

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'informatique



Mémoire de Master

Spécialité

Informatique

Option

Réseaux, Mobilité et Systèmes Embarqués

Thème

« Détection de feux de forêts par drone en mission planifiée »

Présenté par

ZAZOUN Kenza

LALLAL Ahmed

Proposé et Encadré par

M^{rs} DAOUI.M

Devant le jury

M^{rs} LAGHROUCHE.M Président du jury

M^{me} HADDAOUI.R Examinatrice

Soutenu le 27/ 12 / 2020

Remerciements

Nous rendons grâce à Dieu le tout puissant et miséricordieux, qui nous a donné l'aide, la patience, le courage pour accomplir ce modeste travail et nous a maintenu en santé pour mener à bien cette année d'étude.

Nous tiendrons à remercier les personnes qui grâce à eux nous ne pourrions Jamais être là, nos parents et ainsi que toutes nos familles.

*Nous voudrions tout d'abord adresser toute notre reconnaissance et sincères remerciement à notre promoteur M^{rs} **DAOUI Mehammed**, qui a accepté d'encadrer notre thème et pour l'aide continuelle qu'il a fourni et les connaissances qu'il a su nous transmettre. Nous le remercions pour sa patience, sa disponibilité, également pour, la qualité de ses conseils et sa patience et son encouragement tout au long de ce travail.*

Nous remercions les membres du jury, devant qui nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu .

Nos vifs remerciements s'adressent aussi aux prestigieux membres de jurés devant qui nous avons l'honneur d'exposer notre travail et pour l'intérêt qu'ils lui ont porté en acceptant de l'examiner.

Nous tiendrons à remercier Nos parents pour leurs amour, leurs conseils ainsi que leur soutien inconditionnel à la fois moral et économique, ainsi que nos sœurs et frères et toutes nos familles qui nous ont permis de réaliser les études que nous voulions et par conséquent ce mémoire.

Nous remercions vivement tous nos professeurs qui nous ont accompagnés durant notre cursus universitaire et pour leur aide et le savoir aussi précieux qu'ils nous ont apporté au sein de département d'informatique à l'UMMTO. Un grand merci à nous pour avoir cru en nous et pour notre persévérance. Nous réservons ici une place particulière pour remercier vivement tous ceux qui de près ou de loin nous ont aidés et encouragés tout au long de réalisation de ce modeste travail : nos camarades de la promotion M2 2020, RMSE Family.

*Un grand merci et reconnaissance à **Nassim SADALLAH** qui nous à vraiment aidé durant notre réalisation avec ces brillant conseilles.*

Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances aux responsables et au personnel de l'UMMTO.

Nous voudrions exprimer nos reconnaissances envers les amis et camarades de parcours qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Merci à toutes et à tous.

Dédicaces

*On a l'honneur de Dédier ce modeste travail A la mémoire de notre regretté enseignant «Mrs.
HAMDANI Chabane»*

Puisse Dieu, le tout puissant l'avoir en sa sainte miséricorde.

«Sans la science je ne pourrais pas vivre»

*Était sa récitation et son être, Un professeur qui nos a marqué par son dévouement et sa
patience avec nous étudiants et toutes personnes le côtoyant. Il était ce professeur
exemplaire que toutes personnes rêvée d'avoir durant un parcours d'études.*

*Il était si content d'apprendre le thème de notre Mémoire et comme toutes fois ou on lui
demander sur quelques choses ou question, il nous a encouragé et motivé et nous a proposé
son aide.*

*Merci monsieur pour toutes les doses de motivation que vous nous avez injecté durant nos
Tps et débats, pour image que vous avez instauré au sein de nos cœurs et esprits de ce qu'est
la science et ce qu'elle pouvait nos procuré.*

"faut maîtrisé le C"

vous le disiez au moins une fois durant vos cours ou TPs.

*Merci pour tous le savoir que vous nous avez procuré, gratitude et reconnaissance à la qualité
de nos TPs en votre présence Nous espérons que Vous apprécierez cet humble geste comme
preuve de reconnaissance.*

Dédicaces

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout de ce rêve.

Je dédie ce modeste travail à ceux qui m'ont encouragé et soutenu, je cite :

*Particulièrement à **Mes parents** les plus chers au monde qui ont consacré leur existence à bâtir la mienne, pour leur soutien, amour, patience et tendresse, pour tous ce qui ils ont fait pour que je puisse arriver à ce stade. ,que dieu les garde et les Protège. Je prie Dieu le tout puissant de les garder en bonne santé et de les récompenser de toutes les peines et sacrifices données auxquels je ne rendrai jamais assez.*

***A mes sœurs et frère** et aussi à toute ma famille et mes proches qui me souhaitent toujours la réussite.*

***A mes chers copains** qui ne cessent de m'encourager dans les moments difficiles et des moments en ors merci à vous (**Massi, Massi, Lux, Tarik, Lamine Le bras, Farid, Madjid Paradox, Mouhouche, Samir, Samir,...**) et à qui je souhaite plus de succès*
***A mes amis de promotion RMSE** avec qui j'ai passé des moments inoubliables et qui m'ont encouragé.*

A « Mira,Noel, Thanina,Thiziri, Sabrina, l'histoire, Zico, Rafik et Moi-même ! »

A mon binôme Kenza

Et à toute personne ayant participé à l'élaboration de ce travail.

Ahmed

Dédicaces

C'est avec profonde gratitude que je dédie cet humble travail

A mes très chers parents,

Aucun hommage ne saurait exprimer mon amour éternel, ma reconnaissance et ma considération pour les sacrifices que vous avez consentis pour mon éducation et mon bien être.

A la mémoire de mon père

Puisse Dieu, le tout puissant l'avoir en sa sainte miséricorde. Tu nous a quittée subitement et très tôt pour nous, mais ta mission avait pour fin ce jour-là, j'espéré de là ou tu es tu es fière de ce que je suis devenu et des modestes choses que j'ai pu accomplir en t'ayant souvent dans mon cœur et étant sure que tu veille sur moi, chaque ligne que j'écrivais dans ce mémoire je me demander comment ce rapport pouvais être en ta présence toi qui aimé l'écriture et les ouvrages, tes Etudiants étaient chanceux de t'avoir eu comme prof. J'espère que tu apprécieras cet humble geste comme preuve de reconnaissance.

A Ma mère

Qui a sacrifié ses ambitions pour notre réussite, cette femme qui a fait tout son possible pour me donner une bonne éducation, et veillé à ce que je réussie dans tous ce que j'entreprends,

Ma source d'encouragement. Merci ! car ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études

A mes Chères Sœurs

Katia et Lehna qui n'ont jamais cessé de me supporter et de croire en moi. A mon cher Frère Youva Qui a était le père et le frère et qui était à mes coté et qui a n'a jamais cessé de me soutenir et de tout donner pour notre joie, merci d'avoir contribuer à ce que je sois la femme d'aujourd'hui. "Dieu vous garde pour moi je vous souhaite tout le bonheur du monde"

A Moi-même Preuve de persévérance.

A Damya, Melissa, Sabrina

Ce trésor que ce parcours ma offert. Béni de les avoir à mes coté, Merci à Allah qui a fait que nos chemins se croisent les mots ne suffiront pas pour décrire ma gratitude. Que dieu vous protégé et vous comble par son amour #SMDK

A GDG_ Tizi Ouzou Et WTM Family

Je ne peux vous appeler qu'une famille car j'ai partagé avec vous les bon et mauvais moments, qui resteront gravé à jamais dans ma mémoire, moments magiques. Je ne pourrais remercierai jamais assez dieu pour les expériences que on a partagé ensemble " Gratitude ".

A Monsieur Aliane

Mon professeur, Mon grand frère, merci d'être ce que vous êtes, merci d'avoir été là à mes coté quand j'en avait vraiment besoin, vous étiez la toujours à m'encourager pour aller de l'avant et vous creusé au fond de ma personnalisé pour me faire voir ce dont je suis capable, et vaincre mes peurs. Ces remerciements ce n'est rien devant ce que vous avez contribuer durant toutes ces années.

A Mira, Fifi et Dehbia

Merci d'avoir toujours était là et surtout d'un support moral et une bonne écoute.

A Ahmed The Genius *Mon binôme, pour son soutien, sa patience et sa compréhension tout au long de ce projet.*

A Mon oncle Daniel.

Grand Mère, Oncles, Tantes Cousins et Cousines.

A Mes Amis Camarade de promotion

Nos photos durant les TP sans votre permission et les moments partagé avec vous dans labo 9 m'ont marqué, merci à chacun de vous qui de prêt ou loin nous ont aidés particulièrement « Abdessalam, Nawalati, Nina, Tassadit, Thiziri, Tarik, Rafik, Zico »

A la Team L35.

Kenza
Gratitude et Amour

Table des matières

Introduction Générale	1
1 Etat de l'art sur les drones	3
1.1 Introduction	3
1.2 Définition d'un drone	3
1.3 Evolution des drones	6
1.4 Classification des drones	8
1.4.1 Selon la taille	8
1.4.2 Selon le mode de propulsion :	10
1.4.3 Selon le poids maximum au décollage, l'altitude maximale et l'endurance :	12
1.5 Méthodes pour la navigation	13
1.5.1 La navigation par suivi de terrain	13
1.5.2 La navigation par points de passage	13
1.5.3 L'évitement d'obstacles	14
1.6 Les domaines d'utilisation	14
1.6.1 Utilisation militaire des drones	14
1.6.2 Utilisation civile des drones	15
1.7 Règlementation sur l'utilisation des drones	16
1.8 Conclusion	17
2 Principe de fonctionnement d'un drone quad-copter	18
2.1 Introduction	18
2.2 Description générale d'un quad-copter	18
2.3 Fonctionnement	19
2.3.1 Sens de rotation des voilures tournantes	19
2.4 Composants électroniques d'un quad-copter	21
2.5 Les capteurs pour la navigation et la localisation	25
2.5.1 Capteurs proprioceptifs	25
2.5.2 Capteurs extéroceptifs	26
2.5.3 Le système de géolocalisation	27
2.6 Modes de vol :	28
2.6.1 Vol vertical	28
2.6.2 Vol stationnaire	29
2.6.3 Vol de translation	29
2.7 Conclusion	30

3	Planification de missions de drones	31
3.1	Introduction	31
3.2	Contrôle de vol des quad-copter	31
3.2.1	La station de base :	32
3.2.2	Protocole Mavlink	33
3.3	Contrôle d'un UAV Avec Mission Planer	34
3.4	Conclusion	53
4	La détection de feux de forêt	54
4.1	Introduction	54
4.2	Traitement Image	54
4.2.1	Définition :	54
4.2.2	Définition de l'image	54
4.2.3	Image Numérique	55
4.3	Apprentissage automatique	57
4.3.1	L'apprentissage automatique selon la nature des données en entrée :	57
4.3.2	L'apprentissage automatique selon le type de résultat	58
4.4	Méthodes de Classification des images numériques	58
4.4.1	Classification non supervisée	58
4.4.2	Classification supervisée	59
4.4.3	Réseau de neurones artificiels :	59
4.5	Les réseaux de neurones convolutionnels CNN	62
4.5.1	Développement du modèle et implémentation CNN :	64
4.6	Outils utilisés :	72
4.6.1	Anaconda3	72
4.6.2	Jupyter Notebook :	72
4.6.3	Python 3.6	73
4.6.4	Tensorflow 1.13.1 :	73
4.6.5	Bibliothèques utilisées	74
4.7	Conclusion	74
	Conclusion Générale	75
	Annexe	78

Table des figures

1.1	Quelques moteurs	4
1.2	Exemples de voilures.	4
1.3	Exemples d'une cellule.	4
1.4	Exemple d'une caméra.	5
1.5	Le système de drone.	6
1.6	Une libellule.	7
1.7	DH 82 Queen Bee.	7
1.8	Illustration d'un drone FireBee.	8
1.9	Les différents types de drones civils et militaires.	8
1.10	Exemple d'un drone de type HALE.	9
1.11	Exemple d'un drone de type MALE.	9
1.12	Exemple d'un Micro quad-copter.	10
1.13	Exemple d'un drone à voilure fixe.	10
1.14	Exemple d'un mono rotor.	11
1.15	Exemple d'un quadri rotor.	11
1.16	Exemple d'un drone à ailes battantes.	12
1.17	Navigation par suivi de terrain par flux optiques.	13
1.18	Navigation par points de passage	14
1.19	Utilisation des drones dans le secteur militaire	15
1.20	Quelques domaines d'utilisation civile des drones	15
1.21	Quelques domaines d'utilisation civile des drones	16
2.1	Structure générale d'un quad-copter.	19
2.2	Sens de rotation des hélices.	20
2.3	Illustration le mouvement lacet "YAW".	20
2.4	Illustration mouvement roulis et tangage.	21
2.5	illustration d'un ESC.	22
2.6	Motors.	22
2.7	illustration d'une Pixhawk 2.4.8.	23
2.8	illustration du module télémétrie.	24
2.9	Batterie.	24
2.10	Transmetteur TS832 et récepteur RC832.	25
2.11	Une carte IMU.	26
2.12	Capteur télémétrique ultrason pour mesurer la distance avec un objet distant.	27
2.13	illustration du module GPS.	27
2.14	Composants de base du quadricoptère.	28
2.15	Illustration du mouvement vertical.	28
2.16	Illustration du mouvement stationnaire.	29

2.17	Illustration du mouvement de translation.	29
3.1	illustration de notre système.	32
3.2	illustration de la communication entre Mission planner et UAV grâce à MAVlink.	33
3.3	schémas explicatifs de notre système.	33
3.4	charger le firmware sur la Pixhawk.	34
3.5	Type frame.	35
3.6	Calibration des Accéléromètres.	35
3.7	Positionnement du drone pour la calibration des accéléromètres.	36
3.8	Calibration de la boussole.	36
3.9	Calibration Radio.	37
3.10	Calibration des ESCs.	38
3.11	Servo Moteur Configuration.	38
3.12	Full parameter List.	39
3.13	exemple d'une planification de vol avec WayPoints.	40
3.14	tableau d'information WayPoint.	41
3.15	Illustration de la commande Takeoff.	41
3.16	Illustration de la commande Waypoint	42
3.17	Illustration de la commande Spline Waypoint	43
3.18	Illustration de la commande Loiter Time	43
3.19	Illustration de la commande Loiter Turns	43
3.20	Illustration de la commande Loiter unlimited	44
3.21	Illustration de la commande Loiter Launch	44
3.22	Illustration de la commande Land	44
3.23	Illustration de la commande Delay	45
3.24	Illustration de la commande Do_Jump	45
3.25	Illustration de la commande Do_Set_Cam_Trigg_Dist	45
3.26	Illustration de la commande Do_Digicam_Control	46
3.27	Illustration de la commande Do_Set_Roi	46
3.28	Illustration de la commande Condition_Delay	47
3.29	Illustration de la commande Condition_Distance	47
3.30	figure exemple de planification avec auto Grid.	48
3.31	tableau de paramètres "Etat".	49
3.32	charger un script de vol.	49
3.33	Section Action sur Data.	50
3.34	mise en action de Guided Mode.	51
3.35	onglet Data sur Mission Planner.	51
3.36	illustration du système de transmission vidéo	52
4.1	Exemple d'histogramme d'une image.	57
4.2	Réseau de neurone artificiel (a) et profond(b) [2]	60
4.3	Vue simplifiée d'un réseau artificiel de neurones.	60
4.4	Structure simplifié d'un neurone artificiel.	61
4.5	Modélisation d'un perceptron.	62
4.6	Modélisation du perceptron multicouche.	62
4.7	Fonctionnement d'un réseau de neurones convolutifs.	63
4.8	Segmentation des données du modèle séquentiel	65
4.9	Architecture du modèle séquentiel.	66
4.10	Diagramme de précision et de perte du modèle séquentiel.	66

4.11	Resultats de quelques tests d'images avec Model séquentiel.	67
4.12	Segmentation des données modèle inceptionV3.	68
4.13	Résultats du premier entrainement du model InceptionV3.	69
4.14	Résultats d'entrainement de 10 epochs de plus sur notre modèle InceptionV3 . .	70
4.15	Résultats d'entrainement de 48 couches pour 10 epochs	70
4.16	Résultats d'entrainement de 68 couches pour 10 epochs.	71
4.17	résultats d'entrainement de 78 couches pour 10 epochs.	71
4.18	Resultats de quelques tests d'images avec Model InceptionV3 personnalisé . . .	72
4.19	Logo Annaconda.	72
4.20	Logo Jupyter.	73
4.21	Logo Python.	73
4.22	Logo Tenserflow.	73
4.23	Drone quadrirotor avec ses composants.	78

Liste des tableaux

1.1 illustrant une des Classification des drones[d08] 12

Abréviations

ESC Electronic Speed Controller.
HALE High-Altitude Long Endurance.
ICAO International Civil Aviation.
KNN K-Nearest Neighbor.
MALE Medium Altitude Long Endurance.
MEMS Micro electromechanicals.
RPAS Remotely Piloted Aerial System.
RTL Return to launch .
SVM Support vector Machine .
TUAV tactical unmanned aerial vehicle.
UAS Unmanned Aerial System aéronef non habité.
UAVs Unmanned Aerial Vehicle.
UCAV unmanned combat air vehicle.
WP Waypoint .

Introduction générale

Ces dernières années le monde a assisté à la naissance de la robotique aérienne comme un des domaines des systèmes embarqués, qui est un des domaines de recherche le plus actif. Les véhicules aériens autonomes (UAVs en anglais) ou drones aériens, suscitent un très grand intérêt, et ceci s'accroît au fil du temps. Loin d'être fortuite, cette raison est due au fait que les UAVs offrent un grand potentiel pour aider et même voir remplacer l'être humain dans divers domaines d'applications. De grandes avancées technologiques dans le domaine de la robotique, de la miniaturisation et de l'électronique ont vu le jour, d'où la conception et la commercialisation de dispositifs UAV efficaces, à coût raisonnable, et dotés d'importantes capacités, ce qui a rendu la navigation autonome possible. En effet, de nombreuses possibilités d'applications aussi bien pour les professionnels que pour les amateurs s'offrent à nous, entre autres : la cartographie, la prise de vue aérienne, la transmission vidéo, les applications militaires (reconnaissances, combats, surveillances, déminage...), l'Agriculture moderne (pour recueillir des données précises), l'environnement, la livraison des colis, la Modélisation des sols et les bâtiments en 3D, la diffusion Wi-Fi, l'intervention dans des milieux hostiles (zones radioactives, catastrophes naturelles ou zones inaccessibles pour les humains), et tout récemment en Chine et en Russie des « UAV – Taxi ». Les applications pour les prochaines années à venir seront encore plus nombreuses et vont certainement toucher davantage de domaines, au regard des nombreuses recherches technologiques qui y sont dédiées, dans divers endroits du monde.

L'une des problématiques dans le domaine de la robotique aérienne est la conception d'un drone autonome « suivant un plan de vol préétabli et évitant les obstacles » Ce type de drone pourra être utilisé dans des applications diverses civiles et militaires. Par exemple, on peut imaginer un drone qui permet d'explorer un immeuble ou un tunnel contaminé et de réaliser une première observation avant toute intervention humaine.

Dans le cadre de ce travail, nous allons utiliser ce drone dans une application de détection précoce de feux de forêts. Le drone est configuré pour réaliser une mission de reconnaissance de feux. Il prendra des photos et les envoie au centre de traitement. Une application implémentée au centre de traitement examinera les photos et détectera les dépôts de feux éventuels.

*Chapitre 1 : **Etat de l'art sur les drones*** : Dans ce chapitre nous parlerons de ce qu'est un drone en montrant ses différentes parties le constituant ainsi que l'historique (le parcours) d'évolution des drones, et aussi nous allons énumérer les différentes classifications des drones.

*Chapitre 2 : **Principe de fonctionnement d'un drone quad-copter*** : dans ce chapitre seront présentés les bases de fonctionnement d'un quad-copter, les modes de vol possibles, et les composants électroniques qui permettent de concevoir un.

*Chapitre 3 : **Planification de missions de drones*** : Sera consacré à la présentation de la première partie de notre travail : la configuration et la mise en marche de notre quadcopter après l'avoir assemblé et comment nous lui affecterons une mission .

Chapitre 4 : Détection des feux de forêts : dans ce dernier chapitre on parlera de l'objectif principal de notre drone, qui est de répondre à une problématique existante tel que « prévenir les feux de forêt » en développant un modèle intelligent de reconnaissance de feux de forêt.

Chapitre 1

Etat de l'art sur les drones

1.1 Introduction

Le développement des UAVs, est la résultante des progrès scientifiques et technologiques opérés en matière de propulsion, de matériaux, de l'électronique, des ordinateurs, des capteurs, des instruments de navigation, etc.

De nos jours les UAVs jouent un rôle très prépondérant dans l'aviation moderne. En effet, les nouvelles générations des drones, aptes à effectuer un vol stationnaire, sont conçus pour être petits, plus légers et avec certaines fonctions d'autonomies.

Ce chapitre porte sur la revue de littérature. Il définit ce qu'est un drone, décrit les différentes classes des drones et leurs méthodes de navigation, ainsi que les domaines d'utilisation des UAVs et le règlement lié à leurs utilisations.

1.2 Définition d'un drone

Un drone ou UAV est un aéronef¹ sans pilote à bord, et qui peut voler de façon autonome ou être contrôlé à distance depuis le sol, par le moyen d'un système de commande embarqué (radiocommande, smartphone...). Le mot « drone » est une extrapolation d'un terme anglais qui signifie « faux-bourdon » (le mâle de l'abeille). En français, le terme est employé pour désigner des véhicules aériens, terrestres, de surface ou sous-marins, alors que la classification anglo-saxonne distingue tout type d'appareil.[SAD17]

Il y a aussi d'autres appellations pour les drones comme UAS ou encore RPAS qui est utilisé par ICAO² qui est une organisation qui dépend des Nations unies qui a pour rôle de participer à l'élaboration des normes pour la standardisation du transport aérien international.

Les drones ont plusieurs formes et configurations selon la mission qu'ils réalisent. En revanche, ils ont les mêmes parties constituant leur structure qu'on peut représenter comme suit :

1. tout appareil capable de s'élever ou de circuler dans l'air.

2. International Civil Aviation Organization qui est une organisation qui dépend des Nations unies qui a pour rôle de participer à l'élaboration des normes pour la standardisation du transport aérien international. Le lien du site Web : <http://www.icao.int>.

- **Les moteurs**

Ils génèrent l'énergie nécessaire à la portance du drone dans l'air en considérant la masse totale, l'altitude et l'endurance du vol. On trouve plusieurs types de moteurs [1.1](#) tels que les turboréacteurs, les turbines à hélices, moteurs à pistons ou encore des moteurs électriques.



FIGURE 1.1 – Quelques moteurs

- **Les voilures**

Ils assurent le maintien du drone dans l'air ils peuvent être fixes (des ailes) ou tournantes (des hélices) pour l'exploitation en vol stationnaire [1.2](#).



FIGURE 1.2 – Exemples de voilures.

- **La cellule**

Elle porte les moteurs, les systèmes de bord et la charge utile ainsi que le carburant ou la batterie [1.3](#).



FIGURE 1.3 – Exemples d'une cellule.

- **Les systèmes de bord**

Constituée de plusieurs équipements qui sont asservis entre eux, tels que les capteurs pour mesurer les paramètres du vol, des calculateurs qui sont dédiés au pilotage et la navigation, une mémoire contenant la partie programmable du vol, ainsi que des actionneurs qui agissent sur les commandes de vol.

- **La charge utile**

C'est des équipements (capteurs, processeurs, calculateurs, etc.) Spécifiques à la réalisation de la mission, ils assurent trois fonctions qui sont :

1. L'acquisition des données de l'environnement en utilisant des capteurs spécifiques au besoin de la mission, ils peuvent être de type électromagnétique (Radar), électro-optique (caméras), etc.
2. Traitement des données à bord grâce à des calculateurs en vue de leur exploitation en vol ou au sol.
3. La sélection des informations utiles à bord pour les transmettre à la base de contrôle en temps réel ou bien les enregistrer pour un déploiement ultérieur.



FIGURE 1.4 – Exemple d'une caméra.

- **Le Système de transmission**

Il assure l'envoi des données à la base de contrôle et l'acheminement de ces informations, ainsi que la réception des ordres de commande émis par la station de contrôle. La transmission est assurée par la télécommunication de deux manières :

1. En utilisant un relai, qui peut être un autre vecteur aérien (drone ou avion) ou un satellite.
2. En utilisant la portée optique sur des courtes distances (jusqu'à 150 km) qui peut être une diode LED de forte puissance ou un Laser.

- **Intelligence embarquée (dite aussi intelligence Artificielle)**

Elle est constituée d'un ou plusieurs algorithmes implémentés sur le drone accompagné des calculateurs. En fonction des informations captées par les capteurs qui sont à bord du drone, l'intelligence embarquée offre un certain degré d'autonomie au drone vis-à-vis du pilotage et de la navigation d'une part, et la réalisation de la mission d'une autre part.

L'usage des UAVs nécessite une interaction en temps réel avec l'environnement d'évolution ce qui fait que la stabilité et l'optimisation sont des caractéristiques très importantes pour les algorithmes d'intelligence artificielle.

- **Le Système de drones**

C'est l'ensemble des éléments nécessaires pour mettre en œuvre un drone 1.5, ce système est composé de deux (02) segments :

1. Le segment air : C'est le drone, sa charge utile ainsi que son système de transmission.
2. Le segment sol : C'est la base de contrôle, les personnes qui interviennent, ainsi que le matériel dédié au lancement et la récupération du drone tels que les filets et les rails de lancement, le matériel dédié à la maintenance, à la réparation et à la conduite de la mission à partir du sol en assurant la gestion du vol et la navigation, la réception des données, leur décryptage, leur analyse et leur transmission au centre de calcul.

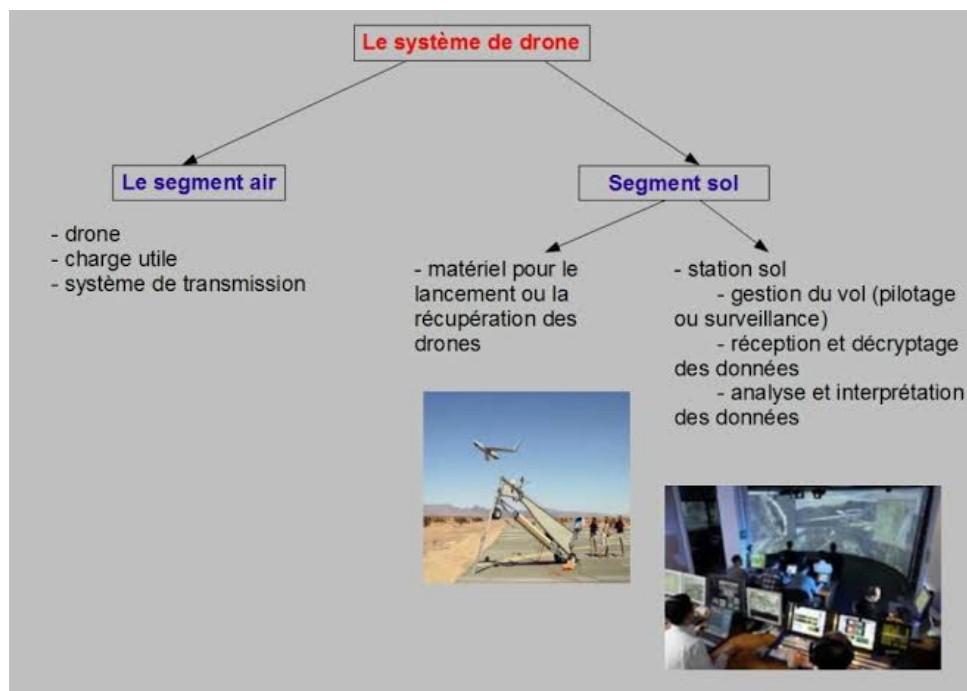


FIGURE 1.5 – Le système de drone.

1.3 Evolution des drones

La nature est la plus grande et vaste source d'inspiration pour la science, et cela s'applique aussi sur les UAVs car les scientifiques se sont inspirés des oiseaux et les insectes volants comme les abeilles et les libellules (figure 1.6)



FIGURE 1.6 – Une libellule.

Ceci dit, quelle que soit la définition ou l'acronyme utilisé, l'histoire nous apprend que le premier drone fut un ballon gonflable développé autour des années 1850. En voici un résumé qui donne l'évolution des drones :

- **Avant 1900** : Des ballons sont utilisés pour lâcher des bombes sur Venise (Italie) par l'Autriche et durant la guerre civile américaine.
- **1910 - 1920** : Les États-Unis produisent le drone « Kettering Bug » durant la première guerre mondiale.
- **1920 - 1930** : Quelques avions de type « Sperry Messenger » sont convertis en bombes volantes et deviennent ainsi les premiers vrais avions téléguidés.
- **1930 - 1940** : Les Britanniques produisent 400 avions téléguidés comme cible de pratique pour les défenses anti-aériennes. Il est nommé « DH.82 Queen Bee » [1.7](#) qui a la particularité de revenir à la base de contrôle d'une façon autonome. DH.82 était le premier avion qui porte le nom « drone » justement [[Lev15](#)].



FIGURE 1.7 – DH 82 Queen Bee.

- **1940 - 1950** : Pendant la deuxième guerre mondiale, Reginald Denny convainc l'armée américaine d'utiliser ses avions téléguidés les « Radioplanes OQ-2 » pour entrainer leur défense anti-aérienne. Les Allemands produisent le bombardier « V-1 » dans le but de bombardier Londres.
- **1950 - 1960** : Le « FireBee » est développé durant les guerres de Corée et du Vietnam pour les missions de surveillance confiées à des drones.
- **1960 - 1970** : L'armée américaine commence la production de son premier drone furtif, le « AQM-34 Ryan FireBee ». La Central Intelligence Agency (CIA) demande la production

d'un drone ultrasecret capable de voler à des vitesses de Mach 3.3 et des altitudes de plus de 27 000m, le « D-21 ».

- **1970 - 1990** : Les forces aériennes israéliennes déploient de nouveaux drones militaires avancés, les plus connus étant les « FireBee1241 » « Scout » et le « Pioneer ».



FIGURE 1.8 – Illustration d'un drone FireBee.

- **1990 – 2000** : Une variété de nouveaux drones sont développés pour des applications militaires et non-militaires comme les drones à énergie solaire « Helios » et « Pathfinder ». C'est aussi la décennie de la mise en service du célèbre « Predator ».
- **2000 - maintenant** : Plus de 32 pays développent des drones. Les nouveaux drones apportent de nombreuses idées et utilisations innovatrices, plusieurs catégories de drones (militaire et civil) partant de quelques grammes à plusieurs tonnes en différentes formes selon les utilisations et missions pour lesquelles ils sont destinés. Quelques exemples sur la figure 1.9. Il suffit de penser au « Fire Scout », au « Yamaha R-MAX », au « Raptor/Talon », au « Global Hawk », au « Black Widow » etc.



FIGURE 1.9 – Les différents types de drones civils et militaires.

1.4 Classification des drones

Les drones peuvent être répartis selon plusieurs critères qui varient selon les pays. Cependant les drones peuvent être classés en plusieurs catégories selon l'altitude, l'endurance qui est le temps que peut passer l'aéronef en vol, la taille ou encore leur voilure. etc. [Poi08]

1.4.1 Selon la taille

On distingue les drones Haute Altitude Longue Endurance (HALE), Moyenne Altitude Longue Endurance (MALE), micro drones et mini drones. Ces catégories sont résumées ci-après :

HALE

Pour “High-Altitude Long Endurance”, ce sont des drones de grande taille, le plus souvent à voilure fixe. Ils sont capables de rester très longtemps en vol et de collecter des informations sur de très longues périodes (entre 12 et 48 heures).



FIGURE 1.10 – Exemple d’un drone de type HALE.

MALE

Pour “Medium Altitude Long Endurance”, sont utilisés pour des vols de longue durée à moyenne altitude opérationnelle, ayant une grande autonomie.



FIGURE 1.11 – Exemple d’un drone de type MALE.

Ces deux types de drones font partie de la classe de grande taille. Ils peuvent embarquer des armes, ce qui nécessite généralement d’avoir un humain dans la boucle, ce dernier doit garder la décision de tir et pouvoir à tout moment annuler la mission.

Mini drones

Ce sont des drones légers et de taille réduite (jusqu’à quelques kilogrammes et d’une envergure allant jusqu’à 1 à 2 mètres) ils ont une autonomie relativement faible (de 10 à 30 minutes) et généralement utilisés pour l’observation de zones difficiles à l’accès.

Micro drones

Ce sont des drones ayant des tailles variant du centimètre à quelques dizaines de centimètres. Généralement propulsés électriquement. Ils permettent de faire des vols à l'intérieur. Ils emportent de faibles charges.



FIGURE 1.12 – Exemple d'un Micro quad-copter.

1.4.2 Selon le mode de propulsion :

Les drones à voilures fixes :

Les drones de cette famille sont des drones utilisant les ailes fixes dans leur mode de déplacement (type avion qui est plus lourd que l'air ou type dirigeable qui est plus léger que l'air). La propulsion est assurée par une ou plusieurs hélices. Parmi les drones à voilure fixe on trouve les HALE, MALE, UCAV³, TUAV⁴.



FIGURE 1.13 – Exemple d'un drone à voilure fixe.

Les drones à voilure tournante

Ce type est caractérisée par le décollage et l'atterrissage vertical, capables de faire du vol stationnaire à basse vitesse et à faible altitude, ils utilisent un ou plusieurs rotors, ils peuvent

3. unmanned combat air vehicle.

4. tactical unmanned aerial vehicle.

réaliser des missions irréalisables par les véhicules à voilure fixe. Ce type de drones englobe plusieurs catégories :

- **Mono rotor** : se compose d'un rotor principal avec barre stabilisatrice qui permet la sustentation et la propulsion.



FIGURE 1.14 – Exemple d'un mono rotor.

- **Birotors contrarotatifs** : se composent de deux rotors tournant en sens opposés et à même vitesse qui permettent la sustentation, la translation.
- **Les quadrirotors** : se compose d'une armature en croix symétrique avec des moteurs et des rotors aux extrémités de chaque tige.



FIGURE 1.15 – Exemple d'un quadri rotor.

- **Les convertibles** : sont des machines munies d'un mécanisme de basculement des rotors qui permet à la fois le vol stationnaire, le décollage et l'atterrissage vertical dans des zones restreintes et difficiles, Le principal inconvénient de ces appareils c'est l'instabilité lors de la phase de transition entre le vol en mode avion et en mode hélicoptère.
- **Les drones à ailes battantes** : Dans ce type le pilotage et la sustentation sont assurés par des ailes battantes comme dans les insectes tel que les mouches, les abeilles... etc. Ces engins permettent des manœuvres agiles car ils ont la capacité de vol avec des trajectoires similaires à celles des insectes et de vol stationnaire à basses vitesses, comme les voilures tournantes, ce qui est intéressant pour des missions dans des espaces réduits.

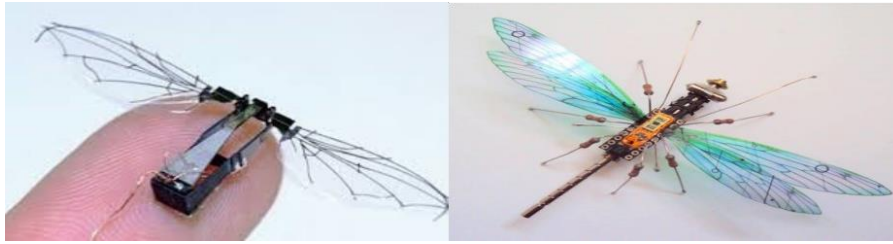


FIGURE 1.16 – Exemple d'un drone à ailes battantes.

1.4.3 Selon le poids maximum au décollage, l'altitude maximale et l'endurance :

Dans le tableau suivant (Tableau 1.1) la classification est basée sur différents paramètres comme le poids maximum au décollage, l'altitude maximale et le temps de vol maximum. Ce tableau est repris de [d08] où l'auteur nous présente un sommaire complet des quatre principales catégories de drones et de leurs acronymes.

Catégorie	Acronyme	Poids maximum au décollage (kg)	Altitude de vol maximale (km)	Endurance (h)	Exemple
Micro/Mini drones	MAV	0.10	0.25	1	<i>Microbat</i>
	Mini	< 30	0.15 - 0.3	< 2	<i>Aladin</i>
Drones tactiques	CR	150	3	2 - 4	<i>R-Max</i>
	SR	200	3	3 - 6	<i>Phantom</i>
	MR	150 - 500	3 - 5	6 - 10	Hunter B
	LR	-	5	6 - 13	<i>Vigilante 502</i>
	EN	500 - 1 500	5 - 8	12 - 24	<i>Aerosonde</i>
	MALE	1 000 - 1 500	5 - 8	24 - 48	<i>Predator-IT</i>
Drones Stratégiques	HALE	2 500 - 12 500	15 - 20	24 - 48	<i>Global Hawk</i>
Drones spécialisés	LET	250	3 - 4	3 - 4	<i>Harpy</i>
	DEC	250	0.05 - 5	< 4	<i>Nulka</i>
	STRATO	À déterminer	20 - 30	> 48	<i>Pegasus</i>
	EXO	À déterminer	> 30	À Déterminer	<i>MarsFlyer</i>

TABLE 1.1 – illustrant une des Classification des drones[d08]

Les mini et micro drones (Micro-Air Vehicle (MAV)) sont les plus petits parmi les quatre, ils sont surtout utilisés pour des applications civiles, scientifiques ou commerciales. Les drones tactiques sont utilisés pour les applications militaires, ils incluent les drones à portées rapprochées (Close Range (CR)), à portées courtes (Short Range (SR)), à portées moyennes (Medium

Range (MR)), à longues portées (Long Range (LR)), d'Endurance (EN) et à Moyenne Altitude et Longue Endurance (MALE). Ensuite, il y a les drones stratégiques qui sont beaucoup plus lourds et gros que ceux des catégories précédentes. Ils sont conçus pour les applications en Haute Altitude et Longue Endurance (HALE). La dernière catégorie est celle des drones spécialisés, ceci inclut les drones Létaux (LET), les leures (Decoys (DEC)), les Stratosphériques (STRATO) et les Exo- stratosphériques (EXO).

1.5 Méthodes pour la navigation

L'un des problèmes les plus important dans la commande des drones est la navigation autonome, un ensemble de techniques localisant le véhicule sans son environnement son objectif de lui permettre de maîtriser ses déplacements et cela d'une façon autonome. Pour cela plusieurs approches de navigation autonome peuvent être employées :

1.5.1 La navigation par suivi de terrain

Dans cette approche le véhicule est maintenu à une altitude fixe par rapport au sol sur le long de la mission comme le montre la figure 1.17, en utilise généralement des RADAR⁵ embarqué à bord du véhicule ce qui lui permet de s'ajuster et suivre le relief au plus près, une caméra remplace le RADAR généralement dans les véhicules de plus petite taille. Cela lui permet de détecter s'ajuster aux différentes variations du terrain.

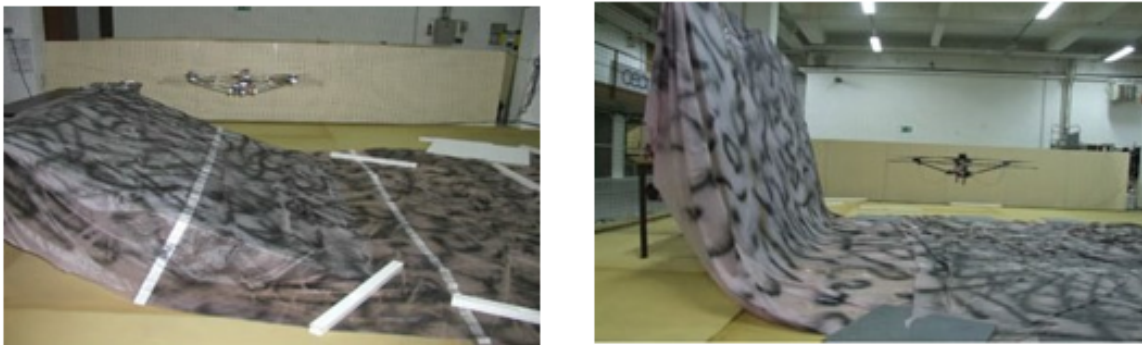


FIGURE 1.17 – Navigation par suivi de terrain par flux optiques.

1.5.2 La navigation par points de passage

Appelée aussi navigation proportionnelle, dans cette méthode on définit des points de références dont la position sur les trois dimensions est définie à priori [Adr13]. Elle est simple à mettre en œuvre mais n'est efficace que lorsque la distance entre l'engin et la cible est suffisamment grande [Pfl06](figure 1.18) est un exemple. La navigation par suivi de trajectoire est issue de cette approche, ou on a le nombre de points de références est très grand voir même infini et le principe est de suivre une cible repérée jusqu'à atteindre l'objectif.

5. Appareil qui permet de déterminer la position et la distance d'un objet par l'émission d'ondes radioélectriques et la détection des ondes réfléchies à sa surface.

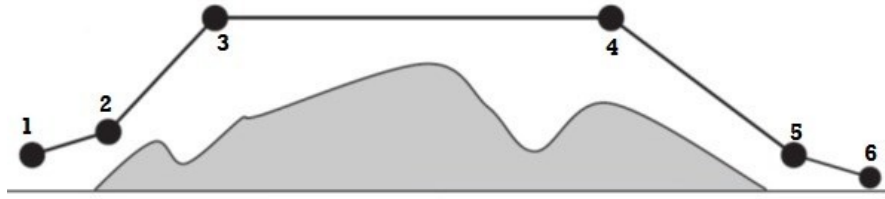


FIGURE 1.18 – Navigation par points de passage .

1.5.3 L'évitement d'obstacles

Aujourd'hui, les drones sont au milieu urbain ce qui implique des stratégies de contournement efficaces dans le plan horizontal ainsi que la détection d'obstacles imprévus et la réaction en temps réel[Pf06], pour cela le véhicule doit disposer d'un ensemble d'équipement électronique et des algorithmes performants

1.6 Les domaines d'utilisation

Les drones aujourd'hui font Object dans de nombreux domaines d'activité vu qu'il ne cesse de se rallonger, deux principales utilisations se distingue :

1.6.1 Utilisation militaire des drones

L'historique précédent nous montre que la guerre a longtemps été la motivation principale pour la conception des drones. Le fait que l'utilisation d'un drone prévienne des pertes humaines inutiles a mené les drones à adopter toutes sortes de missions militaires. En voici une liste :

1. Reconnaissance visuelle, surveillance et identification de cibles.
2. Surveillance avec un radar à synthèse d'ouverture, en temps de paix ou pour le combat.
3. Opérations de déception.
4. Opérations maritimes (support au feu, acquisition de cibles au-delà de l'horizon, classification de navire).
5. Soutien à l'itinéraire et à la reconnaissance pour l'atterrissage.
6. Ajustement du feu indirect et appui aérien rapproché.
7. Confirmation de dommages dus au combat.
8. Opérations spéciales et psychologiques.
9. Mission de météorologie.
10. Guerre électronique et espionnage.
11. Relais radio.



FIGURE 1.19 – Utilisation des drones dans le secteur militaire .

1.6.2 Utilisation civile des drones

Dans les deux dernières décennies, l'utilisation des drones civils s'est étendue rapidement comme présenté sur les figures (1.19 et 1.20) et a permis d'ouvrir une croissance considérable dans des secteurs variés et innovant Voici quelques domaines d'applications civils :

1. Surveillance des frontières.
2. Recherche et sauvetage dans cas de :
 - a. Accidents d'avions.
 - b. Naufrages.
3. Détection des feux de forêt



FIGURE 1.20 – Quelques domaines d'utilisation civile des drones .

4. Application de la loi dans cas
 - a. Contrôle de foule.
 - b. Détection de culture de drogue.
 - c. Relais de communication.
5. Gestions d'urgences et de désastres cas de :

- a. Tremblement de terre.
 - b. Inondations.
 - c. Désastres nucléaires.
 - d. Déversements de pétrole.
6. Recherche scientifique dans différent Domain et sujet tel :
- a. Environnementale.
 - b. Atmosphérique.
 - c. Archéologique.
 - d. Pollution.
 - e. Recensement de la faune.
 - f. Étude des glaces.
7. Application dans le secteur de l'industrie :
- a. Épandage des récoltes.
 - b. Cinéma.
 - c. Photo aérienne, cartographie.
 - d. Reportage pour les canaux de nouvelles.
 - e. Publicité aérienne.
 - f. Sécurité.
 - g. Transport de fret.
 - h. Surveillance des centrales nucléaires, ou des pipelines.



FIGURE 1.21 – Quelques domaines d'utilisation civile des drones .

1.7 Règlementation sur l'utilisation des drones

L'usage des drones, fait l'objet d'une réglementation dans plusieurs pays. Par exemple en Algérie, la loi relative aux drones inclut des dispositions qui portent à la fois sur le droit civil ainsi que le droit pénal. D'autre part, le pilotage des drones est également encadré par le code des transports et le code de l'aviation civile [Rid15]. Tant pour la sécurité des aéronefs que celle des personnes au sol, ou encore pour le respect de la vie privée et du droit à l'image.

Il est très important que les pilotes connaissent et appliquent ces textes. Les fabricants de drones ont l'obligation de fournir une notice d'information concernant les bonnes pratiques et règles à respecter pour l'usage d'un drone avec caméra ou non.

1.8 Conclusion

Dans ce chapitre nous avons découvert que le domaine des drones est un domaine très vaste se base sur la discipline de la robotique, s'évolue rapidement et les différentes catégories des drones existants prouve cette énorme évolution. La réalisation de ce genre de projet nécessite principalement une modélisation de sa dynamique pour étudier les différents mouvements qu'un drone peut faire avant d'entamer la phase de l'implémentation ce qui sera question dans les chapitres suivant.

Chapitre 2

Principe de fonctionnement d'un drone quad-copter

2.1 Introduction

Un quad-copter est un avion soulevé et propulsé par quatre moteurs horizontaux. Les quad-copter sont classés comme giravions¹ ou comme aéronefs à voilure tournante pour les distinguer des aéronefs à voilure fixe, car le quad-copter tire sa source de portance des pales de rotor tournant autour d'un mât. Ils appartiennent à une catégorie plus large d'aéronefs définis comme des multi-copter.

Parmi les différents types des UAV qui existent les quad-copter sont très utilisés dans beaucoup de domaines de notre vie quotidienne. Ce dernier est classé dans la catégorie des systèmes volants les plus complexes vu le nombre d'effets physiques qui affecte son fonctionnement :

- Les effets aérodynamiques.
- La gravité.
- Les effets gyroscopiques.
- Les frottements et le moment d'inertie.

Dans ce chapitre, nous allons définir dans un premier temps qu'est-ce un quad-copter, son principe de fonctionnement, le matériel utilisé pour la mise en œuvre de notre drone, et ce qu'on a utilisé pour établir la communication à la station de base et par la suite les modes de vols.

2.2 Description générale d'un quad-copter

Un quad-copter est un robot mobile aérien à quatre moteurs. Ces 4 moteurs sont généralement placés aux extrémités d'un châssis ou cadre, et l'électronique de contrôle est habituellement placée au centre de ce châssis ou cadre. Afin d'éviter à l'appareil de tourner sur lui-même sur son axe de lacet, il est nécessaire que deux hélices tournent dans un sens, et les deux autres dans l'autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélice tournant dans le même sens soit placé aux extrémités opposées d'une branche du châssis ou du cadre.

1. Aéronef dont la sustentation est assurée par des voilures tournantes.

Le fonctionnement d'un quad-copter est assez particulier. En faisant varier astucieusement la puissance des moteurs, il est possible de le faire monter/descendre, de l'incliner à gauche/droite (roulis) ou en avant/arrière (tangage) ou encore de le faire pivoter sur lui-même (lacet) (figure 2.1).

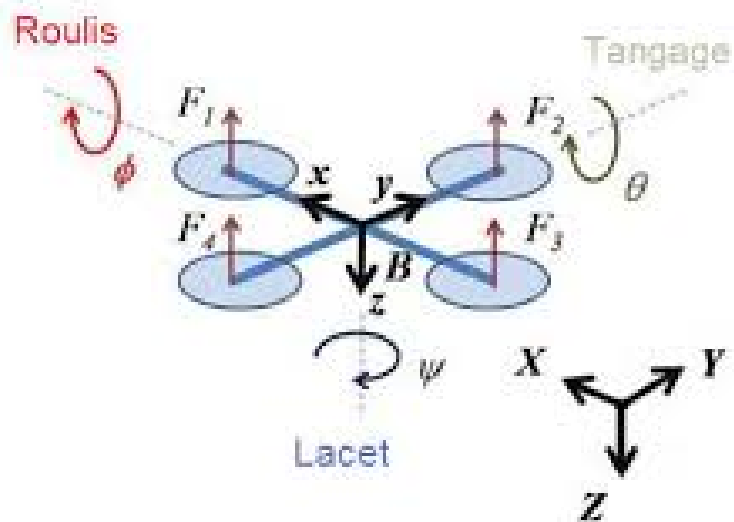


FIGURE 2.1 – Structure générale d'un quad-copter.

2.3 Fonctionnement

Le fonctionnement d'un quad-copter est assez particulier. On distingue quatre mouvements possibles : le vertical, le lacet, le roulis et tangage (respectivement en anglais : throttles, yaw, roll and pitch).

2.3.1 Sens de rotation des voilures tournantes

Afin d'éviter un mouvement de lacet (l'appareil tourne sur lui-même), il est nécessaire que deux moteurs tournent dans un sens et que les deux autres tournent dans l'autre sens. En plus, pour diriger l'appareil, les moteurs tournant dans le même sens doivent être placés l'un en face de l'autre (figure 2.2).

De plus, pour faciliter la gestion des commandes et les calculs, un des rotors (le rotor numéro 1 dans la figure 2.2) est choisi comme l'avant du quad-copter.

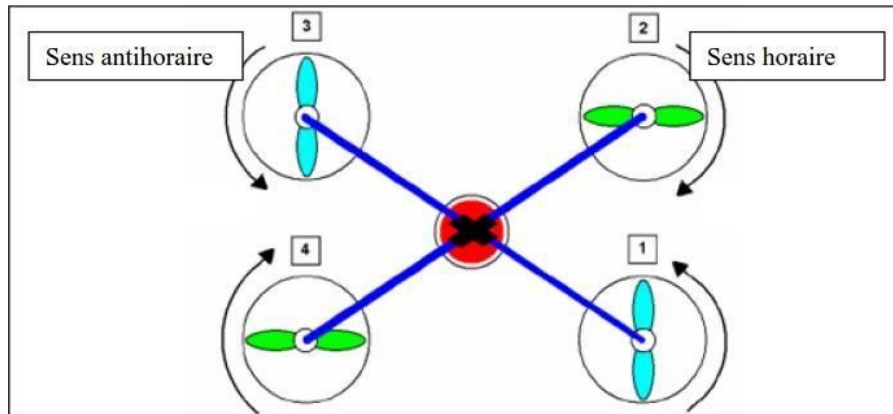


FIGURE 2.2 – Sens de rotation des hélices.

Le mouvement vertical (Throttle)

Le mouvement vertical correspond tout simplement à la montée/descente du quad-copter.

La montée est obtenue en augmentant la vitesse des quatre moteurs. La descente s'obtient par la réduction de la vitesse des moteurs.

Le mouvement lacet (Yaw)

Le mouvement de lacet sert à faire tourner le quad-copter sur lui-même. Il est obtenu en augmentant la vitesse des hélices à pas normal et en diminuant proportionnellement la vitesse des hélices à pas inversé.

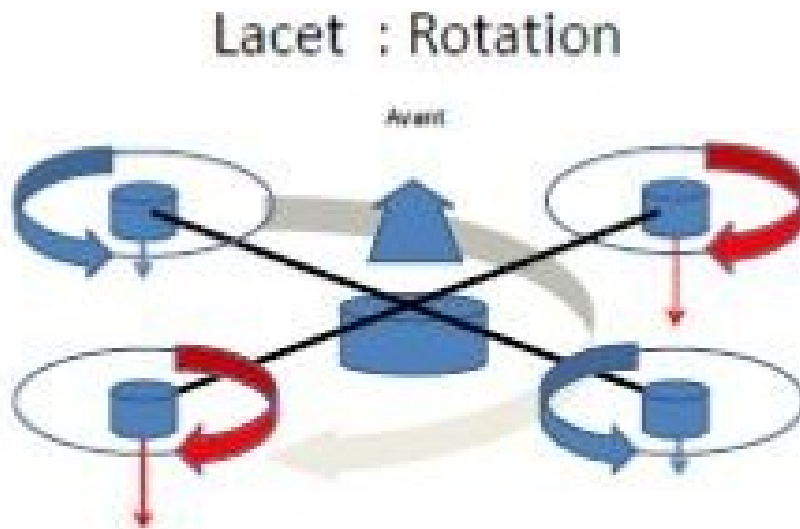


FIGURE 2.3 – Illustration le mouvement lacet "YAW".

Le mouvement roulis et tangage (Roll and Pitch)

Le roulis et le tangage sont des mouvements assez similaires visant à pencher le quad-copter sur un axe ou sur un autre.

Ce mouvement est obtenu en augmentant la vitesse d'une hélice et en abaissant proportionnellement la vitesse de l'hélice opposée (hélice du même couple).

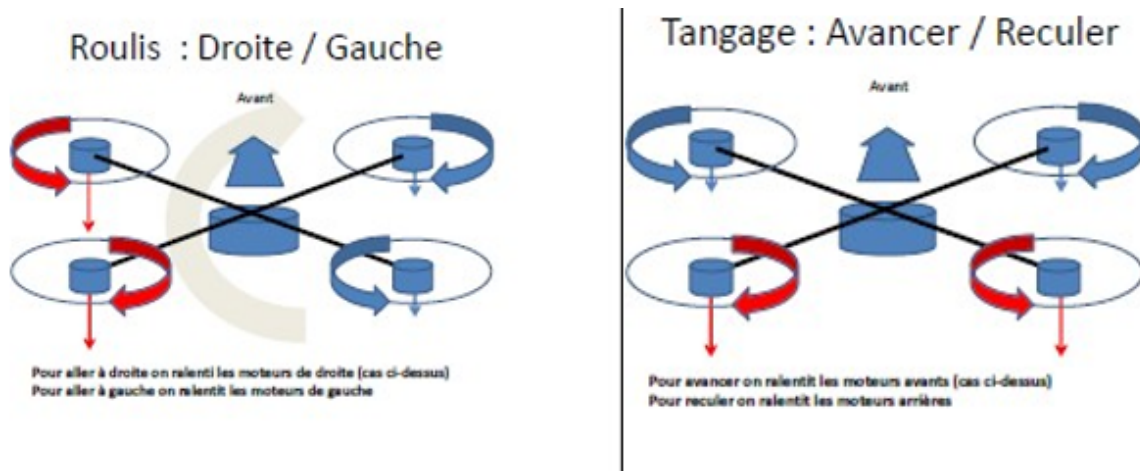


FIGURE 2.4 – Illustration mouvement roulis et tangage.

2.4 Composants électroniques d'un quad-copter

La fabrication d'un quad-copter nécessite un nombre important de composants. Nous pouvons citer entre autres :

- Le châssis,
- 04 ESC (contrôleurs électroniques des moteurs brushless),
- 04 moteurs brushless,
- 04 hélices (02 pour le sens horaire et 02 autres pour le sens anti-horaire),
- Le contrôleur de vol,
- Des capteurs (accéléromètres, GPS ... etc),
- Batterie pour l'alimentation électrique,
- 01 récepteur selon la radiocommande utilisée,
- Eventuellement une caméra embarquée,

Dans notre cas on a utilisé le type de composants tels :

Les Contrôleurs de vitesse électroniques ESCs

Étant donné que chaque moteur du quad-copter doit tourner à différentes vitesses, chaque moteur à son propre ESC, Un contrôle de vitesse électronique ou ESC est un circuit électronique qui a pour but de faire varier la vitesse d'un moteur électrique, sa direction et éventuellement aussi pour agir comme un frein dynamique. Les ESC sont généralement connectés aux batteries via un tableau de distribution d'alimentation dans le châssis du multirobot et reçoivent un signal d'entrée du contrôleur de vol.

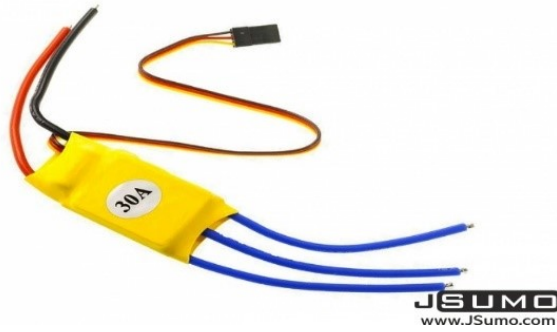


FIGURE 2.5 – illustration d'un ESC.

Moteur

Permettent au quad-copter d'avoir la portance nécessaire pour voler. Il existe des centaines de modèles plus ou moins adaptés à l'utilisation. Dans notre cas, les moteurs choisis sont des moteurs Brushless (sans balais) 1000 KV.

Les moteurs sans balais pour drones utilisent une convention de dénomination qui transmet leurs caractéristiques.

Par exemple, "A 22121000 Kv" signifie ce qui suit :

22 est le diamètre du moteur.

12 est la hauteur de la boîte (conteneur) du moteur.

1000 est le nombre de tours par minute, à 1 Volt, sans charge.

« KV » fait référence à la vitesse constante d'un moteur (à ne pas confondre avec « kV », l'abréviation de kilovolt).

Il est mesuré par le nombre de tours par minute (tr / min) qu'un moteur tourne quand 1 V (un volt) est appliqué sans charge attachée à ce moteur. Plus la valeur nominale en kV est élevée, plus les moteurs tournent rapidement à une tension constante.



FIGURE 2.6 – Motors.

Microcontrôleur :

Le contrôleur de vol est l'esprit ou « cœur » du drone. C'est lui qui va stabiliser l'engin en contrôlant le firmware dont les ESC qui contrôlent la vitesse à laquelle chaque moteur tourne. Mais aussi effectuer différentes tâches plus ou moins complexes comme suivre un plan de vol, atterrir automatiquement, etc. Le microcontrôleur prend les entrées d'un récepteur ou d'un

ordinateur compagnon et ajuste le régime du moteur en conséquence.

Les systèmes de contrôleur de vol contiennent également des capteurs supplémentaires qui permettent d'améliorer le contrôle et la stabilité. La plupart des contrôleurs de vol contiennent un gyroscope, un accéléromètre et un baromètre. Un module GPS est généralement ajouté à l'extérieur pour fournir des informations de coordonnées et d'altitude.

Il existe plusieurs cartes pour le contrôle du vol comme les cartes Raspberry, les cartes Arduino, les cartes STM. Dans notre cas nous avons utilisé une carte Pixhawk 2.4.8 spécialement conçue pour le contrôle de vol. Elle est décrite ci-dessus.

Pixhawk 2.4.8 :

Pixhawk Est Un contrôleur de vol de niveau intermédiaire qui a comme caractéristiques :

- Basé sur le processeur Cortex M4F 32 bits.
- 168 MHZ, 2 Mo de mémoire flash, 256 Ko de RAM.
- Coût bas et léger (38g).
- Micro logiciel via le planificateur de mission.
- Contrôle jusqu'à 14 moteurs PWM.
- Accepte jusqu'à 8 canaux radio.
- 5v UART, I2C, CAN, SPI. Gyroscope, accéléromètre, Baromètre magnétomètre.
- Alimentation et sauvegarde redondantes.
- Fonctionne avec le logiciel Mission Planner.
- Carte microSD pour l'enregistrement des données.



FIGURE 2.7 – illustration d'une Pixhawk 2.4.8.

Émetteur et récepteur

L'émetteur et le récepteur radio sont utilisés pour contrôler le quad-copter. Au minimum pour qu'un quad-copter fonctionne, quatre canaux (accélérateur, lacet, roulis, tangage) sont nécessaires.

Cependant, la plupart des émetteurs et récepteurs ont plus que les quatre canaux requis, ce qui permet des fonctionnalités supplémentaires, telles que la commutation entre les modes de

vol ou le contrôle d'un cardan de caméra embarqué.

Dans notre cas on a juste utilisé un module Radio télémétrie qui permet de communiquer avec la station de contrôle, il contient un module aérien et un autre terrestre.



FIGURE 2.8 – illustration du module télémétrie.

Batteries

Les batteries au lithium polymère (LiPo) semblent être la source d'alimentation la plus préférée pour alimenter les multicopters actuellement.

Cependant, pour les multicopters plus grands, même les batteries LiPo ne permettent pas d'étendre durée de vol.



FIGURE 2.9 – Batterie.

Émetteur et récepteur vidéo :

Le transmetteur vidéo TS832 assure l'envoi de la vidéo (l'information) captée par la caméra, l'information sera reçue par un récepteur RC832 qui est connecté à un écran avec une liaison vidéo.

Pour assurer l'envoi de la vidéo, il suffit de sélectionner la même fréquence et le même canal pour le transmetteur et le récepteur.



FIGURE 2.10 – Transmetteur TS832 et récepteur RC832.

2.5 Les capteurs pour la navigation et la localisation

Les drones embarquent différents capteurs assurant une navigation, une stabilité et une localisation dans l'environnement d'évolution, on distingue principalement deux catégories :

2.5.1 Capteurs proprioceptifs

Les capteurs de ce type permettent de mesurer la vitesse, la position et l'accélération de l'engin par rapport à un état de référence. Leur fonctionnement ne nécessite pas une connaissance de l'environnement dans lequel l'engin évolue. Cependant, ces informations se dégradent au cours du temps ce qui les rend inutilisable à une référence long terme [KOE12].

Les capteurs proprioceptifs les plus utilisés sont :

- **Les accéléromètres** : des capteurs qui calculent l'accélération linéaire² de l'engin sur les trois axes orthogonaux [KOE12].
- **Les Gyroscopes et Gyromètres** : les deux capteurs permettent de définir la vitesse et la position angulaire³ de l'engin dans l'air tel que le gyroscope mesure la position angulaire sur l'axe où il a été monté, par contre un gyromètre permet de mesurer la vitesse instantanée de rotation autour des axes du repère mobile (la vitesse de roulis, de tangage et lacet) [Rid15].
- **Centrales inertielles (IMU : Inertiel Measurement Unit)** : C'est un système de navigation fournissant l'altitude, la vitesse et la position de l'engin composée principalement de trois accéléromètres et trois gyromètres. La technologie MEMS (Micro-Electro-Mechanical Systems) est la plus utilisée actuellement puisqu'elle intègre des capteurs simples de petite taille tels que les accéléromètres, gyromètres ou encore les magnétomètres dans un circuit électronique comme illustré à la 2.11 [Rid15].

2. Le changement d'accélération d'un objet mobile selon une ligne orthogonale, le résultat est donné en m/s^2 .

3. c'est la vitesse de rotation d'un engin exprimée en Radians par secondes (rad.s^{-1}), plus souvent en tours par minute (tr/min).



FIGURE 2.11 – Une carte IMU.

Comme présenté précédemment, les informations acquises par les capteurs proprioceptifs ne sont pas utilisables à une référence de long terme, c'est pourquoi il est indispensable de les associer à un système permettant de recalibrer périodiquement la position absolue de l'engin et qui sont les capteurs extéroceptifs.

2.5.2 Capteurs extéroceptifs

Ce type de capteurs utilise les informations de l'environnement extérieur à l'engin (drone) dans le but de connaître sa situation par rapport à certains repères tel que le repère visuel. Ils sont nécessaires pour la localisation ainsi que pour la détermination de l'environnement dans lequel le drone évolue.

Les informations acquises par ces derniers sont indépendantes du déplacement de l'engin et l'erreur acquise par les capteurs proprioceptifs ce qui les rend utilisables dans des références à long terme, [KOE12].

Les plus utilisés dans cette catégorie sont :

- **Le compas magnétique (magnétomètre) :** Donne la référence de la direction (le nord) magnétique, fonctionne sur le principe de la boussole dont l'orientation dépend des lignes du champ magnétique terrestre.
- **Le gyrocompas :** Donne la référence de la direction nord géographique qui est toujours stable par rapport à celui magnétique.
- **Le compas satellitaire :** Basé sur le système de positionnement global (GPS) dont le principe consiste à avoir deux antennes aux extrémités de l'engin équipées des récepteurs GPS, ainsi, il permet de définir l'orientation du vecteur aérien.
- **Les capteurs télémétriques :** Permettent de connaître l'environnement direct du drone, ils mesurent la distance entre le plus proche objet et l'engin et cela par mesure

du temps entre l'émission d'une onde telle que l'onde acoustique (utilisation des ondes ultrasonore sur une courte distance), les ondes optiques (télémètres lasers qui utilisent des ondes lumineuses) et sa réception La 2.12 illustre un exemple d'un capteur extéroceptif.



FIGURE 2.12 – Capteur télémétrique ultrason pour mesurer la distance avec un objet distant.

2.5.3 Le système de géolocalisation

C'est le GPS (Global Positioning System) qui est un système de positionnement par balises actives (les satellites), composé de 24 satellites en orbite autour de la terre. Le système de géolocalisation est utilisé pour acquérir la position absolue de l'engin hébergeant un récepteur GPS à la surface du globe ou dans une région délimitée [KOE12].



FIGURE 2.13 – illustration du module GPS.

Une représentation visuelle des composants de base est illustrée à 2.14

BASIC COMPONENTS OF A QUADCOPTER

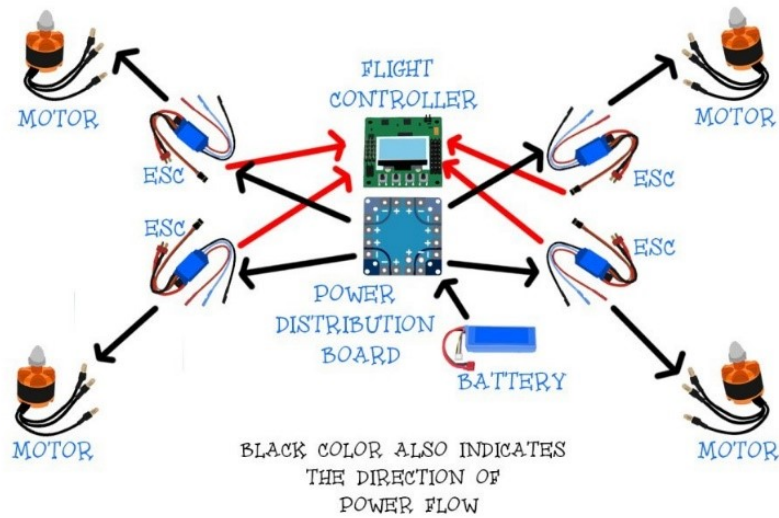


FIGURE 2.14 – Composants de base du quadricoptère.

2.6 Modes de vol :

En se basant sur les mouvements possibles, le drone peut effectuer trois modes de vol :

- Vol vertical.
- Vol stationnaire.
- Vol de translation.

2.6.1 Vol vertical

Quand les hélices appliquent Soient deux forces ayants la même direction mais de sens opposé, le quad-copter peut monter ou descendre, selon que suivant l'effet aérodynamique soit supérieur ou inférieur au poids de l'appareil.

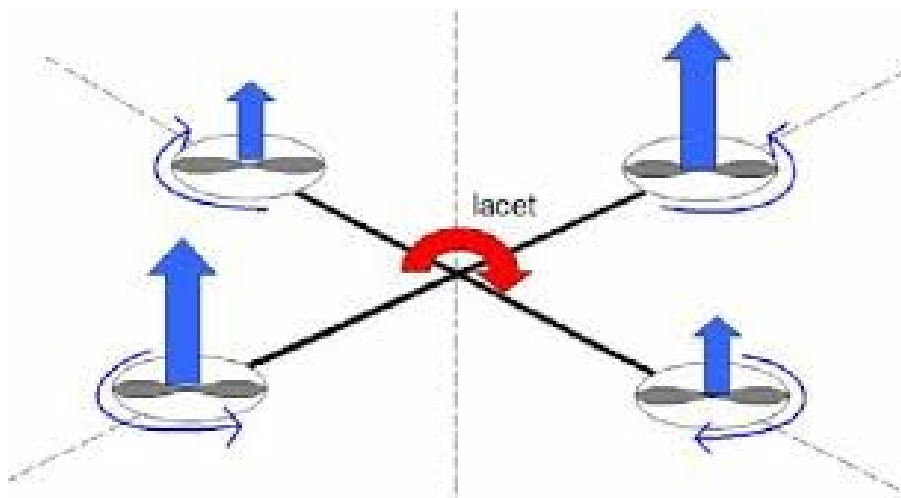


FIGURE 2.15 – Illustration du mouvement vertical.

2.6.2 Vol stationnaire

Quand la force de Portance, et celle de la pesanteur sont égales et opposées et que tous les moteurs tournent à la même vitesse : le quad-copter reste immobile. On parle de vol stationnaire.

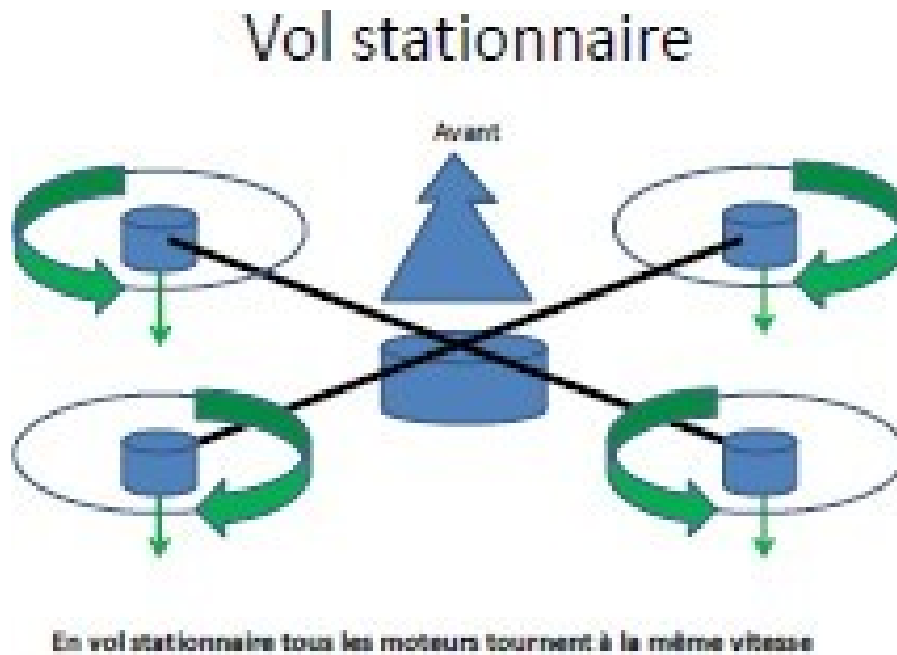


FIGURE 2.16 – Illustration du mouvement stationnaire.

2.6.3 Vol de translation

Le vol de translation correspond à la navigation de Quad-copter sur un plan horizontal. Il est assuré en se basant sur les mouvements d'inclinaison tangage, et roulis.

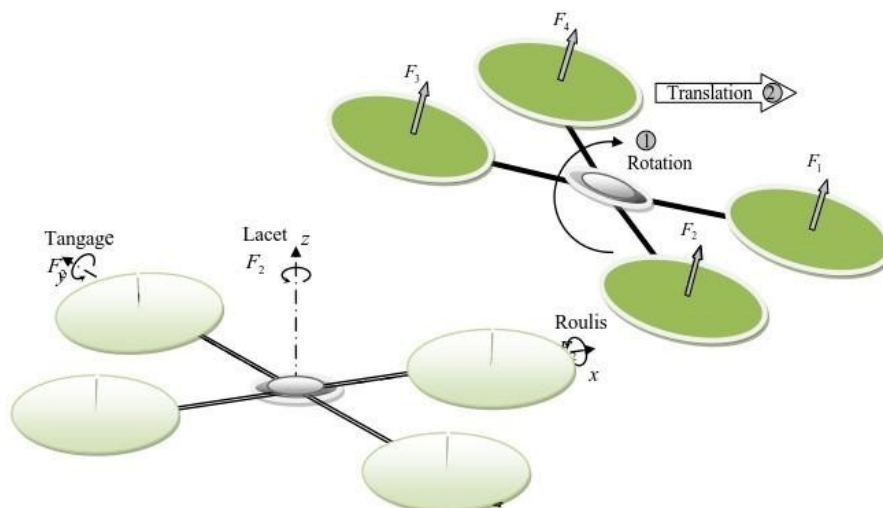


FIGURE 2.17 – Illustration du mouvement de translation.

2.7 Conclusion

Les points abordés dans ce chapitre ont essentiellement porté sur le fonctionnement du drone quad-copter, avec des détails sur les différents mouvements possibles ainsi que les modes de vol, en se basant sur le tangage, roulis, lacet. Le chapitre suivant donne un aperçu sur l'exploitation d'un drone en introduisant entre autres la planification de missions.

Chapitre 3

Planification de missions de drones

3.1 Introduction

La planification automatique de la trajectoire des robots sans collision dans des environnements dynamiques a fait l'objet de plusieurs travaux de recherche. Leur application dans les UAVs a connue récemment plusieurs évolutions et améliorations, leur permettant de jouer un rôle important dans de multiples domaines comme la livraison de différents types de produits, la communication en situation d'urgence, la reconnaissance, la lutte contre les incendies, la détection, le sauvetage et bien d'autres domaines d'application.

La mise en œuvre des techniques de planification de trajets dans le contexte des robots en général et des drones en particulier revient à leur fournir le moyen de gérer leur déplacement dans la zone cible en toute sécurité en temps opportun. Autrement dit, il faut trouver et déterminer un chemin optimal et sans collision qui permet à un robot de se déplacer et de changer la direction tout en évitant les obstacles existants dans l'environnement en partant d'une position de départ et arrivant à celle de destination.

Dans ce chapitre nous présenterons en premier lieu les différents aspects de la planification. Ensuite, nous présenterons la définition de la notion de planification et comment l'utiliser.

3.2 Contrôle de vol des quad-copter

Dans but d'exploration et d'analyse en temps réel d'un environnement, nous déploieront notre drone, qui se chargera de capturer des données, des informations et les transmettre à la station de base à laquelle il est connecté pour être traité. Cette section nous allons décrire notre système et le moyen de communication entre ces différentes parties le composant. Comme la montre là [3.1](#)



FIGURE 3.1 – illustration de notre système.

3.2.1 La station de base :

Pour exploiter notre drone nous avons besoin d'une station de base pour communiquer avec lui et le contrôler à distance, il existe beaucoup d'application qui effectuent le rôle d'une station de base comme APM Planner, Ground contrôler, Mission Planner, etc. Dans notre projet nous avons utilisé Mission Planner pour toutes les configurations initiales de notre drone, la planification du vol, la récolte des données, etc.

Mission planner :

Mission Planner est une application complète gratuite, open source et prise en charge par la communauté pour les projets de pilote automatique Ardupilot plus particulièrement à propos du firmware⁵ ArduCopter (désormais appelé APM :Copter), il est constamment amélioré par une communauté de développeurs du monde entier.

Mission Planner est une station de contrôle pour avion, hélicoptère et Rover, peut être utilisée comme utilitaire de configuration ou comme complément de contrôle dynamique pour un véhicule autonome. Voici quelques actions que nous pouvons effectuer avec Mission Planner :

- Charger le logiciel (firmware¹) dans la carte du pilote automatique (c'est-à-dire la série Pixhawk) qui contrôle le véhicule.
- Configurer et régler le véhicule pour des performances optimales.
- Planifier, enregistrer et charger des missions autonomes dans le pilote automatique avec une simple entrée de points de passage (WayPoints) pointer-cliquer sur Google map ou d'autres cartes.
- Télécharger et analyser les journaux de mission créés par le pilote automatique.
- Interface avec un simulateur de vol sur PC pour créer un simulateur d'UAV hardware-in-the-loop complet.
- Avec le matériel de télémétrie approprié, nous pouvons :

1. est un programme intégré dans un matériel informatique (ordinateur, photocopieur, automate (API, APS), disque dur, routeur, appareil photo numérique, etc.). Pour qu'il puisse fonctionner le APM :Copter est un micrologiciel de vols qui peut être utilisé sur différentes cartes électroniques, pour la plupart open-hardware, comme l'APM 2.6, le Pixhawk, le Pixhawk 2 ou même sur certains modèles de Parrot.

- Surveiller l'état du véhicule pendant son fonctionnement.
- Enregistrer les journaux de télémétrie qui contiennent beaucoup plus d'informations que les journaux du pilote automatique embarqué.
- Afficher et analyser les journaux de télémétrie.

3.2.2 Protocole Mavlink

MAVLink est un protocole de communication open source de télémétrie binaire conçu pour les systèmes à ressources limitées et les liaisons à bande passante réduite. MAVLink est déployé dans deux versions majeures : v1.0 et v2.0, qui est rétro compatible (les implémentations v2.0 peuvent analyser et envoyer des paquets v1.0).



Les flux de données de télémétrie sont envoyés dans une communication multicast tandis que les aspects de protocole qui modifient la configuration du système et qui nécessitent une livraison garantie, comme le protocole de mission ou le protocole de paramètres, sont de type point à point avec retransmission. MAVLink normalise le protocole de communication pour l'envoi d'informations en standardisant les types de messages pouvant être envoyés et permet une communication bidirectionnelle entre le poste de contrôle au sol et le drone.



FIGURE 3.2 – illustration de la communication entre Mission planner et UAV grâce à MAVlink.

Donc au final notre system peut se résumer selon le schéma explicatif 3.3

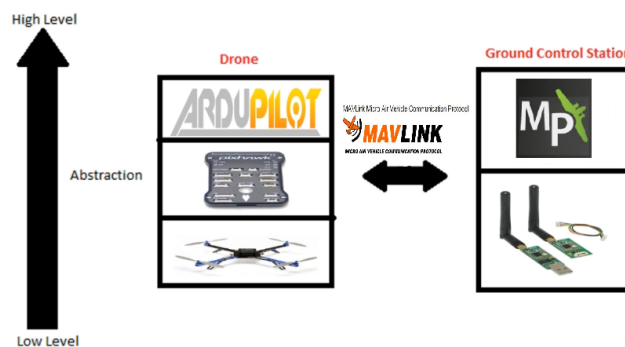


FIGURE 3.3 – schémas explicatifs de notre système.

3.3 Contrôle d'un UAV Avec Mission Planner

Pour le bon fonctionnement de notre drone on chargera le micrologiciel Arducopter adéquat et spécialement développé pour le type de multiroteur « quad-copter » et la carte dont nous disposons « Pixhawk2.4.8 ». Ce dernier offre la possibilité de programmer des missions à l'aide d'un logiciel avec interface intuitive (Mission Planner). Le système ArduCopter + Mission Planner a été élaboré en gardant à l'esprit la simplicité d'utilisation tout en laissant la possibilité d'ajouter des fonctionnalités « customisés » pour des utilisations spéciales, de la recherche ou de l'éducation.

Les logiciels Copter, Plane, Rover ou AntennaTracker que nous trouverons sur Mission Planner fonctionnent sur une grande variété de matériels embarqués (y compris des ordinateurs Linux complets), généralement constitués d'un ou plusieurs microcontrôleurs ou microprocesseurs connectés à des périphériques capteurs utilisés pour la navigation. Inclure des gyroscopes MEMS et un accéléromètre au minimum, nécessaires pour le vol du multiroteur et la stabilisation de l'avion. Les capteurs comprennent généralement, en plus, une ou plusieurs boussoles, altimètres (barométriques) et GPS, ainsi que des capteurs supplémentaires en option tels que des capteurs de débit optique, des indicateurs de vitesse anémométrique, des altimètres ou télémètres laser ou sonar, des caméras monoculaires. Les capteurs peuvent être sur la même carte électronique ou externes. Dans ce qui suit nous s'intéresserons aux étapes de configuration d'un UAV et plus exactement un Quad-copter et à la calibration de tous les capteurs et modules de bases connectés à ce dernier.

La configuration d'un UAV

La configuration d'un UAV passe par 6 étapes de bases que l'on notera essentielles :

Etape 1 : charger firmware Arducopter

Ça consiste à charger le firmware « Arducopter V4.0.4 Quad » sur notre Pixhawk « mission planner peut détecter automatiquement ou juste grâce à deux trois question la carte ou le microcontrôleur en question et télécharger le firmware adéquat à ce dernier 3.4.

Il suffit de brancher l'autopilote (drone) au port com et choisir baud à 152000 et accéder à l'onglet « PLAN » et puis « Install Firmware LEGACY ».

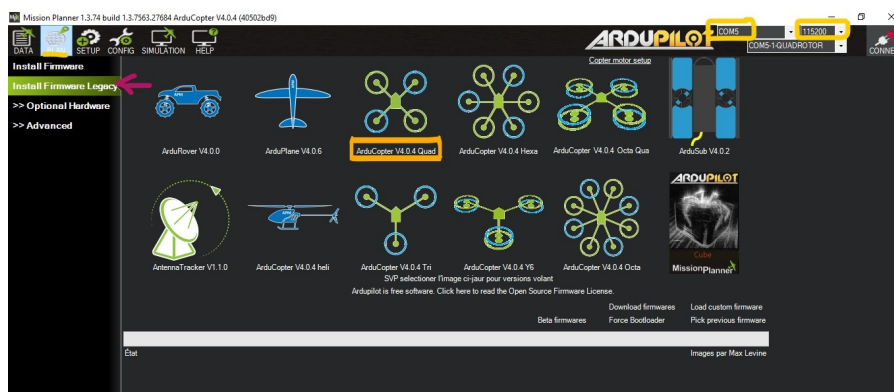


FIGURE 3.4 – charger le firmware sur la Pixhawk.

Une fois le firmware est chargé sur la carte on connecte la carte à mission planner en cliquant sur « Connect » et on peut consulter Frame Type et vérifier que le chargement a été bien fait comme le montre la figure si dessous :

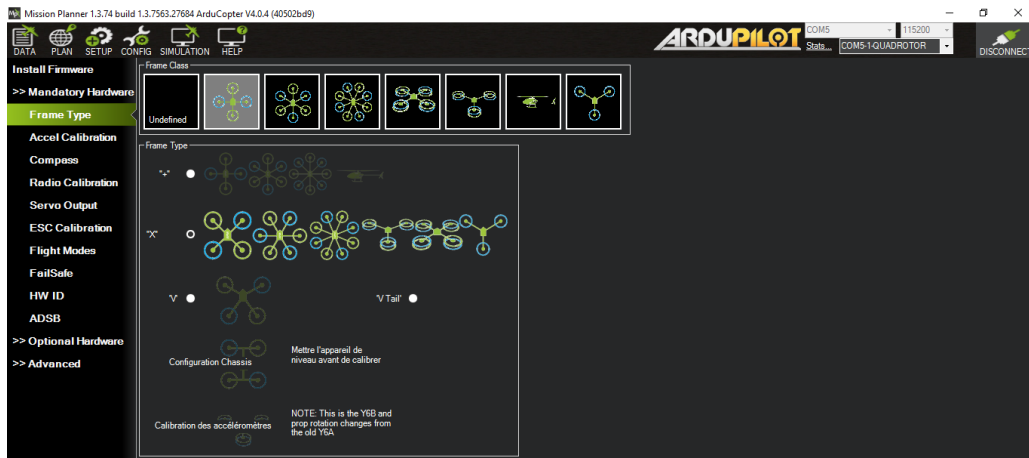


FIGURE 3.5 – Type frame.

Etape 2 : Calibration des accéléromètres.

Dans cette étape on Mission Planner nous invitera à placer le véhicule à chaque position d'étalonnage. il suffit d'appuyer sur n'importe quelle touche pour indiquer que le pilote automatique est en position, puis passez à l'orientation suivante.

Les positions d'étalonnage sont : de niveau, à droite, à gauche, vers avant, vers 'arrière et sur le dos. Comme le montre la figure ci-dessous :

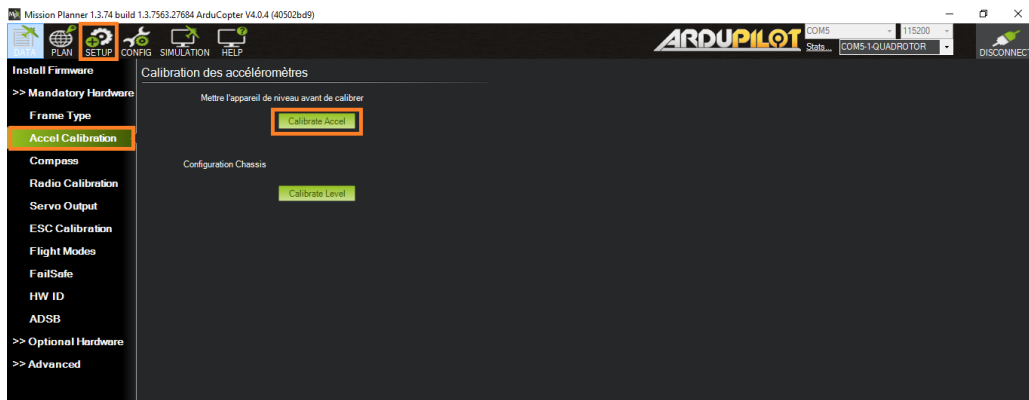


FIGURE 3.6 – Calibration des Accéléromètres.

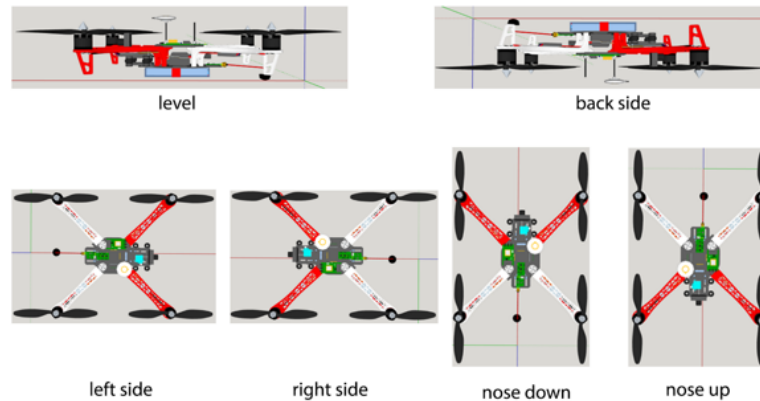


FIGURE 3.7 – Positionnement du drone pour la calibration des accéléromètres.

Etape 3 : Calibration de la boussole « Compass Calibration »

Ça consiste à Tenir le véhicule en l'air et le faire pivoter de manière à ce que chaque côté (avant, arrière, gauche, droite, haut et bas) pointe vers le bas pendant quelques secondes. Envisager un virage complet à 360 degrés, chaque virage indiquant une direction différente du véhicule vers le sol. Il en résultera 6 tours complets plus éventuellement du temps supplémentaire et des tours pour confirmer l'étalonnage ou réessayer s'il ne réussit pas initialement. Lorsque le véhicule tourne, les barres vertes doivent s'étendre de plus en plus vers la droite jusqu'à ce que l'étalonnage soit terminé. Une fois terminé, trois tonalités montantes seront émises et une fenêtre « Veuillez redémarrer le pilote automatique » apparaîtra et vous devrez redémarrer le pilote automatique avant qu'il soit possible d'armer le véhicule.

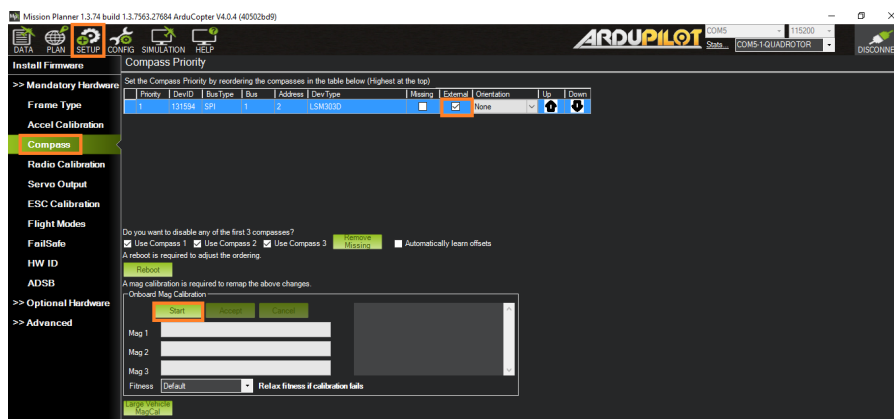


FIGURE 3.8 – Calibration de la boussole.

Etape 4 : Calibration du contrôleur Radio

Cette étape consiste à calibrer la télécommande (Remote Controller) du drone. En Vérifiant le mappage des canaux dans l'émetteur (c.-à-d. Vérifiez quels canaux d'entrée sont contrôlés par les bâtons, commutateurs et boutons de l'émetteur) en déplaçant les bâtons, les boutons et les commutateurs et en observant quelles barres vertes (le cas échéant) se déplacent.

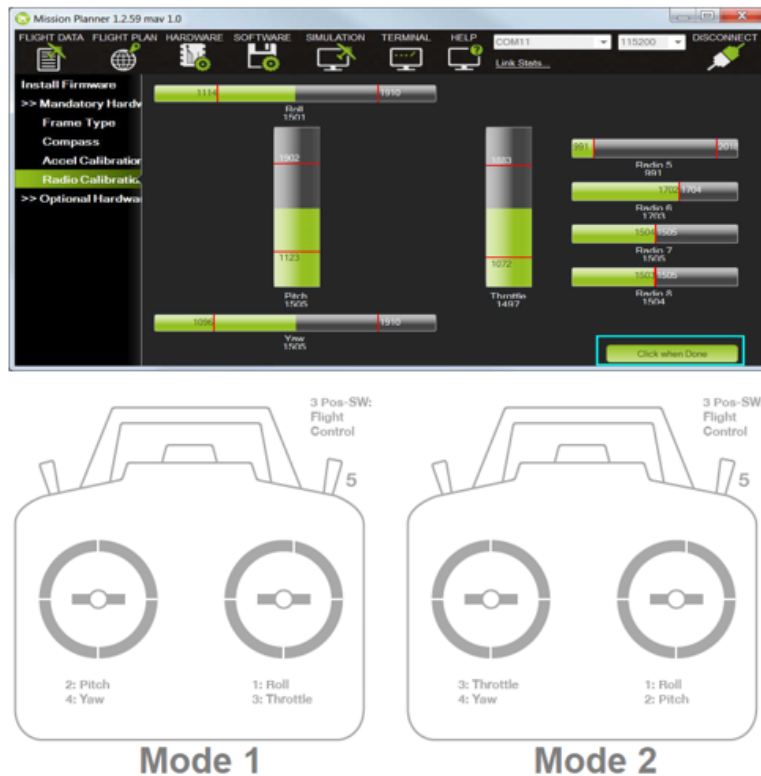


FIGURE 3.9 – Calibration Radio.

Si c'est la première fois que l'émetteur est utilisé avec ArduPilot, il est probable que le Mapping des canaux de l'émetteur devra être modifié et cela se fait normalement sur l'émetteur lui-même à l'aide de son menu de configuration intégré

Déterminez si votre émetteur est Mode1 ou Mode2 (voir ci-dessus)

Le bâton de roulement doit contrôler le canal 1.

Le Pitch Stick doit contrôler le canal 2.

Le manche des gaz doit contrôler le canal 3.

Le bâton de lacet doit contrôler le canal 4.

Un commutateur à 3 positions (pour contrôler le mode de vol) doit être configuré pour contrôler le canal 5 (si vous utilisez Copter) ou le canal 8 (si vous utilisez Rover ou Avion).

Déplacez les manches de roulis, de tangage, d'accélérateur et de lacet de l'émetteur et assurez-vous que les barres vertes se déplacent dans la bonne direction : pour les canaux de roulis, d'accélérateur et de lacet, les barres vertes doivent se déplacer dans la même direction que les manches physiques de l'émetteur, pour la hauteur, la barre verte doit se déplacer dans la direction opposée au manche physique de l'émetteur.

Etape 5 : Esc Calibration

Calibration d'un ESC se fait avec un joystick et cela consiste à apprendre à l'ESC à quelle plage d'entrées d'accélérateur il doit répondre. Il doit savoir quelle valeur PWM sur le canal d'accélérateur correspond à la commande d'arrêt du moteur et quelle valeur PWM correspond à plein régime. Ces valeurs sont stockées dans l'ESC.

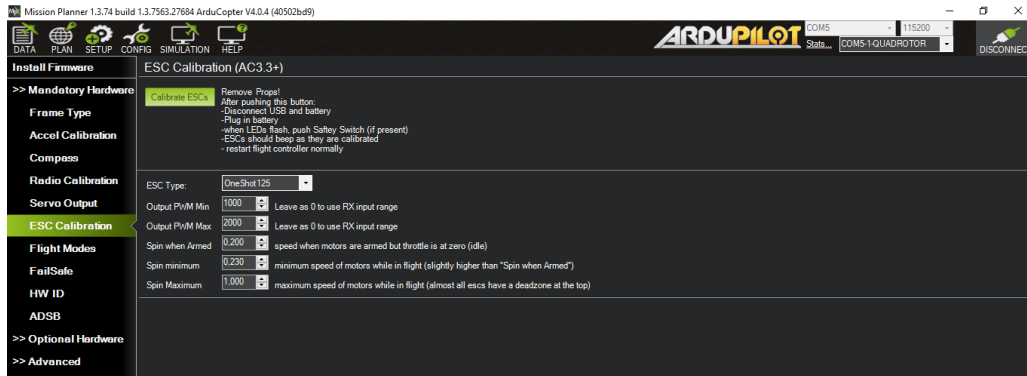


FIGURE 3.10 – Calibration des ESCs.

Remarque : Calibration des ESCs et radio se fait avec un joystick ce qui n'est pas notre cas vu le nôtre est un UAV autonome et on a pas eu à configurer le contrôleur de vol (remote controller) .

Etape 6 : Servo Output :

Configuration des moteurs dans cette partie on consulte la configuration standard pour un vol stabilisé ou PWM varie entre 1100 et 1900 dans notre cas

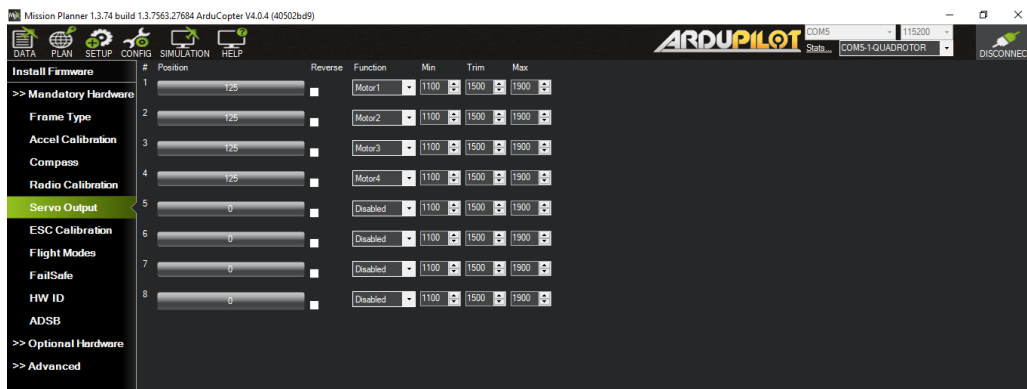


FIGURE 3.11 – Servo Moteur Configuration.

Full Paramètres List

Flight Mode	Command	Value	Units	Options	Desc	Rev
Geofence	MOT_BAT_CURR_MAX	0	A	0.200	Maximum current over which maximum throttle is limited (0 = Disabled)	
Basic Tuning	MOT_BAT_CURR_TC	5	s	0.10	Time constant used to limit the maximum current	
Extended Tuning	MOT_BAT_VOLT_MAX	0	V	0 First battery 1 Second battery	Which battery monitor should be used for doing compensation	
Standard Params	MOT_BAT_VOLT_MIN	0	V	6.35	Battery voltage compensation maximum voltage (voltage above this will have no additional scaling effect on thrust)	
Advanced Params	MOT_BOOST_SCALE	0		0.5	Battery voltage compensation minimum voltage (voltage below this will have no additional scaling effect on thrust)	
User Params	MOT_HOVER_LEARN	2		0 Disabled 1 Learn 2 Learn and Store	Recommended 4.8 cell count 0 = Disabled	
Full Parameter List	MOT_PWM_MAX	2000	PWM	0.2000	Booster motor output scaling factor vs main throttle. The output to the Boost Throttle servo will be the main throttle times this scaling factor. A higher scaling factor will put more of the load on the booster motor. A value of 1 will set the Boost Throttle equal to the main throttle.	
Full Parameter Tree Planner	MOT_PWM_MIN	1000	PWM	0.2000	Enable/Disable automatic learning of hover throttle	
	MOT_PWM_TYPE	2		0 Normal 1 OneShot 2 OneShot 125 3 OneShot 4 OneShot 10 5 OneShot 6 OneShot 7 OneShot 1200	This sets the max PWM value in microseconds that will ever be output to the motors. 0 = use input RC3_MAX	
	MOT_SAFE_DISARM	0		0 PWM enabled while disarmed 1 PWM disabled while disarmed	This sets the min PWM output value in microseconds that will ever be output to the motors. 0 = use input RC3_MIN	
	MOT_SAFE_TIME	1	s	0.5	This selects the output PWM type, allowing for normal PWM continuous output, OneShot, brushed or DShot motor output	
	MOT_SLEW_DN_TIME	0	s	0.5	Disables motor PWM output when disarmed	
	MOT_SLEW_UP_TIME	0	s	0.5	Time taken to disable and enable the motor PWM output when disarmed and armed	
	MOT_SPIN_ARM	0.2		0.0 Low 0.1 Default 0.2 High	Time in seconds to slew output from full to zero. This is used to limit the rate at which output can change. Range is constrained between 0 and 0.5.	
	MOT_SPIN_MIN	1		0.0 Low 0.95 Default 1.0 High	Time in seconds to slew output from zero to full. This is used to limit the rate at which output can change. Range is constrained between 0 and 0.5.	
	MOT_SPIN_MAX	0.23		0.0 Low 0.15 Default 0.3 High	Point at which the motor start to spin expressed as a number from 0 to 1 in the entire output range. Should be lower than MOT_SPIN_MIN	
	MOT_SPOOL_TIME	0.5	s	0.2	Point at which the thrust saturates expressed as a number from 0 to 1 in the entire output range	
	MOT_THST_EXPO	0.65		0.25 0.8	Point at which the thrust starts expressed as a number from 0 to 1 in the entire output range. Should be higher than MOT_SPIN_ARM	
	MOT_THST_HOVER	0.6875		0.2 0.8	Time in seconds to spool up the motors from zero to main throttle	
	MOT_YAW_HEADROOM	200	PWM	0.500	Motor thrust curve exponent (from 0 for linear to 1.0 for second order curve)	
	PILOT_THR_BHV	0		0.500 1.500	Motor thrust needed to hover expressed as a number from 0 to 1	
	PSC_ACCE_P	0.5		0.0 No Netters 2 Rps 3 Simple Mode	Yaw control is given at least this pwm in microseconds range	
					Break containing various throttle stick options. TX with spring throttle can set PILOT_THR_BHV to "1" so motor feedback when landed starts from mid stick instead of bottom of stick	
					Acceleration (vertical) controller P gain. Converts the difference between desired vertical acceleration and actual acceleration into a motor output	

FIGURE 3.12 – Full parameter List.

On trouve sur mission planner dans l'onglet Config « Full paramètres List » qui nous permet d'avoir une vision détaillée de tous les paramètres qui concernent le vol, la configuration de chaque composant ou module relié au microcontrôleur, et aussi la configuration des interfaces d'entrées de ce dernier, tels que les ports de communication. Par exemple sur la figure ?? on peut bien changer les paramètres concernant la pwm min/ max sera attribuer à un moteur.

Planifier une mission avec des Point d'acheminement « WayPoint » sur Mission Planner Mode manuel

Pour planifier une mission il suffit d'ouvrir Mission Planner, cliquer sur l'icône « Plan » une carte va être affichée ensuite on suit les étapes suivantes :

1. Designer la position d'origine

Avant de commencer la planification d'une mission il faut d'abord designer la position d'origine (home). Pour les Copter, Plane et Rover, la position d'origine est définie comme l'emplacement où le véhicule était armé. Cela signifie que si on exécute un RTL (Return To Launch), il retournera à l'endroit où il a été armé, donc il suffit d'armé le véhicule à l'endroit où on veut qu'il revienne où utiliser un point de ralliement pour configurer un autre point de retour. Sur Mission Planner il suffit de naviguer sur la carte et cliquer avec le bouton droit de la souris ou on veut situer notre point d'origine et cliquer sur Set Home Here.

2. Créer les WayPoints

D'abord il faut faire un décollage (TAKEOFF) avec un clic droit sur la carte puis sélectionner Takeoff puis entrer l'altitude. Ensuite il suffit de créer les WayPoints selon le plan de vol désiré avec un clic sur la carte pour créer chaque WP. Une fois on a tous les WayPoints à leurs places on donne la commande à notre drone d'atterrir (Land) avec un clic droit sur la carte puis sélectionner Land et le drone va atterrir ou on a cliqué, ou bien on peut faire RTL et dans ce cas le drone retournera au point d'origine. .

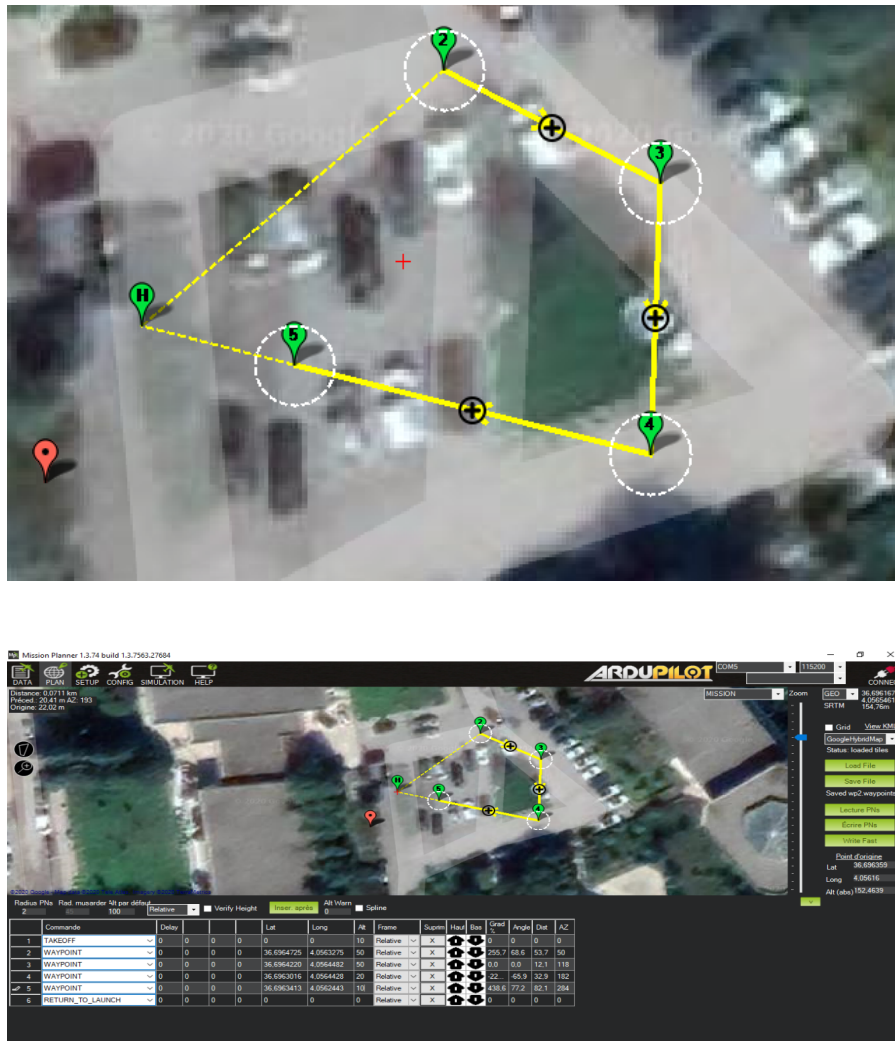


FIGURE 3.13 – exemple d’une planification de vol avec WayPoints.

Dans la figure 3.13 nous avons créé un plan de mission de tel sorte que le quad-copter va commencer par un auto Takeoff à 10 mètres d’altitude, puis il va vers le WP2 (WayPoint 2) Qui est à 50 mètres d’altitude, ensuite vers WP3 puis il va descendre à WP3 qui est à 20 mètres d’altitude, ensuite vers WP5 et enfin il va faire an auto RTL ou il va retourner vers la position d’origine.

Au fur et à mesure d’ajout de WayPoints un tableau au-dessous de la carte qui nous montre toutes les informations concernant ces derniers comme :

- La longitude, l’attitude du WayPoint.
- L’altitude du WayPoint.
- La commande a effectué l’orsque le drone arrivera à ce WayPoint (c’est-à-dire est ce que c’est un Takeoff, RTL, Land, etc.).
- Etc. [Voir 3.14]

Radius PNs Rad. musarder Alt par défaut

245100RelativeVerify HeightInscr. aprèsAlt Warn0Spline

	Commande	Delay				Lat	Long	Alt	Frame	Suprim	Haut	Bas	Grad %	Angle	Dist	AZ
1	TAKEOFF	0	0	0	0	0	0	10	Relative	X			0	0	0	0
2	WAYPOINT	0	0	0	0	36.6964725	4.0563275	50	Relative	X			255,7	68,6	53,7	50
3	WAYPOINT	0	0	0	0	36.696422	4.0564482	50	Relative	X			0,0	0,0	12,1	118
4	WAYPOINT	0	0	0	0	36.6963016	4.0564428	20	Relative	X			-22...	-65,9	32,9	182
5	WAYPOINT	0	0	0	0	36.6963413	4.0562443	100	Relative	X			438,6	77,2	82,1	284
6	RETURN_TO_LAUNCH	0	0	0	0	0	0	0	Relative	X			0	0	0	0

FIGURE 3.14 – tableau d'information WayPoint.

Quelques commandes de points d'acheminement :

Il existe 3 types de commandes de point d'acheminement

1. Commandes de navigation « NAV » :

Ce type de commande affectent directement le comportement du quad-copter durant son vol. Elles permettent d'acheminer un itinéraire ou une trajectoire. Parmi ces commandes on a Take Off, WayPoint, SplineWayPoit, Loiter_Time, Loiter_Turns, Loiter unlimited Return_To_Launch, land et delay expliquées ci-dessous.

• Takeoff



FIGURE 3.15 – Illustration de la commande Takeoff.

Le véhicule montera tout droit de son emplacement actuel à l'altitude spécifiée (en mètres). Cela devrait être la première commande de presque toutes les missions. Si la mission est commencée alors que le quad-copter est déjà en train de voler, le véhicule montera droit jusqu'à l'altitude spécifiée, si le véhicule est déjà au-dessus de l'altitude spécifiée, la commande de décollage sera ignorée et la mission passera immédiatement à la commande suivante.

• Waypoint

Le quad-copter volera en ligne droite jusqu'à l'emplacement spécifié comme latitude, longitude et altitude (en mètres). L'argument **Delay** spécifie le temps d'attente (en

secondes) au waypoint avant de passer à la commande suivante. L'argument **Hit Rad** (non pris en charge dans cette commande) destiné à maintenir la distance (en mètres) du point cible qui qualifiera le waypoint comme complet. Cette commande n'est pas prise en charge. Au lieu de cela, le paramètre WPNAV_RADIUS doit être utilisé (voir le champ « WP Radius » dans la capture d'écran ci-dessous ou ajuster via la liste des paramètres standard).

L'argument **Yaw Ang** (non pris en charge dans cette commande) destiné à maintenir l'angle de lacet résultant en degrés (0 = nord, 90 = est).

Les arguments **Lat**, **Lon** spécifie les cibles de latitude et de longitude. S'il est laissé à zéro, il contiendra l'emplacement actuel.

L'argument **Alt** spécifie l'altitude cible au-dessus de la position d'origine en mètres. S'il est laissé à zéro, il conservera l'altitude actuelle.



FIGURE 3.16 – Illustration de la commande Waypoint

• Spline Waypoint

Les spline waypoints prennent les mêmes arguments que les WayPoints normaux (lat, lon, alt, delay) mais lorsqu'ils sont exécutés, le véhicule suivra des trajectoires lisses (à la fois verticalement et horizontalement) au lieu de lignes droites. Les spline WayPoints peuvent être mélangés avec des points de cheminement en ligne droite (WayPoints), comme indiqué dans l'image ci-dessous.

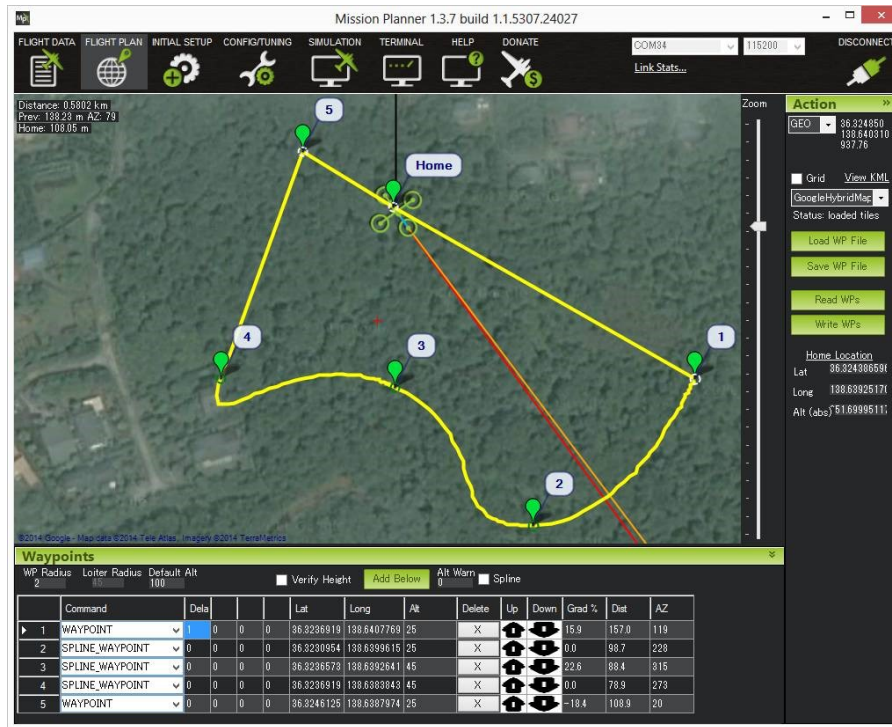


FIGURE 3.17 – Illustration de la commande Spline Waypoint

- **Loiter _Time**



FIGURE 3.18 – Illustration de la commande Loiter Time

Le quad-coptre s'envolera et attendra à l'emplacement spécifié pendant le nombre de secondes spécifié. C'est la même chose qu'une commande WayPoints mais avec une durée de temps.

L'argument **Time** spécifie le nombre de secondes pour rester à l'emplacement.

- **Loiter _Turns**

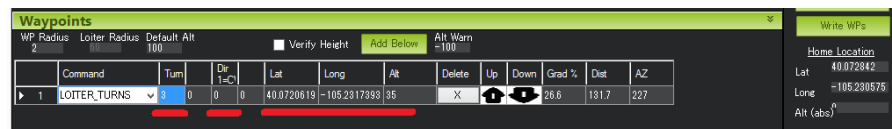


FIGURE 3.19 – Illustration de la commande Loiter Turns

Le quad-coptre fera un cercle autour de la latitude, de la longitude et de l'altitude spécifiée (en mètres). Le rayon du cercle est contrôlé par le paramètre CIRCLE_RADIUS (c'est-à-dire qu'il ne peut pas être défini dans le cadre de la mission mais il est défini dans la fenêtre Config/Full Parameters List).

L'argument **Turn** spécifie le nombre de rotations complètes à effectuer autour du point.

L'argument **Dir 1=CW** spécifie la direction pour tourner autour du point. -1 = sens antihoraire, +1 = sens horaire.

- **Loiter_unlimited**

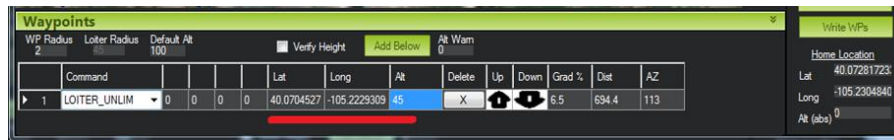


FIGURE 3.20 – Illustration de la commande Loiter unlimited

Le quad-copter s'envolera et attendra indéfiniment à l'emplacement spécifié. La mission ne dépassera pas ce commandement.

- **Return_To_Launch**



FIGURE 3.21 – Illustration de la commande Loiter Launch

Mission équivalente au mode de vol RTL. Le véhicule monte d'abord à l'altitude spécifiée du paramètre RTL_ALT (la valeur par défaut est 15 m) avant de rentrer chez lui. Le domicile est celui où le véhicule a été armé pour la dernière fois (lorsque le véhicule est déjà en vol il retournera directement vers la position d'origine ou le domicile(home)).

Cette commande ne prend aucun paramètre et devrait généralement être la dernière commande de la mission.

- **Land**



FIGURE 3.22 – Illustration de la commande Land

Le véhicule atterrira à son emplacement actuel (si les arguments lat et lon sont à 0) ou aux coordonnées lat / lon fournies. C'est l'équivalent du mode de vol LAND.

- **Delay**

Le véhicule restera à son emplacement actuel jusqu'à ce qu'un délai spécifié en nombre de secondes se soit écoulé ou que l'heure absolue soit atteinte. La première colonne (« Seconds ») contient le nombre de secondes à retarder. Mis à -1 si ce champ doit être ignoré. Les champs « heure UTC », « minute UTC » et « seconde UTC » peuvent être utilisés pour spécifier une heure absolue (UTC). L'exemple ci-dessus



FIGURE 3.23 – Illustration de la commande Delay

fera décoller le véhicule à 13h30 UTC. Notez que le champ heure et / ou minute peut également être ignoré en le définissant sur -1.

- **Do_Jump**



FIGURE 3.24 – Illustration de la commande Do_Jump

Faire un saut vers la commande de mission spécifiée le nombre de fois spécifié avant de poursuivre la mission. Dans l'exemple ci-dessus, le véhicule effectuait des allers-retours entre les WayPoints # 1 et # 2 un total de 3 fois avant de voler au point de cheminement # 4.

L'argument **WP #** spécifie la commande de mission à laquelle il faut sauter.

L'argument **Repeat** spécifie le nombre maximum de fois où nous devrions effectuer le saut. -1 signifie toujours sauter.

Remarque : Do Jump est un cas exceptionnel de commandes, son nom et sous forme « Do_Action » mais c'est une commande de type navigation d'où les commandes conditionnelles telles que **CONDITION_DELAY** n'affecteront pas Do-Jump, il effectuera toujours le saut dès qu'il atteindra cette commande.

2. Commandes « DO_Action » : permet de donner un ordre d'effectuer une tâche au niveau d'un point d'acheminement, elles n'affectent pas la trajectoire du quad-copter. Parmi celles-ci :

Do_Set_Cam_Trigg_Dist Do_Digicam_Control et Do_set_Roi,ect.

- **Do_Set_Cam_Trigg_Dist**



FIGURE 3.25 – Illustration de la commande Do_Set_Cam_Trigg_Dist

Déclenchez l'obturateur de l'appareil photo à intervalles réguliers. Par exemple, la commande ci-dessus provoquera le déclenchement de l'obturateur de la caméra tous les 5 mètres parcourus par le véhicule.

L'argument **Dist** spécifie l'intervalle de distance en mètres. Pour empêcher la caméra de se déclencher, ajoutez un DO_SET_CAM_TRIGG_DIST plus tard dans la mission avec Dist = 0.

- **Do_Digicam_Control**



FIGURE 3.26 – Illustration de la commande Do_Digicam_Control

Déclenche l'obturateur de l'appareil photo une fois. Cette commande ne prend aucun argument supplémentaire.

- **Do_Set_Roi**



FIGURE 3.27 – Illustration de la commande Do_Set_Roi

Dirige le nez du véhicule ou du cardan de la caméra vers la « région d'intérêt ». Dans l'exemple ci-dessus, le nez ou la caméra serait pointé vers le marqueur rouge.

Le nez continuera à pointer vers le marqueur rouge jusqu'à la fin de la mission ou si le véhicule retourne à son comportement par défaut (c'est-à-dire pointant vers le prochain waypoint) en ajoutant une deuxième commande DO_SET_ROI plus tard dans la mission avec tout zéro pour Lat, Lon et Alt.

3. Commandes conditionnelles : commandes qui affecte sur des commandes de type « do_action » tels que : Condition_Delay, Condition_Distance, etc.

- **Condition_Delay**

WP	Radius	Loiter Radius	Default Alt	Verify Height	Add Below	Alt Warn	Command	Time (sec)	Dist (m)	Grad %	Dist	AZ
1	2		100				TAKEOFF	0	0	0	0	0
2							WAYPOINT	0	0	0	40.0720948	-105.2307200
3							CONDITION_DELAY	5	0	0	0	0
4							DO_SET_ROI	0	0	0	40.0724888	-105.2299476
5							WAYPOINT	0	0	0	40.0725956	-105.2288210

FIGURE 3.28 – Illustration de la commande Condition_Delay

Retarde le démarrage de la prochaine commande « Do » pendant le nombre de secondes spécifié. Dans l'exemple ci-dessus, la commande n ° 4 (Do-Set-Roi) est retardée de sorte qu'elle ne démarre que 5 secondes après que le véhicule ait franchi le waypoint n ° 2.

Remarque : cette commande n'arrête pas le véhicule. Notez également que la minuterie de retard n'est active que jusqu'à ce que le prochain WayPoint soit atteint, donc si la commande « Do » n'a pas été déclenchée d'ici là, elle ne le sera jamais.

L'argument **Time (sec)** spécifie le temps de retarder la prochaine commande « Do » en secondes.

- **Condition_Distance**

WP	Radius	Loiter Radius	Default Alt	Verify Height	Add Below	Alt Warn	Command	Time (sec)	Dist (m)	Grad %	Dist	AZ
1	2		100				TAKEOFF	0	0	0	0	0
2							WAYPOINT	0	0	0	40.0720948	-105.2307200
3							CONDITION_DISTANCE	0	50	0	0	0
4							DO_SET_ROI	0	0	0	40.0724888	-105.2299476
5							WAYPOINT	0	0	0	40.0725956	-105.2288210

FIGURE 3.29 – Illustration de la commande Condition_Distance

Retarde le démarrage de la prochaine commande « Do » jusqu'à ce que le véhicule se trouve dans le nombre de mètres spécifié du prochain WayPoint. Dans l'exemple ci-dessus, la commande n ° 4 (Do-Set-Roi) est retardée de sorte qu'elle ne démarre que lorsque le véhicule se trouve à moins de 50 m du WayPoint n ° 5.

Remarque : cette commande n'entraîne pas l'arrêt ou la pause du véhicule pendant la mission, elle n'affecte que les commandes « Do ».

L'argument **Dist** spécifie à quelle distance le véhicule doit être du prochain WayPoint (en mètres) avant de commencer la prochaine commande « Do ».

4. Sauvegarder le plan de la mission

Après avoir créé les WayPoints la création du plan est achevée, donc on enregistre en cliquant sur le bouton Save File pour ne pas recréer le plan chaque fois qu'on ouvre une nouvelle session, il suffit juste d'importer le fichier que nous avons déjà créé en cliquant sur le bouton "Load File ". Maintenant on clique sur le bouton "Write Fast "pour uploader le plan de notre mission au contrôleur de vol (le pc doit être connecté à la Pixhawk).

Planifier une mission avec des WayPoints sur Mission Planner à l'aide d'auto Grid :

On peut également demander au Mission Planner de nous créer une mission, ce qui est utile pour des fonctions telles que des missions de cartographie, où le véhicule doit simplement aller et venir dans un modèle de « tondeuse à gazon » sur une zone pour collecter des photographies.

Pour ce faire, dans le menu du clic droit, sélectionnez Polygone et dessinez un cadre autour de la zone que vous souhaitez mapper. Sélectionnez ensuite Auto WP, Grid. Suivez le processus de la boîte de dialogue pour sélectionner l'altitude et l'espacement. Le planificateur de mission générera alors une mission exemple cas de la figure 3.30.

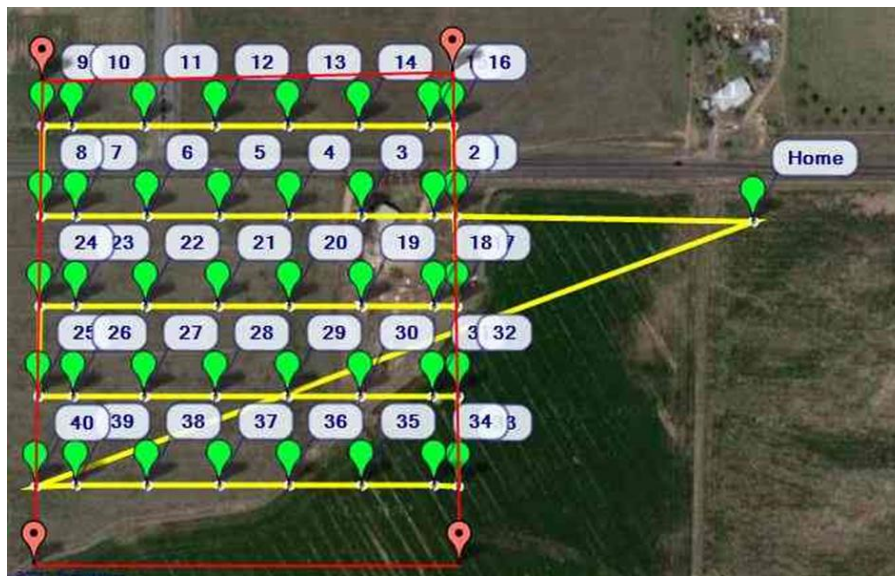


FIGURE 3.30 – figure exemple de planification avec auto Grid.

Remarque :

Mission Planner peut créer plusieurs autre model que le Grid comme Wp Circle, spline circle, area,etc.

Planifier une mission avec un script en Python sur mission planner

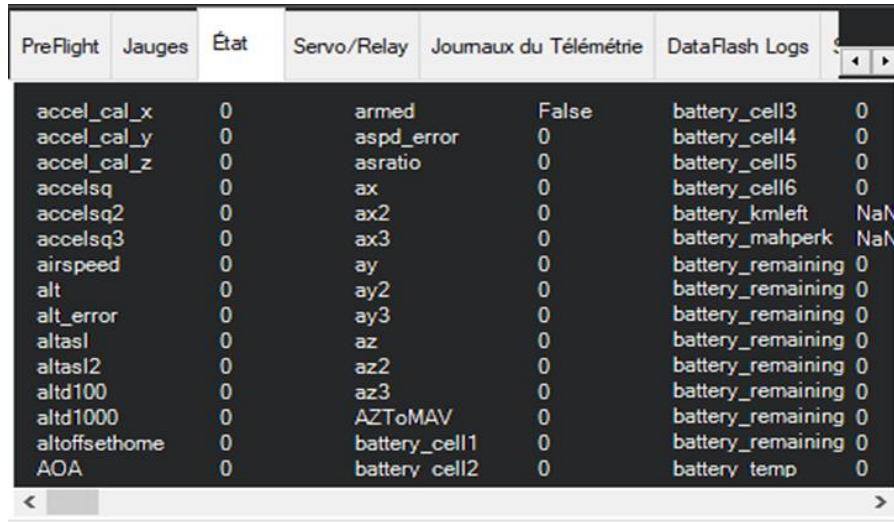
L'une des fonctionnalités intéressantes de Mission Planner c'est qu'il peut exécuter des scripts Python, ce qui est un moyen facile d'étendre les fonctionnalités du programme au-delà de ses fonctions intégrées.

On peut facilement programmer votre drone pour faire des missions basées sur des scripts (Mission Planner utilise une implémentation d'IronPython en interne), voici quelque exemple de commande qu'on peut utiliser sur mission planner outre que les commande python classique :

- Script.Sleep(ms)
- Script.ChangeParam(name,value)
- Script.GetParam(name)
- Script.ChangeMode(mode)
- Script.WaitFor(string,timeout)

- `Script.SendRC(channel,pwm,sendnow)`

On peut utiliser n'importe quelle variable de l'onglet d'état [3.31](#).



PreFlight	Jauges	État	Servo/Relay	Joumaux du Télémétrie	DataFlash Logs
accel_cal_x	0	armed	False	battery_cell3	0
accel_cal_y	0	aspd_error	0	battery_cell4	0
accel_cal_z	0	asratio	0	battery_cell5	0
accelsq	0	ax	0	battery_cell6	0
accelsq2	0	ax2	0	battery_kmleft	NaN
accelsq3	0	ax3	0	battery_mahperk	NaN
airspeed	0	ay	0	battery_remaining	0
alt	0	ay2	0	battery_remaining	0
alt_error	0	ay3	0	battery_remaining	0
altasl	0	az	0	battery_remaining	0
altasl2	0	az2	0	battery_remaining	0
altd100	0	az3	0	battery_remaining	0
altd1000	0	AZToMAV	0	battery_remaining	0
altffsethome	0	battery_cell1	0	battery_remaining	0
AOA	0	battery_cell2	0	battery temp	0

FIGURE 3.31 – tableau de paramètres "Etat".

Il suffit de créer un script python, ensuite il faut y aller vers l'onglet « Scripts dans la fenêtre data sur Mission Planner (en bas à gauche) [3.32](#),

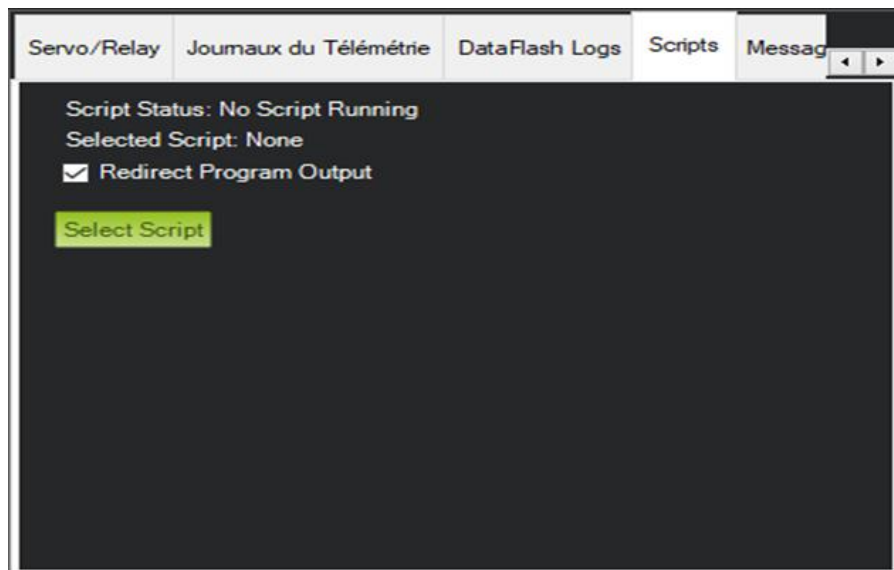


FIGURE 3.32 – charger un script de vol.

cliquer sur Select Script pour sélectionner le fichier et l'exécuter.

Exécution du plan de la mission et contrôle du drone

Avant l'exécution de notre plan de mission faut d'abord savoir que l'on a plusieurs modes de vols que pourra exécuter au cours de notre mission.

- Les modes de vol

Mission Planner nous offre plusieurs modes de vol pour notre drone comme le mode Auto, le Mode Guided, le Mode Stabilize, le Mode RTL, Le Mode Loiter, etc. Pour activer un mode il suffit d'y aller à l'onglet "Data" (figure 3.35), sélectionner Action dans le menu en bas à gauche (Figure 3.33) et là on sélectionne le mode que nous voulons et on le définit en cliquant sur Déf. Mode.



FIGURE 3.33 – Section Action sur Data.

- **Le mode Stabilize**

Le mode Stabilize nous permet de piloter le quad-copter manuellement, et il fait l'équilibration automatique de l'axe de roulis et de tangage (Quand on arme le drone il vaut mieux être dans ce mode).

- **Le mode Loiter**

Le mode Loiter maintient automatiquement la position, la direction et l'altitude actuelle. Si on active ce mode le drone va maintenir sa position dans l'air jusqu'à ce que on le désactive.

- **Le mode RTL (Return_To_Launch)**

Ce mode fait naviguer le quad-copter à partir de sa position actuelle pour survoler la position d'origine. Dans ce mode le véhicule retournera à la position d'origine.

- **Le mode Auto (Automatique)**

En mode Auto, le quad-copter suivra un plan ou un script de mission préprogrammé stocké dans le pilote automatique qui est composé de commandes de navigation ou de WayPoints (c'est-à-dire des points de cheminement).

- **Le Mode Guided**

Le mode Guided n'est pas un mode de vol traditionnel qui serait attribué à un commutateur de mode comme les autres modes de vol (c'est-à-dire qu'on

ne peut pas activer ce mode avec la télécommande mais uniquement avec une station de base ou station au sol). La fonction du mode Guided est activée à l'aide d'une application de station au sol (comme Mission Planner) et d'une radio de télémétrie (comme une radio de télémétrie SiK). Cette capacité vous permet de commander de manière interactive au quad-copter de se rendre à un emplacement cible en cliquant sur un point sur la carte des données de vol du planificateur de mission. Une fois l'emplacement atteint, l'quad-copter planera à cet emplacement, en attendant la prochaine cible (figure 3.34).

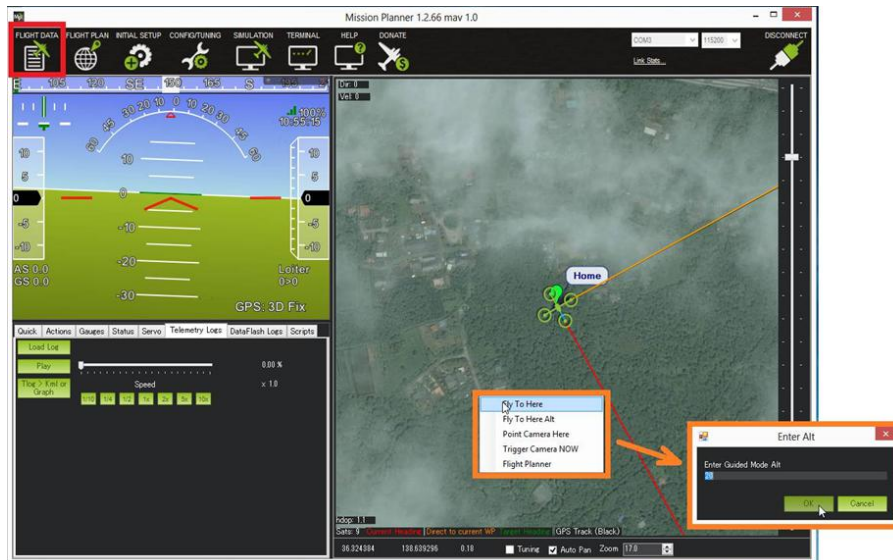


FIGURE 3.34 – mise en action de Guided Mode.

Après avoir uploadé le plan de notre mission a la Pixhawk, il faut l'exécuter. Pour cela on va vers l'onglet « Data » (Figure 3.33). Toutes les informations sur notre drone et l'environnement dans lequel on est son afficher :

- Les coordonnées GPS du drone et son emplacement sont affichés sur la carte.
- La vitesse de l'aire, l'altitude et toute l'information retournée par les capteurs embarqués.
- Etc.

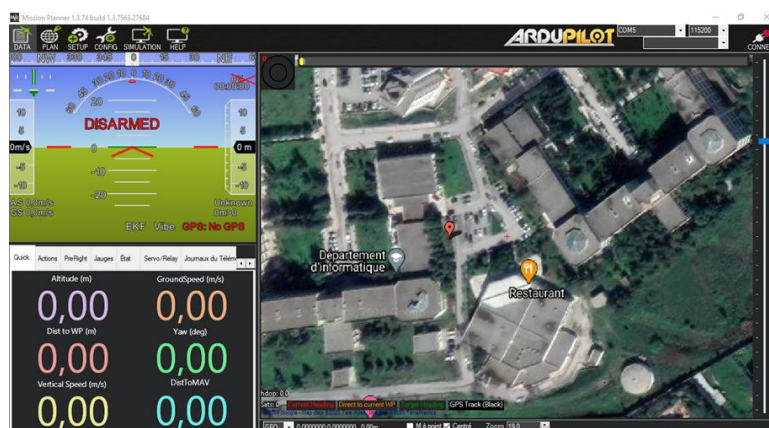


FIGURE 3.35 – onglet Data sur Mission Planner.

Maintenant pour exécuter notre mission on se concentre sur la fenêtre en bas à gauche (Figure 3.35) et on clique sur Action (Figure 3.33). Ensuite on arme le drone en cliquant sur le bouton Arm/Disarm. Puis on sélectionne le Mode Auto (Automatique) et on change la commande à effectuer à "Start_Mission ". Le drone démarrera sa mission et on peut le suivre sur la carte et ainsi récolter les données de la mission. Ceci dit, si par exemple on veut modifier l'itinéraire de notre quad-copter durant la mission on sélectionne le mode Loiter et dans ce mode le véhicule restera stable et immobile dans l'air et on change le WayPoints restant puis on retourne au mode Auto.

On peut aussi utiliser le mode Guided pour contrôler dynamiquement le quad-copter, il suffit d'un clic droit sur la carte puis sélectionner Takeoff puis entrer l'altitude voulue et le véhicule Volera jusqu'à cette altitude et il reste en mode Loiter jusqu'à ce qu'on lui donne un nouvel ordre. Puis on clique avec le bouton droit de la souris sur l'endroit où on veut y aller dans la carte et sélectionner "Fly To Here Alt", puis entrer l'altitude désiré et le véhicule se déplacera et ainsi de suite. On peut aussi sélectionner " Fly To Coords" mais dans ce cas il faut lui donner les coordonnées de la destination voulue (figure 3.34).

Transmission de l'image et de la vidéo à la station de base :

Pour que notre système puisse transmettre à la station de base les images et les vidéos pour les traiter et ce en temps réel. Le quad-copter est équipé d'une caméra et d'un transmetteur vidéo pour assurer la capture d'images La station de base fait les traitements nécessaires sur les images et les vidéos reçus grâce au récepteur vidéo (3.36).

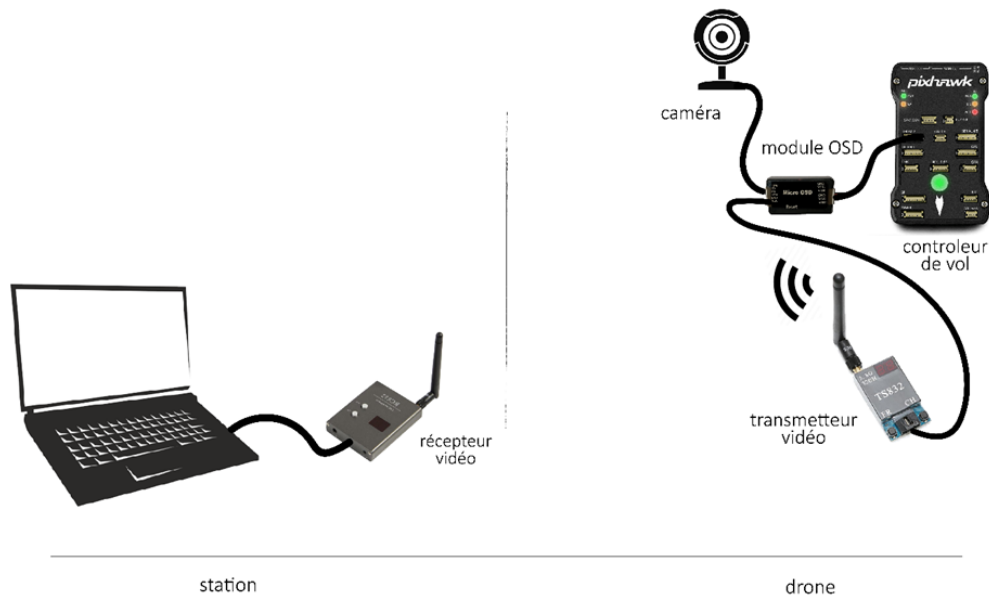


FIGURE 3.36 – illustration du système de transmission vidéo

3.4 Conclusion

Dans ce chapitre, nous avons utilisé *mission planner* pour planifier une mission dans l'environnement en question en prenant des photos qui seront transmises à la station de base et pourront être traitées et analysées pour un intérêt particulier. Par exemple l'exploration d'une zone hostile, la récupération d'information en temps réel. Un exemple d'exploitation sera le sujet du prochain chapitre intitulé « Un drone pour la détection feux de forêts »

Chapitre 4

La détection de feux de forêt

4.1 Introduction

Le monde souffre d'un fléau sans précédent dont l'être humain essaye toujours de faire face et d'y remédier. C'est les feux de forêt qui se déclenchent sans avertissement et se propagent sur une étendue boisée. Plusieurs difficultés sont fréquemment rencontrées quand on fait face à cette situation. Nous pouvons citer quelques obstacles :

- Approche difficile du foyer sur terrains accidentés, isolés ou éloignés ;
- Foyers étendus ou multiples ;
- Incendies criminels, moins prévisibles ;
- Progression rapide du feu, qui peut encercler les secours ;
- Contextes de sécheresses où l'eau peut manquer pour les secours.

Le point qui fera changer la situation c'est le temps d'intervention, vu que si on a l'information dès le début du feu, on pourra intervenir rapidement et éliminer le danger tant qu'il est faible. C'est là que l'utilisation d'un drone autonome d'intervient pour nous permettre d'avoir cet environnement de gestion et intervention. Il faudra aussi traiter rapidement les données (images) envoyées par le drone en utilisant une application de détection précoce de feux reposant sur les algorithmes de reconnaissance. Statistique en algerie!!?

4.2 Traitement Image

4.2.1 Définition :

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leurs qualités ou d'en extraire de l'information. La segmentation d'images est une étape importante et primordiale dans le processus de traitement et d'analyse d'images qui a pour but de partitionner une image en régions homogènes et regrouper des pixels ayant des propriétés communes suivant des critères prédéfinis.

4.2.2 Définition de l'image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui,

après affichage sur l'écran, a une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction de brillance analogique continue, définie dans un domaine borné, tel que sont les coordonnées spatiales d'un point de l'image et est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation [ANO12].

4.2.3 Image Numérique

Les images manipulées par un ordinateur sont numériques (représentées par une série de bits). L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

La numérisation d'une image est la conversion de celle-ci en matrice bidimensionnelle de valeurs numériques $I(n,m)$ où : n,m sont représentée par des coordonnées cartésiennes d'un point de l'image $I(n,m)$ et le niveau de gris ou de couleur en ce point.

Caractéristiques d'une image numérique

L'image numérique est déterminée par paramètres suivants :

1. Pixel

Le pixel est la contraction de l'expression anglaise 'Picture elements'. Le pixel, étant le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Dans une image couleur (R.V.B), un pixel peut être représenté sur trois octets, un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

2. Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multipliée par le nombre de colonnes nous donne le nombre total de pixels dans une image.

3. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les écrans d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur, plus grand est ce nombre, meilleure est la résolution.

4. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. Pour un observateur lointain, le mot luminance est substitué au mot brillance qui correspond à l'éclat d'un objet [11]. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

5. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste est défini par le rapport

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (4.1)$$

6. Bruit

Un bruit dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Le bruit peut provenir de différentes causes :

- Qualité de l'échantillonnage,
- Capacité des capteurs ou une mauvaise utilisation de ces derniers,
- Environnement lors de l'acquisition.

7. Images en couleur

Nous pouvons représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, etc.) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B).

8. Image à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images en niveaux de gris, on attribue à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel est représenté par un octet.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la « couleur » de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux. [Cst16]

On peut obtenir une image à niveaux de gris à partir d'une image en couleur en calculant la somme pondérée des couleurs comme dans la formule suivante :

$$gris = int(round(0.299 Rouge + 0.587 Vert + 0.114 Bleu)) \quad (4.2)$$

9. Histogramme de l'image

On veut souvent avoir une information sur la distribution des valeurs des pixels (niveaux de gris) dans une image, pour cela on utilise souvent une table qui donne le nombre de pixels de chaque niveau de gris dans l'image. Cette table, souvent représentée graphiquement, est appelée Histogramme de l'image, notée $h(x)$. Dans cet exemple, on veut représenter l'image de taille 256 x 256 pixels par son histogramme et pour cela, on commence par calculer la table histogramme ensuite on trace L'histogramme de l'image comme montré dans la 4.1

x : présente la valeur du pixel dans l'image. $y=h(x)$: présente le nombre de fois répété de cette valeur du pixel dans l'image. [D.B05]

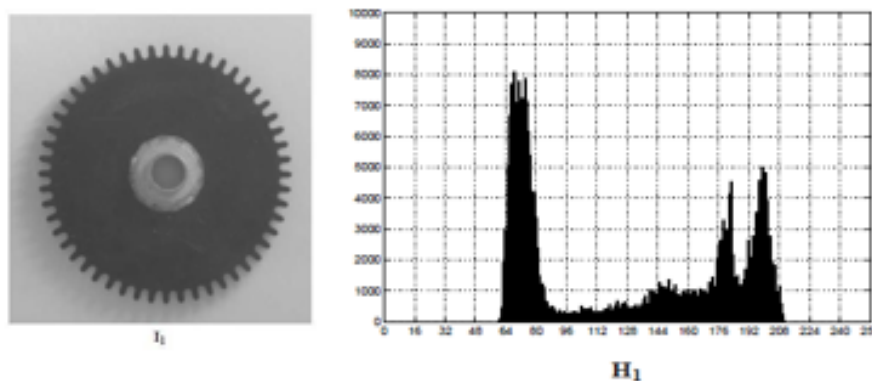


FIGURE 4.1 – Exemple d'histogramme d'une image.

4.3 Apprentissage automatique

L'apprentissage automatique concerne le développement d'algorithmes permettant de rendre une machine capable d'accomplir des tâches complexes sans avoir été explicitement programmée dans ce but. Son objectif est d'extraire et exploiter automatiquement l'information présente dans un jeu de données [Den18]

4.3.1 L'apprentissage automatique selon la nature des données en entrée :

Apprentissage supervisé

C'est une forme d'apprentissage automatique qui crée des modèles d'intelligence artificielle en se basant sur des données d'apprentissage « étiquetées ».

Apprentissage non supervisé

Désigne la situation d'apprentissage automatique où les données ne sont pas étiquetées. Il s'agit donc de découvrir les structures sous-jacentes à ces données non étiquetées.[1]

Apprentissage par renforcement

L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage.

4.3.2 L'apprentissage automatique selon le type de résultat

La classification

La classification est un type d'apprentissage supervisé, qui identifie la catégorie à laquelle un nouvel élément appartient sur la base d'un data set d'entraînement de données contenant des observations dont la catégorie est connue [16]. Aussi appelé partition des données, elle permet de grouper des objets (observations ou individus) dans des classes (clusters) de manière à ce que les objets appartenant à la même classe soient plus similaires entre eux qu'aux objets appartenant aux autres classes. Il existe plusieurs types d'algorithmes de classification :

- **Arbre de décision** : classe l'utilisation d'une structure d'arbre avec des règles if-then, exécutant l'entrée via une série de décisions jusqu'à ce qu'elle atteigne une condition de terminaison. Capable de modéliser des processus de décision complexes et très intuitif, mais peut facilement amplifier les données.
- **Forêt aléatoire** : un ensemble d'arbres de décision, avec sélection automatique de l'arbre le plus performant. Fournit la force de l'algorithme d'arbre de décision sans problème de surapprentissage.
- **Naïve Bayes classifieur** : un classificateur basé sur les probabilités. Calcule la probabilité que chaque point de données existe dans chacune des catégories cibles. Simple à mettre en œuvre et précis pour un large éventail de problèmes, mais sensible à l'ensemble des catégories sélectionnées.
- **K-voisin les plus proches** : classe chaque point de données en analysant ses voisins les plus proches parmi les exemples d'apprentissage. Simple à mettre en œuvre et à comprendre, efficace pour de nombreux problèmes, en particulier ceux à faible dimensionnalité. Fournit une précision inférieure par rapport aux algorithmes supervisés, et nécessite beaucoup de calculs.

4.4 Méthodes de Classification des images numériques

La classification d'image est le processus de segmentation des images en différentes catégories en fonction de leurs caractéristiques. Une caractéristique peut être les bords d'une image, l'intensité des pixels, la modification des valeurs de pixels et bien d'autres.

4.4.1 Classification non supervisée

La classification non supervisée (en anglais *unsupervised learning*) se base sur des algorithmes qui permettent de classer en regroupant les individus similaires entre eux sans passer par la phase d'apprentissage. Parmi ces algorithmes on trouve le K moyennes (K-means) et celui de carte Kohonen (réseaux de neurones).

4.4.2 Classification supervisée

Contrairement à la classification non supervisée, la classification supervisée (en anglais supervised learning), le système observe des couples de types entrée-sortie et apprend une fonction qui permet d'aboutir à la sortie à partir de l'entrée. Cette phase est appelée phase d'apprentissage ou d'entraînement. Les données de sortie correspondantes sont appelées classes.

Il existe différents algorithmes pour la classification supervisée. Parmi ces algorithmes on cite, le K plus proche voisin (KNN : K-NearestNeighbor), les arbres de décision et les machines à vecteur de support (SVM).

Dans notre cas on s'intéressera à l'implémentation d'un modèle de réseau de neurones Artificielle

4.4.3 Réseau de neurones artificiels :

La conception des réseaux de neurones artificiels s'appuie sur la structure des neurones biologiques du cerveau humain. Les réseaux de neurones artificiels peuvent être décrits comme des systèmes composés d'au moins deux couches de neurones - une couche d'entrée et une couche de sortie - et comprenant généralement des couches cachées intermédiaires (« hiddenlayers »). Les réseaux de neurones artificiels sont un système d'apprentissage supervisé constitué d'un grand nombre d'éléments simples, appelés neurones ou perceptrons. Chaque neurone peut prendre des décisions simples et remonte ses décisions vers d'autres neurones, organisés en couches interconnectées. Ensemble, le réseau neuronal peut émuler presque toutes les fonctions et répondre à pratiquement toutes les questions, avec suffisamment d'échantillons de formation et de puissance de calcul. Un réseau neuronal « peu profond » ne comporte que trois couches de neurones comme illustré en la figure (Figure 4.2 (a)) :

- Une couche d'entrée qui accepte les variables ou entrées indépendantes du modèle.
- Une couche cachée.
- Une couche de sortie qui génère les prédictions.

Un réseau neuronal profond a une structure similaire, mais il a deux ou plusieurs « couches cachées » de neurones qui traitent les entrées (voir (Figure 4.2.(b)) Goodfellow, Bengio et Courville [3] ont montré que si les réseaux neuronaux peu profonds sont capables de résoudre des problèmes complexes, les réseaux de neurones profonds sont plus précis et améliorent la précision à mesure que de nouvelles couches de neurones sont ajoutées. Des couches supplémentaires sont utiles jusqu'à une limite de 9 à 10. Aujourd'hui la plupart des modèles et implémentations de réseaux neuronaux utilisent un réseau profond entre 3 à 10 couches de neurones.

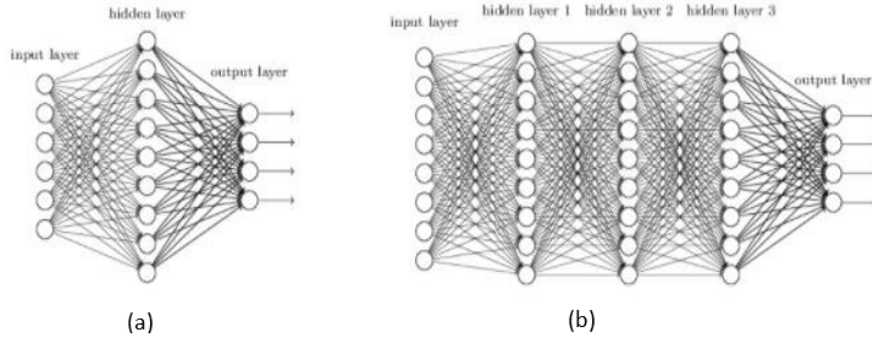


FIGURE 4.2 – Réseau de neurone artificiel (a) et profond(b) [2]

Les domaines d'application des réseaux neuronaux sont souvent caractérisés par une relation entrée-sortie de la donnée d'information :

- La reconnaissance d'image.
- Les classifications de textes ou d'images.
- Identification d'objets.
- Prédiction de données.
- Filtrage d'un set de données.

Au sein d'un réseau de neurones artificiels, le traitement de l'information suit toujours la même séquence : les informations sont transmises sous la forme de signaux aux neurones de la couche d'entrée, où elles sont traitées. À chaque neurone est attribué un « poids » particulier, et donc une importance différente. Associé à la fonction dite de transfert, le poids permet de déterminer quelles informations peuvent entrer dans le système.

À l'étape suivante, une fonction dite d'activation associée à une valeur seuil calculent et pondèrent la valeur de sortie du neurone. En fonction de cette valeur, un nombre plus ou moins grand de neurones sont connectés et activés.

Cette connexion et cette pondération dessinent un algorithme qui fait correspondre un résultat à chaque entrée. Chaque nouvelle itération permet d'ajuster la pondération et donc l'algorithme de façon à ce que le réseau donne à chaque fois un résultat plus précis et fiable.

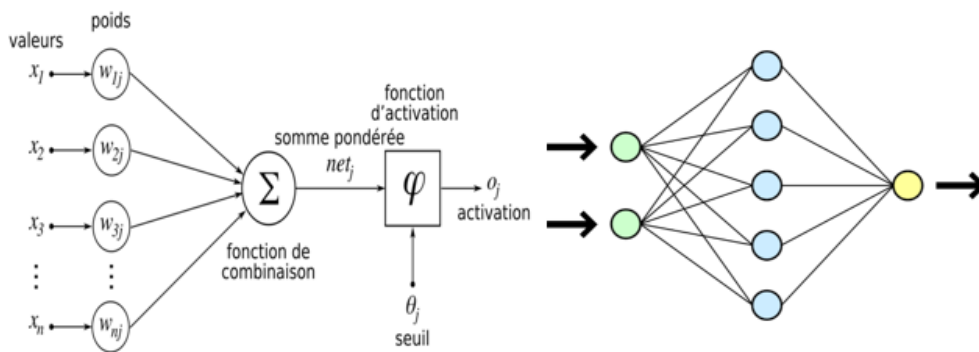


FIGURE 4.3 – Vue simplifiée d'un réseau artificiel de neurones.

Les fonctions d'activation

Une fonction d'activation est une équation mathématique qui détermine la sortie de chaque élément (perceptron ou neurone) dans le réseau neuronal. Comme illustré en (Figure VI-4), il prend l'entrée de chaque neurone et la transforme en sortie, généralement entre un 0 ou -1 et 1. Les fonctions d'activation classiques utilisées dans les réseaux de neurones incluent la fonction lineaire, step sigmoid, tanh et Softmax. De nouvelles fonctions d'activation, destinées à améliorer l'efficacité des réseaux de neurones, incluent ReLu et Swish '[4].

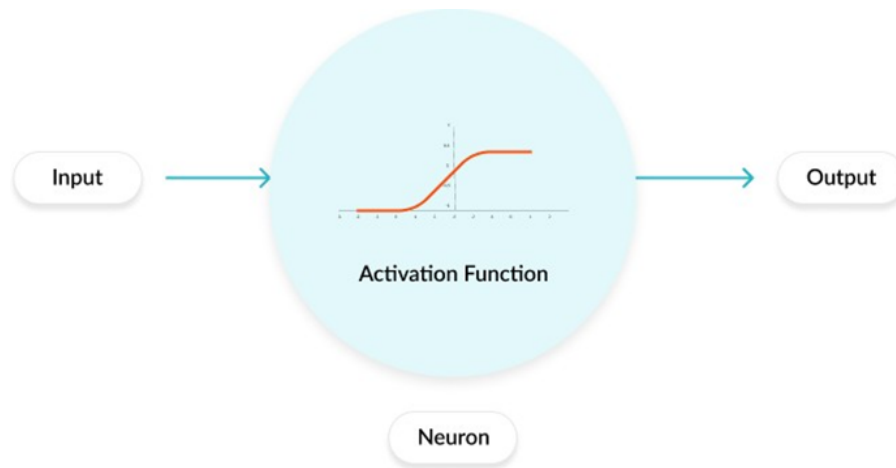


FIGURE 4.4 – Structure simplifié d'un neurone artificiel.

Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée.

Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif

. Les architectures de réseaux neuronaux peuvent être divisées en 4 grandes familles :

- Réseaux de neurones propagation avant « Feed forwarded »
- Réseaux de neurones récurrent (RNN)
- Réseaux de neurones à résonance
- Réseaux de neurones auto-organisés.

Les réseaux de neurones feed-forward sont des réseaux où les neurones ne sont connectés que dans un sens, le sens orienté de l'entrée vers la sortie.

Les réseaux de neurones récurrents sont des réseaux où les neurones de sortie par exemple peuvent voir leur sortie utilisée comme entrée d'un neurone d'une couche précédente (nous verrons ce que sont les couches) ou de la même couche.

Un perceptron

Est un algorithme d'apprentissage supervisé de classifieurs binaires (c'est-à-dire séparant deux classes) [19], considéré être un réseau de neuronal simple monocouche qui ne contient

aucun cycle (il s'agit d'un réseau de neurones à propagation avant) .

Il existe 2 types de perceptrons : les perceptrons feed-forward et les perceptrons récurrents. Les perceptrons récurrents sont ceux qui alimentent leurs entrées avec leurs sorties, alors que les perceptrons feed-forward non.

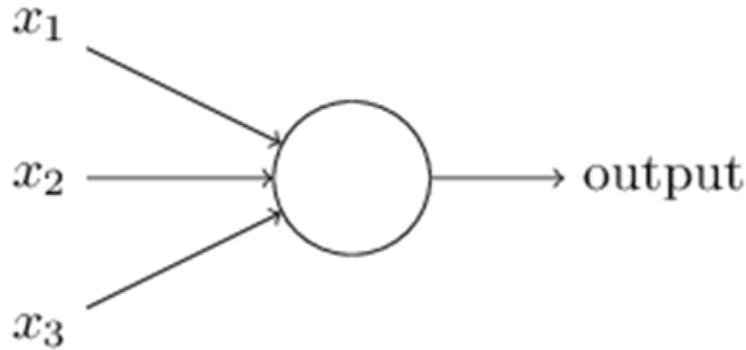


FIGURE 4.5 – Modélisation d'un perceptron.

Un perceptron multicouche MLP Il s'agit maintenant de généraliser le perceptron. Nous pouvons ainsi disposer les neurones en plusieurs couches. Ainsi les informations en entrée sont connectées à tous les neurones de la première couche, tous les neurones de la première couche sont connectés à tous les neurones de la seconde couche, et ainsi de suite jusqu'à la dernière couche, appelée couche de sortie. Toutes les couches exceptée la couche de sortie sont considérées comme "couches cachées".

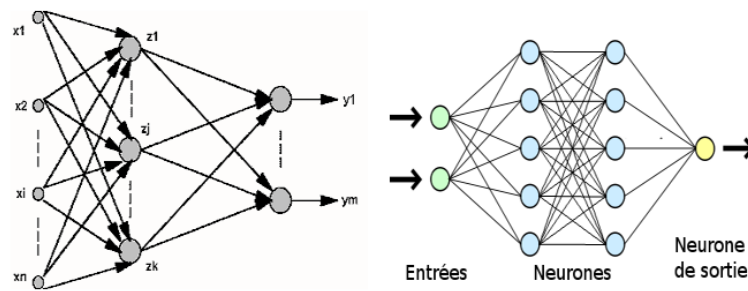


FIGURE 4.6 – Modélisation du perceptron multicouche.

4.5 Les réseaux de neurones convolutionnels CNN

Les réseaux de neurones convolutionnels sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais « Convolutional Neural Network », ils consistent en un empilage de MLPs, dont le but est de prétraiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Les CNN comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local.

Au final, les cartes de convolutions sont mises concaténées en un vecteur de caractéristiques, appelé code CNN.

Dans un CNN Les couches sont peu connectées ou partiellement connectées plutôt que complètement connectées dans un MLP.

Un CNN est un réseau profond composé de quatre types de couches : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche fully connected.

- **Couche de convolution** : c'est la couche la plus importante et le cœur des éléments constitutif du réseau convolutif, et c'est aussi elle qui effectue le plus grand des calculs lourds. Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée.
- **Couche de mise en commun (pooling)** : est souvent placée entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling (ou sub-sampling) consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.
- **La couche de correction ReLU** : remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation.
- **La couche fully-connected** : constitue toujours la dernière couche d'un réseau de neurones, convolutifs ou non, elle n'est donc pas caractéristique d'un CNN. Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

La figure suivante montre l'architecture d'un réseau de neurone de type CNN :

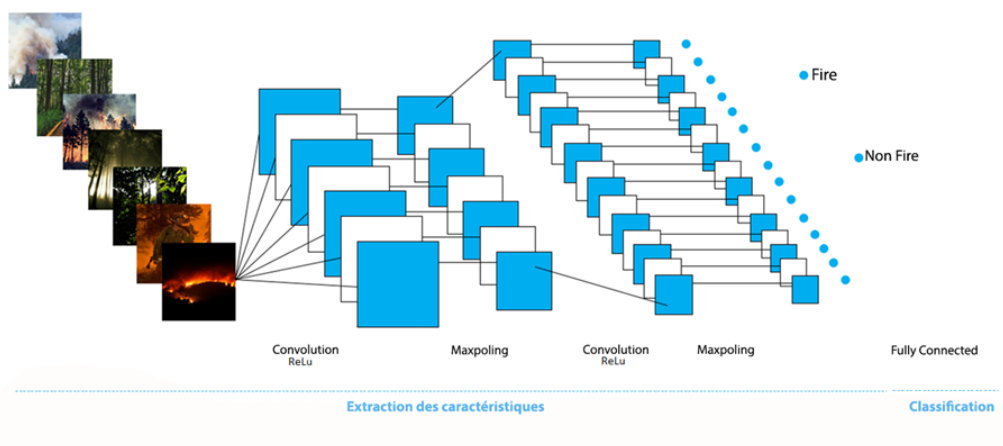


FIGURE 4.7 – Fonctionnement d'un réseau de neurones convolutifs.

4.5.1 Développement du modèle et implémentation CNN :

Les progrès récents dans le traitement d'image ont permis aux systèmes basés sur la vision de détecter les incendies à l'aide de réseaux de neurones convolutionnels. Dans notre projet nous avons implémenté deux architecture CNN de détection des feux de forêt, pour cela nous avons créé un modèle Séquentiel et un autre modèle InceptionV3 personnalisé.

Implémentation du model séquentiel :

Dans cette implémentation nous avons créé une architecture CNN pour classifier les images en deux catégories (images de feu, images dont il n'y a pas de feu), pour cela nous avons suivis les étapes suivantes :

- **Préparation de l'ensemble de données (segmentation de données)**

Pour l'implémentation de notre model nous avons utilisé un ensemble de données (Dataset)

contenant un total de 1219 images, nous avons devise cet ensemble en 2 sous ensemble (deux dossier) le premier contient des images utilisées pour l'entraînement de notre model, le deuxième contient des images utilisées pour la validation du même model.

Cela bien dit les images des deux sous ensemble sont complètement diffèrent, c'est-à-dire quel que soit une image appartenant à un sous ensemble elle ne doit pas appartenir à l'autre pour assurer que notre segmentation soit pertinente.

Le premier sous ensemble (dossier) appelé « Train » contient 980 images en total, ces images sont devisées en 2 classes, la première classe contient des images de feu ou d'incendie (un dossier contenant 694 images appelé « Fire »), la deuxième classe contient des images dont il n'y a pas d'incendie (un dossier contenant 286 images appelé « NoFire »).

Le deuxième sous ensemble (dossier) appelé « Validation » contient 239 images en total, ces images sont devisées en 2 classes, la première classe contient des images de feu ou d'incendie (un dossier de 171 images appelé « Fire », la deuxième classe contient des images dont il n'y a pas d'incendie (un dossier contenant 68 images appelé « NoFire »).

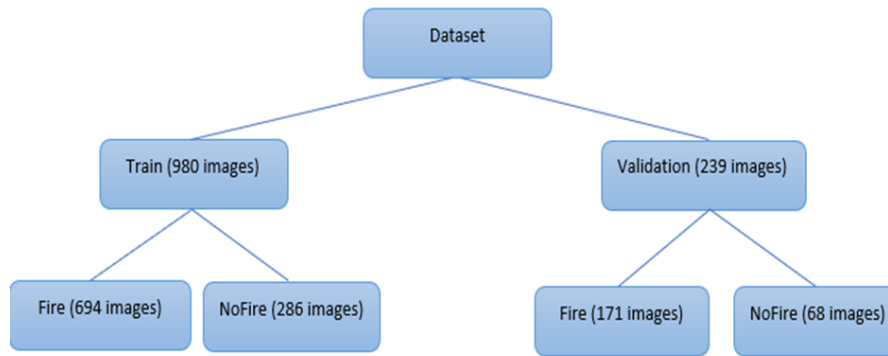


FIGURE 4.8 – Segmentation des données du modèle séquentiel

- **Prétraitement :**

Pour cette étape nous avons utilisé la fonction `ImageDataGenerator` offert par Keras qui nous permet de :

- Redimensionner nos images pour avoir les mêmes dimensions pour toutes les images (mêmes hauteur et largeur « 224X224 »).
- Faire la normalisation des données (images), c'est-à-dire assurer que les valeurs des pixels de nos images soient assez petites pour faciliter leurs traitements, dans notre cas chaque pixel contient un vecteur de trois valeurs allant de 0 à 255 vu que nos images sont en format RGB ou RVB (rouge vert et bleu) en français, donc nous avons divisé ses valeurs sur 255 pour avoir des nouvelles valeurs entre 0 et 1.
- Augmenter l'ensemble de données, c'est-à-dire augmenter le nombre d'image dans notre ensemble de données, pour cela nous avons appliqué trois techniques : le retournement horizontal (horizontal flipping), la rotation et le changement de hauteur (height shifting).
- Etiqueter nos images pour avoir deux class d'images (Fire, NoFire).

- **Création du modèle :**

L'architecture de notre modèle contient 12 couches (layers) plus une couche Flatten en total :

- Trois paires de couches « Conv2D, MaxPooling2D » :
Conv2D : permet de faire des convolutions sur des images (chaque couche contient 96, 256, 384 neurones respectivement avec une fonction d'activation « relu » pour chacun).
MaxPooling2D : Réduit la taille des données.
- Après les trois couches précédentes on a fait un Flatten, c'est-à-dire rendre toutes les données à une dimension.
- Deux paires de couches « Dense, Dropout » suivis d'une couche « Dense » :
Dense : une couche régulière de neurones dans un réseau neuronal (chaque couche contient 2048, 1024 neurones respectivement avec une fonction d'activation « relu »

pour chacun sauf la dernière).

Dropout : réduit la complexité de notre modèle pour éviter le sur apprentissage.

- La dernière couche est la couche Dense contient 2 neurones avec une fonction d'activation « Softmax », elle nous donne la distribution de probabilité pour les deux classes « Fire et Nonfire ». On peut également utiliser la fonction d'activation « sigmoïde » à la dernière couche en changeant le nombre de classes à 1.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 54, 54, 96)	34944
max_pooling2d (MaxPooling2D)	(None, 26, 26, 96)	0
conv2d_1 (Conv2D)	(None, 22, 22, 256)	614656
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 256)	0
conv2d_2 (Conv2D)	(None, 6, 6, 384)	2457984
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 384)	0
flatten (Flatten)	(None, 1536)	0
dropout (Dropout)	(None, 1536)	0
dense (Dense)	(None, 2048)	3147776
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
dropout_2 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 2)	2050
Total params: 8,355,586		
Trainable params: 8,355,586		
Non-trainable params: 0		

FIGURE 4.9 – Architecture du modèle séquentiel.

- **Résultats obtenus :** Après l'entraînement pour 50 epochs (parcourir toutes les images de l'ensemble de données 50 fois) nous avons obtenu une précision d'entraînement de 96,83% et une précision de validation de 94,98%. La perte de formation et de validation est de 0,09% et 0,13% respectivement.

Dans la (Figure 4.10) on a 2 graphes qui nous montrent la précision et la perte d'entraînement et de validation durant les epochs de notre entraînement.

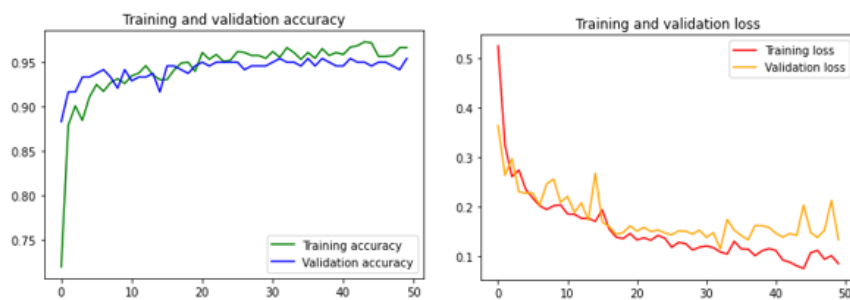


FIGURE 4.10 – Diagramme de précision et de perte du modèle séquentiel.

Implémentation du model InceptionV3 :

Les résultats que nous avons obtenus avec le model séquentiel étaient assez bonne pour l'entraînement et la validation, mais lorsque nous l'avons testé avec des images qui n'appartenaient pas à notre Dataset il n'était pas vraiment précis et s'est trempé dans la classification de quelques images, nous l'avons testé sur un ensemble d'image que nous avons choisi aléatoirement (un ensemble de 145 images) et nous avons obtenu une précision de 64.6%.Voici quelques images avec leur prédiction :

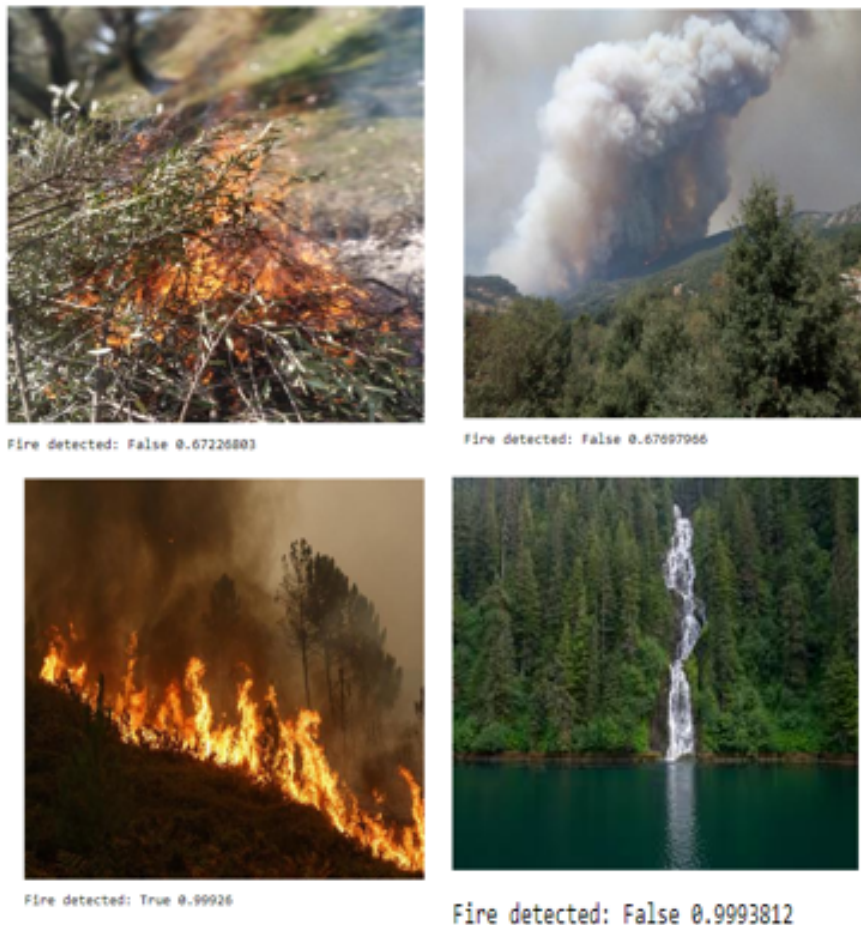


FIGURE 4.11 – Resultats de quelques tests d'images avec Model séquentiel.

Dans la figure ci-dessus on remarque que notre model séquentiel a réussi à faire une bonne prédiction pour les deux images en bas mais il s'est trempé dans les deux en haut.

Pour remédier à ce problème nous avons opté pour une nouvelle technique qui consiste à utiliser un model déjà entraîné offert par Tensorflow, ce modèle est appelé InceptionV3.

Inceptionv3 est un réseau de neurones convolutionnels développé pour aider à l'analyse d'images et à la détection d'objets, il a fait ses débuts en tant que module pour Googlenet. Il s'agit de la troisième édition du Inception Convolutional Neural Network de Google, il est initialement présenté lors du Défi de reconnaissance ImageNet.

Tout comme ImageNet Inception peut être considéré comme une base de données d'objets

visuels classifiés, Inception aide à la classification des objets dans le monde de la vision par ordinateur. Une de ces utilisations est dans la biologie, où il aide à la recherche sur la leucémie. Le nom (Inception) a été nommé de cette façon après qu'une phrase sur Internet est devenue populaire « we need to go deeper » (Nous devons aller plus loin), cette réplique vient du film Inception de Christopher Nolan.

- **Préparation de l'ensemble de données :**

Cette fois ci nous avons utilisé une Dataset contenant un total de 2000 images, nous avons divisé cet ensemble en 2 sous-ensembles (deux dossiers) le premier contient des images utilisées pour l'entraînement de notre model, le deuxième contient des images utilisées pour la validation du même model. Et bien sûr comme la première fois les deux sous ensemble sont complètement différents pour assurer une segmentation pertinente. Le premier sous ensemble (dossier) appelé « Validation » contient 1800 images en total, ces images sont divisées en 2 classes, la première classe contient des images de feu ou d'incendie (un dossier contenant 900 images appelé « Fire »), la deuxième classe contient des images dont il n'y a pas de feu (un dossier contenant 900 image appelé « Neutral »). Le deuxième sous ensemble (dossier) appelé « Test » contient 200 images en total, ces images sont divisées en 2 classes, la première classe contient des images de feu ou d'incendie (un dossier de 100 images appelé « Fire », la deuxième classe contient des images dont il n'y a pas de feu (un dossier contenant 100 image appelé « Neutral »).

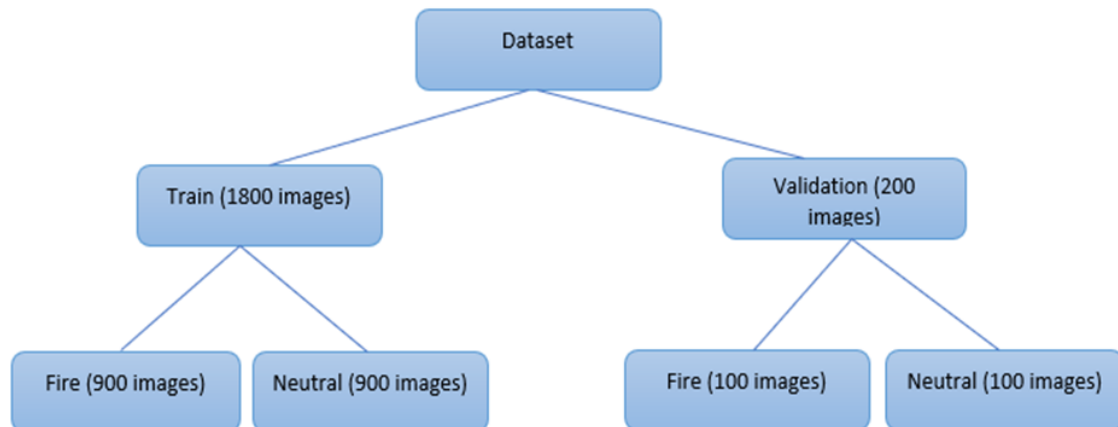


FIGURE 4.12 – Segmentation des données modèle inceptionV3.

- **prétraitement :**

Pour ce modèle, les prétraitements sont les même avec le model séquentiel sauf l'augmentation des données car nous n'avons utilisé que deux techniques vues que notre Dataset et plus grand, le retournement horizontale (horizontal flipping) et le Zoom (agrandir l'image).

- **Création du model InceptionV3 personnalisé :**

Nous avons importé le model InceptionV3 de keras (une API écrite en python), puis nous avons ajouté quelque couches (layers) à la fin du model pour avoir un total de 317couches,

Chaque couche a comme entrée.

La sortie de la couche précédentes, elles sont enchainées comme suit :

- Une couche « GlobalAveragePooling2D » ayant la sortie du model InceptionV3 comme entrée, elle réduit la taille de données.
- Une couche « Dense » contenant 2048 neurones avec une fonction d'activation « ReLu », suivi d'une couche « Dropout ».
- Une couche « Dense » » contenant 1024 neurones avec une fonction d'activation « ReLu », suivi d'une couche « Dropout ».
- Une dernière couche « Dense » contenant 2 neurones avec une fonction d'activation « softmax ».

• Résultats obtenus :

Nous avons entrainé le model pour 20 epochs et nous avons obtenues les résultats suivants (la figure 4.13) :

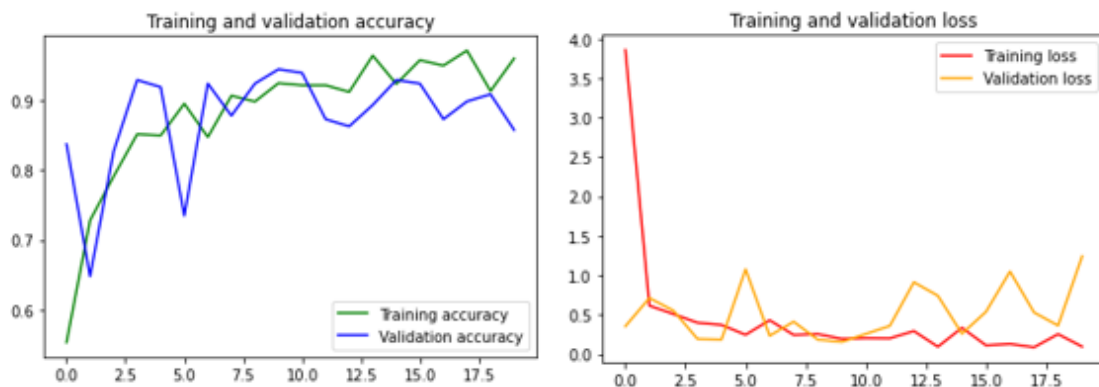


FIGURE 4.13 – Résultats du premier entrainement du model InceptionV3.

En analysant les résultats obtenus nous avons remarqué que durant quelques epochs il y a une grande marge de différence entre la précision d'entrainement et la précision de validation (même chose pour la perte). Nous avons essayé d'entrainer notre model encore plus, donc nous l'avons entrainé pour 10 epochs de plus (donc un total de 30epochs), mais nous avons eu un Overfitting (surentrainement), il y a une grande marge de différence entre la précision d'entrainement et la précision de validation (même chose pour la perte (Figure 4.14)).

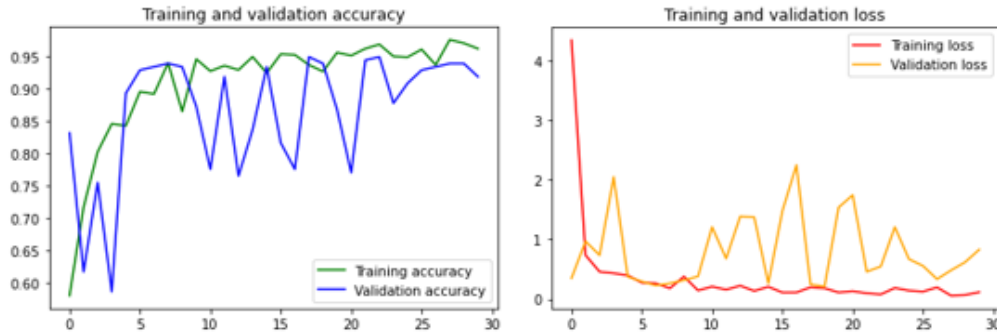


FIGURE 4.14 – Résultats d’entraînement de 10 epochs de plus sur notre modèle InceptionV3

Pour remédier à ce problème nous avons opté pour une autre technique, au lieu d’entraîner notre modèle pour une deuxième fois pour 10 epochs, nous avons essayé d’entraîner que quelques couches (layers). L’architecture de notre model contient 317 couches (layers), alors nous avons essayé d’entraîner que les dernière couches, en premier temps nous avons entraîné que les dernières 48 couches, puis nous avons augmenté le nombre de couches entraîne jusqu’à 68 puis, ensuite nous avons encore augmenté jusqu’à 78, nous avons obtenu les résultats suivants :

— Entraînement des dernières 48 couches :

Nous avons lancé un entraînement de 10 epochs pour les 48 dernières couches de notre model, nous avons obtenu de bons résultats par rapport aux précédents, c’est-à-dire la marge de différence entre les résultats d’entraînement et de validation a baissé (Figure 4.15)

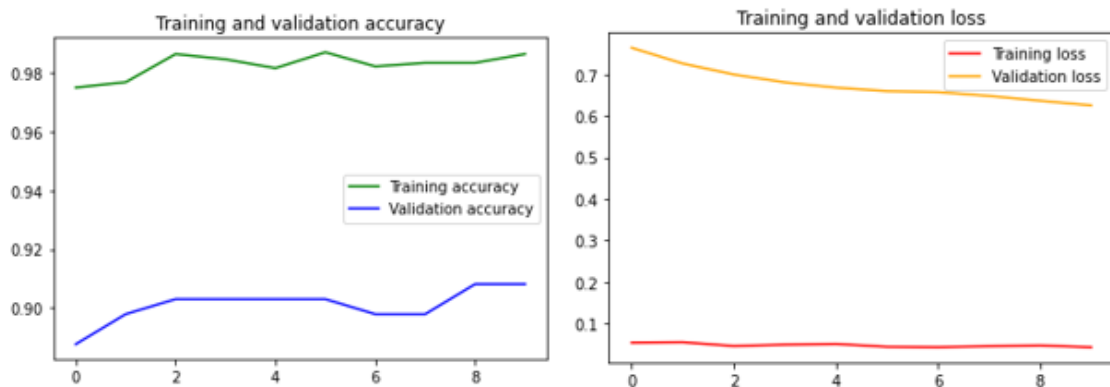


FIGURE 4.15 – Résultats d’entraînement de 48 couches pour 10 epochs

— Entraînement des dernière 68 couches :

Nous avons lancé un entraînement de 10 epochs pour les 68 dernières couches de notre model, nous avons obtenu des résultats encore meilleurs que les précédents, c’est-à-dire la marge de différence entre les résultats d’entraînement et de validation a baissé encore plus (Figure 4.16)

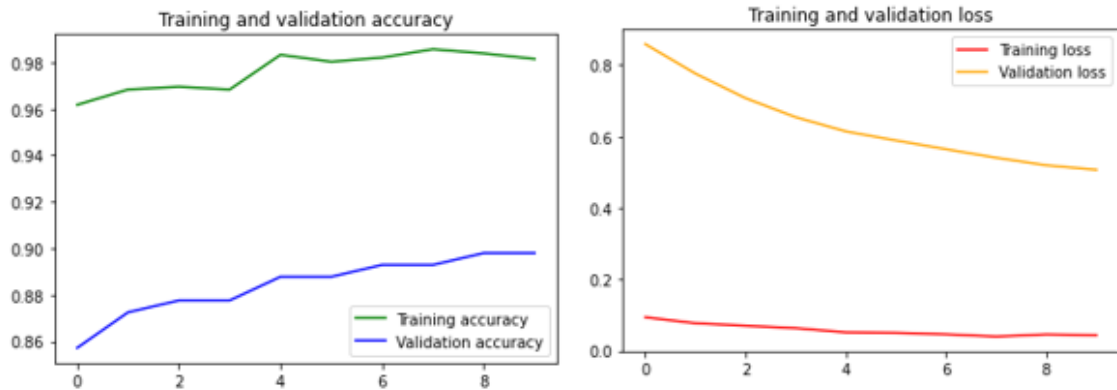


FIGURE 4.16 – Résultats d’entraînement de 68 couches pour 10 epochs.

— Entraînement des dernières 78 couches :

Cette fois-ci nous avons lancé un entraînement de 10 epochs pour les 78 dernières couches, mais nous avons obtenu des résultats moins bons que les précédents, c’est-à-dire la marge de différence entre les résultats d’entraînement et de validation a augmenté (Figure 4.17).

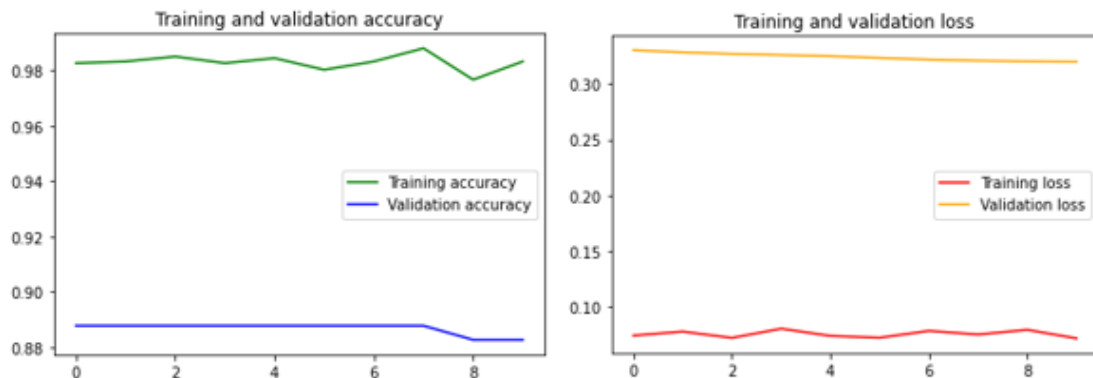


FIGURE 4.17 – résultats d’entraînement de 78 couches pour 10 epochs.

Durant cette expérience nous avons remarqué qu’en augmentant le nombre des dernières couches entraînées nos résultats étaient de mieux en mieux, mais cela n’est valide que si le nombre est inférieur ou égal à 68, car lorsque le nombre des dernières couches entraînées est supérieur à 68 nos résultats sont dégradés. Donc nous avons conclu que le meilleur choix était de faire un deuxième entraînement uniquement pour les dernières 68 couches pour 10 epochs, alors nous avons opté pour l’utilisation du modèle obtenu pour le traitement des images envoyées par notre drone pour les classer comme image de feu ou image de non feu.

D’ailleurs nous l’avons testé sur le même ensemble d’image que nous avons choisi aléatoirement (un ensemble de 145 images pour le test) et nous avons obtenu une précision de 91.03%. Voici les résultats obtenus après avoir testé sur les mêmes images que dans la Figure 4.18 :

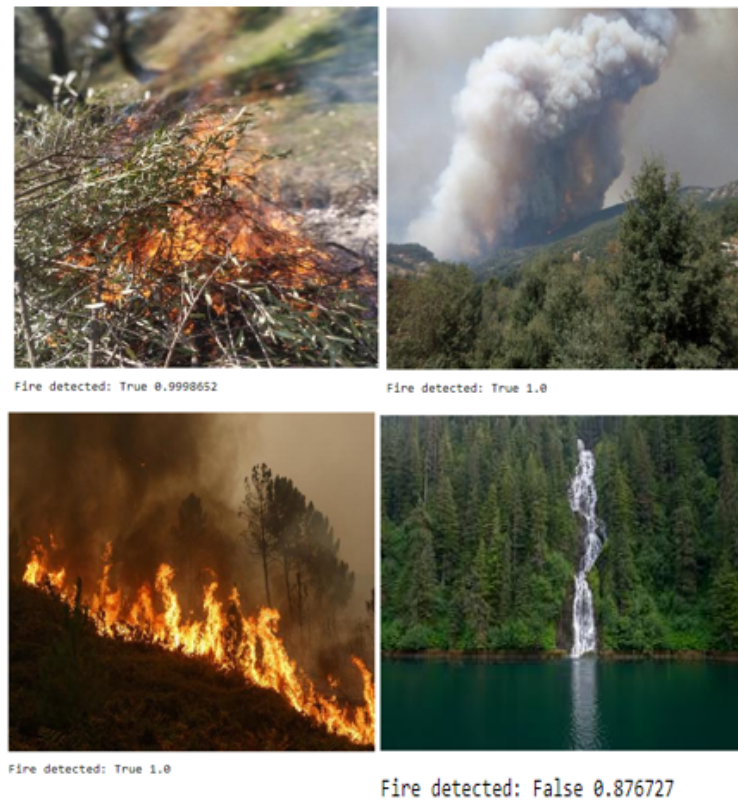


FIGURE 4.18 – Resultats de quelques tests d’images avec Model InceptionV3 personnalisé

4.6 Outils utilisés :

Durant l’implémentation de notre réseau de neurones nous avons utilisées les outils suivants :

4.6.1 Anaconda3

Une distribution libre et open source des langages de programmation Python et R appliqué au développement d’applications dédiées à la science des données et à l’apprentissage automatique.



FIGURE 4.19 – Logo Annaconda.

4.6.2 Jupyter Notebook :

Le bloc-notes Jupyter est une application Web open source qui vous permet de créer et de partager des documents contenant du code en direct Accessible via Anaconda.



FIGURE 4.20 – Logo Jupyter.

4.6.3 Python 3.6

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire.

Il prend en charge de multiparadigme de programmation, y compris orientée objet, fonctionnelle, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation.



FIGURE 4.21 – Logo Python.

4.6.4 Tensorflow 1.13.1 :

TensorFlow est une bibliothèque open source d'apprentissage automatique, créée par Google, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning et regroupe un grand nombre de modèles et d'algorithmes de ces applications. Nous avons utilisé cette bibliothèque pour la reconnaissance d'image.



FIGURE 4.22 – Logo Tenserflow.

4.6.5 Bibliothèques utilisées

1. Numpy :

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion).

2. Matplotlib :

est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter, wxPython, Qt ou GTK +.

3. Keras :

est une bibliothèque open source écrite en python et permettant d'interagir avec les algorithmes de réseaux de neurones profonds et de machine Learning, notamment Tensorflow.

4. Pandas :

pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python. Pandas est un projet sponsorisé par NumFOCUS. Cela contribuera au succès du développement de pandas en tant que projet open source de classe mondiale. [Ajouter quelques exemples d'images avec étiquette feu, et non feu selon le résultat fourni par le modèle]

4.7 Conclusion

Dans ce chapitre, nous avons introduit quelques concepts de base qui servent de fondement à la compréhension des différentes techniques de traitement d'image, à savoir quelques définitions et caractéristiques sur les images numériques tout en passant par les notions de filtrage (linéaire et non-linéaire).

Les pré-traitements d'images permettent d'améliorer la qualité de l'image et de supprimer les bruits en vue de traitements ultérieurs.

Après avoir vu quelques notions de base sur l'apprentissage automatique et les différentes méthodes de classification d'images numériques, nous avons expliqué en détails les différentes étapes que nous avons suivies à fin d'élaborer notre modèle de classification, A savoir les pré-traitements, la création, et nous avons représenté et discuté nos résultats sous forme de graphe.

Enfin nous avons présenté les différents outils et environnement utilisés afin de réaliser notre system de classification.

Conclusion Générale

Grâce au développement des technologies de communication, des systèmes embarqués, des logiciels et des systèmes intelligents, le drone prend de plus en plus d'ampleur dans la vie civile et s'infiltrer pratiquement dans tous les domaines. Notre travail consiste à implémenter un système intelligent sur un drone quadcopter autonome qui permettra de détecter les feux de forêts en temps réel. Au début nous avons présenté un état de l'art sur les drones, leurs classification et domaines d'utilisation, par la suite nous nous sommes penchés sur les drones quad-copter en mettant en évidence leur fonctionnement, leurs différents composants ainsi que les modes de vol existants. Après cela nous avons présenté notre réalisation en deux parties distinctes ; la configuration et la planification de mission pour notre quad-copter et enfin le développement de notre modèle de reconnaissances de feux forêts.

Avec ce travail, nous sommes arrivés à atteindre notre objectif qui était l'implémentation d'un drone intelligent en développant un module de caméra embarquée capable de détecter les feux dans les forêts et d'assurer une intervention rapide et précise sur le terrain par les organismes concernés. Ce travail nous a permis la découverte d'un nouveau domaine qu'est les drones, considéré comme un domaine très vaste et innovant. En plus, il nous a apporté énormément de connaissances dans plusieurs disciplines.

Nous espérons que notre travail soit à la hauteur des attentes, Pour ce qu'est des perspectives à réaliser, plusieurs voies peuvent être envisagées, A titre indicatif, nous proposons d'examiner et de développer les points suivants :

- Basculer vers un system 100%embarqué en effectuant tous les traitements au niveau du system air. Par exemple en utilisant une carte Raspberry Pi ce qui permettra non seulement une sécurisation de nos données puisque on n'aura pas à les transmettre en temps réel à la station sol, mais aussi un gain de performance et d'autonomie.
- Améliorer notre modèle des reconnaissances en utilisant une Dataset plus volumineuse et plus variées pour l'entraînement et qui englobera les différentes situations climatiques (vu que dans notre cas, le modèle n'est pas assez entraîné pour différencier entre une fumée et un brouillard.
- Travailler sur l'aspect sécurisation des données.

Bibliographie

- [Adr13] D Adrien. *Stratégies de commande pour la navigation autonome d'un drone*. 2013.
- [ANO12] Hinde ANOUAL. *Modélisation, Observation et Commande d'un Drone Miniature Birotor Coaxial*. Thèse de Doctorat, 2012.
- [Cst16] Hector Csta. *Prutivil data analyse*. Packt publishing, 2016.
- [d08] Bento M. d. *Unmanned Aerial Vehicles : An Overview*. U.S, 2008.
- [D.B05] D.Boukhoulouf. *Généralités sur le traitement d'images*. Mémoire de Master, 2005.
- [Den18] Thierry Dencœux. *Introduction à l'apprentissage automatique-heudiasyc*. Université de Technologie de Compiègne Département Génie Informatique Heudiasyc, 14 septembre 2018.
- [KOE12] A. KOEHL. *Application à la détection des plaques d'immatriculation marocaines*. 2012.
- [Lev15] C Levaux. *Démesures. Une histoire du drone des 1960 à nos jours (Vol. 7)*. Intervalles, 2015.
- [Pfl06] J.-M Pflimlin. *Commande d'un minidrone à hélice carénée : De la stabilisation dans le vent à la navigation autonome*. 2006.
- [Poi08] D Poinot. *Commande d'un drone en vue de la conversion vol rapide - vol Stationnaire*. Ecole Nationale Supérieure de l'aéronautique et de l'espace, 2008.
- [Rid15] M. M. Rida. *Observation et Commande de Drones Miniatures à voilures Tournantes*. Tlemcen, 2015.
- [SAD17] SADAALLAH.N. *Planification de trajectoires pour un véhicule aérien autonome (UAV)*. Mémoire de Master, université (2017).

webographie

- [1] : https://fr.wikipedia.org/wiki/Apprentissage_non_supervis%C3%A9 (consulté le 12 juillet 2020)
- [2] : <https://blog.varonis.fr/classification-des-donnees-kesako/> (consulté le 15 juillet septembre 2020)
- [3] : <https://hal-amu.archives-ouvertes.fr/hal> (consulté le 20 juillet 2020)
- [4] : https://fr.wikipedia.org/wiki/Fonction_d%27activation (consulté le 3 septembre 2020)
- [5] : <https://fr.wikipedia.org/wiki/Perceptron> (consulté le 16 septembre 2020)
- [6] : <https://intelligence-artificielle.agency/tensorflow-notes-sur-le-tutoriel-3c-de-magnus-erik-hvass-pedersen-keras-api/> (consulté le 5 septembre 2020)
- [7] : <https://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones/#LIV> (consulté le 16 septembre 2020)
- [8] : <https://towardsdatascience.com/convolution-neural-network-for-image-processing> (consulté le 18 septembre 2020)
- [9] : <https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-par> (consulté le 22 septembre 2020)
- [10] : <https://ardupilot.org/planner/> (consulté le 28 septembre 2020)

Annexe

Description

Un quadri-rotor (ou quadcopter) est un drone qui appartient à la famille des voilures tournante, il est composé de quatre moteurs munis d'hélices, disposés aux quatre extrémités d'une croix en métal, ou autre matériau. Chaque moteur est relié à un contrôleur de vitesse, ces quatre étant pilotés par une carte de commande, qui reçoit ses ordres d'un récepteur de radio-commande ou d'un programme chargeable à l'intérieur. **A. Hardware**

La figure 4.23 ci-dessous montre notre drone quadri rotor avec les différents composants le constituant.

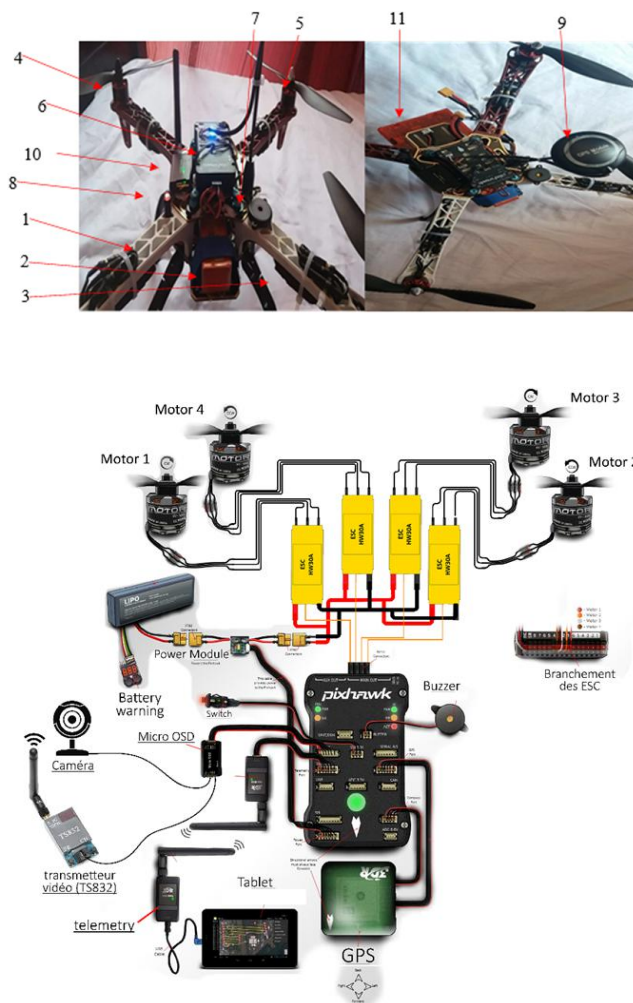


FIGURE 4.23 – Drone quadrirotor avec ses composants.

La description de chaque composant est comme suit :

1. Le châssis

C'est la structure du drone, c'est sur lui que l'on monte le reste des composants, de lui dépend le comportement général du drone et son autonomie. Des bras longs amélioreront la stabilité alors que des bras plus courts permettront des figures plus acrobatiques. On en trouve principalement à base d'aluminium ou de fibre de carbone, ou même de plastique comme dans notre cas, il existe des dizaines de formes différentes.

2. La batterie

C'est la source d'énergie du drone, dans notre cas nous avons utilisé une batterie en Lithium Polymère (LiPo), c'est une batterie avec une bonne capacité énergétique et un bon taux de décharge.

3. Les ESCs

Ce sont des petits boîtiers qui vont permettre de contrôler électroniquement la vitesse des moteurs.

4. Les moteurs

Permettent au quadri-rotors de voler, il existe des centaines de modèles plus ou moins adaptés à l'utilisation qui va être faite, dans notre cas, les moteurs choisis sont des moteurs Brushless 1000 KV (destinés pour un vol stable).

5. Les hélices

Elles doivent être adaptées à la taille du drone, et aux moteurs choisis, de grandes hélices assure un vol plus stable et pouvant soulever plus de poids et de petites hélices permettent d'avoir un vol Nerveux plus rapide et nécessite un châssis plus léger, elles sont fabriquées généralement du plastique, du bois et de la fibre du carbone. Dans notre cas nous avons utilisé des hélices en plastique dur.

6. Contrôleur de vol

C'est le cœur du drone. C'est lui qui va stabiliser l'engin mais aussi effectuer différentes tâches plus ou moins complexes comme suivre un plan de vol, atterrir automatiquement, etc. il existe plusieurs cartes pour le contrôle du vol comme Cartes Raspberry, cartes Arduino, cartes STM et carte Pixhawk, dans notre cas nous avons utilisé une carte Pixhawk.

7. Buzzer

C'est un biper qui permet de générer les différents sons du système.

8. Safety switch

Un bouton de démarrage sécurisé.

9. Module GPS

C'est un capteur qui permet au drone de se localiser géographiquement via le système GPS.

10. Radio télémétrie

C'est la composante qui permet de communiquer avec la station de contrôle, elle contient un module aérien et un autre terrestre.

B. Software

Pour les différents systèmes et logiciels nécessaires au fonctionnement, nous résumons comme suit :

La station de contrôle

Nous avons installé Mission Planner qui est une station de contrôle développée par la communauté Ardupilot. Pour l'installation il suffit de télécharger le fichier zip sur leur site et le décompresser (<http://firmware.ardupilot.org/Tools/MissionPlanner/>)