronautics and space studies . In particular, the design and implementation of the interface between the wildfire monitoring architecture and the UAV control system has been developed with the Aircraft laboratory of the institute of Aeronautics and space studies , Blida, Algeria. previous research papers have been used to achieve this work , mentioning :

- Linear Commands Comparison in a Real Time Simulation of a Quadrotors Unmanned Aerial Vehicle , K. Choutri, M. Lagha, L. Dala, M. Lipatov,International Conference on Control, Automation and Diagnosis (ICCAD), March 2018.
- UAV Aerial Image-Based Forest Fire Detection Using Deep Learning ,institute of aeronautics and space studies , university of blida 1 , Akila Keddous , 2020.
- Quadrotors UAVs Swarming Control Under Leader-Followers Formation , K Choutri, M Lagha, L Dala, M Lipatov , 22nd International Conference on System Theory, Control and Computing (ICSTCC), November 2018.
- Quadrotors attitude estimation and control with Arduino implementation,Khettal Fairouz , Doumi Nouha Wissem ,institute of aeronautics and space studies , university of blida 1 , 2020.

Problem statement

Advances in machine learning and deep learning are enabling the development of new methods to analyses big data including image classification and segmentation , complex scenarios and make predictions. Today's environmental problems, such as forest fires, are areas of great interest. However, existing approaches still present various practical problems for their operational use. The use of drone-based systems and a practical deep learning architecture for forest fire detection and localization circumvents these problems and provides solutions based only on image data.
Although existing research demonstrates the possibility and potential benefits of using drones to detect forest fires, the development of such systems, including hardware, software and related application strategies, is still minimal in the limited amount of previous

research. Further research is needed on all aspects of their use, including appropriate system platforms, remote sensing sensors and different algorithms for GNC, as well as remote sensing techniques. In addition, the combination of UAV and remote sensing techniques is also a particular challenge. This thesis attempted to study an appropriate and dynamic fire detection system in order to achieve its implementation on real Pixhawk based quadcopter to create awareness in the need to prevent forest fires.

Outline

The remainder of the thesis is structured as follows:

In Chapter 1 , it will be given a brief overview of the thesis subject, Reviews and enumeration of the scientific contributions.

In Chapter 2, a novel method of UAV-based forest fire detection using deep learning networks is addressed . In order to improve the accuracy of fire detection, different algorithms are adopted and different network are chosen. Network structures will be covered. Furthermore, techniques regarding training a neural network such as training a CNN using a pre-trained models are evaluated. The construction of the model and how it is trained are introduced, followed by an explanation of the evaluation process. The training process is further explained by showing the structure of the data-set, the pre-processing steps and the data augmentation performed on the data-set. Finally, a comparative study is made using several models based on some important points.

In Chapter 3, the architecture of the stereo system used for depth extraction after fire detection is included, based on the Triangulation method. Then, the relative position of the fire is calculated based on the different Camera calibration parameters .The camera calibration process is discussed due to its importance in providing an accurate position and several techniques are provided to help increase the accuracy of the position.

In Chapter 4, the use of pixhawk flight-controller for navigation is established in the design and modeling of a quadcopter.The process of the operation is detailed followed by test results of the behavior of the UAV in different phases. Finally, the conclusions of this thesis are drawn and the design of the complete system is presented as a future work, with some ideas on alternative versions and applications of the model.

Chapter 1

State of the art

1.1 Introduction

The field of aerial robotics is huge and interdisciplinary. Drones are reusable, unmanned flying objects that are controlled remotely. The term "unmanned aerial vehicle" (UAV) refers to a mechanical, electronic, and computer system that is programmed or controlled to carry out tasks that are either dangerous or difficult to be done directly by human beings (7).
In this chapter, we will present a brief definition of a drone, followed by a description of UAV technologies, fire detection using UAVs, a review of the existing literature on the use of UAVs in fire forest detection and finally forest fire detection and localization methods.

1.2 Background

1.2.1 UAV technologies

The history of UAVs is one of the cyclical developments that are frequently centered on military conflicts, but its design has been imagined since the ancient era, with a text dating back to 425 B.C. describes a mechanical bird built by Archytas the Tarantine that can fly 200 meters (41). Leonardo Da Vinci developed various aircraft in the 15th century that preserve some of the ideas of today's helicopters, despite the fact that they could not fly at the time. Several engineers struggled to fly their own aircraft during the ensuing centuries, with little to no success due to undeveloped engine, aerodynamic, and control technologies. On this day in history, the First World War, flying robots known as unmanned aerial vehicles (UAVs) make their debut. Drones have primarily been employed by the military for combat training. The "Kettering Bug" drone was created by the United States during World War I after ten years. Some "Sperry Messengers" were turned into flying bombs within ten years, making them the first true unmanned aircraft. British manufacturers created 40 unmanned aircraft between 1930 and 1940 to use as a target practice for anti-aircraft defenses. Their appellation, "Queen Bees," served as the basis for the term "drone."

Numerous aircraft designs appeared in the early half of the twentieth century, but it was not until its last decades that UAVs took their position in aviation. Because of advances in computational power and simplification of electronic devices, as well as improved and smaller communication equipment and sensors, their use for reconnaissance and intelligence operations has become extremely important.

The move to civil UAVs began at the turn of the century. Everyone had access to exact positioning everywhere in the planet thanks to GPS. Miniaturization of electronics, less power-hungry Central Processing Units, and lighter Inertial Measurement Units are all possibilities. UAVs were able to continue their route to autonomy because to enhanced battery storage density. It has varying degrees of autonomy, such as remote control and full autonomy, and can carry military payloads depending on the mission (11). The capacities required in any mission are affected by size and weight. These vehicles are distinguished by sensors and equipment, such as a camera, a thermal sensor, and so on, which are used to collect data in the environment of a targeted mission. Furthermore, they are supplied with GPS to identify the location information that reveals the mission's path (32).

1.2.2 Fire detection using UAVs

The essential parts of a general UAV-based forest fire surveillance and fighting system, which performs the functions of monitoring (finding a potential fire), detection, diagnosis (determining the fire location and extent and tracking its evolution), and prognosis (predicting the future evolution of the fire based on real-time wind and firefighting conditions), can be shown in figure 1.1. This is based on a review of the existing literature and research. A single UAV or a group of UAVs (each with a different sort of sensor) working with a centralized ground station do these tasks.
The goals are to employ UAVs to track fires, predict their evolution, and offer real-time data to human firefighters, as well as (or) to conduct firefighting with UAVs. There are various meanings of the terms used for monitoring and detecting forest fires throughout the literature, therefore they are not yet well defined. Following established customs in the more general field of condition monitoring, fault detection, and diagnosis, definitions for forest fire monitoring, detection, diagnosis, and prognosis are offered in this review to prevent misunderstanding (45).
Monitoring for forest fires is defined as keeping an eye out for potential fires before they start, whereas fire detection refers to finding a fire that has already started. Smaller flames are simpler to contain and put out, thus detection must happen as soon as possible.
The goal of fire diagnostic is to learn as much as possible about the fire, such as its location and extent. The goal of fire prognosis is to observe and anticipate the course of a fire in real time utilizing data from onboard remote monitoring sensors mounted on UAVs. UAV-based forest fire monitoring, detection, diagnosis, and prognosis systems typically employ the following components to achieve these objectives:

- Various frames and sensors, including global positioning system (GPS) receivers, inertial measurement units (IMUs) , and cameras, all of which aid in firefighting;
- Specific algorithms and strategies for fire monitoring, detection, diagnosis, and prognosis;
- GNC systems for both single and multiple UAVs;
- Cooperative localization, deployment, and control systems for UAV fleets to optimally cover fire areas such systems are based on the real-time information provided by the onboard visual (for daytime) and infrared (for both nighttime and daytime) monitoring sensors and their associated image and (or) signal processing algorithms);
- A dedicated ground station that includes equipment for communication, ground computation, visualization of fire detection, tracking, and prediction with automatic fire warning or alarm, as well as all equipment necessary for the safe and efficient operation of UAVs.

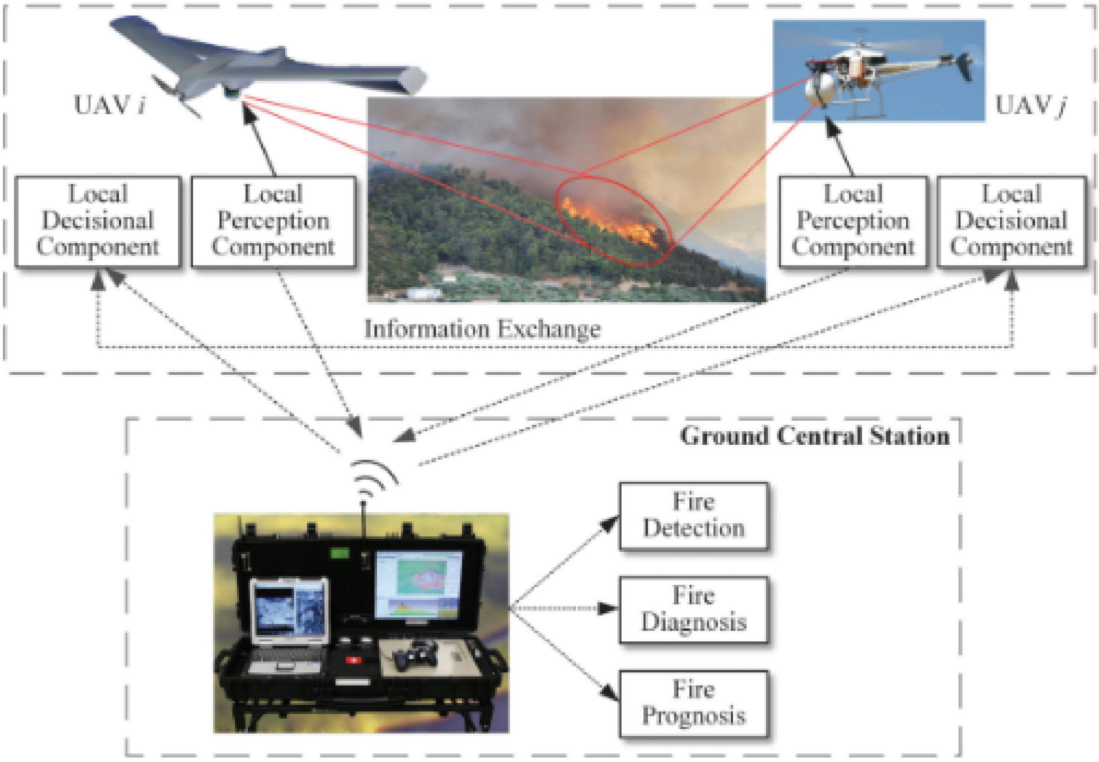


Figure 1.1: Fire detection system general Architecture

1.3 Literature reviews

1.3.1 UAV Based Forest Fire Monitoring and Detection Systems

Preliminary work on unmanned aircraft technologies for remote sensing of forest fires began in the early 2000s. This period is characterised by the use of high-altitude, high-endurance unmanned aerial vehicles (HALE UAVs) operated by research agencies as a complement to existing satellite-based monitoring systems. HALE UAVs have great capabilities: They can fly for hours at high altitudes and carry very heavy payloads, but they are expensive systems and do not provide more accurate data than satellites. Smaller drones fly much closer to the ground and are potentially capable of detecting more detailed information about forest fires.

The algorithm is then evaluated against a basic strategy of circling around the current fire outline. A mixed reality experiment, with a real drone and a simulated fire, was performed to test the proposed algorithm.

As noted in the article (40) the expansion of drone applications in the fire domain is primarily in the area of remote sensing of forest fires. Aerial monitoring of forest fires is costly and very risky, especially when it comes to uncontrolled fires.

An autonomous forest fire monitoring system is described by (35), whose objective is to track a set of hot spots as quickly as possible. Given a realistic simulation of the expected evolution of a forest fire. The authors introduce an algorithm that guides the drone towards these hot-spots. The algorithm is then evaluated against a basic strategy of circling around the current fire outline. A mixed reality experiment, with a real drone and a simulated fire, was performed to test the proposed algorithm.

Other tasks beyond remote sensing can be performed by drones, such as aerial prescribed fire lighting discussed by (5), the latter task remaining to be developed in operational contexts, as it requires transporting huge amounts of water and fire retardant.

In the work of (15), a great framework was created , based on mixed learning techniques composed of yoloV4 deep network and LiDAR technology , which was cost effective and practical sense the control of the developed UAV based fire detection system was able to fly over the burned region and deliver accurate information.

(Praveena et al.) proposed to develop a real-time forest fire monitoring system employing a UAV and remote sensing technique. The drone is provided with sensors, a mini processor and camera. Data from different sensors and therefore the images taken by the camera are processed on-board.

The most common mission in remote sensing of forest fires is fire mapping, which produces a map of an area highlighting the location of fires at a given time from Geo-referenced aerial images. In addition, fire maps can be processed to determine the current fire perimeter and provide an estimate of its position in unobserved areas. When mapping is carried out on an ongoing basis, for example to track the perimeter of a fire and provide regular updates of the fire map, this is called monitoring. Drones are able to generate rich and accurate fire data using high-resolution cameras, which can be used to characterise fire geometry. Remote 3D reconstruction of a forest fire can provide a wealth of information to firefighters, who can safely assess the severity of the fire at a given location.

Authors	Mission type	Onboard sensors(camera)	Information processing	UAV class	fielded
R.C. Skeele et al.2016	Front tracking	IR	On-board	quadcopter	Yes
D. Twidwell,2016	Detection and mapping	visual	Off-board	hexacopter	No
E. Beachly et al,2018	Mapping and tracking	visual	Off-board	UAS	Yes
R. N. Haksar et al,2018	Detection and suppression	Not mentioned	On-board	quadcopter	No
Kasyap et al,2022	early detection	visual	On-board	quadcopter	No
Praveena et all,2022	detection	visual	On-board	quadcopter	Yes

Table 1.1: review on UAV based forest monitoring and detection system

1.3.2 Vision Based Automatic Forest Fire Detection Techniques

The advantages of vision-based techniques, including intuitive, informative and reliable real-time data capture, wide detection range and convenient verification and recording capabilities, have made them a major research topic in the field of forest fire monitoring and detection (43).

Over the past decade, a series of research studies have been conducted using vision-based UAV systems for forest fire monitoring and detection in quasi-operational field trials, although actual firefighting trials remain rare. In addition, there are a number of other studies of other offline platforms and videos used to monitor and detect fires.

Although these methodologies were not originally designed for UAV application and do not take into account the problems associated with UAVs, e.g. flight-induced image vibrations and cooperative control of multiple UAVs, they nonetheless offer potentially transferable insights into UAV-based wild land firefighting applications due to their common problems in fire detection using vision-based technologies. In order to reduce the cost of equipment and personnel and to save experimentation time, the effectiveness of various fire detection approaches is normally tested and verified in advance on the basis of forest fire videos. Only then are the specific practical problems (image vibrations, etc.) affecting the on-board images taken into account and solved. For this reason, it is quite possible that these detection methodologies are also applicable to UAVs carrying out operational and/or quasi-operational forest fire detection missions.

As forest fires are very sophisticated and unstructured, it is crucial to use several sources of information in different locations. In addition, the rate at which this information is updated may be unsatisfactory if a single drone is deployed for a single large-scale forest fire or for several forest fires.

Colour, motion and geometry of the detected fire are commonly used in existing papers, with colour mainly used to segment fire areas by extracting color feature of the fire using a trained network . Considerable effort has been devoted to the development of offline forest fire detection algorithm which is fast in the detection process and can maintain performance accuracy, this article proposes a new framework for fire detection based on combining color-motion-shape features with machine learning technology. The characteristics of the fire are not only red but also from their irregular shape and movement that tends to be constant at specific locations (13).

The paper of (38) has introduced the multi-UAV system utilized in this examination for agreeable forest fire detection . The article has exhibited the equipment parts of the helicopter type UAV considered in the paper. A technique for vision-based forest fire detection in clear range pictures, which utilizes both color and movement examinations, is produced for the UAV-based observation application. Algorithms for FFD in ultraviolet and visual cameras have being displayed.

(16) proposed a robust forest fire detection system that requires precise and efficient classification of forest fire imagery against no-fire. They had curated a novel data-set (Deep-Fire) containing diversified real-world forest imagery with and without fire , a training based on VGG-19 transfer learning to achieve improved prediction accuracy was made with a comparison of the performance of several machine learning approaches.

Meanwhile (Chopde et al.) proposed a deep-learning-based forest fire detection model that can detect the forest fire from satellite images, Due to the limitations of the CNN model, RCNN has been used to reduce the prediction time. The model was able to differentiate between images with and without fire with an accuracy of 97.29%.This approach focuses on observing the forests without steady human supervision.

Author	Detection Method	adopted features	Resolution	Object detection
Chi yuan et al.2015	remote sensing	multiple	Not mentioned	flame
S.Sudhakar et al 2020	real time process	Color, motion	320*240	flame and smoke
Agus Harjoko et al 2022	real time process	Color and motion	not mentioned	flame
Khan,Ali et al,2022	real time process	Color and motion	not mentioned	flame
Chopde et al,2022	real time process	multiple	not mentioned	flame

Table 1.2: Review on vision vased automatic forest fire detection

1.3.3 Forest Fire Geo-localization Techniques

Vision-based technologies for automatic forest fire diagnosis and prognosis The most common mission of forest fire diagnosis by UAV is to produce Geo-located alarms, while the most common mission of fire prognosis is to predict fire spread, which is sufficient to meet the requirements of operational forest fire fighting. (12) paper proposed a mathematical models for the positioning of detectors that are created to have high spatial resolution in detecting the coordinates of forest fires by using the libraries of Google Maps APIs in the cloud. The geolocation of the fire and behavior of the fire inside the model are then simulated visually on the map portal, thanks to an extraordinary standalone software program called FireAnalyst. The proposed system was implemented for the Faruk Yalçın Zoo and Botanical Park in Darıca, Turkey.

(6) paper show planning and control methodology for forest fire localization. Improving the localization mission by a decision-based strategy resulting from a probabilistic model based on the temperature in order to estimate the distance towards the forest-fire. The UAV optimizes its trajectory according to the state of the forest-fire knowledge by using a map to represent its knowledge and updates it at each step of its exploration.

Some fusion techniques have been used to provide information before, during and after forest fires, using onboard vision and telemetry sensors and navigation units (GPS, IMU) (Shahbazi et al. 2014). Meanwhile, the location of the drones themselves is known through the use of GPS. The orientation of the UAV camera is calculated by composing the orientation angles of the orientation and tilt system with the orientation angles of the UAV cell, which are estimated by IMUs and compasses. In (33) work researchers proposed a unmanned aerial vehicle (UAV) that can map a region in order to clear fuel to prevent the spread of fire . they developed a multi-sensor payload consisting of cameras, LiDAR, GPS with onboard processing and SLAM system to understand the 3D structure of the environment and a semantics system to identify fuel and other features in the environment .This approach provides a 3D map of the environment and geo-registered maps describing the locations of fire.

also (23) presented a lightweight model, NanoDet, which was applied as a detector to identify and locate fires in the vision field. After capturing 2D images with fires from the detector, the binocular stereo vision is applied to calculate the depth map, where some algorithms were proposed to eliminate the interference values when calculating the depth of the fire area.

Meanwhile, (8) proposed in this paper, the overall dynamic model of the system is developed . The aerial vehicle is designed with vertical and lateral rotors. A model-based controller is developed to guarantee stable flight of the aerial vehicle. A localization method is proposed, where the aerial vehicle's relative position with respect to its base is computed by including a force sensor on the vehicle .

Authors	Detection method	Geolocation	Spectral bands	Resolution Image	stabilization
Shahbazi et al. 2014	Data fusion	yes	Multi-spectral	Not mentioned	Not mentioned
Belbachir.2015	probabilistic model	yes	Visual	Not mentioned	yes
YUCEL GUL-LUCE,2020	mathematical modeling algorithms	yes	multispectral infrared detectors	Not mentioned	not mentioned
Russell et al.2022	probabilistic model	yes	Visual	not mentioned	yes
Kangjie Lu,2022	mathematical model	yes	Visual	not mentioned	not mentioned
Chaikalis et al. 2022	dynamic model	yes	not mentioned	not mentioned	yes

Table 1.3: Review on forest fire geolocalization techniques

1.4 Forest fire detection and localization Methods.

1.4.1 Deep CNN-based fire detection system

Accurate identification of fires can contribute to their control. Traditional methods of fire detection rely mainly on temperature or smoke detectors. These detectors are sensitive to damage or interference from the external environment. Image-based fire detection methods mainly include traditional colour-based methods and deep learning. Colour-based methods need to take into account the colour of the flames, and different types of fires produce different colours of flames, as mentioned earlier. Meanwhile, convolutional neural networks (CNNs) have achieved state-of-the-art performance in image classification and other computer vision tasks.

The conventional fire detection algorithm uses the characteristics of flames to identify fires. However, it is difficult to define the characteristics of flames. Recently, convolutional neural networks (CNN) (22) have been widely used for object detection. In this regard, (24) used a CNN for fire and smoke detection in video. In the early stages of fire detection, researchers trained fire images using a CNN built based on scaling constraints However, the CNN requires a huge amount of data to be trained and takes a lot of time where the problem was discussed by (22) in his two research papers . In this view, transfer learning has been used to design neural networks in addition to data augmentation which facilitates the reconstruction of large databases, and AlexNet based CNN (36) and YOLO based CNN have been proposed to perform fire detection.

The application of CNN to fire detection systems will greatly improve the accuracy of detection, which will minimize fire disasters, especially in remote hard-to-reach areas, and reduce the ecological and social consequences. However, the main problem with CNN-based fire detection systems is their implementation in real-world monitoring networks, due to their high memory and computing requirements for inference.

1.4.2 Stereo vision-based fire localization system

A fire location system based on stereo vision, which can automatically record the position of the fire and extinguish the source. The image data captured by the cameras is analysed by an image processing algorithm to detect the flames of the fire in real time, and then the calibrated stereo vision system is implemented to obtain the 3D location of the fire or more precisely a relative position of the fire according to the camera itself.

The accurate location of the fire is an important prerequisite for a quick response of the fire-fighting team and for the accurate injection of water into the fire, especially in forests.

Mainly, a stereo vision-based system which relays on disparity map generation is proposed, and then, by calculating the accurate calibration data of the camera, the 3D coordinates of the fire in the real world can be extracted. As a result, more and more research is focusing on the positioning of fire and its three-dimensional modelling using stereo vision sensors. (37) used the stereo vision sensor system for 3D fire location and successfully applied it in a coal chemical company.

(19) used a stereo infrared camera and laser radar to capture fire images in smoky scenes (19) (18) (20),the fusion sensor achieved good fire positioning in a clean and complex environment, but the effective working distance is less than 10m .

Similar studies were conducted by (27),limited by the working range of the infrared camera and the base distance of the stereo vision system, the system is only applicable for short distance fire identification and positioning. (39) built a stereo vision system with a base distance of 100mm for 3D modelling and geometric analysis of fire. Experimental results in the outdoor environment show that reliable fire localization can be achieved when the depth distance is less than 20m.

The stereo vision system as presented in figure 1.2 for fire positioning is itself faced with some challenges discussed by (28) ; (46). For example, the accuracy of the stereo vision calibration directly affects the results of the light positioning, the positioning accuracy decreases with increasing distance, and the system is not very adaptable to the positioning of the light at different distances. Therefore, several techniques and solutions were mentioned and adopted in order to evaluate better results .

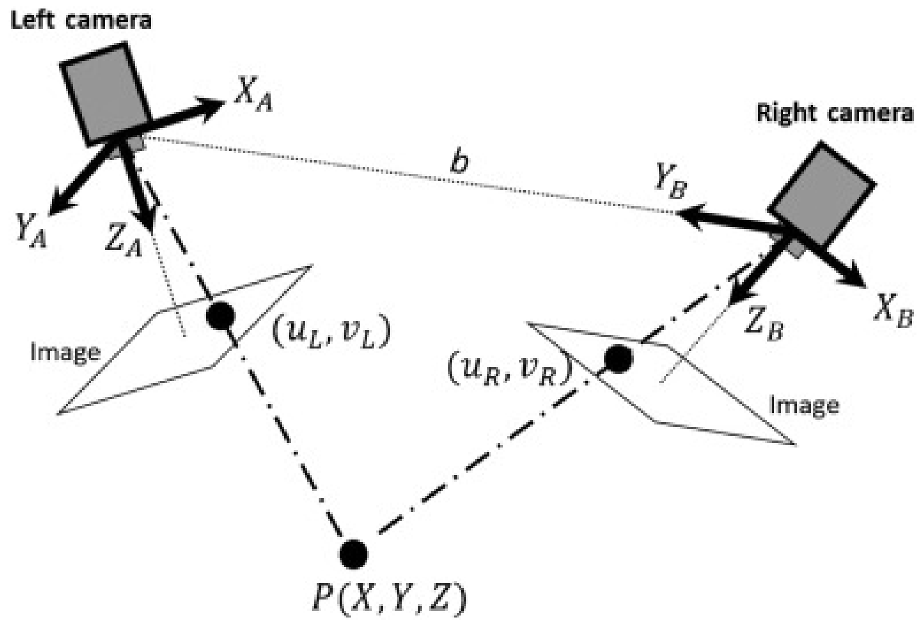


Figure 1.2: Typical Stereo vision system

1.5 Conclusion

We have devoted this chapter to the presentation of UAV technologies, their architecture and the latest literature reviews and methods in order to detect forest fire using computer vision and localization system using stereo vision.

Chapter 2

Fire detection using computer vision

2.1 Introduction

As a new fire detection technology, image fire detection has recently played a crucial role in reducing fire losses by alarming users through early fire detection. Image fire detection is based on an algorithmic analysis of images. Novel image fire detection algorithms based on the advanced object detection CNN models of Faster-RCNN, SSD, YOLO V2 and YOLO v4 are proposed in this chapter.

2.2 Research designing

The primary purpose of this thesis is to achieve fire detection , more specifically forest fire detection .As shown in figure 2.1, the flowchart of fire detection process through pointed steps is represented, those steps are divided into four sub chapters as the structure of this chapter is denoted.

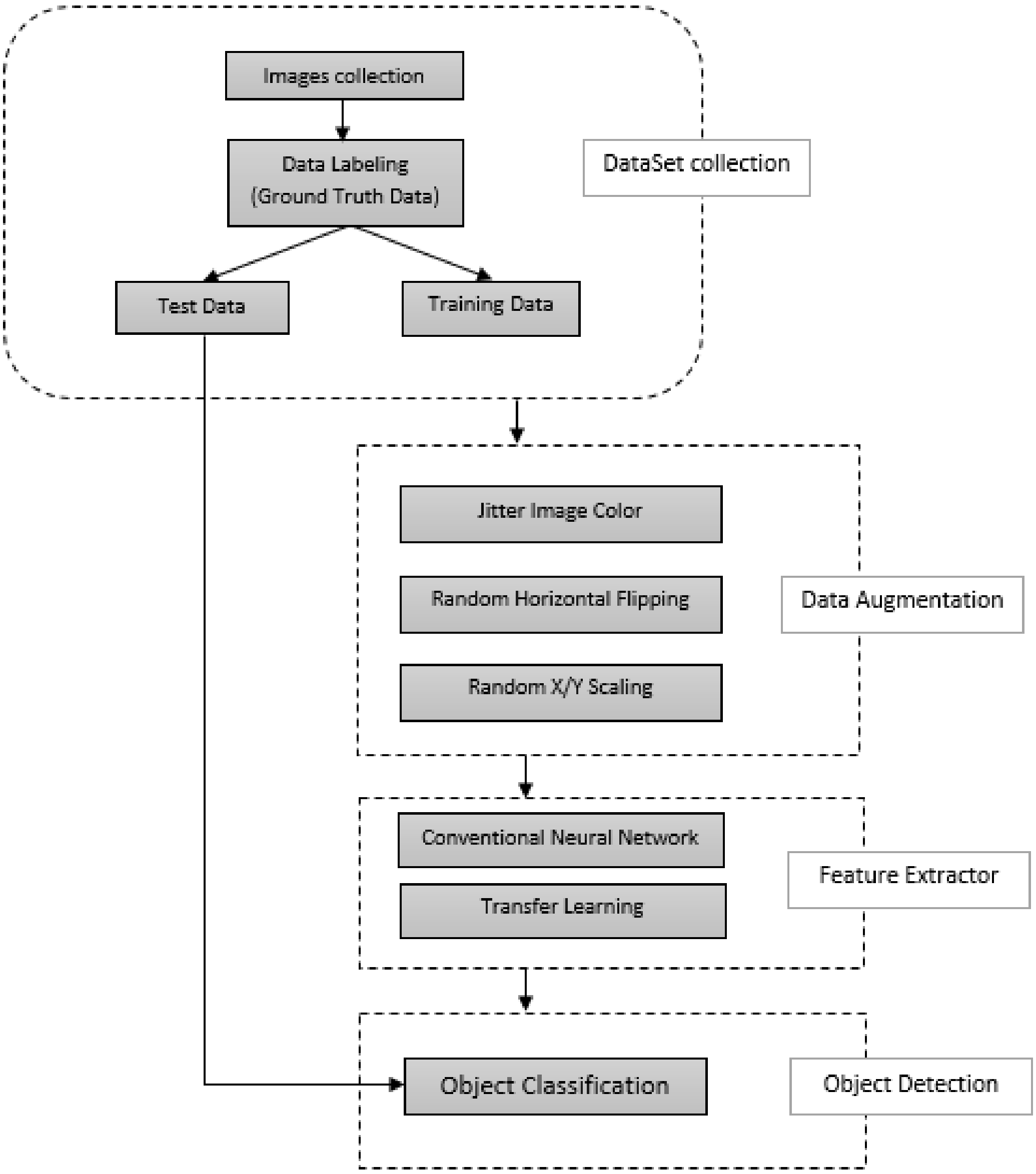


Figure 2.1: Flowchart of the steps of forest fire detection

The first part in this chapter is Data-set preparation which includes Data collection and augmentation due to the necessity of Big data in deep learning.The second part is Feature Extractor which was designed as the leading part in this structure, considering the complexity of our problem and the limitation of our computer hardware, we relayed on Transfer learning .The following part is object detection which includes the results and discussion.

2.3 Data selection and pre-processing

2.3.1 Data selection

Even after extensive research and various implementations on fire detection, not much data is openly available. Most of the existing implementations worked on data-sets of videos and not all of this data can be put into use as the images obtained through this approach are a series of frames from these videos.
The network needs to learn the outdoor environment features.

The collection of the data-set mainly relayed on the famous FLAME data-set , which is a fire images data-set collected by drones during a prescribed burning piled detritus in an Arizona pine forest.
The data-set included a video recording which was transformed into a set of 25018 frame. Pre-processes were applied to ensure the availability and suitability of the aerial images of the study area. After elimination of images not containing Fire and ones with fire not visible, besides computer hardware limitation, we ended up with 2906 images labeled Manually.

Figure 2.2: set of images from the FLAME dataset containing fire

Figure 2.3: a set of successive images not containing fire.

2.3.2 Data pre-processing

2.3.2.1 Image labeling:

After the collection of the data-set, the images were resized to a common resolution of 416×416. The labeling process was manual using MATLAB Image Labeler App.

The process is so simple, it is based on specifying the images file, defining the Name of the class (i.e. Fire Label) and than creating manual bounding boxes surrounding the fire in each image.

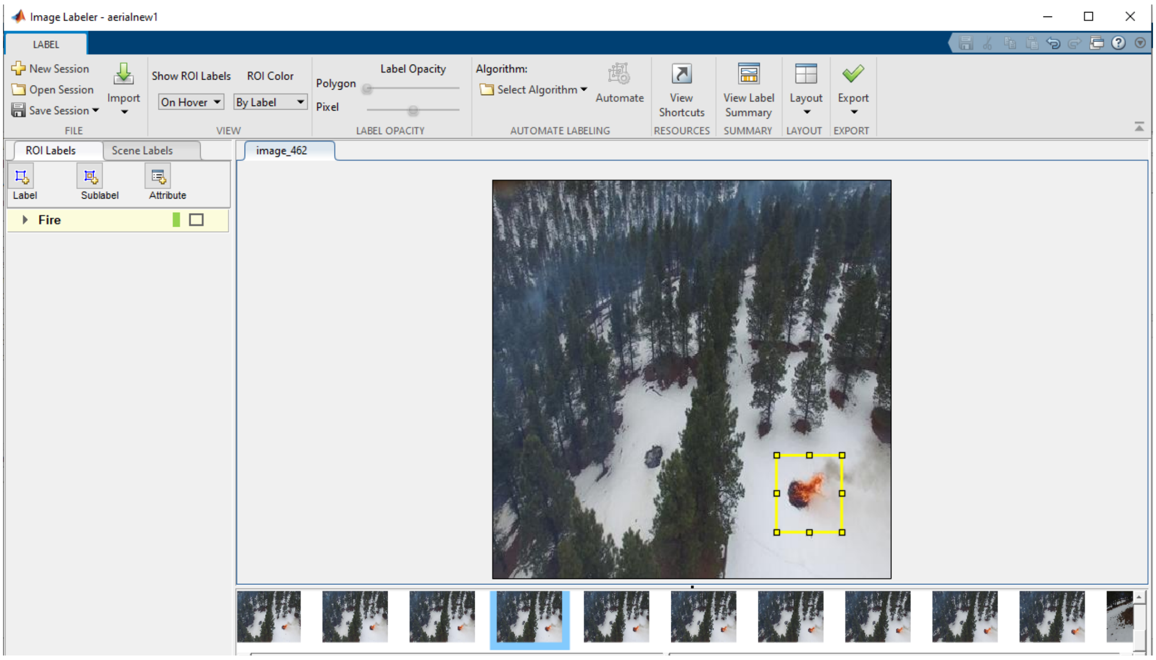


Figure 2.4: image labeler session

Finishing the labeling process requires the generation of Ground Truth Data containing information about image filenames and its associated bounding box coordinates .

2.3.2.2 Data splitting:

The data-set with all the features was ready to be fed into the machine learning algorithms. Before proceeding to use the algorithm, it was advised to apply data splitting.

This technique was used to split the existing data into two or three subsets. These subsets are usually called training, validation and testing sets and are used to develop feature sets that model can learn from the data.

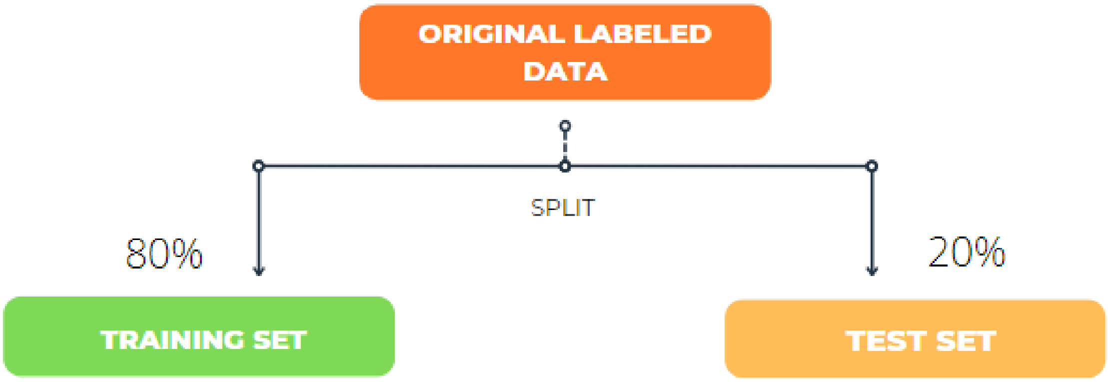


Figure 2.5: Test training and validation sets

There are several ways to split the data into training and testing sets. The most common approach is to use random sampling. Complete random sampling is a straightforward strategy to implement, and usually protects the model from being biased towards any characteristic of the data. However, this approach can be problematic when the response is not evenly distributed across the outcome. A less risky splitting strategy would be to use a stratified random sample based on the outcome.

This approach ensures that the frequency distribution of the outcome is approximately equal within the training and test sets. In the case of many number of images, the model requires a lot of training, then a larger chunk of the data is used for training purposes.

a.Training set : A training set is a subset from the original data-set. The model uses a training set for learning the features from the data-set, fit the parameters and adjust weights to understand the empirical relationship between these features. 80% of the original data-set is split and used as training set.

b.Test set : A test set was used to test the model hypothesis. This data is left untouched in any of the model training even though it is a subset of the original data-set. Usually, this data is never seen by the network until it finishes training. The final model is applied on the test data to get an accurate measure of how it would perform when deployed on real-world data. A mini-test set of 581 images was used to analyze the performance of different networks.

2.3.3 Data augmentation

We could say that Data augmentation is another common pre-processing technique that involves augmenting the existing data-set with perturbed versions of the existing images, scaling, rotations, and other affine transformations are typical. Augmentation is applied to enlarge the data-set and expose the neural network to a wide variety of variations of images.

This makes it more likely that the model recognizes objects when they appear in any form and shape:

Jitter image color: ColorJitter is a type of image data augmentation where we randomly change the brightness, hue, contrast and saturation from the HSV color space of an image , as mentioned:

- Hue Jitter Hue specifies the shade of color, or a color's position on a color wheel. As hue varies from 0 to 1, colors vary from red through yellow, green, cyan, blue, purple, magenta, and back to red. Hue jitter shifts the apparent shade of colors in an image.
- Saturation Jitter Saturation is the purity of color. As saturation varies from 0 to 1, hues vary from gray (indicating a mixture of all colors) to a single pure color. Saturation jitter shifts how dull or vibrant colors are.
- Brightness Jitter Brightness is the amount of hue. As brightness varies from 0 to 1, colors go from black to white. Brightness jitter shifts the darkness and lightness of an input image.

Figure 2.6: Left image representing the original picture , right one representing jitterColor augmentation

Random horizontal flipping: Horizontal axis flipping is much more common than flipping the vertical axis. This augmentation is one of the easiest to implement and has proven useful on data-set, the idea of chosen the random horizontal flip came since the fire is generally burning upwards , the flipping is made with certain probability.

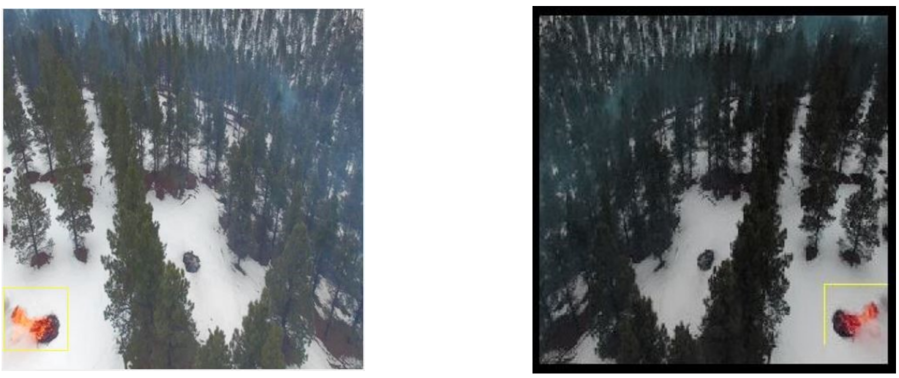


Figure 2.7: Left image representing original image , right one representing the result of horizontal flipping

Random X/Y scaling: The image is scaled outward and inward along X/Y respectively. An object in new image can be smaller or bigger than in the original image by scaling. Image Scale Augmentation is an augmentation technique where we randomly pick the short size of a image within a dimension range. One use case of this augmentation technique is in object detection tasks.

Figure 2.8: Left image representing original image , right one representing object scaling along X

To be noted that the data augmentation process in this thesis was performed only on training data to avoid overlapping and to increase the training precision . After performing data augmentation on the training Data , we ended up with 9300 images for training .

2.4 Artificial Neural Networks

A neural network consists of artificially created groups of neurons, called layers. An artificial neural network usually consists of an input layer, a number of hidden layers and an output layer, as shown in the figure below. Each layer consists of a number of artificial neurons, and the flow of information between the neurons is indicated by the black lines.The first layer shown in the figure is the input layer. This layer is considered passive, which means that it does not modify the data. The neurons in the input layer receive values on their input channel and transmit this information to their individual connections. Unlike the input layer, the hidden layers are considered active layers, meaning that they can modify the incoming data. In the figure , each value is sent to all hidden neurons (represented by the arrows), this is called a fully interconnected structure. The number of hidden layers differs for each network and depends strongly on the type of problem the network is trying to solve. The hidden layers can also use different types of transfer functions, such as ReLU, Tanh and Sigmoid. The problem to be solved and the data to be processed are two decisive factors in choosing the right transfer function.

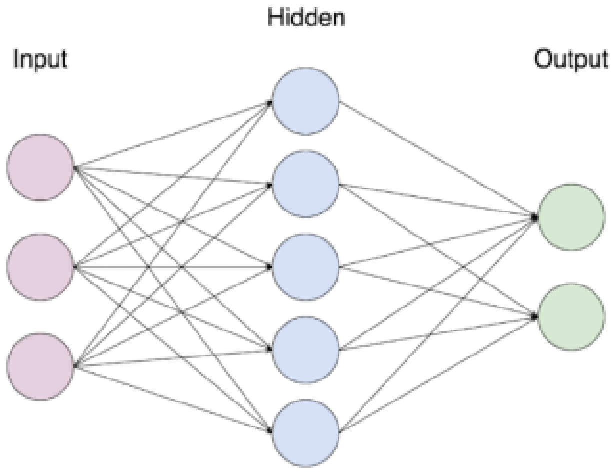


Figure 2.9: An example of a three layered neural network , compromised of an input layer, a hidden layer and an output layer.

2.5 Convolutional neural network

Deep learning networks are generally distinguished from shallow neural networks by their depth. A neural network that consists of more than one hidden layer is generally defined as a deep neural network. Convolutional neural networks are similar to the neural network in that they consist of neurons with learn able weights and biases, and the entire network always expresses a single differentiable score function. However, the convolutional neural network is generally defined by the type of hidden layers it uses, such as convolution, fully connected, normalization, and pooling layers.
Using neural networks for image pattern recognition works well for small, single-layer images consisting of simple shapes. However, using fully connected neural networks on high resolution, three color channel images would result in a considerable number of parameters to train. Due to the increased amount of parameters, the model will likely tend to over-fit.
The neurons in the convolutional neural network are arranged in a three-dimensional structure: width, height and depth. Rather than focusing on a single pixel at a time, the convolutional network uses square parcels of pixels and passes them through a filter, called a kernel. The goal of this process is to transform the image into a form that is easier to process, without losing the features that are essential for prediction.

2.5.1 Convolution

The convolution operation is the main building block of a convolutional neural network. The convolution takes a filter or kernel of a specified size and slides it over the input with a given stride, that specifies how many columns the filter should move on the input image. A dot product between the filter and section of the input (the same size as the filter) is computed in each step, shown in Figure. The output generated from each step are summed into a feature map which is put together as a final output .In purely mathematical terms, the operation is defined as a linear operator that transforms data from one domain to another:

$$(f.g(t)) = \int_{-\infty}^{\infty} f(\pi)g(t-\pi)d\pi \tag{2.1}$$

The input image I is on the left side of Figure 2.10 and the filter K is in the middle. Due to the shape of the filter, it is annotated as a 3x3 convolution. To produce the feature map, element-wise matrix multiplication is applied at every location, and the result is the sum, annotated as I * K on the right side of the Figure.

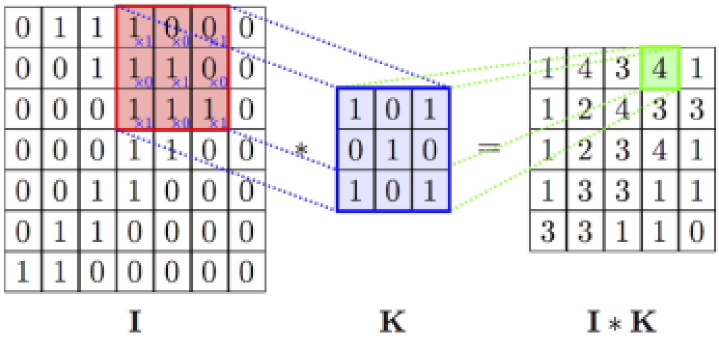
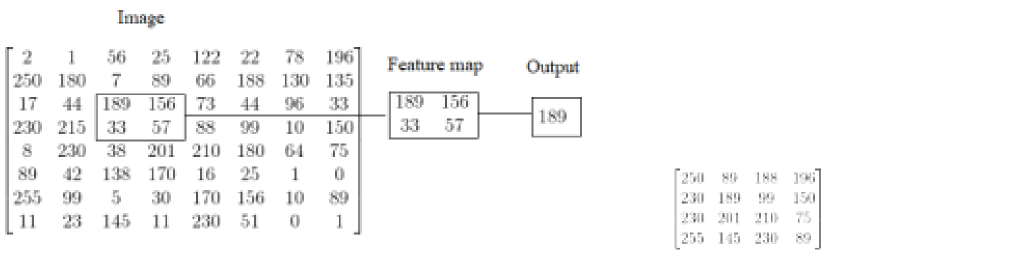


Figure 2.10: An example of a convolution where I is the input, K the kernel and I * K is the output.

2.5.2 pooling

Pooling, or down-sampling is used to reduce the dimensionality of feature maps. Pooling with a 2x2 feature map, for example, takes four pixel values as input and output a value for those pixels. As shown in Figure below, 2x2 pooling on an 8x8 image will result in 16 feature maps presented in a 4x4 matrix. There are different types of pooling such as average, mean, max, min and stochastic pooling. The pooling types are distinguished by how they choose pixel values to create their feature maps.
The less obvious of the pooling types; stochastic pooling picks a value randomly, where high pixel values have a higher chance to be picked .



(a) 2x2 Max-Pooling example (b) 2x2 Max-Pooling result

Figure 2.11: (a) Example of a 2x2 max-pooling with stride = 2, where (b) is the resulting output.

2.5.3 Prevent overfitting

Overfitting models are one of the most common problems encountered by researchers and companies in the field of deep neural networks. One reason is that many state-of-the-art architectures have a large number of parameters to learn during training. Overfitting occurs when a trained model has high validation accuracy on training data, but does not perform well on test data. Specifically, the trained model learns the behavior of the noise patterns in the training data, which creates a large difference between the training error and the test error, which is the definition of overfitting and is visualized in the figure 2.12.

Deep neural networks typically have a large number of trainable parameters. A complex model that contains more parameters than warranted by the amount of training data leads to an overfitted model. This is one of the major pitfalls of big data, which is becoming increasingly important. In the figure, the model fits the data perfectly during training, but performs poorly in samples outside the training data.
To avoid overfitting, one can either increase the amount of training data or improve the generalization ability of the networks by applying different techniques explained below:

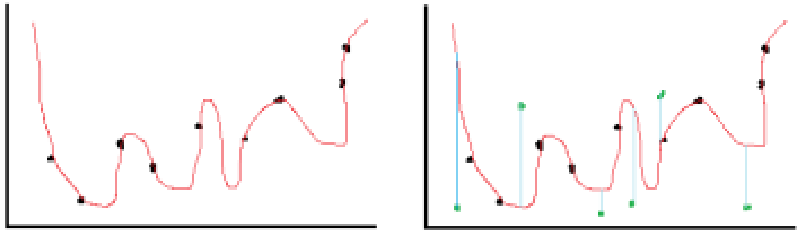


Figure 2.12: Left - a figure where the line intersects the black dots perfectly(overfitted), and the right image when the model performs poorly on the test data(green dots), where the error is huge(blue lines).

Regularization : Regularization is a technique used in order to prevent overfitting. The most commonly used regularization algorithms are L1, L2 and dropout. The L1 and L2 regularization updates the cost function by adding a regularization term which decreases the values of the weight matrices. The assumption is that a network with smaller weights matrices leads to simpler models. The difference between L2 and L1 regularization is that L2 decay the weights towards zero, but not exactly zero, while L1 regularization could be reduced to zero .

$$Costfunction(L2) = Loss + \frac{\lambda}{2m} \times \sum \|W\|^2 \tag{2.2}$$

The L1 regularization penalize the absolute value of the weights, meaning that a neuron with a high weight will cost more than a neuron with a low weight, therefore L1 could be useful while trying to compress the model .

$$Costfunction(L1) = Loss + \frac{\lambda}{2m} \times \sum \|W\| \tag{2.3}$$

Dropout : is a commonly used technique to enhance the network's ability to generalize the data. The method randomly drop neurons in the neural network and temporarily removes the incoming and outgoing connections. The dropped neurons contribution to the activation of downstream neurons are temporally removed. Neighboring units will have to compensate for the removed unit and handle that specific representation of the missing neuron. The effect of the dropout is that the network becomes less dependent of the specific weighted neurons and therefore enhance the networks ability to generalize .

Loss : When optimizing an algorithm, a function is used to evaluate a candidate solution, also called an objective function. By applying an objective function such as cost and loss functions one seek to maximize or minimize the objective function. Normally in deep learning, it is preferred to minimize the objective function which calculate a possible solution, in this case a set of weights with the lowest cost. The value returned by the loss function is called "loss" .

The loss is a measurement that represent the classification models performance. For example, a model that predict a probability of 0.1 when the observation labels value is 1, would result in a high loss. Some of the functions that are commonly used as loss functions are cross-entropy and mean square error. When calculating the cross-entropy, one seeks the set of the model weights that minimize the difference between the model's predicted probability distribution given the dataset and its distribution of probabilities .
Depending on the task of the neural network, binary cross-entropy or categorical cross-entropy can be applied. If the network is trying to classify multiple classes from the dataset, applying categorical cross-entropy is sufficient. If the dataset contains only two classes one can apply binary cross-entropy that predicts the probability of the test dataset belonging to one class .

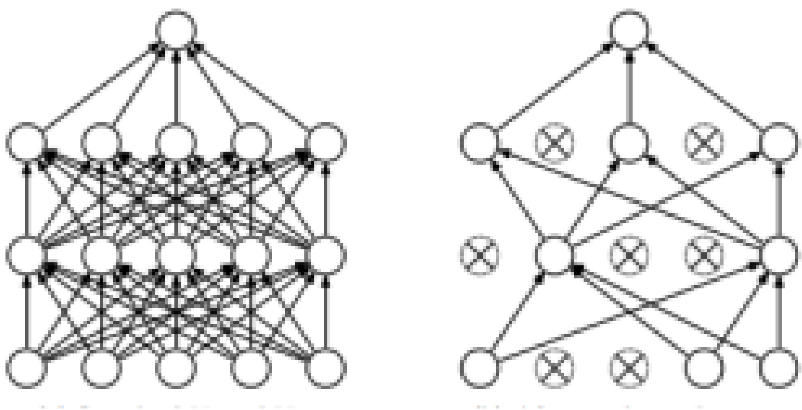


Figure 2.13: Figure shows the effect of dropout. The left network is the original neural network. The right network is the result after applying dropout .

2.6 Convolutional neural network architectures

The discovery that a convolutional neural network (CNN) could be used to extract higher- and higher-level representations of image content was a breakthrough in building models for image classification which lead to a significant improvement in image classification in several topics, such as fire detection. Rather than pre-processing the data to extract features such as textures and shapes, a CNN uses the image's raw pixel data as input and "learns" how to extract these features and, in return, infer what object they represent (21).

In practice, there are two types of mainstream object detection algorithms.

Algorithms like R-CNN and Faster R-CNN use a two-step approach - first to identify regions where objects are expected to be found and then detect objects only in those regions using convnet. On the other hand, algorithms like YOLO (You Only Look Once) and SSD (Single-Shot Detector) use a fully convolutional approach in which the network is able to find all objects within an image in one pass (hence 'single-shot' or 'look once') through the convnet.

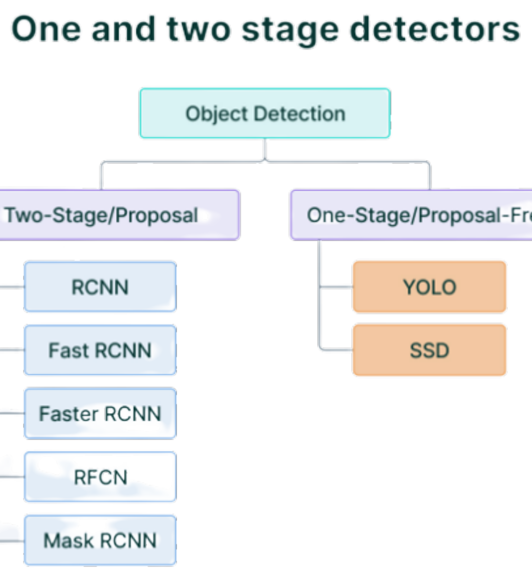


Figure 2.14: Types of mainstream object detection algorithms.

The region proposal algorithms usually have slightly better accuracy but slower to run, while single-shot algorithms are more efficient and has as good accuracy and that's what we are going to focus on in this section. (4)

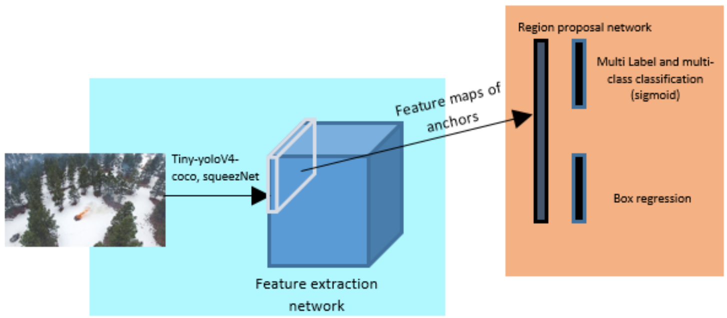


Figure 2.15: plan representing Convolutional neural network architecture " YOLO "

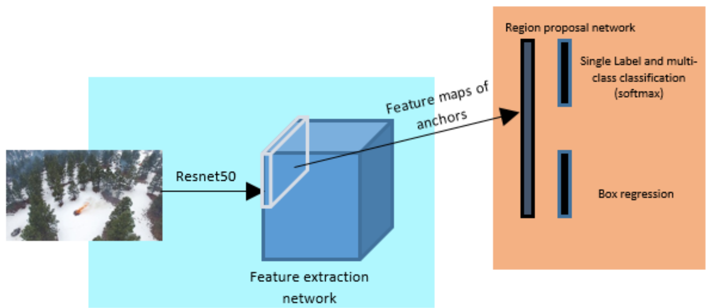


Figure 2.16: plan representing Convolutional neural network architecture " SSD "

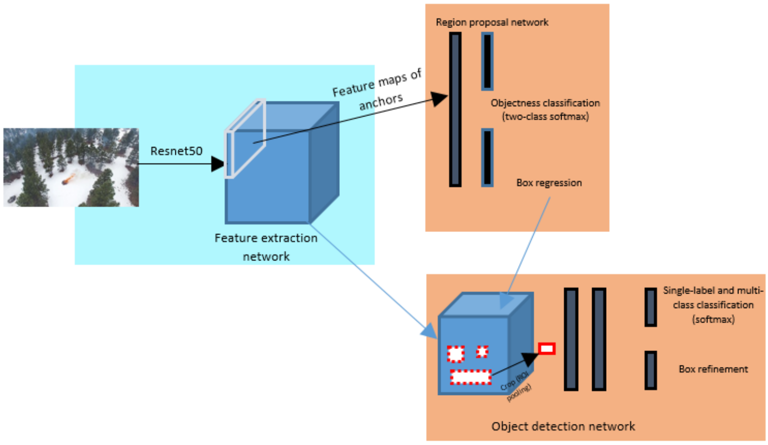


Figure 2.17: plan representing Convolutional neural network architecture " Faster R-CNN

2.6.1 YOLOV2

You look only once (YOLO) is a deep learning model for object detection. YOLO can detect the location of multiple classes at one time. It is accurate and fast, which meets the requirements for real-time processing, YOLOv2 is the second version, which has several enhancements to YOLO. Indeed, YOLO has two drawbacks for object detection. The first weakness is that it is inaccurate to locate and position the classes to be detected in the images. The second problem is a low recall rate when it compares to the regional based detectors. YOLOv2 resolved these issues, thus increasing the accuracy and the speed of the architecture.

YOLOv2 made a number of iterative improvements on top of YOLO including Batch-Norm, higher resolution, and anchor boxes (21).

2.6.2 YOLOV4

it is a one-stage detector with several components to it. YOLOv4 outruns the existing methods significantly in both terms “detection performance” and “superior speed”. in (21) it is mentioned “speedily operating” object detector that can be trained smoothly and used in production systems. the different building blocks of YOLOv4:

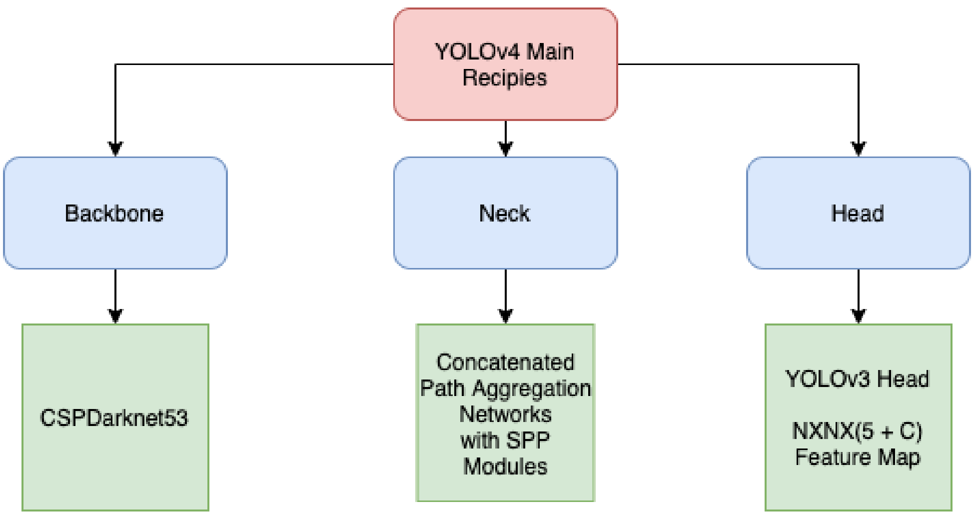


Figure 2.18: Architecture of the latest update on YOLOv4

2.6.3 Faster R-CNN

Faster RCNN is an object detection architecture presented by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015, and is one of the famous object detection architectures that uses convolution neural networks (17)

Faster RCNN is composed from 3 parts

Part 1 : Convolution layers: In this layers we train filters to extract the appropriate features the image, for example let's say that we are going to train those filters to extract the appropriate features for a human face, then those filters are going to learn through training shapes and colors that only exist in the human face.

Part 2 : Region Proposal Network (RPN): RPN is small neural network sliding on the last feature map of the convolution layers and predict weather there is an object or not and also predict the bounding box of those objects.

Part 3 : Classes and Bounding Boxes prediction: Now we use another Fully connected neural networks that takes as an input the regions proposed by the RPN and predict object class (classification) and Bounding boxes (Regression).

2.6.4 Single-Shot Detector:

Single-Shot Detector (SSD) has two components: a backbone model and SSD head. Backbone model usually is a pre-trained image classification network as a feature extractor, The SSD head is just one or more convolutional layers added to this backbone and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layers activation's (3).

In the figure below 2.19, the first few layers (white boxes) are the backbone, the last few layers (blue boxes) represent the SSD head.

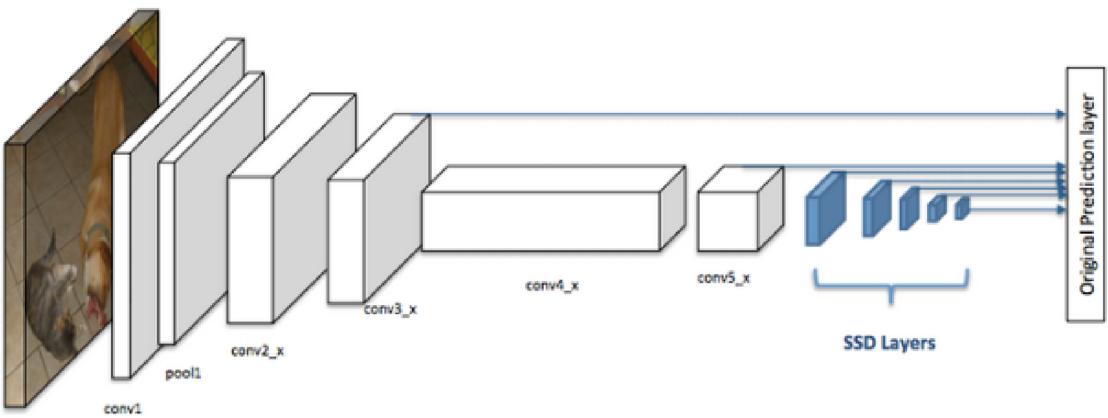


Figure 2.19: Architecture of a convolutional neural network with a SSD detector

2.6.5 Transfer learning

Many resources are required for training a deep learning model. In order to fasten the learning process, a common practice is used , what is known as transfer learning. Humans usually are able to learn new tasks by using the learn knowledge from previous tasks. Transfer learning uses the same idea by training a network on a general task, such as object classification, and then using its pre-trained weights as an initialization for the network of the task to be done, as could be object counting. This avoids wasting resources on training the model from scratch each time to learn the basic feature mapping and achieves a faster convergence of our model. For example, a Convolutional Neural Network can use transfer learning by using the pre-trained weights for object recognition and avoid learning how to recognize lines, curves, etc. Instead, it can focus its resources on computing the gradient for more specific feature mappings required for its task.
When using transfer learning on convolutional neural networks, it is important to know how are distributed the recognized features throughout the network. The convolution filters in the first layers of the network provide a general analysis of the given input like, as previously mentioned, the detection of lines, curves, corners, etc. Nonetheless, as we get further into the inner layers of the network, the features start being more specific and task-oriente.
The benefit of transfer learning is that one can acquire a pre-trained model with its weights. By removing the output layer and adding their own desired output one reduce the problem with limited data-sets. In the project, transfer learning was used for tiny-yolov4-coco, Resnet50 and SqueezeNet for yoloV4 , SSD and Faster R-CNN , and yoloV2 respectively.

2.7 Experimental results

In this part of the chapter, we will talk about the experimental results .We first discussed the selection criteria for data-sets and methods in the first part of this chapter, then we will talk about the selected methods' parameters, and finally, we will talk about the evaluation and post-processing of these methods. The implementation of these methods is mainly discussed due to how they influence the method selection. Prior to the commencement of the training process, the following steps have been completed that are essential for the training of the algorithm:

2.7.1 Data-set

Our data-set is explained in the beginning of this Chapter. In this section, we will discuss the short-comings and ways to improve it. The data-set showed before is considered a small size data-set with only 2906 images. We improved our data-set by applying several features:

- data augmentation. Which made the volume of images four times bigger and gave a bit of diversity for the set.
- Bounding Box annotation helps the model to know where in the image the object is focus.
- Resized all images to (416x416) pixels since this is the preferred input for our model.
- Split the set into subsets: Training and Test .

2.7.2 Training platform

For this thesis, we choose to use a VM in the Microsoft azure platform,due to its low cost and availability . it has the following specifications :

CPU	Intel Xeon 8370C 2.70 (4 vCpus)
RAM	16 Gb DDR4 3200Mhz
Disk	2HDD 7200 RPM
GPU	8191 Mb.

Table 2.1: Virtual machine specifications

The choice was made based on the experience of previous work with CNN and Artificial Intelligence also described as a problem in a previous master thesis in the same research area .When implementing a system based on CNN, dependencies are a major problem. Many of the tools needed to perform the desired work are too version specific, and multiple systems need to interact with each other to perform the task at hand.
The advantage of implementing such systems in a VM environment is that it is easier to start with "clean sheets" if we have dependency issues, and we can try out different versions until we find the combination that works as expected, without the tedious installation and removal of packages. We expect this tool to be very common in the field of machine learning and artificial intelligence.

2.7.3 Classification metrics

In object classification, the average accuracy (AP) is the most commonly used metric as it assesses the overall accuracy of the model. The AP measures how accurate the model is on a specific label (class). For all models, we want to calculate the AP for the label "Fire", which is based on the precision-recall curve. The different concepts included in the evaluation are detailed in the subsection.
The confidence score is predicted by the classifier and represents the probability that there is an object in the anchor box. Intersection over Union (ior) is defined by the intersection area divided by the union of a predicted anchor box and a ground truth box. It is used to determine whether the detection is TP (true positive) or FP (false positive) . When evaluating our models, we follow the threshold of 0.5 (50%). For a detection to be TP, it must satisfy three conditions:

- Confidence score > threshold
- Predicted bounding box has an IoU greater then a threshold with the ground-truth.
- The predicted label(class) matches the label of a ground-truth.

If either of the two last conditions is not fulfilled, we get a FP. When the confidence score drops below the threshold, we get a FN (False Negative).

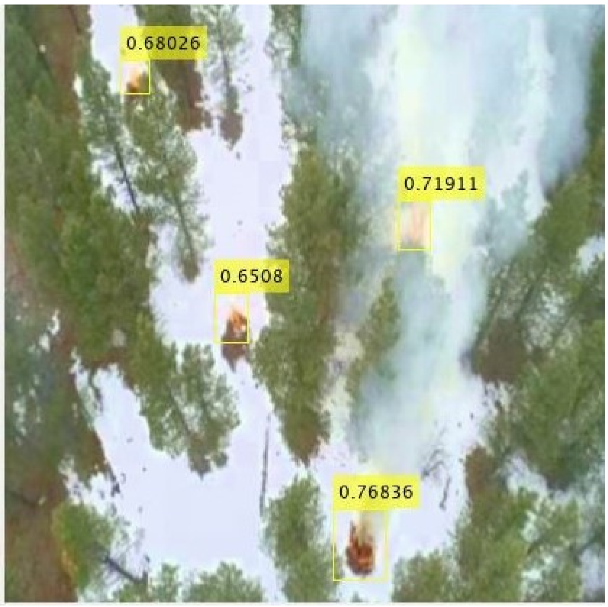


Figure 2.20: example of confidence score diversity using yolov2 detector.

Figure 2.21: Example on confidence score using yolov4 detector

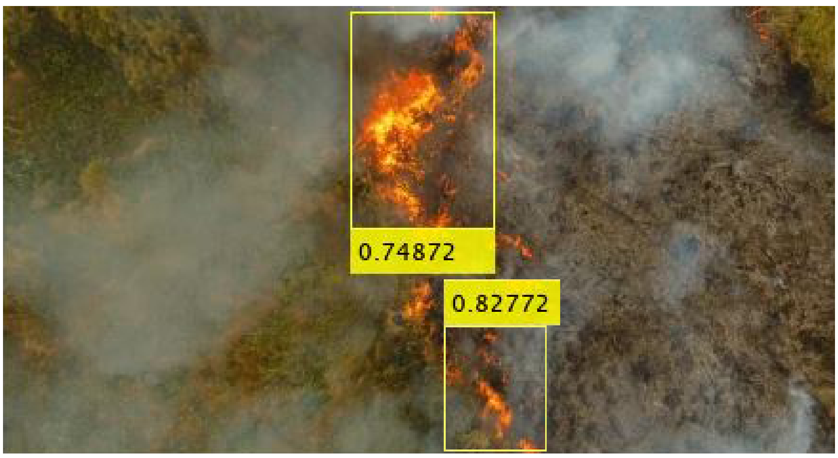


Figure 2.22: example of confidence score diversity using Faster R-CNN detector.

Precision is the baseline that reveals how accurate our model is in a specific class. It is defined by the number of TP divided by the sum of TP and FP

$$Precision = \frac{TP}{TP + FP'} \tag{2.4}$$

Recall is defined by the number of TP divided by the sum of TP and FN (this is known as the ground-truths).

$$Recall = \frac{TP}{TP + FN'} \tag{2.5}$$

Precision and recall curve : Precision and recall are inversely proportional. As precision increases, recall decreases and vice versa. It is best to achieve a balance between the two. This can be achieved by creating a precision-recall curve. The values of the precision-recall curve are calculated for subsets of detection's. The highest detection scores are added first, then the recalls, with detection scores added in descending order. Plotting all detection's on a graph gives the precision-recall curve. The precision score is between 0 and 1 on the vertical line, and the recall score is between 0 and 1 on the horizontal line. The PA is to find the area under the precision-recall curve. Since both precision and recall are between 0 and 1 (0-100%), the AP is also included in the parameters. The environment is static in all images, which is considered a major advantage for the final result, as it minimises interference. When testing with different images, we encountered a problem that needs to be taken into consideration. Since the data-set contains only images with a single class in each image, we know that the training is performed on single objects. Using all these elements, the results of the three algorithms were compared in order to compare their performance. The results are as follows.

2.7.4 Results comparison

Evaluation of the four models shows that performance in regards to AP is above 80% for both YOLOv4 and YOLOv2. We suspect that the combination of bounding box annotation, RGB images, and the static environment are all factors that contribute to the high precision of the model, while for SSD it couldn't reach more than 42% and for Faster RCNN we came up with a precision of 66% .
The model is trained on resized images (416×416)pixels, as this is the recommended input image size for the YOLOv4-tiny model and it was generalized for other networks to provide equivalence between training complexity and accuracy. We tested the model on the original size and resized image, which did not affect the model's accuracy. It performed equally on both inputs. Classification speed is hard to define for the model, as it is tested on a virtual machine , as multiple other applications are running simultaneously in the background.

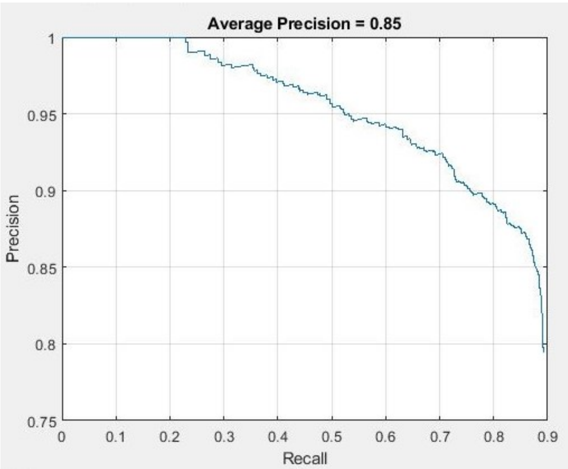


Figure 2.23: average precision for yolov2

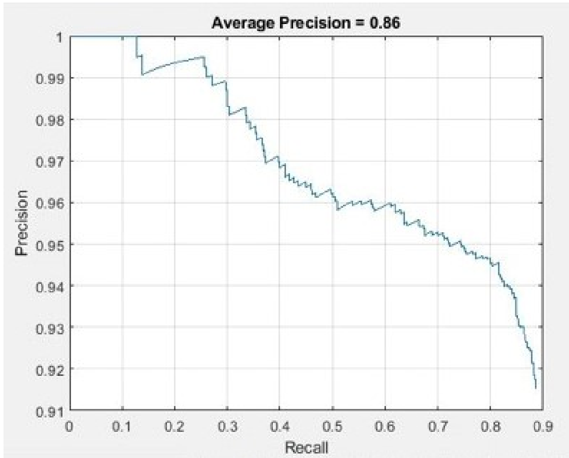


Figure 2.24: average precision for yolov4

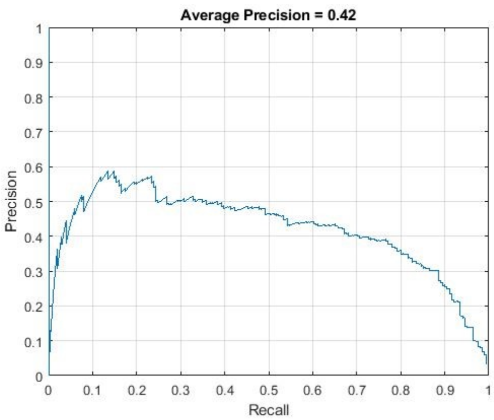


Figure 2.25: average precision for SSD

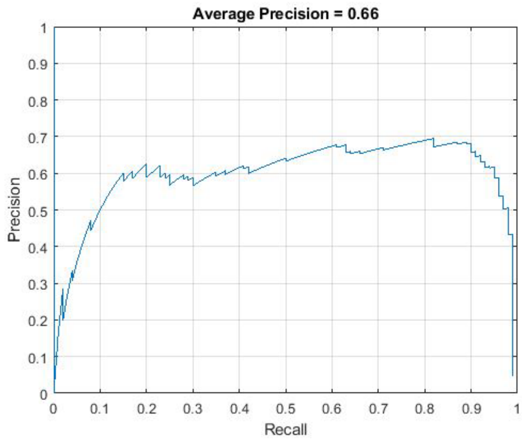


Figure 2.26: average precision for Faster r-cnn

2.7.5 Discussion

Following were some limitations that were observed in the three models

SSD: When it comes to smaller objects, the performance of SSD is much worse than that of Faster R-CNN. The main reason for this disadvantage is that in SSD, the high-resolution layers are responsible for detecting small objects. However, these layers are less useful for classification as they contain lower level features such as colour spots or edges, which reduces the overall performance of the SSD. Another limitation of this method, which can be inferred from the complexity of the SSD data augmentation, is that the SSD requires a large amount of data for training purposes. This can be quite expensive and time consuming depending on the application.

Faster R CNN: The accuracy of this algorithm comes at a cost in terms of time complexity. It is significantly slower than algorithms like YOLO. Despite improvements over RCNN and Fast RCNN, it still requires multiple passes over a single image unlike YOLO (this despite the fact that we used shuffling after each epoch). FRCNN has many components: the convolutional network, the region of interest (ROI) clustering layer and the region proposal network (RPN). Each of these can act as a bottleneck for the others.

YOLO: YOLOv4 was one of the best modifications that had been done to an object detection system even with the introduction of Tiny-yolov4-coco . This modified update was received very well among the critics and other industrial professionals. But it had its own shortcomings. Though YOLOv4 is still considered to be a veteran, the complexity analysis showed flaws and lacked optimal solutions to the loss function. It was later rectified in an optimized model of the same in other updates and was later used and tested for functionality enhancements . A better version of a given software is the best to analyze the faults in the former. After analyzing YOLOv4 we can see that version 3 used to fail when the image had multiple features to be analyzed but they weren't the highlight of the pic.
The lack of accuracy was always an issue when it came to smaller images. It was basically useless to use version 3 to analyze small images because the over fitting problem that faced us during the training , so that the accuracy was around 21% (already tested at the same data of other algorithms). Another matter to be looked at is that the use of tiny-yolov4-coco net as it uses only 66% of the number of parameters that version 3 used to use but gives a better result which enhanced speed and accuracy .
For yolov2 it was shown that it is not quite different from yolov4 since it showed lot of similarity with it , except for the training time where yolov2 is slower and it requires many more epochs number to compute higher precision . The precision-recall curves plotted using the confidence score, AP, allowed us to deduce how well these four models perform object detection. Graphs were plotted for each model
However, of these, F R-CNN is more accurate than SSD, as discussed above, while SSD is more efficient for big size object detection , according to previous research , YOLO (with its two versions V4 and V2) is clearly the most efficient of all.

algorithm	AP	Number of epochs	Training time
Yolov4	0.86	30	14h
Yolov2	0.85	60	35h
Faster R-cnn	0.66	100	60h
SSD	0.42	90	72h

Table 2.2: Comparaison between the different architectures

2.8 Conclusion

This chapter compared the latest and most advanced CNN-based object detection algorithms. Without object detection, it would be impossible to analyze the hundreds of thousands of images that are uploaded to the internet every day . Technologies like self-driving vehicles that depend on real-time analysis are also impossible to realize without object detection.
All the networks were trained with the same data-set extracted from the FLAME data-set , to ensure a homogeneous baseline. It was found that Yolo-v4 is the fastest with yolov2 following closely, than Faster RCNN in the third place and SSD coming in the last place. However, it can be said that the use case influences which algorithm is picked; if you are dealing with a relatively small data-set and don't need real-time results, it is best to go with Faster RCNN.
Yolo-v4 is the one to pick if you need to analyses a live video feed as it is fast and ore efficient . Meanwhile, SSD provides a bad balance between speed and accuracy. Additionally, Yolo-v4 is the most recently released of the four and is actively being contributed to by the vast open-source community.
Hence, in conclusion, out of the four Object Detection Convolutional Neural Networks analyzed, YoloV4 shows the best overall performance. This result is similar to what some of the previous reports have obtained. A great deal of work can still be done in the future in this field. Also, each field—aviation, autonomous vehicles (aerial and terrestrial), industrial machinery, etc. are suited to different algorithms. These subjects can be explored in detail in the future.

Chapter 3

Localization using stereo vision

3.1 Introduction

The camera is one of the most essential tools in computer vision. It is the mechanism by which we can record the world around us and use its output photographs.
Camera calibration is frequently used word in image processing or computer vision field. The camera calibration method is intended to identify the geometric characteristics of the image creation process. This is a vital step to perform in many computer vision applications, especially when metric information on the scene is needed. The camera is often categorized on the basis of a set of intrinsic parameters such as skew of the axis, focal length, main point in these applications, and its orientation expressed by extrinsic parameters such as rotation and translation. Linear or nonlinear algorithms are used to estimate intrinsic and extrinsic parameters utilizing known points in real-time and their projections in the picture plane.
in this chapter We will discuss camera geometry in more detail. Particularly, we will outline parameters that are important to several key computer vision tasks and must be computed (calibrated) using approaches that we will discuss.

The flowchart bellow represents the procedures we followed through the chapter organisation

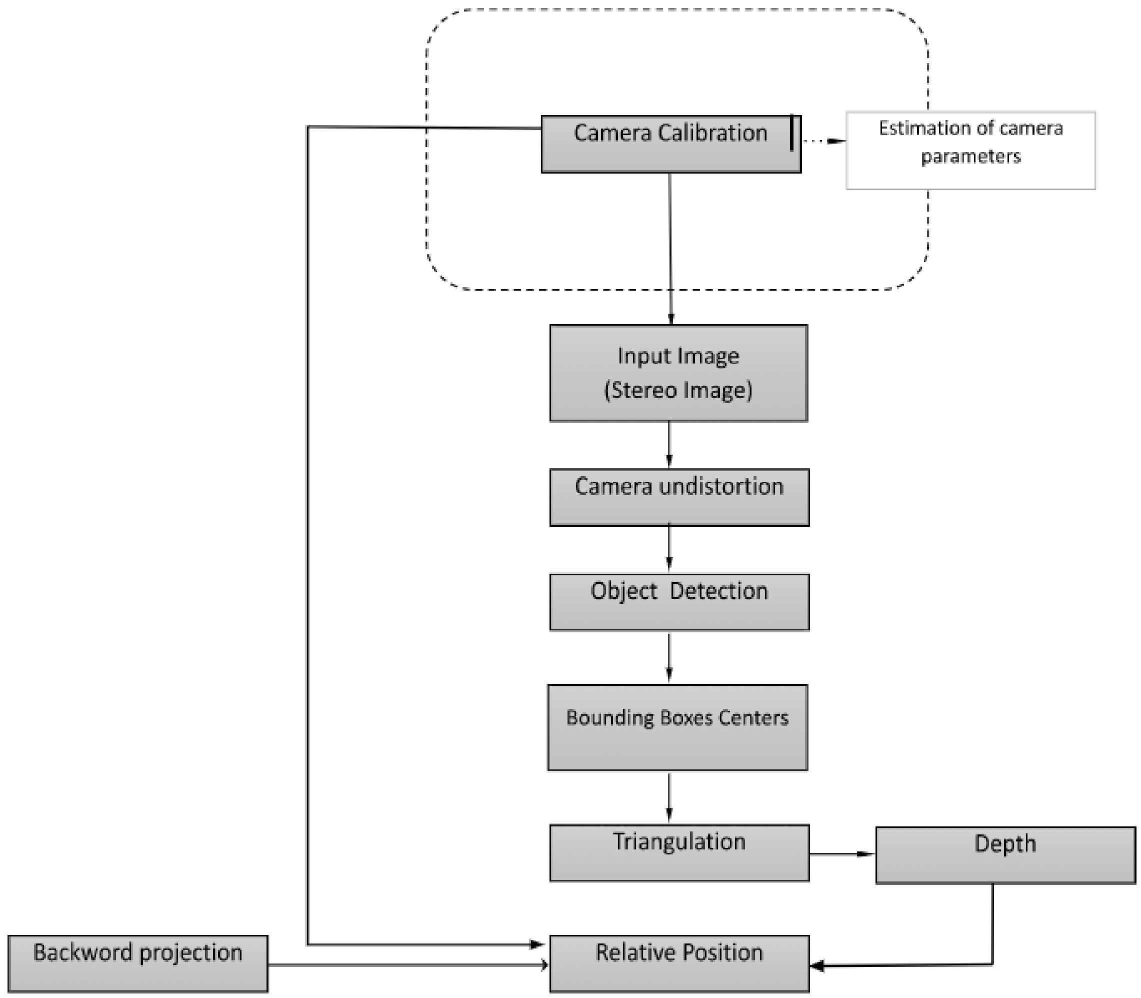


Figure 3.1: Diagram representing a description of localization procedure

3.2 Description of localization procedure

As a summary of our localization approach , we mention the important steps performed : As the main step to an accurate localization , camera calibration process should be performed carefully respecting several condition to extract camera parameters including intrinsic and extrinsic matrices .
The pair images collected from stereo system created by two identical USB cameras , will be processed by undistortion algorithm identified by MATLAB , this aims to have a clear , net and the closest image model to reality .
An object detection algorithm developed in Chapter 2 was used in order to identify "fire " and to compute bounding boxes coordinates . After defining the bounding boxes in each image , the coordinates of each center of the bounding boxes are calculated and extracted in order to be used with the triangulation formula that computes depth (in our case the depth was calculated relatively to the first camera) .
The matter of getting the exact position of the object in real world coordinates is not an easy task , therefor , we had to improvise with some condition mentioned in part 3.7.4 , that allow us to extract simplified equations of perspective projection which calculates (X, Y, Z) coordinates of the object relatively to the camera.

3.3 Definitions

Definition 1 *Frame of reference: a measurements are made with respect to a particular coordinate system called the frame of reference.*

Definition 2 *World Frame: a fixed coordinate system for representing objects (points, lines, surfaces, etc.) in the world.*

Definition 3 *Camera Frame: coordinate system that uses the camera center as its origin (and the optic axis as the Z-axis).*

Definition 4 *Image or retinal plane: plane on which the image is formed, note that the image plane is measured in camera frame coordinates (mm).*

Definition 5 *Image Frame: coordinate system that measures pixel locations in the image plane.*

Definition 6 *Intrinsic Parameters: Camera parameters that are internal and fixed to a particular camera/digitization setup.*

Definition 7 *Extrinsic Parameters: Camera parameters that are external to the camera and may change with respect to the world frame.*

3.4 Pinhole Camera

Definition 8 *A pinhole camera is a simple camera without a lens and with a single small aperture. Light rays pass through the aperture and project an inverted image on the opposite side of the camera (14).*

The pinhole camera model is the basic camera model used in computer vision. Its name originates from the concept of pinhole camera and it models perspective projections. Actually the idea of pinhole projection and pinhole camera dates back to 500 BC, there were Chinese philosophers who were writing about this concept way back then.And somewhere around 1000 AD, the Arab physicist Alhazen, he wrote a book called Kitab al-Manazir one of the first optic books, he describe, in great deal this concept of the pinhole camera. in the figure above 3.2is a sketch by Gemma Frisius, the Dutch mathematician where we can see the first illustration of the pinhole camera model. (14)

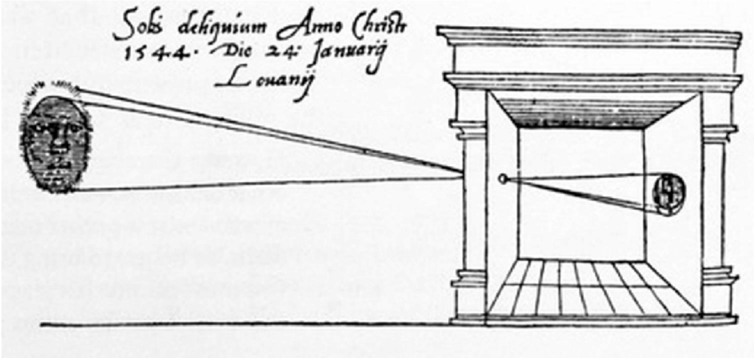


Figure 3.2: First illustration of pinhole camera model

3.4.1 principle of the pinhole camera model

It's a system that can record an image of an object or scene in the 3D world. This camera system can be designed by placing a barrier with a small aperture between the 3D object and a photographic film or sensor. As Figure shows 3.3, each point on the 3D object emits multiple rays of light outwards. Without a barrier in place, every point on the film will be influenced by light rays emitted from every point on the 3D object. Due to the barrier, only one (or a few) of these rays of light passes through the aperture and hits the film. Therefore, we can establish a one to-one mapping between points on the 3D object and the film. The result is that the film gets exposed by an "image" of the 3D object by means of this mapping (14).

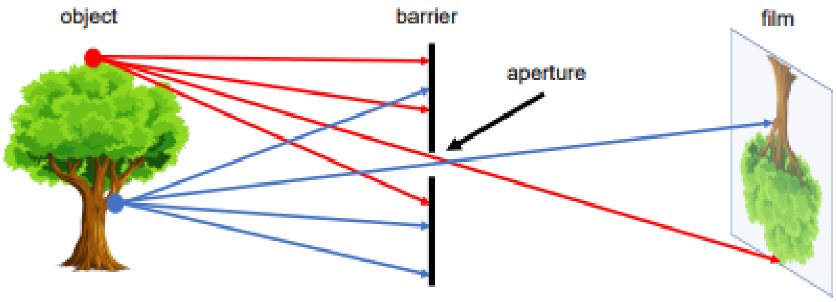


Figure 3.3: Working of a pinhole camera

A more formal construction of the pinhole camera is shown in Figure. In this construction, the film is commonly called the image or retinal plane. The aperture is referred to as the pinhole O or center of the camera. The distance between the image plane and the pinhole O is the focal length f.

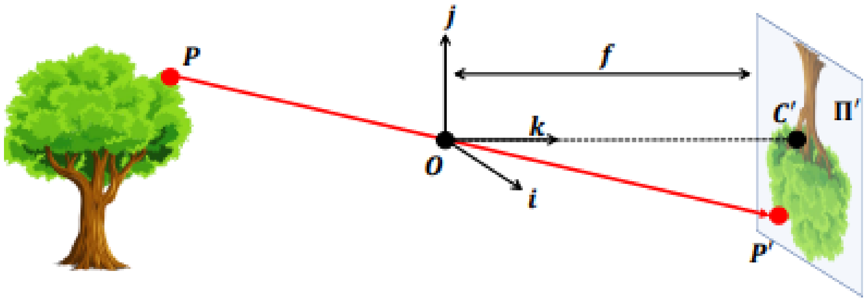


Figure 3.4: A formal construction of the pinhole camera model

The two most important parameters in a pinhole camera model:

- a. Focal length: the distance between the pinhole and the image plane it affects the size of the projected image and the camera focus when using lenses.
- b. Camera center: The coordinates of the center of the pinhole.

3.4.2 Perspective projection

Coordinate frames

Definition 9 *In geometry, a coordinate system is a system that uses one or more numbers, or coordinates, to uniquely determine the position of the points or other geometric elements on a manifold (29)*

In our chapter we have 3 coordinate systems:

- The coordinate system (c x y) for the image plane is defined such as the origin is at the point "c".
- The coordinate system (C X Y Z) for the three dimensional space as indicated where the origin Is at the point "C" point , this system is called the standard coordinate system of the camera. The camera coordinate system is a 3D Cartesian coordinate system with the origin at the focus point of the camera and the Z-axis is this optical axis of the camera.

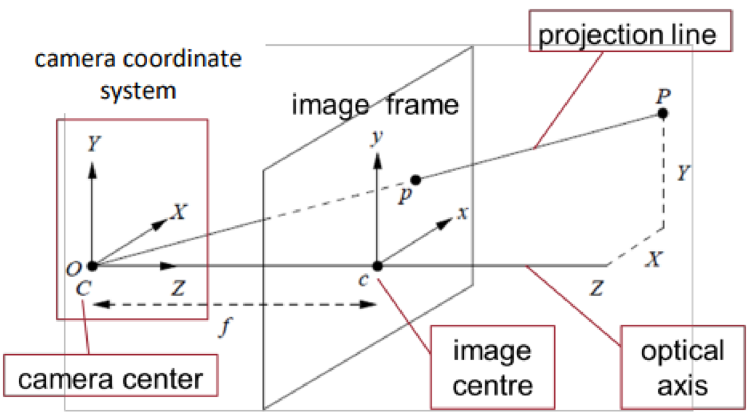


Figure 3.5: Camera and image coordinate frames

Also we have:

- Pixel Coordinate System : Pixel coordinates are just the position of each pixel in an image from the top left corner.

From 3D to 2D: From the above definition of the camera and image coordinate system, it is clear that the relationship between 2D image coordinate and 3D space coordinates can be written as the equation that follows.
Select a coordinate system (X,Y,Z) for the three-dimensional space to be imaged

- Let (x,y) be the retinal plane
- Then, the two are related by:

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \tag{3.1}$$

Let $P = (XYZ)^T$ be a point on some 3D object visible to the pinhole camera. P will be mapped or projected onto the image plane , resulting in point $p = (xy)^T$

$$(X, Y, Z)^T \to (x, y)^T = (\frac{fX}{Z}, \frac{fY}{Z}) \tag{3.2}$$

the above equation represent the simplest form of perspective projection
$(X, Y, Z)^T$: Euclidean Coordinates 3D World frame.
$(x, y)^T$: Euclidean coordinates 2D Image plane.
Which is written linearly in homogeneous coordinates as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.3}$$

From 2D(mm) to 2D(pixel): The effect we must account for that the points in digital images are expressed in pixels, while points in image plane are represented in physical measurements (e.g. centimeters). In order to accommodate this change of units, we must introduce two new parameters S_x and S_y. These parameters, whose units would be something like $\frac{pixels}{mm}$, correspond to the change of units in the two axes of the image plane. Note that S_x and S_y may be different because the aspect ratio of a pixel is not guaranteed to be one.

If $S_x = S_y$, we often say that the camera has square pixels. We adjust our previous mapping to be:

$$x = S_x * f(mm) * \frac{X(mm)}{Z(mm)} \tag{3.4}$$

$$y = S_y * f(mm) * \frac{Y(mm)}{Z(mm)} \tag{3.5}$$

The camera sensor have S_x pixels along the horizontal and S_y pixels along the vertical direction, therefore the units of S_x and S_y are $\frac{pixels}{mm}$

The Second parameters, c_x and c_y, describe how image plane and digital image coordinates can differ by a translation. Image plane coordinates have their origin (c) at the image center where the Z axis intersects the image plane. On the other hand, digital image coordinates typically have their origin at the lower-left corner of the image. Thus, 2D points in the image plane and 2D points in the image are offset by a translation vector $(c_x, c_y)^T$. therefore,

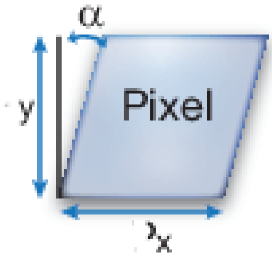


Figure 3.6: pixel illustration

$$x = S_x * f * \frac{X}{Z} + c_x \tag{3.6}$$

$$y = S_y * f * \frac{Y}{Z} + c_y \tag{3.7}$$

putting it all together, the final perspective projection model is:

$$x = f_x * \frac{X}{Z} + c_x \tag{3.8}$$

$$y = f_y * \frac{Y}{Z} + c_y \tag{3.9}$$

with: $S_x * f = f_x$
$S_y * f = f_y$
writing the model in a matrix form :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = p = K * P \tag{3.10}$$

3.5 Camera calibration

To precisely know the transformation from the real, 3D world into digital images requires prior knowledge of many of the camera's intrinsic parameters. If given an arbitrary camera, we may or may not have access to these parameters. We do, however, have access to the images the camera takes. Therefore, can we find a way to deduce them from images? This problem of estimating the extrinsic and intrinsic camera parameters is known as camera calibration.

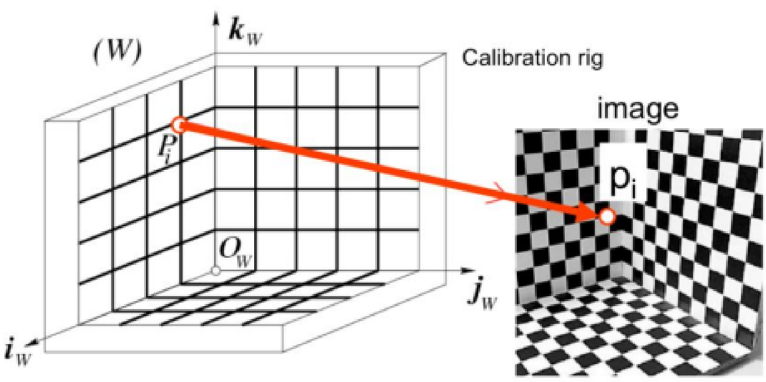


Figure 3.7: The setup of an example calibration rig

3.5.1 Camera calibration method

The calibration models for machine and computer vision have traditionally employed reference grids, the calibration matrix K being determined using images of a known object point array (e.g. a checkerboard pattern). Commonly adopted methods are those of Tsai, (1987), Heikkila Silven (1997) and Zhang (2000). These are all based on the pinhole camera model and include terms for modelling radial distortion (34).
in this chapter we are going to use Zhang method , it requires a planar checkerboard grid to be placed at different orientations (more than 2) in front of the camera. The developed algorithm uses the extracted corner points of the checkerboard pattern to compute a projective transformation between the image points of the n different images, up to a scale factor. Afterwards, the camera interior and exterior parameters are recovered using a closed-form solution, while the third- and fifth-order radial distortion terms are recovered within a linear least-squares solution.

3.5.1.1 Plane based camera calibration:

The MATLAB APP used to perform stereo camera calibration with this method, the idea is to present the camera for a set of images of a flat surface.
The most common method is to use a checkerboard as a flat surface. It is important that the camera or the checkerboard is fixed in space .
The camera characteristics are the same for all images, only the orientation and position, R and t, of the camera change. Knowing the size and structure of the 2D pattern, in this case the checkerboard, we can define a coordinate system on the checkerboard that describes where each corner is in the real world.
The origin of the world coordinate system is the upper left corner of the plane, the X and Y directions move to the right and down respectively. and Z is orthogonal to the plane. To map known world points to image points, it is common to use a corner detector to identify the corners of each square in the image, and then associate each corner with the corresponding world point. Since the origin of the world coordinates is set to be in the upper left corner of the checkerboard, the plane is fixed at Z = 0.

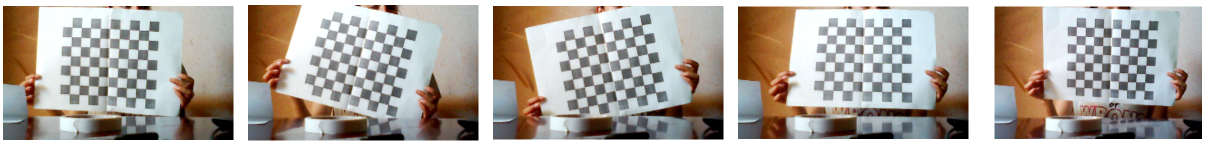


Figure 3.8: multiple translation and orientations images of the check board from right camera

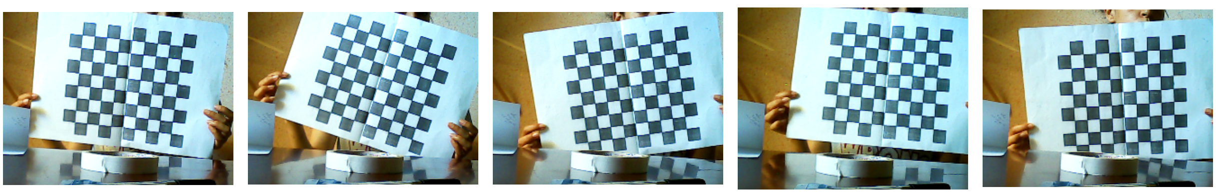


Figure 3.9: multiple translation and orientations images of the check board from left camera

This means that any point in the plane has the coordinates :

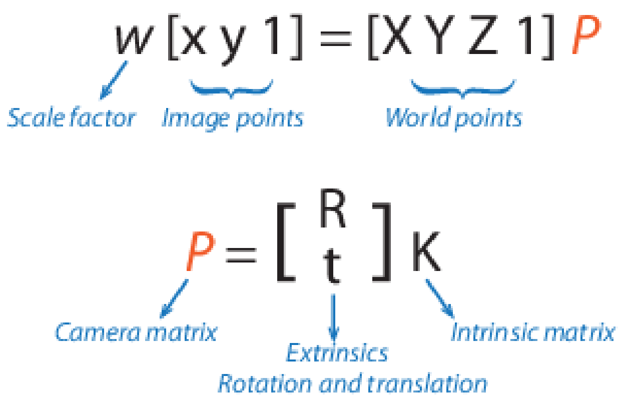


Figure 3.10: Camera and image coordinate frames

Where is the planar homograph that maps points on the checkerboard to the corresponding pixel points in the image.

3.5.1.2 Camera calibration matrix

The perspective transformation provides the relationship between the position of target point of interest in camera frame (Pixel coordinates) with its location in inertial frame (3D world coordinates). But this accounts only the external parameters (positional orientation) not the internal characteristics of the camera, which also defines where a 3D real world points would be mapped in the camera image frame. The internal characteristics of a camera can be defined Camera Calibration Matrix (C_c) as mentioned below:

$$k = \begin{bmatrix} f_x & f_\theta & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

where, C_c is the Camera Calibration Matrix, f_x and f_y is the focal length measured along width pixel and height pixel of image plane in meters, f_θ is the skew of the pixel in meters, c_x and c_y is the principal point of the camera in pixels.

3.6 Camera parameters

The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system.
The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates. The extrinsic parameters of a camera depend on its location and orientation and have nothing to do with its internal parameters such as focal length, the field of view, etc. On the other hand, the intrinsic parameters of a camera depend on how it captures the images. Parameters such as focal length, aperture, field-of-view, resolution, etc govern the intrinsic matrix of a camera model.

3.6.1 Intrinsic and Extrinsic parameters

$$k = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

K only depends on the intrinsic camera parameters like its focal length, principal axis and thus defines the intrinsic parameters of the camera.

Now if the camera does not have its center of projection at (0, 0, 0) and is oriented in an arbitrary fashion (not necessarily z perpendicular to the image plane), then we need a rotation and translation to make the camera coordinate system coincide. Let the camera translation to origin of the XYZ coordinate be given by T(T_x, T_y, T_z). Let the the rotation applied to coincide the principal axis with Z axis be given by a 3 × 3 rotation matrix R. Then the matrix formed by first applying the translation followed by the rotation is given by the 3 × 4 matrix

$$p = K(R|T) * P \tag{3.13}$$

These parameters R and T are known as the extrinsic parameters because they are external to and do not depend on the camera.
This completes the mapping from a 3D point P in an world reference system to the image plane. To reiterate, we see that the full projection matrix consists of the two types of parameters introduced above:
intrinsic and extrinsic parameters.
All parameters contained in the camera matrix K are the intrinsic parameters, which change as the type of camera changes.
The extrinsic parameters include the rotation and translation, which do not depend on the camera's build. Overall, we find that the 3 × 4 projection matrix M has 11 degrees of freedom: 5 from the intrinsic camera matrix, 3 from extrinsic rotation, and 3 from extrinsic translation.

3.7 Depth retrival

Calculating the distance of various points in the scene from the camera position is an important task for a computer vision system. A common method of extracting this depth information from intensity images is to acquire a pair of images using two cameras offset from each other by a known distance. Alternatively, two or more images taken by a moving camera can also be used to compile depth information.
In contrast to intensity images, images in which the value of each pixel is a function of the distance from the corresponding point are called distance images.
These images are acquired directly using remote imaging systems. Two of the most commonly used principles for obtaining such distance images are radar and triangulation. In our case, triangulation was used and this method is described in this chapter.

3.7.1 Stereo Imaging

The geometry of the binocular stereo is shown in Figure 3.12 . The simplest model is that of two identical cameras separated only in the x direction by a baseline distance b. The image planes are co-planar in this model. One element of the scene is seen by the two cameras at different positions in the image plane.

The offset between the positions of the two elements in the image plane is called the disparity.

The plane passing through the centres of the cameras and the point of the scene element is called the epipolar plane. The intersection of the epipolar plane with the image plane defines the epipolar line. In the model shown in the figure, each element in one image is on the same line in the second image.

In practice, there may be a vertical mismatch due to poor registration of the epipolar lines. Many formulations of binocular stereo algorithms assume zero vertical disparity.

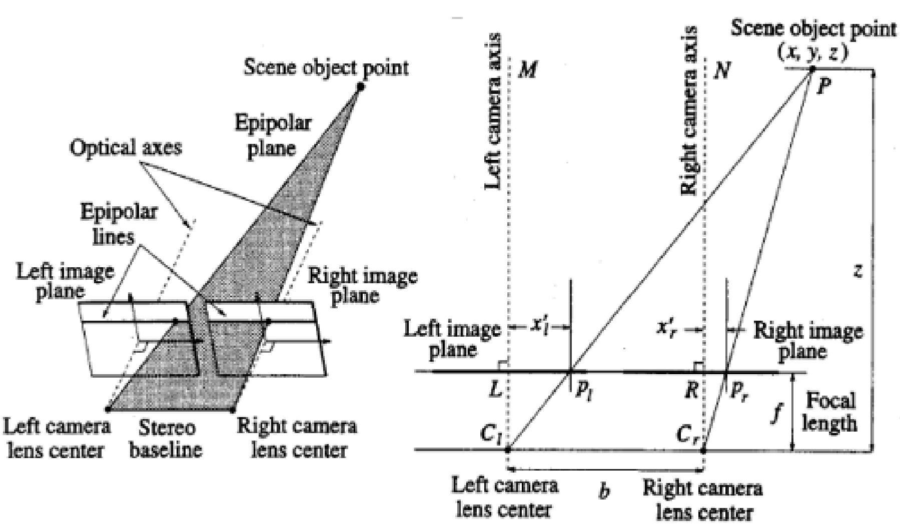


Figure 3.11: Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images, called a conjugate pair. the displacement between the position of the two points is called the disparity

As at least two images are taken, with an offset from the baseline, the object whose depth we want to know will have moved in the different images. The relative change in the position of the object in the image plane, assuming we have a calibrated camera and can restore the image coordinates to the Coordinates to real-world coordinates, gives us a clue to the depth.

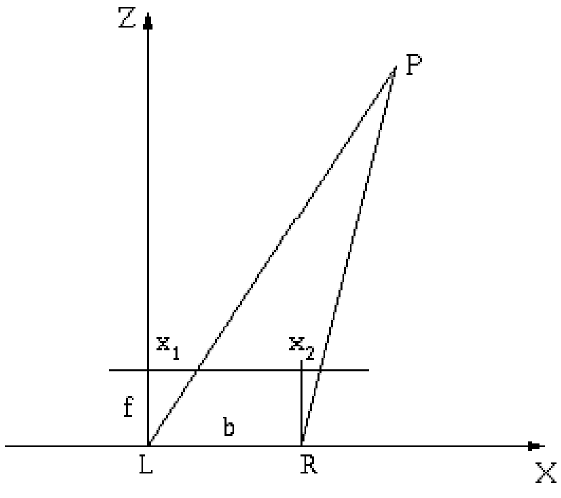


Figure 3.12: Stereo camera geometry: L and R represent the left and right camera respectively, f is the focal length, b the baseline width of the camera placement and P is a common point in both images

From the previous figure, the formulas for stereo triangulation can be deduced. With a point P located on the image coordinates (x_1, y_1) and (x_2, y_2) respectively in two images, and XY being the plane of the world reference system parallel to the image plane of the cameras and XZ being the plane where the optical axes are located, we have the following relationships:

$$Z = \frac{b}{x_1 - x_2} \tag{3.14}$$

$$X = x_1 \frac{Z}{f} \tag{3.15}$$

$$Y = y_1 \frac{Z}{f} \tag{3.16}$$

This suggests that objects with a large offset in the horizontal direction will be closer and objects with a small offset will be further away, since Z increases as the x1x2 denominator decreases, the depth being in direct correlation with the disparity between two images.

This makes the problem of depth determination essentially a problem of finding the disparity between the two images, so that we can then calculate the distance to each object, or even a point, using the above relationships.

The problem can then be redefined as a task of finding the correct correspondence of the object from one camera to another. Early depth reconstruction methods were based on the extraction of a set of features.

These features were identified using a kernel that reliably detected, for example, corners, which could then be matched in the corresponding image.

3.7.2 Image Undistortion

Almost every image we see is distorted in some way. It is especially when using lenses that distortions become clearly visible.
This is the case when you put on another person's glasses, when you look through a peephole or a fish eye in a doorway, or when you are near a webcam. Professional camera lenses are optimised to minimise these effects, but small, cheap lenses or lenses designed for purposes other than imaging (such as the laser focus lens in the integrated welding head) often suffer from severe distortion.

To generate clear and sharp images, the diameter of the aperture (hole) of a pinhole camera should be as small as possible. If we increase the size of the aperture, we know that the rays coming from several points of the object will be incident on the same part of the screen, thus creating a blurred image.

On the other hand, if we reduce the size of the aperture, only a small number of photons hit the image sensor. As a result, the image is dark and noisy.
So, the smaller the aperture of the pinhole camera, the sharper the image, but also the darker and noisier. With a larger aperture, the image sensor receives more photons (and therefore more signal). This results in a bright image with a low amount of noise.

By using a lens, we get better quality images, but the lens introduces distortion effects. There are two main types of distortion effects: Radial distortion: This type of distortion would be explained in the part below , There are two type of radial distortion effect

- 1. Barrel distortion effect, which corresponds to negative radial displacement.
- 2. Pincushion distortion effect, which corresponds to a positive radial displacement.
- 3. Tangential distortion: This usually occurs when image screen or sensor is at an angle with reference to the lens. Thus the image seem to be tilted and stretched.

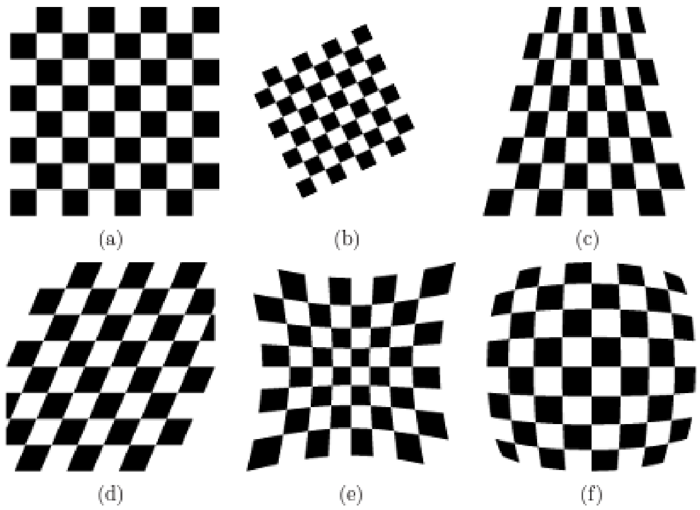


Figure 3.13: Several information:
a)Original image
b)linear conformal
c) projective/ perspective
d)affine/skew
e)Radial:pincushion
f)Radial:Barrel(Negative pincushion distortion

The main distortions in an image are radial distortions, where pixels are moved from or towards the optical centre by an amount related to their distance from that centre. Radial distortions cause straight lines in the real world to be displayed as a curve in the camera image, as shown in the figure below.
Fitting a straight line through a curve will give inaccurate results, as the fitted line will always deviate from the line in the image in several places. Another problem caused by radial distortions is that two objects in the centre of the image may be 10 pixels apart, while two other objects, equally far apart in the real world, may appear to be 15 pixels apart when placed at the edge of the image.
The figure below illustrates this phenomenon: the equally sized squares in the figure below appear larger at the edges of the image after distortion. This means that a line at the edge is not only curved, it will also be offset. It is therefore important to correct any radial distortion in the images before proceeding with feature extraction, such as joint detection.

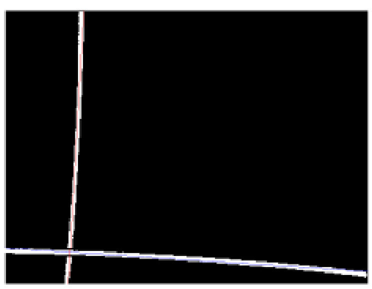


Figure 3.14: The distorted image shows curved lines, which are difficult to fit accurately

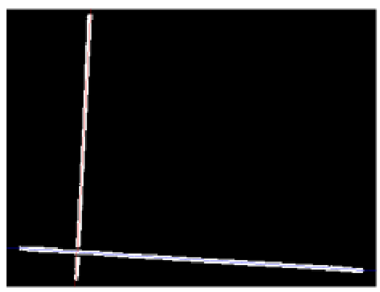


Figure 3.15: After the images is undistorted the lies are straight again which makes accurate fitting possible

Figure 3.16: The problem of extracting features from radially distorted images

3.7.3 Image Undistortion Algorithm

The image warping (undistortion) algorithm matches the pixel locations of the undistorted output image to the pixels of the distorted input image using an inverse mapping technique. This diagram shows the steps in the algorithm. No need for bi-linear interpolation

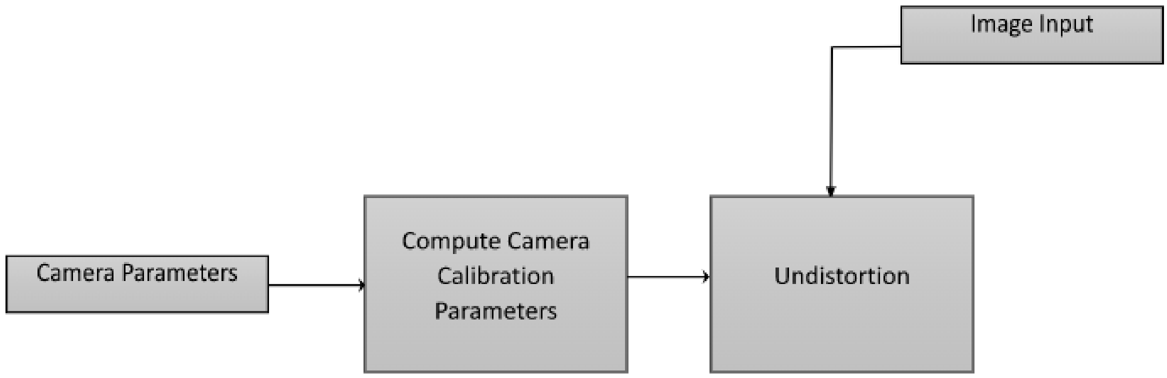


Figure 3.17: diagram shows steps in the algorithm

.

As explained in the second part of this chapter , Camera calibration estimates the lens and image sensor parameters of a video or image camera. These parameters can be used to correct lens distortion, measure the size of an object in world units or determine the location of the camera in the scene. Applications such as machine vision, robotic navigation systems and 3D scene reconstruction use these operations to detect and measure objects. Camera parameters include intrinsics, extrinsics and distortion coefficients. This step calculates these parameters from a given camera parameters object as input and passes them to the distortion cancellation step.

This step removes the distortion using the distortion coefficients and the intrinsic matrix of the camera. These equations model the removal of distortion. Let (u,v) be the coordinates of the input camera image and (x,y) the locations of the undistorted pixels. Normalise x and y from the pixel coordinates by moving them to the optical centre and dividing by the focal length in pixels, which was first explained in the camera model at the beginning of this chapter. Note that , After computing the Camera Calibration parameters we have noticed that the values of tangential coefficients are equivalents to zero in which it will simplify the whole undistortion process .

$$U_{radial} = x(1 + K_1 r^2 + K_2 r^4) \tag{3.17}$$

$$V_{radial} = y(1 + K_1 r^2 + K_2 r^4) \tag{3.18}$$

Where :

$$r^2 = x^2 + y^2 \tag{3.19}$$

K_1 , K_2 are radial distortion coefficients resulted from the camera calibration,Then, calculate the final coefficient values by combining the radial components. $u = u_{radial}$ $v = v_{radial}$ Undistortion defines a mapping between the coordinates of the output undistorted image (u, v) and the coordinates of the distorted input camera image, (x, y).

3.7.4 3D position estimation

The final goal in our Thesis is to learn how to calculate real world coordinates of an object recognized by neural network . In Computer Vision, there are many ways to find the 3D location of an object like using a Lidar, Radar, etc. also , it is possible to achieve 3D perception with a stereo camera .
There are conditions to find the 3D location using stereo camera where are ,the size of the object in the picture whose position needs to be estimated should be known and the distance between the camera and the object should be estimated . The basic of this approach is the inverse of perspective projection explained earlier in the first part of this chapter there is how it works:

- We calculate the parameters that relates between the 3D world in the Camera coordinate system and the 2D pixel coordinates in the image. It is the result of camera calibration, more specifically intrinsic parameter calculation.
- Then we use an object segmentation algorithm to find the object of interest.
- We estimate the depth using stereo vision .
- We use the depth and the pixel position of the center of the object to estimate the 3D position using the intrinsic parameters obtained in step 1.

Before proceeding any further actions , we need to put on some assumption to make the process of finding the position more accurate and understandable .

- Center of projection is the same as the origin of the world
- The camera axis is aligned with the Z axis
- We assume that the image plane is in front of the center of projection .
- The camera coordinates system and world coordinates system share same axis. Those assumption were made in order to simplify the derivation of perspective projection equations.

Mapping between image plane and camera plane: From equations (3.1), (3.2) above we get:

$$x_{camera} = \frac{u - c_x}{f_x} \tag{3.20}$$

$$y_{camera} = \frac{v - c_y}{f_y}$$

Mapping between camera plane and real world coordinates:

$$X_{world} = x_{camera} \times Z \tag{3.21}$$

$$Y_{world} = y_{camera} \times Z$$

$$Z_w = Z$$

With Z is the depth. (u ,v) are the coordinates of the center of the bounding box in pixel, (f_x, f_y) are the components of the focal length,(c_x, c_y) are the coordinates of the principle point .

3.8 Experimental results

3.8.1 Calibration process

The calibration process is based on a MATLAB Application which is Stereo-vision Camera calibrator that uses OpenCV library in addition to other libraries which are widely used to manage complex mathematical operations, and to perform special functions on single or multi-dimensional arrays and to manage and store the data retrieved from the various calibration processes, while OpenCV library is used to perform the core of the calibration process, this library is a very powerful computer vision library including various functions related to calibration processes and different tools that help during the development.
Also Matlab toolboxes (25) (26) have been used during this thesis, this software provides two simple and easy-to-use app designed to help and improve speed of the calibration process (both intrinsic and extrinsic). The results obtained using Matlab camera calibration toolbox have been shared in further parts .

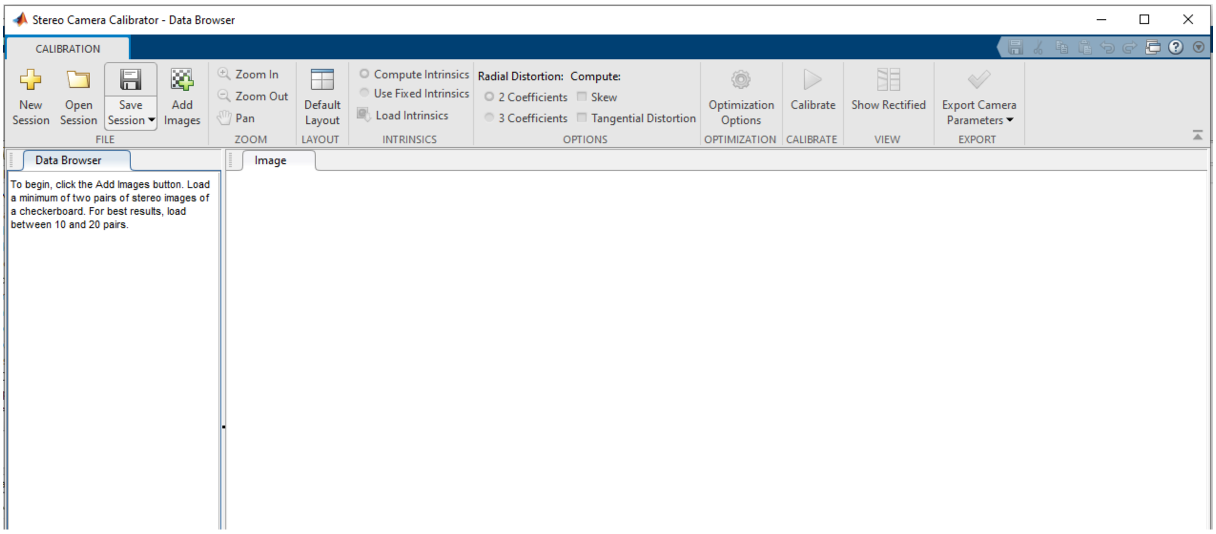


Figure 3.18: Stereo camera calibration window

Our stereo system is composed of two identical cameras , which means that they do share same specifications , separated from each other with fixed distance .
As a first step , the choice of calibration images was mentioned earlier in Camera calibration Part , as it is shown in figure below 3.19 , the dimension of chessboard squares must be known (in our case it was 28 mm) as an input for camera calibration

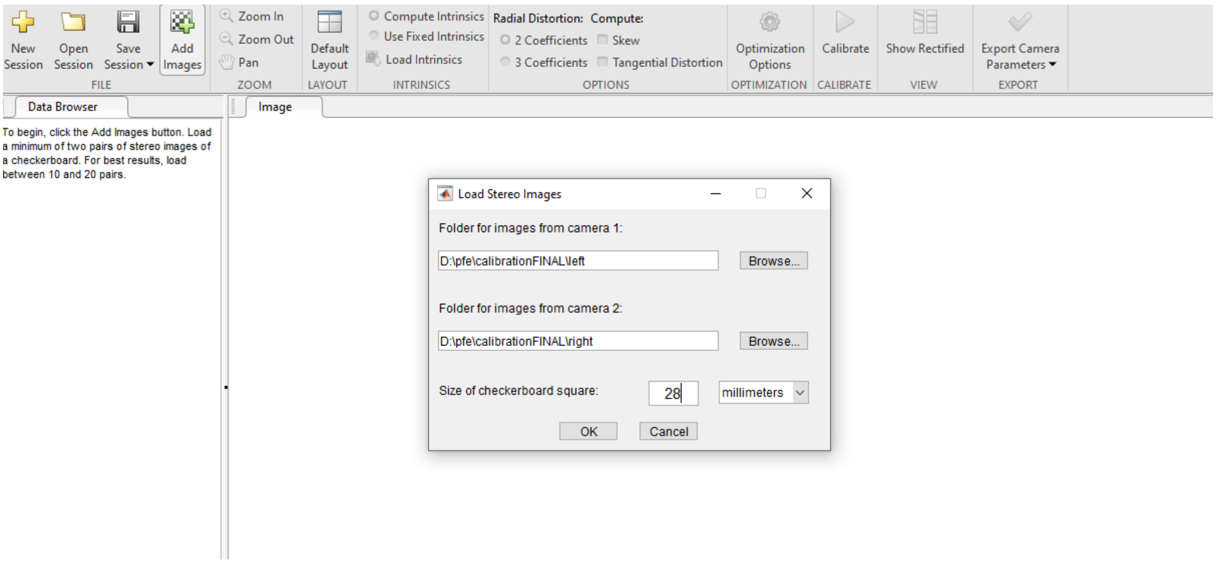


Figure 3.19: Stereo camera calibration window

After choosing images , the origin of the chessboard and X,Y directions would be specified automatically , after that we must export the camera parameters to the workspace of MATLAB in order to use it for the object Localization.

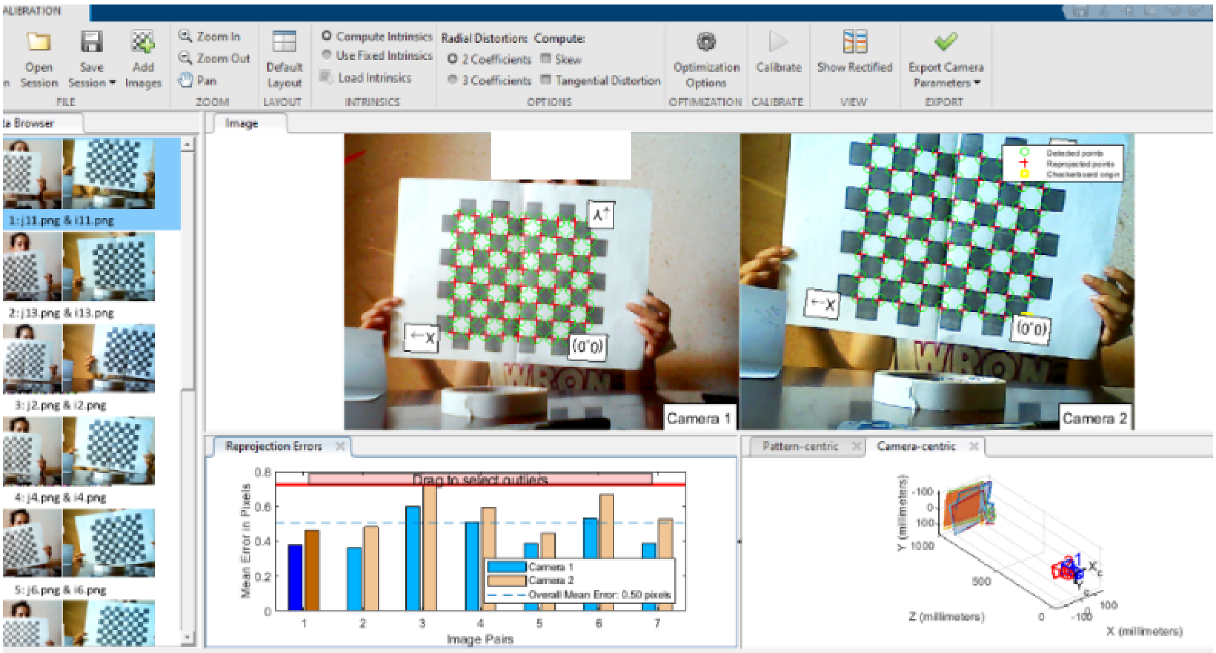


Figure 3.20: Calibration process

From the simulation results, camera parameters including focal length, principal point, radial distortion, mean projection error and intrinsic parameters matrix are calculated. The focal length, principal point are stored in a 2 ×1 vector and radial distortion is stored in 3×1 .

The intrinsic parameters are stored in the 3 × 3 matrix along with mean projection error is calculated. These parameters are enlisted in the Table 4.4 (Appendix B) .

1.Camera-centric : Camera or pattern-centric view, specified as Camera Centric or Pattern Centric. The view input sets the visualization for the camera extrinsic parameters. In our case ,we kept our camera stationary while moving the calibration pattern, that is why we set the view to 'CameraCentric' which is shown in figure below. Camera centric gives us information about the distance separating both cameras and the position of each one to the other , added to that , the distance between the camera itself and images used for calibration , which gives information about the accuracy of our calibration process.

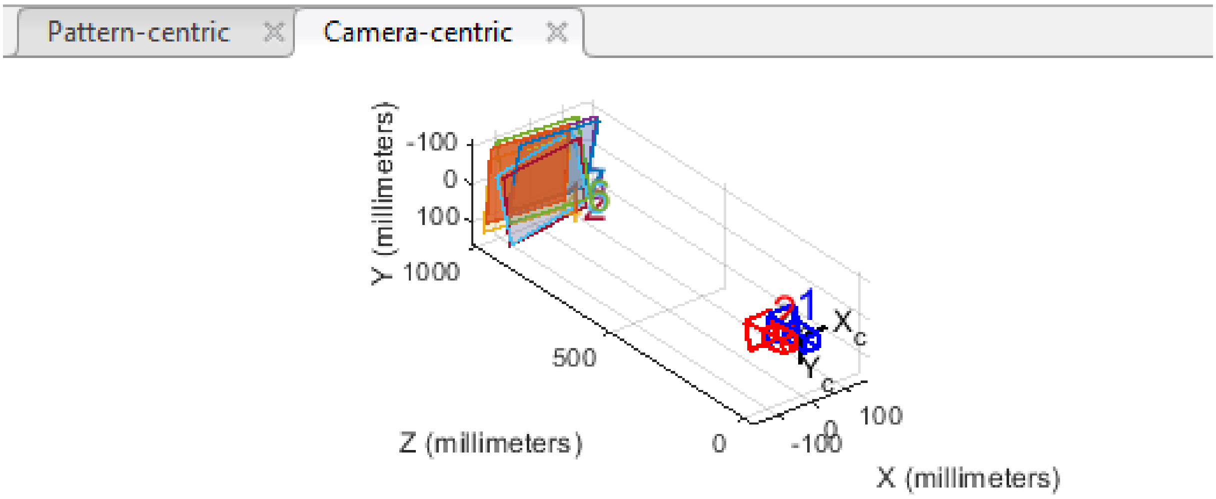


Figure 3.21: Camera centric

2. Reprojection Errors: Re-projection errors provide a qualitative measure of accuracy. A re-projection error is the distance between a pattern key point (corner points) detected in a calibration image, and a corresponding world point projected into the same image. The calibration application provides a useful visualization of the average re-projection error in each calibration image. If the overall mean re-projection error is too high, excluding the images with the highest error and re-calibrating is the most important step for reducing it . Re-projection errors depends on Camera resolution and lenses . a higher resolution with wide lenses cause higher errors and vice versa . Generally, the mean re-projection error of less than one pixel is acceptable . The mean re-projection error per image in pixel and overall mean error of selected images is shown in Figure 3.22 below.

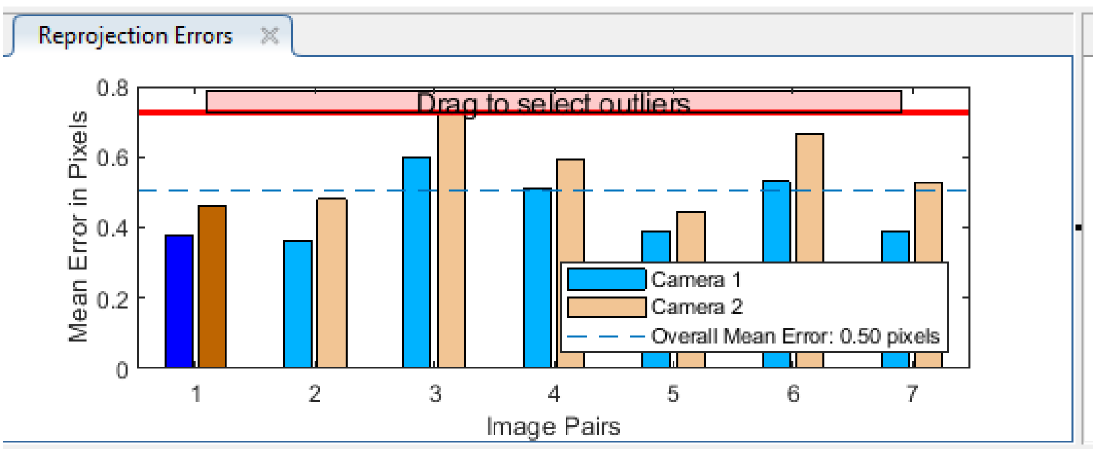


Figure 3.22: re-projection errors

3.8.2 Object detection :

Our thesis deals with specifically fire detection, taking it as an example in our process, we used the trained network mentioned before in Chapter 2 to perform fire detection which is quite simple :

Resizing the images : using the function "imresize" function is essential since the input images size captured by the stereo system is 640×480 , while the input-size of the trained network layer is 416×416 , the results is the following figure 3.23 :

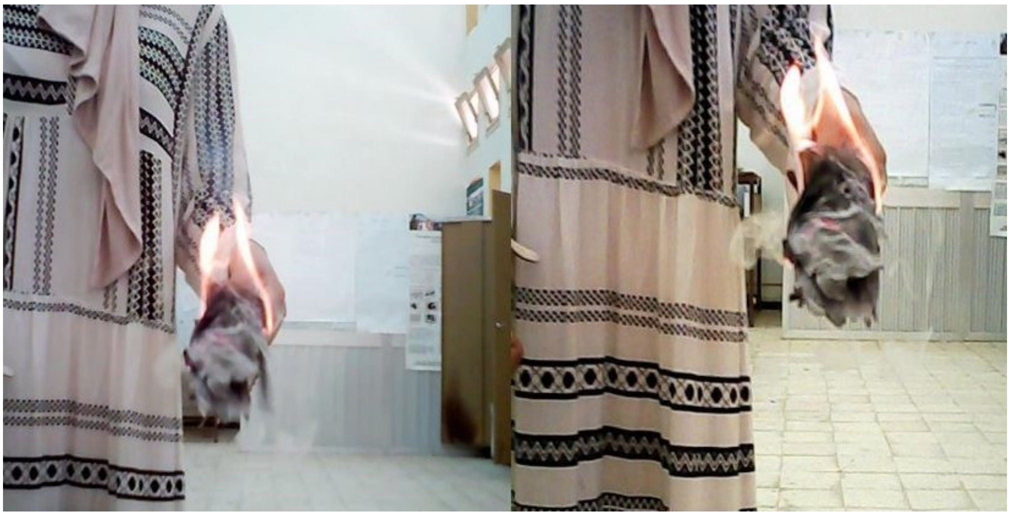


Figure 3.23: Input images from the 2 cameras

Loading the detection : Network resulted from the training , which is contained in a ".mat" file and using the detection function " DetectYoloV4" , adding to that displaying the detected object in bounding boxes as showed in figure 3.24 below:

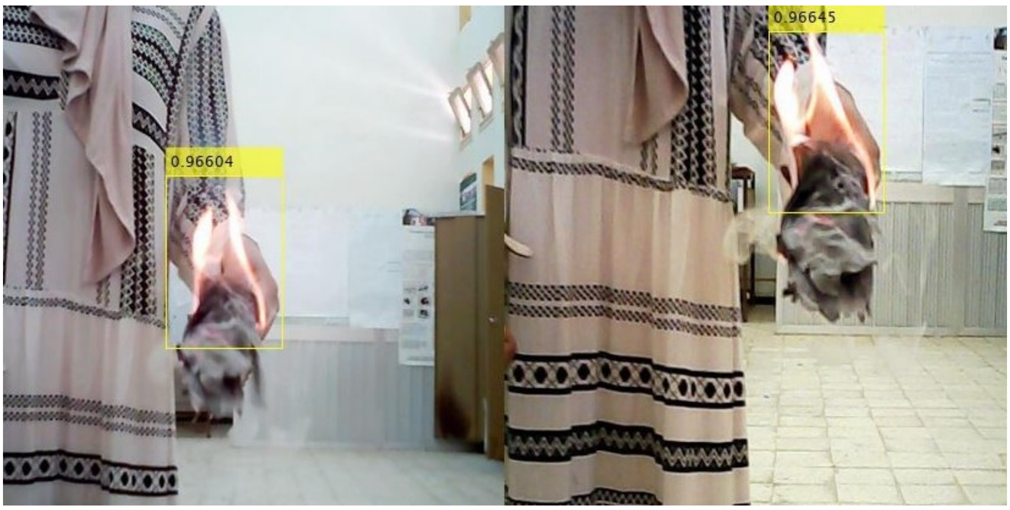


Figure 3.24: Fire detected in images adding bounding boxes

3.8.3 Depth Extraction

It passes by the following steps :

- Image undisortion In order eliminate the deformation of the image due to the lenses which would make the depth estimation more accurate. Figure 3.23 shows the input image which is taken from USB webcam. After removing lens distortion, the undistorted image and the fire detection are obtained as shown in Figure 3.24
- Computing the center coordinates (X,Y) of each bounding boxes.
- Using the "triangulate" MATLAB function to compute distance and to display the values as follow :

Figure 3.25: Distance extraction

As the picture 3.25 above shows, the result of depth is 0.56 meters, the measured real distance was 0.65 meters, the error was of 9 cm which is considered as an acceptable result due to the camera resolution, camera calibration error and camera orientation.

Real measured distance (m)	Depth result (m)
0.65	0.56

Table 3.1: Comparison between real measured distance and depth extracted

3.8.4 object relative position to the camera

After the depth extraction ,it is an easy task to calculate object relative position to the camera since the inverse of perspective projection equations are available in the pinhole camera part of this chapter.
We based our test on close position of the fire to simplify the calculation , where we tried to make the fire at the same level of the first camera so that we can estimate the approximate (X Y Z) coordinates before the calculation phase.
the results were as follow :

	Camera coordinates	World coordinates (mm)
X	-0.1621	-91.3771
Y	-0.0312	-17.5973
Z		563.7638

Table 3.2: Object relative position to the camera

As the table above 3.2 shows the results of computing the relative position of the object to the first camera, as discussed before the depth value or the Z coordinate was holding errors due to several factors, since other coordinates (X Y) are related to the depth the error is generalized to X and X , with some notes :

- The object was at the same level of the camera , which means that X coordinates should be as small as possible .
- The camera was placed in the right direction relatively to the object , which explains the minus sign in the Y coordinate , with a deviation of 15 cm .

3.9 Conclusion

Throughout this chapter, we have discussed the basic details of the stereo vision system and the importance of camera calibration throughout the process.
We have discussed the main use cases and applications as well as the conceptual functioning of the system. As we have seen previously, there are many complex challenges when using stereos for any application.
The solutions to these challenges are not magic solutions and require some research to find the solution that works best for the chosen application. Many experts have introduced ready-made stereo vision systems of reasonable accuracy to save researchers the effort of setting up a good stereo system themselves, but in our case this was not possible. So we created our personal stereo system from two USB cameras, which can be used for personal projects.
Many solutions have been developed to speed up the depth estimation process. The approach proposed in this chapter allows the triangulation method to be used with more confidence and pays attention to camera calibration and perspective projection.
This chapter has been an attempt to cover most of the fundamental concepts that govern the operation of stereo vision systems and to give an alternative to rapid depth estimation techniques. The intention was to give enthusiastic students enough information on the subject at the end of the chapter, so that they would be able to dig deeper into advanced research.

Chapter 4

Design and control of quadcopter using PIXHAWK autopilot

4.1 Introduction

Drones have numerous advantages in the twenty-first century, which is why they are becoming increasingly popular among the general public. Drones come in a variety of varieties, such as quad-copters, hexa-copters, or octa-copters based on the number of propellers utilized. In this project, quad-copters are of particular importance.
A quad-copter is a one-of-a-kind Unmanned Aerial Vehicle (UAV) that can take off and land vertically. Which is often referred to as a quad-rotor. Due to its basic structure, it is one of the most often used designs for small UAVs.

In this chapter, a quad-copter is built and controlled remotely from the ground using the PIXHAWK autopilot. We will also define and demonstrate all of the essential components for a successful UAV building, including common parts and electrical components, as well as technical explanations and pre-flight calibrations.

Figure 4.1: Quadcopter

4.2 System description

Any project involving an autonomous vehicle will typically include the following parts:

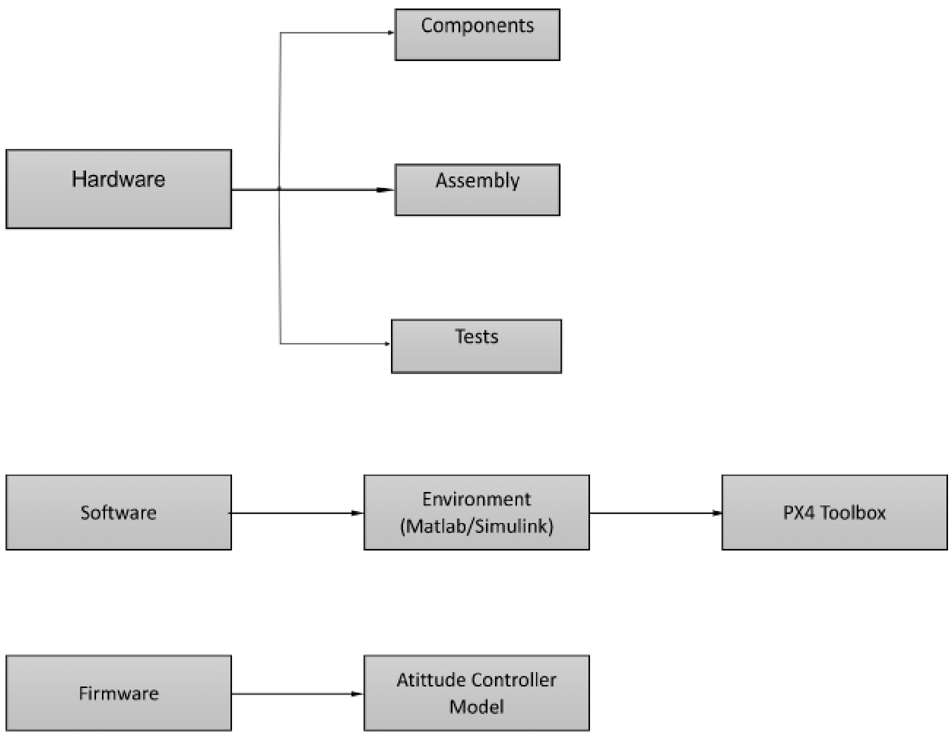


Figure 4.2: The process of quadcopter design and control

Hardware: a set of sensors, controllers, and output devices that enable the drone to perceive its environment and take appropriate action in light of the circumstances. It essentially consists of every part that makes up the drone physically.

Firmware: the code that runs on the controller and ensures dependability, stability, and features by exploiting sensor inputs, elaborating them through a processing unit, and delivering outputs to various peripherals like as ESCs, servos, and so on. The firmware enables the integration of hardware components.

Software: This is the controller's user interface, also known as the Ground Control Station (GCS). Both PCs and mobile devices can run software. A GCS enables users to set-up, configure, test, and fine-tune the vehicle configuration by simply and directly acting on the firmware. Autonomous mission planning, execution, and post-mission analysis are possible with advanced packages or add-ons. The operator often uses this to program the mission.

4.3 Hardware components of a Quad-copter

4.3.1 Frame

The frame is in charge of linking all of the subsystems and preserving physical integrity. The frame is also what holds the quad-rotor's components together. Arms, landing gears, rotor mounts, and center plates are all included. Most center plates provide two functions. It's where the landing gears and arms connect, as well as the power distribution wiring that powers the remaining components. The F450 frame employed within this project has a central plate that power all the ESCs.

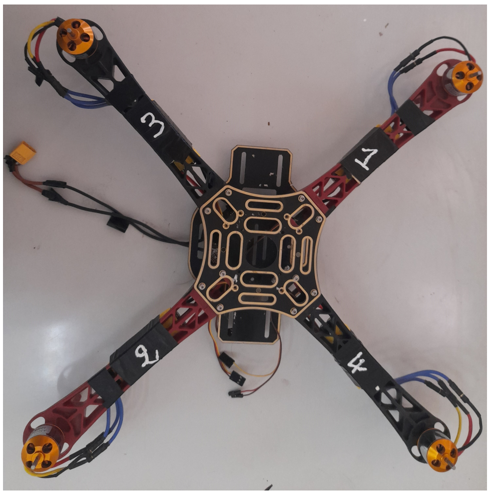


Figure 4.3: F450 frame

4.3.2 Motors

The selection of motors is critical in the UAV market since they must be strong enough to lift the entire drone. Brushed motors and brushless motors are the two primary types: Brushed rotors are made up of an armature, brushes, a field magnet, an axle, and a commutator.
Brushless motors, on the other hand, do not have brushes. They are surrounded by permanent magnets, usually more than four in number. These rotors make use of a

control circuit. They are powered by direct current energy via an inverter or switching power source, which generates alternating current to regulate the rotor phases using a closed loop controller.

For illustration, the motor used in this project has a KV of 2200 Brushless rotors, two revolving clockwise and the other two rotating counter-clockwise.

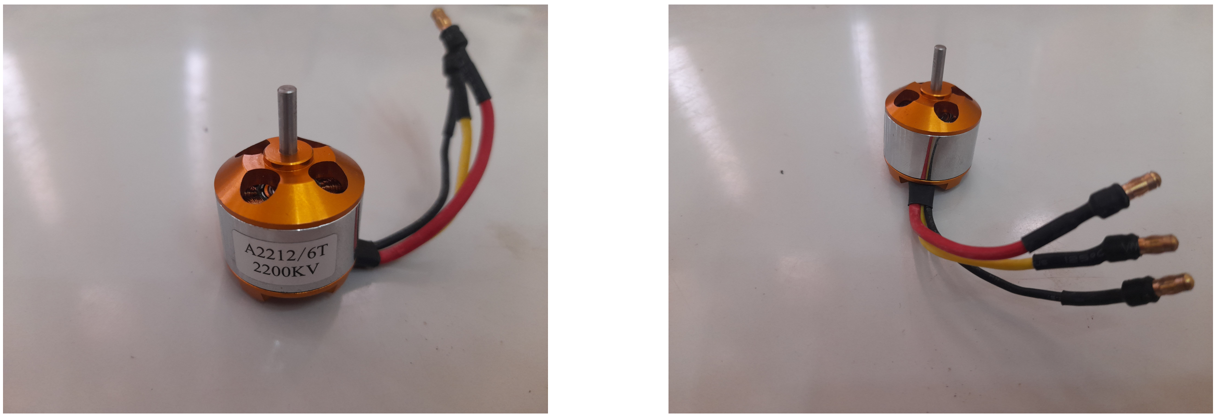


Figure 4.4: Brushless rotor

4.3.3 Propellers

Propellers are the top-end component of all rotors. A motor cannot provide the amount of thrust that it will create in order to lift a UAV. As a result, propellers must be provided. Each of the motors has a propeller attached to it. They are classified into two types: clockwise and counterclockwise.

Figure 4.5: Propellers 1045

4.3.4 Electronic speed controller

These components are really electrical circuits that provide a three-phase electric energy source. They are used to control the speed of a radio-controlled model's brushless motor. A standard ESC used in aeromodeling contains three connections and five wires that are each utilized for a specific purpose.

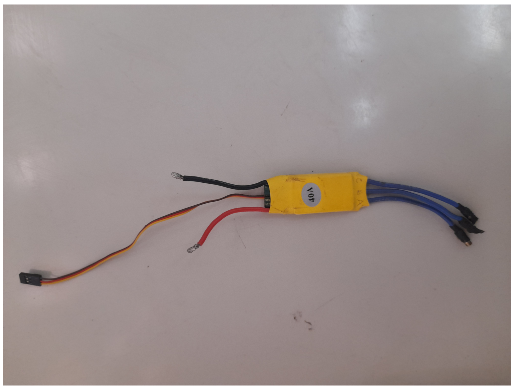


Figure 4.6: Electronic speed controller

4.3.5 Flight controller

All quad-copter behaviors must be under the control of the flight controller. A Pixhawk 2.4.8 is included in this project. Pixhawk is an independent open hardware project that offers UAV autopilot functions that are often used for commercial, industrial, and military applications. The project that was generated and developed is the alternate flight controller that is now available.

Figure 4.7: Pixhawk 2.4.8

4.3.6 Other components

4.3.6.1 Global Positioning system

A GPS sensor is required to provide an accurate reading of the UAV's present position in 3D space in order to execute autonomous activities such as waypoint-guided flights or to save home location. It also provides a more precise measurement of the UAV's current height above sea level, and GPS data enhance the UAV's current position and altitude holding in windy settings.

Figure 4.8: External GPS

4.3.6.2 Radio receiver

Nowadays the most used system to establish a communication with the vehicle is the radio control. Its great advantage is its continuous updates. The radio transmitter and receiver allow us to control the quad-copter. sending orders to the quad-copter.The receiver (on the quad-copter) receives these orders and make a reference signal to the control system in the controller . the Radiolik T8FB RC transmitter used in this project has 8 channels. This channel could be also used for servo motor control, which is commonly used when a camera is mounted as a payload on a quad-copter.

Figure 4.9: Radiolink transmitter and the receiver

4.3.7 Camera

It comes with a two axis gimbal that can be controlled with the remote controller, we used an FPV camera because of its real-time video transmission, it may have medium video quality but with high-speed data transmission to the ground station.

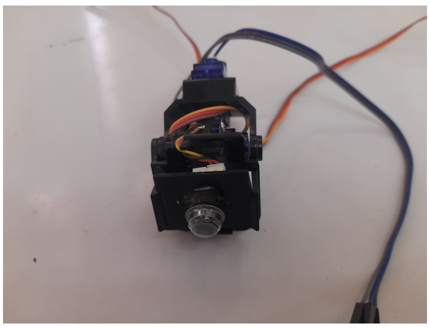


Figure 4.10: Camera

4.4 Software platform (Development environment)

As our thesis is concerned with academic research, the MATLAB/Simulink environment is frequently used as a tool for system modelling and control design. In particular, MATLAB/Simulink is the standard tool for exploiting the model-based design approach of developing embedded software, starting from block models.
The most interesting step of the UAV development cycle for the purpose of this thesis is the Processor-in-the-Loop (PIL) simulation: in this phase, the model-derived production code dedicated to the control of the aerial system is tested on the real autopilot board with a real-time simulation using Simulink, in order to check its robustness and to evaluate performance and potential optimisations, before proceeding to real flight tests.

To exploit this framework, in the past there has been a great effort to provide the possibility to automatically translate the algorithms developed in MATLAB/Simulink on the Pixhawk autopilot series.
Today, thanks to MathWorks' advances in automated embedded coding, PX4 development is able to support system models and control algorithms, designed with a Model-Based Design approach, without the need for developers to master low-level programming.

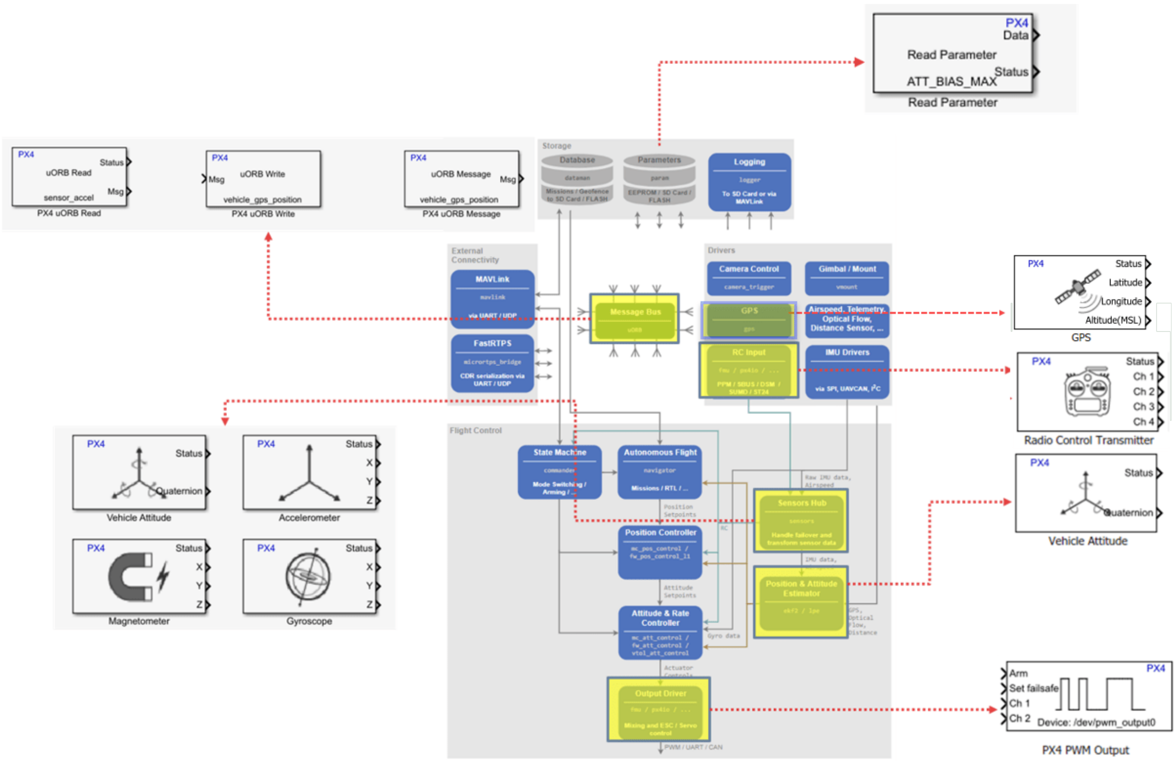


Figure 4.11: Supported Simulink Blocks that Interface with the PX4 Modules

Concretely, one of the practical objectives of this thesis has been to explore the potentiality of the means made available by the UAV Toolbox Support Package for PX4 autopilots to implement the quad-rotor model and controller design directly on Pixhawk 2.4.8 with automatic code generation.
The UAV Toolbox Support Package for PX4 autopilots is available since the 2018b MATLAB/Simlulink release under the name "Embedded coder Support package for PX4 Autopilot". This package is directly derived from the Simulink Pilot Support Package, used for the previous studies "Iris+ quad-rotor design and control" taken as reference for this work. (47)
This development environment allows access to the autopilot devices from the MATLAB/Simulink environment and to generate C++ code using the PX4 software stack, to

build and deploy algorithms while incorporating data from the onboard sensors. Interfaces for the components of the PX4 architecture are provided by Simulink blocks that function as inputs and outputs to the model.

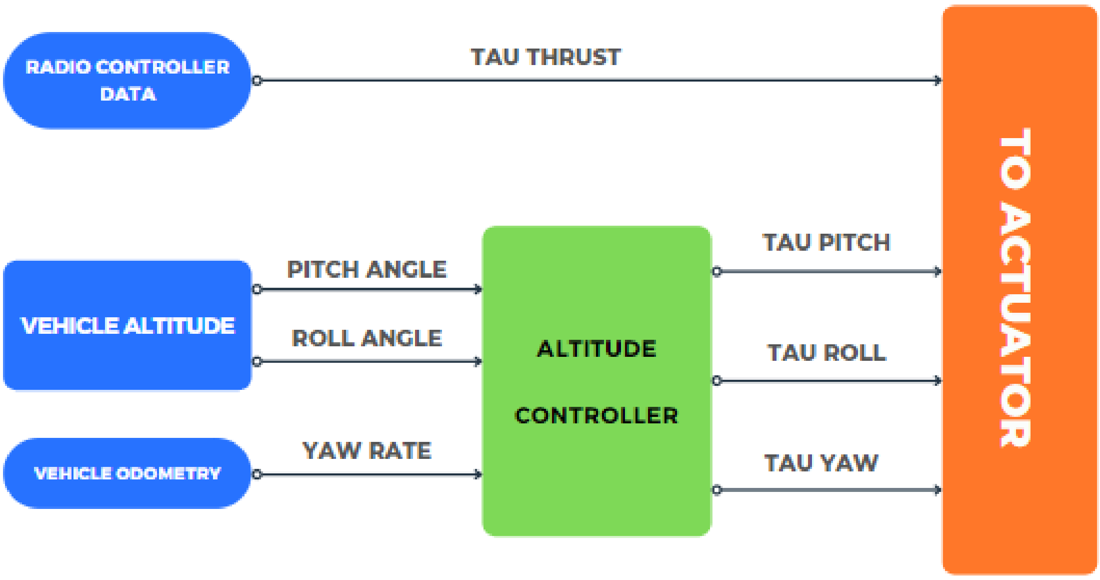


Figure 4.12: Diagram represent Control for X-Configuration Quadcopter

Using these capabilities, the position controller and attitude velocity controller modules of the general PX4 architecture are replaced by user-defined algorithms, which is our case : this is achieved through a custom boot script, which must be copied to the micro sd-card mounted on the Pixhawk autopilot .

This script, run just after bootstrapping (NuttX), disables the default PX4 Navigator and Commander modules, replacing them with a module, called "px4 simulink app", which acts as a wrapper for the generated code. The embedded coder, based on the "CMake" constructor, generates and compiles code from the models developed in Simulink using blocks.

This code is then executed by the px4 simulink app module in the PX4 software stack. Simulink blocks in general they provide the ability to subscribe to or publish (uORB) topics to retrieve read sensors or to impose a control output.

This allows a model to be built that refers directly to the peripherals, sensors and controls of the autopilot board. For example, the Vehicle Attitude block reads the uORB topic vehicle odometry and outputs attitude measurements from the Pixhawk hardware. With its own frequency, the block representing the software module checks whether a new message is available on the vehicle odometry topic.

This information is calculated in an attitude control system which, in order to follow a reference signal, outputs control signals which, via a mixing matrix, are supplied to a pulse width modulation (PWM) block. The attitude control system and mixing matrix must be selected, designed and set up for the particular air-frame being used.

The PWM block configures the PWM outputs for the servo motors: the block accepts the controller signals as input and writes these values to the selected channels, which are in turn subscribed by the motor control modules. In the figures shown in the following parts of this chapter, one can see the interconnection between the PX4 blocks and the attitude controller: the model is ready for the build process and deployment on the selected Pixhawk Series flight controller, which is to be installed on the real drone for flight testing.

At the end of this chapter, the simulation potential of the Support Package is explored, and the controller implementation exploiting these blocks for flight testing is also mentioned and explained.

4.5 Assembly and final build

In this section, we will go through all of the previously described components and connections.

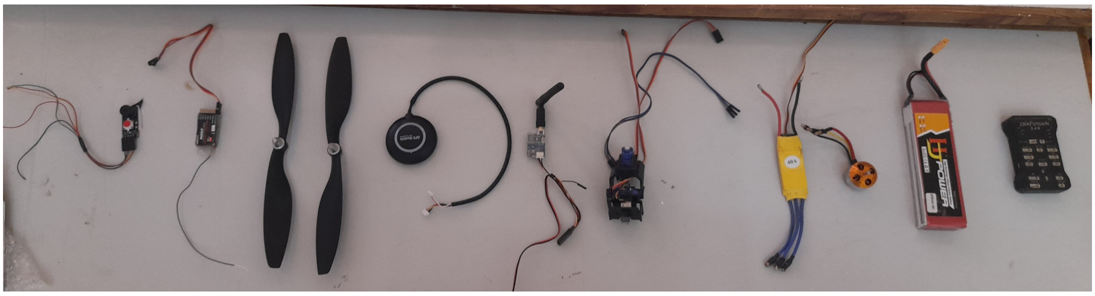


Figure 4.13: components

As mentioned above, two rotors spin clockwise and the other two spin counterclockwise. For that instance, the blades need to have the corresponding shape.

4.5.1 Final model

the custom-built UAV totally configured and ready to fly, including all the previous described components. To help at UAV landings, a landing gear was installed.

Figure 4.14: Final UAV build

4.6 UAV flight tests and results

4.6.1 Flight tests

for the testing multiple simulations were run on our simulink model by making the necessary changes in PID controller and obtaining the desired results in MATLAB 2021a, then we deployed and load the model on the board (PX4).

before any flight the quad-copter needs calibration. After that we make sure that the quad-copter is in safe conditions as an example we check if all propellers are well mounted and secure. The tests are showed in figures bellow:

Figure 4.15: Flight test 1 of the quadcopter

Figure 4.16: Flight test 2 of the quadcopter

The aim of this chapter is to detect aerial view fires .Due to the lack of hardware materials , we tried to memic aerial footage in the laboratory and compose our discussion based on it . We tested our algorithm on a set of real images. For each experiment we show four images: the original image , the undistorted images where the fire was detected; image with its depth displayed in meters , and a final image containing real world coordinated of the object detected . pretest steps : before proceeding the test phase, we had to measure some important parameters in order to estimate the accuracy of the provided algorithm , which are :

- 1. Measuring the distance between the fire and cameras approximately .
- 2. Measuring the initial height of the camera
- 3. Measuring the orientations and the translation of the cameras along the three axis (for our case the relative position was calculated according to the first camera only so we were interested with its parameters)

To be noted that resizing the images in the fire detection phase could play a major factor in affecting the accuracy of depth extraction .

4.6.2 Fire detection and localization results

4.6.2.1 First test:

The Test steps are similar to the one mentioned before in chapter 3 results .The only difference is the height added to the camera to perform aerial images .

Figure 4.17: Original image before detection

Figure 4.18: Detection test 1

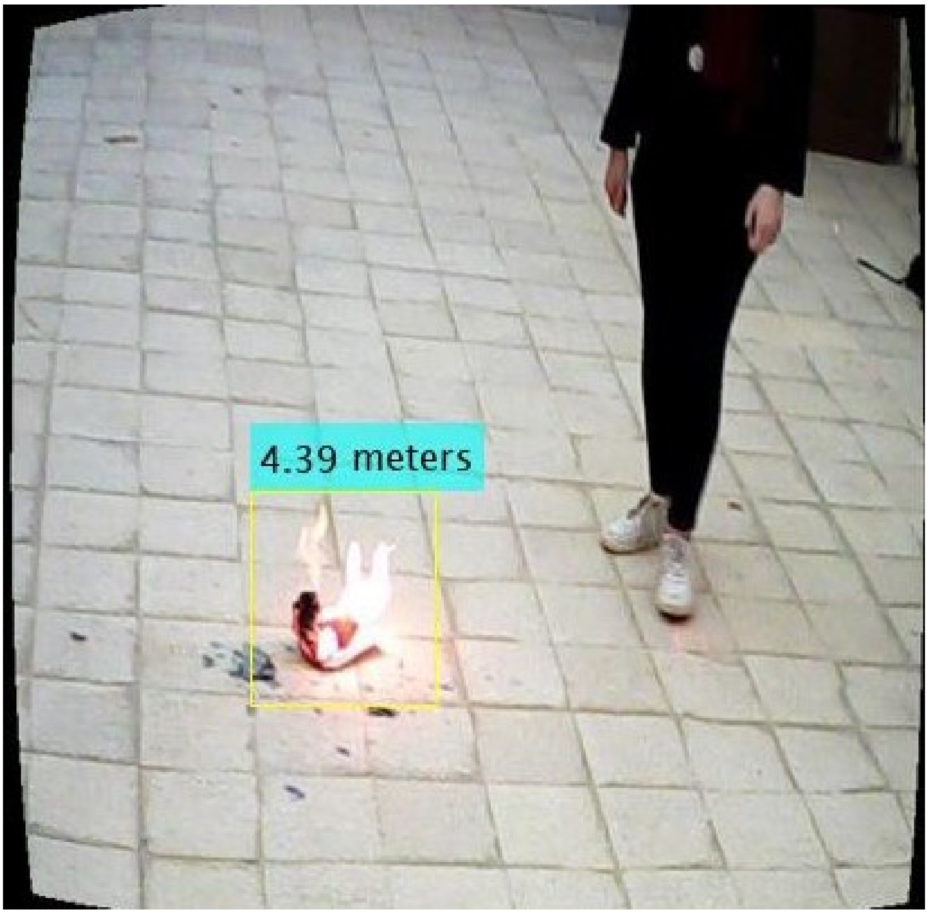


Figure 4.19: Depth extraction test 1

4.6.2.2 Second test:

Figure 4.20: Detection test 2

Figure 4.21: Depth extraction

Camera coordinates	World coordinates (mm)
-0.1219	-386.4144
-0.0150	-1592.6
-	3174.3

Table 4.1: Object relative position test 2

Camera coordinates	World coordinates (mm)
-0.1977	-868.1972
-0.0249	-1704.381
-	4391.02

Table 4.2: Object relative position test 1

4.6.3 Results discussion

Flight tests: Multiple simulations were run on the simulink model before making the first test, from the above experiments we can say that the quad-copter can realize the basic flight actions. the UAV was manually flown at different heights above the fire location.

Detection and localization tests: The results represented in this section are just examples from the precedent chapters.
we gave in the previous chapters detailed steps on how to get the fire detection using different CNN architectures and the procedure to get the relative position.

Simple result images along with their originals are illustrated in figures above 4.18 4.19 contains fire images captured from a close distance. as the results show, the fire detection and monitoring algorithm proved to be able to correctly identify the fire as expected even with the presence of red color of man made Objects 4.21 within the same color range.
Using stereo cameras a detected object and MATLAB function to extract the depth displays the real world coordinates of an object in our case Fire. The tables 4.2 4.1 represent the coordinates results.

4.7 Conclusion

This chapter is divided into three main objectives , first we aimed to develop a firmware for a quad- rotor UAV using MATLAB software .
A firmware is composed by a lot of functionalities and the time needed to develop them in the best way can be enormous, For example, the first project of pixhawk was in 2009 , since then and till today it stills under development . Knowing that, our work was limited to the development of a basic, but stable, version of a personal firmware that has to deal with pre-defined hardware.Than , we have done the hardware implementation of the UAV using pixhawk autopilot , this process took a lot of time phases to test each component on its own and to insure the coordination between each one . We did a simulation of the entire system both on the computer and in real time.
Finally, a part of object detection and depth estimation with relative position calculation was computed in order to be tested on aerial images to present the efficacy of algorithms proposed in previous chapters. However , due to the lack of time and materials , real images from the made Quad-rotor was not able to be token , so we had to improvise in this case and create our own aerial images .

General conclusion

In this thesis, current research progress related to forest fire detection are well investigated. Besides, several reliable and accurate UAV-based forest fire detection methodologies are also developed, these include:

- A comprehensive literature review on the existing UAV-based forest fire detection systems as well as vision-based forest fire detection and localization techniques are provided.

- A method of vision-based forest fire detection in visible range images, which makes use of CNN in image segmentation , is developed for the UAV-based forest fire detection system . The proposed method based on neural network fire detection approach with standard computation requirement is first designed to effectively extract the suspicious fire regions with high accuracy .Experimental verification are conducted in two scenarios, one is a real aerial forest fire image gathered by an aircraft and the other is a collected by a like-aerial camera footage in an indoor environment. Experimental results have demonstrated that the designed forest fire detection approach is able to achieve satisfactory performance with greatly improved reliability and accuracy in forest fire detection applications.

- Choosing the best algorithm depends on the used application and system requirement. Detecting drones requires high accuracy, long range, and low false positives. This is why we had to perform a comparison study , Faster RCNN , SSD , YOLOv2 and YOLOv4 were the most used algorithms along years , with different results , we ended up with YOLOv4 as the object detection algorithm that meets our requirements .

- Another matter to deal with , was the issue of depth extraction .Due to the complexity of the problem , lack of researches was present. The stereo-vision based method applied in this study had a great

- In order to achieve an effective UAV control , a mix of careful assembly of the quad-copter based on pixhawk autopilot and a personal firmware was deployed in the hardware . The effectiveness of the proposed methodology is verified and the experimental results show that the quad copter passed the stability test with ups and downs through the process (the need of PID fine tuning .Unfortunately , we haven't made the phase of combining the three main parts of the thesis to make the real time Fire detection system and localization implemented on UAV.

This thesis studied " **Hardware implementation of a forest fire detection system using UAV's "**
it was only a first step of many other projects and can be used in future researches.

4.8 Further work

The following propositions may be an area of future research based on the work done in this ongoing research in this thesis:

- the accuracy of Fire detection could be increased using data merging , this can be computed using more than one camera , an approach to be proposed is the use of 3 cameras aligned with (X Y Z) axis in order to get the full scene of the burning area , than merging the received data with strong and robust algorithms for the detection aim.

- Smoke from wildfires is visible from a great distance, whereas this is not always the case for flames. So, in order to have a system capable of detecting a forest fire in its early stages, it is necessary to also search for smoke. It is therefore interesting to study the combination of of different fire characteristics such as smoke and flames.

- Forest fire propagation path is an important parameter to be put in the consideration . The development of a method for this purpose can be of great help to fire fighters who want to control forest fires.

- A CNN-based location algorithm could improve the accuracy of finding the position of the fire in the forest with precise coordinates, this method could help in early fire alarms and people evacuation , therefore saving more lives .

- In order to become operational, a system must pass many field tests. As previously stated, the main objective of this thesis was to mount the system on a UAV and test it in more practical application scenarios, but due to the conditions imposed, this did not happen. Therefore,a recommended course of action is to perform extensive field testing on a UAV. The tricky point could be the high-frequency vibrations of the drones that can make the images blurred and thus , the fire detection task would be more difficult .

Appendix

Appendix A

Component	Weight	Characteristics
Frame	282 g	Dimension: 450mm Voltage Distributor: Yes
Motor	52 g	KV(Rpm/V): 2200 Battery Opernating :2-3 Lipo Working.Amps :14-22 A Thrust at 3S with 1045 propeller: 1200 g
Electronic Speed Control	35g	Max Current: 40A Lipo: 2-4 S BEC: 3A / 5V
Propellers	14g	Length: 10 inch / 254 mm
Battery	243g	Capacity: 4200mAh Voltage: 11.1 V Max continuos discharge: 25c
Pixhawk 2.4.8	40 g	Supply Voltage: 7V Processor: 32 Bit Cortex M4 Core Sensors: Gyrometer, Accelerometer, Barometer ,Magnetometer
Radiolink Receiver R8EF	7g	Channels: 8 CH Signals: SBUS/PPM,PWM Operating Voltage: 3 12V Control distance: 2000 meters

Table 4.3: Table representing components weight and Characteristics

Appendix B

Camera	Focal length	Principal point	Intrinsic matrix	Radial distortion
Camera parameters 1	[932.1819 929.1559]	[338.5335 246.8962]	$\begin{bmatrix} 923.1819 & 0 & 0 \\ 0 & 29.1559 & 0 \\ 338.5335 & 246.8962 & 1 \\ \end{bmatrix}$	[0.0509 -1.0153 20.1072]
Camera parameters 2	[1.1652e+03 1.1760e+03]	[355.8581 330.2815]	$\begin{bmatrix} 1.1652e+03 & 0 & 0 \\ 0 & 1.1760e+03 & 0 \\ 335.8581 & 330.2815 & 1 \\ \end{bmatrix}$	[0.0240 -0.1004 2.4647]

Table 4.4: Camera parameters

Bibliography

[1] algeria press service (2022). Forest fires: more than 100,000 hectares ravaged in 21 wilayas during the summer of 2021. `shorturl.at/hMRU6/`. [Online; Tuesday, May 10, 2022].

[2] Alkhatib, A. A. (2014). A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, 10(3):597368.

[3] API, A. (2021). Image classification and object detection. `https://developers.arcgis.com/python/guide/how-ssd-works/`. [Online; 2021].

[4] Bandyopadhyay, H. (2022). YOLO: Real-Time Object Detection Explained. `https://www.v7labs.com/blog/yolo-object-detection/`. [Online; july 19, 2021].

[5] Beachly, E., Detweiler, C., Elbaum, S., Duncan, B., Hildebrandt, C., Twidwell, D., and Allen, C. (2018). Fire-aware planning of aerial trajectories and ignitions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 685–692. IEEE.

[6] Belbachir, A., Escareno, J., Rubio, E., and Sossa, H. (2015). Preliminary results on uav-based forest fire localization based on decisional navigation. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 377–382. IEEE.

[7] Boudguiga, O. (2016). *Commande à saturation pour le contrôle de la position d'un robot volant de type quadrotor.* PhD thesis, École de technologie supérieure.

[8] Chaikalis, D., Evangeliou, N., Tzes, A., and Khorrami, F. (2022). Design, modelling, localization, and control for fire-fighting aerial vehicles. In *2022 30th Mediterranean Conference on Control and Automation (MED)*, pages 432–437. IEEE.

[Chopde et al.] Chopde, A., Magon, A., and Bhatkar, S. Forest fire detection and prediction from image processing using rcnn.

[10] Curt, T., Aini, A., and Dupire, S. (2020). Fire activity in mediterranean forests (the algerian case). *Fire*, 3(4):58.

[11] Danilov, A., Smirnov, U. D., and Pashkevich, M. (2015). The system of the ecological monitoring of environment which is based on the usage of uav. *Russian journal of ecology*, 46(1):14–19.

[12] GÜLLÜCE, Y. and Celik, R. N. (2020). Fireanalyst: An effective system for detecting fire geolocation and fire behavior in forestsusing mathematical modeling. *Turkish Journal of Agriculture and Forestry*, 44(2):127–139.

[13] Harjoko, A., Dharmawan, A., Adhinata, F. D., Kosala, G., and Jo, K.-H. (2022). Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. *Fire*, 5(1):23.

[14] Hata, K. and Savarese, S. (2017). Cs231a course notes 1: Camera models.

[15] Kasyap, V. L., Sumathi, D., Alluri, K., Reddy Ch, P., Thilakarathne, N., and Shafi, R. M. (2022). Early detection of forest fire using mixed learning techniques and uav. *Computational intelligence and neuroscience*, 2022.

[16] Khan, A., Hassan, B., Khan, S., Ahmed, R., and Abuassba, A. (2022). Deepfire: A novel dataset and deep transfer learning benchmark for forest fire detection. *Mobile Information Systems*, 2022.

[17] KHAZRI, A. (2019). Faster RCNN Object detection. `https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4`. [Online; Tuesday, May 9, 2019].

[18] Kim, J.-H., Jo, S., and Lattimer, B. Y. (2016). Feature selection for intelligent firefighting robot classification of fire, smoke, and thermal reflections using thermal infrared images. *Journal of Sensors*, 2016.

[19] Kim, J.-H., Starr, J. W., and Lattimer, B. Y. (2015). Firefighting robot stereo infrared vision and radar sensor fusion for imaging through smoke. *Fire Technology*, 51(4):823–845.

[20] Kim, J.-H., Sung, Y., and Lattimer, B. Y. (2017). Bayesian estimation based real-time fire-heading in smoke-filled indoor environments using thermal imagery. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5231–5236. IEEE.

[21] Li, P. and Zhao, W. (2020). Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, 19:100625.

[22] Liu, H., Liu, T., Zhang, Z., Sangaiah, A. K., Yang, B., and Li, Y. (2022). Arhpe: Asymmetric relation-aware representation learning for head pose estimation in industrial human-machine interaction. *IEEE Transactions on Industrial Informatics*.

[23] Lu, K., Xu, R., Li, J., Lv, Y., Lin, H., and Liu, Y. (2022). A vision-based detection and spatial localization scheme for forest fire inspection from uav. *Forests*, 13(3):383.

[24] Luo, Y., Zhao, L., Liu, P., and Huang, D. (2018). Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimedia Tools and Applications*, 77(12):15075–15092.

[25] Mathwork (2021a). Computer Vision Toolbox. `https://www.mathworks.com/products/computer-vision.html`. [Online; 2021].

[26] Mathwork (2021b). Image Processing Toolbox. `https://www.mathworks.com/products/image.html`. [Online; 2021].

[27] McNeil, J. G. and Lattimer, B. Y. (2015). Real-time classification of water spray and leaks for robotic firefighting. *International Journal of Computer Vision and Image Processing (IJCVIP)*, 5(1):1–26.

[28] McNeil, J. G. and Lattimer, B. Y. (2017). Robotic fire suppression through autonomous feedback control. *Fire technology*, 53(3):1171–1199.

[29] Meyer, W. (2006). Coordinate frame. `https://www.sciencedirect.com/topics/mathematics/coordinate-frame/`. [Online;2006].

[Praveena et al.] Praveena, S. M., Akshaya, B., Devipriya, B., Divya, U., and Mirudhula, K. Forest fire detection usingdrone.

[31] press service, A. (2018). National forest heritage. `https://cutt.ly/5CmZgYj/`. [Online;October 25,2018].

[32] Rabbath, C. A. and Léchevin, N. (2010). *Safety and reliability in cooperating unmanned aerial systems*. World Scientific.

[33] Russell, D. J., Arevalo-Ramirez, T., Garg, C., Kuang, W., Yandun, F., Wettergreen, D., and Kantor, G. (2022). Uav mapping with semantic and traversability metrics for forest fire mitigation. In *ICRA 2022 Workshop in Innovation in Forestry Robotics: Research and Industry Adoption*.

[34] Salvi, J., Armangué, X., and Batlle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. *Pattern recognition*, 35(7):1617–1635.

[35] Skeele, R. C. and Hollinger, G. A. (2016). Aerial vehicle path planning for monitoring wildfire frontiers. In *Field and Service Robotics*, pages 455–467. Springer.

[36] Son, G.-Y. and Park, J.-S. (2018). A study of kernel characteristics of cnn deep learning for effective fire detection based on video. *The Journal of the Korea institute of electronic communication sciences*, 13(6):1257–1262.

[37] Song, T., Tang, B., Zhao, M., and Deng, L. (2014). An accurate 3-d fire location method based on sub-pixel edge detection and non-parametric stereo matching. *Measurement*, 50:160–171.

[38] Sudhakar, S., Vijayakumar, V., Kumar, C. S., Priya, V., Ravi, L., and Subramaniyaswamy, V. (2020). Unmanned aerial vehicle (uav) based forest fire detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, 149:1–16.

[39] Toulouse, T., Rossi, L., Akhloufi, M., Pieri, A., and Maldague, X. (2018). A multimodal 3d framework for fire characteristics estimation. *Measurement Science and Technology*, 29(2):025404.

[40] Twidwell, D., Allen, C. R., Detweiler, C., Higgins, J., Laney, C., and Elbaum, S. (2016). Smokey comes of age: unmanned aerial systems for fire management. *Frontiers in Ecology and the Environment*, 14(6):333–339.

[41] Valavanis, K. P. (2008). Advances in unmanned aerial vehicles: state of the art and the road to autonomy.

[42] Wikipedia (2022). Kabylie. `https://fr.wikipedia.org/wiki/Kabylie/`. [Online;august 1,2022].

[43] Yuan, C., Zhang, Y., and Liu, Z. (2015). A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian journal of forest research*, 45(7):783–792.

[44] Zhang, L., Wang, M., Fu, Y., and Ding, Y. (2022). A forest fire recognition method using uav images based on transfer learning. *Forests*, 13(7):975.

[45] Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252.

[46] Zhao, P. and Wang, N.-H. (2010). Precise perimeter measurement for 3d object with a binocular stereo vision measurement system. *Optik*, 121(10):953–957.

[47] Zhong, F. W. (2015). Implementation of simulink controller design on iris+ quadrotor. *Naval Postgraduate School Thesis, Monterey, California.*