

الجمهورية الجزائرية الديمقراطية الشعبية

LA REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Saad Dahlab Blida 1  
Institut d'Aéronautique et des Études Spatiales  
Département Construction Aéronautique



**Mémoire de fin d'études**

En vue de l'obtention du diplôme de

**Master en Aéronautique**

*Option : Avionique*

**THEME**

**Reconnaissance par apprentissage automatique de signatures  
acoustique de drone**

Proposé et dirigé par :

**Dr. CHEGGAGA Nawal**

Réalisé par :

**Mr. BENABIDA Lalouani,**

**Mr. HAMDANI Younes;**

*Soutenue devant le jure composer de :*

**Dr. AZMADROUB Boussad**

**Président**

**Dr. CHOUTRI Kheirddine**

**Examineur**

**Promotion : 2022 / 2023**

## ***Remerciement :***

Au nom d'ALLAH le tout-Puissant, nous adressons un immense remerciement pour sa guidance divine qui nous a accordé la foi, la volonté, des moyens et du courage nécessaires pour atteindre ce niveau. Nous lui sommes reconnaissants de nous avoir permis d'accomplir cette thèse.

Tout d'abord, nous tiens d'exprimer notre gratitude à notre encadreur Mme CHEGAGA NAWAL qui nous a guidés pendant notre travail par ses conseils, ses encouragements et ses remarques pertinentes. Ainsi que votre expertise et votre vaste connaissance dans le domaine et votre capacité à inspirer ont été essentielles pour mener à bien notre projet dans les meilleures conditions.

Nous remercions également les membres du jury qui ont généreusement accepté d'évaluer ce travail, qui a eu l'honneur de présider le jury de ce mémoire.

Nous tenons à exprimer nos sincères remerciements à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail. Votre soutien et votre implication ont été d'une importance capitale et nous sommes profondément reconnaissants envers vous.

## *Dédicace :*

*Je dédie ce modeste travail, a ceux qui mon donné la vie, tendresse, qui se sont sacrifiée pour mon bonheur et ma réussite à mes très chers parents **MOURAD** et **SAMIA***

*A mes frères **Zaki,Ishak** et ma sœur **Nour** aussi mon oncle **Younes** qui se réjouit toujours de mon succès et de mes progrès dans mes études, et à toute la famille **Benabida**.*

*A mon binôme Younes et à mes amis Fares , Imad Aissa, Raouf, Noufel, Ziad, Acil, Hamza, Aymen, Hocine.*

*Enfinement je dédie se modeste travail a mes chères collègues de spécialité Avionique Master 02 Promo 2022/2023.*

***Benabida Lalouani***

## **Dédicace**

*Ce projet fin d'étude est le fruit de mon travail acharné, de mes efforts et mes dévouement. Mais il est également le résultat du soutien inconditionnel et des encouragements que j'ai reçus des gens qui m'entourent.*

*Je dédie ce modeste travail à mes chers parents. A ma mère **KHENICHE DJAHEDA**, mon pilier de force pour ton amour inépuisable, tes sacrifices pour moi et ta patience infinie. Tes encouragements et ton dévouement ont été ma boussole dans ce voyage.*

*Au meilleur des pères **HAMDANI MOHAMED**, dont la sagesse et l'exemple de persévérance m'ont guide. Tes conseils, ta patience et ta confiance en mes capacités m'a poussé à repousser mes limites.*

*A mon frère **ISLEM** et ma sœur **IBTISSEM**, mes complices de toujours, vous avez été mes meilleurs allies. Votre soutien indéfectible et vos encouragements ont été des sources de motivation principale.*

*A toute ma famille, surtout mes deux oncles **OMAR** et **TOUFIK**. Vous êtes à la place de mon père. Je vous adresse mes sincères remerciements pour votre soutien continu et vos conseils.*

*Je tiens également à exprimer ma gratitude à mon binôme **ALOUANI** et à tous Mes camarades au long mon parcours universitaire **HAMZA, RAOUF, ZIAD, AISSA, ALLA, ACIL, KARIM, AYMEN, NIZAR, SADIK** et **ZAOUI**. Votre présence et votre amitié sincère ont illuminé mes journées et m'ont aidé à garder le cap, même dans les moments les plus difficiles.*

*Cette dédicace est un témoignage de ma gratitude éternelle envers vous tous. Merci d'avoir été là pour moi au long de ce voyage.*

**Younes Hamdani**

**Résumé:**

Le développement des drones et les menaces qu'ils représentent pour les sites sensibles rendent critiques leur localisation et identification. Plusieurs modalités sont disponibles pour traiter ce problème comme l'imagerie optique ou radar, la détection de communications radiofréquence ou encore l'acoustique. C'est sur cette dernière modalité que ce travail s'articule. Il s'agit de développer un module acoustique pour la détection et l'identification de sources acoustiques de type mini- drone. Afin de protéger des sites sensibles contre les drones artisanaux. Les résultats obtenus au cours de cette recherche démontrent clairement l'efficacité de réseaux de neurones spécifiquement CNN dans la classification de sons de drones transformés sous forme d'image de spectrogramme. Avec une précision de 98%.

**Mots clés :** Drones, détection acoustique, réseaux de neurone convolutifs, spectrogramme.

**Abstract:**

The development of drones and the threats they pose to sensitive sites make their location and identification critical. Several modalities are available to treat this problem such as optical or radar imaging, detection of radio frequency communications or acoustics. It is on this latter modality that this work is articulated. The aim is to develop an acoustic module for the detection and identification of mini-drone acoustic sources. To protect sensitive sites from artisanal drones. The results obtained during this research clearly demonstrate the effectiveness of CNN-specific neural networks in classifying transformed drone sounds as spectrogram images. With an accuracy of 98%.

**Keywords:** Drones, acoustic detection, convolutive neural networks, spectrogram.

إن تطوير الطائرات بدون طيار والتهديدات التي تشكلها على المواقع الحساسة تجعل موقعها وتحديد هويتها أمرًا بالغ الأهمية. تتوفر عدة طرائق لمعالجة هذه المشكلة مثل التصوير البصري أو الراداري أو اكتشاف اتصالات الترددات الراديوية أو الصوتيات. وبناءً على هذه الطريقة الأخيرة، يتم توضيح هذا العمل. الهدف هو تطوير وحدة صوتية لاكتشاف وتحديد المصادر الصوتية الصغيرة للطائرات بدون طيار. لحماية المواقع الحساسة من الطائرات الحرفية بدون طيار. تُظهر النتائج التي تم الحصول عليها خلال هذا البحث بوضوح فعالية الشبكات العصبية الخاصة بشبكة CNN في تصنيف أصوات الطائرات بدون طيار المحولة على أنها صور طيفية. بدقة 98٪.

**الكلمات الرئيسية:** طائرات بدون طيار، اكتشاف صوتي، شبكات عصبية تلافيفية، مخطط طيف.

## Table des matières

Remerciement

Dédicace

Résumé

Table des matières

Liste des figures

Liste des tableaux

Abréviations/Acronymes

Introduction générale

Chapitre 01: Généralité sur les drones

1.1 Introduction.....	4
1.2 Historique.....	4
1.3 Classification des drones.....	5
1.3.1 Classification selon P.A.E.....	5
1.3.2 Classification selon la configuration aérodynamique .....	7
1.4 Architecture et Composants (Partie Mécanique & Electronique).....	9
1.4.1 Composants .....	9
1.4.2 Hélices.....	10
1.4.3 Moteurs électriques .....	11
1.4.4 Contrôleurs de vitesse (ESC) .....	11
1.4.5 Contrôleur de vol.....	12
1.4.6 Le châssis .....	12
1.4.7 Les Batteries.....	13
1.4.8 La Caméra embarquée :.....	13
1.4.9 La Radio-commande (RC) .....	14
1.5 Domaine d'application.....	15
1.5.1 Applications militaires (marine, armée et force aérienne) .....	15
1.5.2 Applications civil .....	15
1.5.3 Les utilisations négatives des drones .....	15
1.6 Les systèmes de détection.....	16
1.6.1 Détection radar .....	17

1.6.2 Détection Radio Fréquence: .....	17
1.6.3 Détection optique .....	18
1.6.3.1 Caméras électro-optiques (EO) .....	18
1.6.3.2 Caméras infrarouges/thermiques .....	19
1.6.4 Détection acoustique .....	19
1.7 Les avantages et les inconvénients des systèmes de détection : .....	20
1.8 Conclusion .....	22
2 .1 Introduction.....	24
2.2 L'apprentissage en profondeur (Deep learning).....	24
2.3 Réseaux de neurones et son principe de fonctionnement .....	25
2.4 L'apprentissage :.....	26
2.4.1 Apprentissage supervisé.....	27
2.4.2 Apprentissage non supervisé.....	27
2.4.3 Apprentissage semi-supervisé.....	27
2.5 Réseau de neurones à convolution (CNN).....	28
2.6 Réseaux Neuronaux Récurents (RNN).....	30
2.7 Réseaux Long Short-Term Memory (LSTM).....	31
2.8 Les Réseaux Siamois .....	33
2.8.1 L'architecture des réseaux siamois .....	33
2.8.2 Formation des Réseaux Siamois(Training) .....	34
2.9 L'utilisation des réseaux de neurones .....	35
2.10 Les avantages et les inconvénients des réseaux de neurone .....	36
2.10.1 Les Avantages des réseaux de neurones .....	36
2.10.2 Les Inconvénients des réseaux de neurones .....	36
2.11 Conclusion .....	37
3.1 Introduction.....	39
3.2 Logiciel de travail .....	39
3.3 La collecte des données (dataset).....	39
3.4. La visualisation des sons.....	40
3.4.1. La représentation temporelle .....	40
3.4.2. La représentation fréquentielle.....	44
3.4.3. La représentation spectrale.....	46
3.5. Type et structure de réseau de neuronal.....	49
3.5.1 Bibliothèque utilisé .....	49
3.6 Algorithme de manipulation .....	50

3.6.1 Lecture des données .....	50
3.6.2 Visualisation des images paires.....	50
3.6.3. Création de réseau de neurone .....	51
3.6.4 L'entraînement de réseau .....	52
3.6.5 Renforcement de la dataset .....	54
3.6.6 Création de réseau CNN.....	55
3.6.7 Apprentissage de modèle .....	57
3.6.8 Partie de validation.....	60
3.7 Conclusion .....	61
Bibliographie .....	65



## Liste des figures

### Chapitre 01 : Généralité sur les drones

Figure 1. 1 : les drones donne l’histoire.....	4
Figure 1. 2 : Spectre de masse des différents drones selon le classement américain (en Kg) . .....	6
Figure 1. 3: Drone à aile battante.....	7
Figure 1. 4: Drone à voilure fixe.....	7
Figure 1. 5: Drones mono-rotors.....	8
Figure 1. 6: Les différents types de drone birotors .....	8
Figure 1. 7: Drone tri-rotor. ....	9
Figure 1. 8: Exemples des drones à rotors multiples. ....	9
Figure 1. 9: Exemples d’hélices pour drone. ....	10
Figure 1. 10: Les deux types d’hélices pour drone. ....	10
Figure 1. 11: Exemple d’un moteur Brushless.....	11
Figure 1. 12: Electronic Speed Controler (ESC) .....	11
Figure 1. 13: Types de châssis (Configuration en « X » et en configuration en « + »). 13	
Figure 1. 14: Batterie LiPo.....	13
Figure 1. 15: Exemple de caméra à embarquer sur quadricoptère.....	14
Figure 1. 16: Principe de fonctionnement d’une radiocommande.....	14
Figure 1. 17 : Système de détection de drones avec Radar.....	17
Figure 1. 18: Système de détection de drones Radio Fréquence .....	18
Figure 1. 19: Système de détection de drones par Caméras électro-optiques (EO).....	18
Figure 1. 20: Schéma synoptique de la méthode de détection acoustique [15]. .....	19

### Chapitre 02 : Les Réseaux De Neurone Artificiels

Figure 2. 1: Schéma de décomposition du domaine de l’intelligence artificielle et de ces sous-domaines.....	24
Figure 2. 2: Topologie de réseau de neurones avec une seule sortie. ....	26
Figure 2. 3: Réseau de neurones avec de nombreuses couches convolutives.....	28
Figure 2. 4: Exemple explicative sur l’opération de convolution.....	29
Figure 2. 5: (a) pooling maximal, (b) pooling moyen.....	29
Figure 2. 6: Couche fully-connected.....	30
Figure 2. 7: Architecture de RNN.....	31
Figure 2. 8: Architecture de Long-Short Term Memory Unit .....	31
Figure 2. 9: Gated Reccurent Unit .....	33
Figure 2. 10 : L’architecture des réseaux siamois .....	34

## Liste des figures

### Chapitre 03 : Détection acoustique de drone

Figure 3. 1 : Liste des sons collectée. ....	40
Figure 3. 2: Résultats de visualisation des sons de drones dans le domaine temporel.	41
Figure 3. 3 : Résultats visualisation des sons de Bruit dans le domaine temporel (stéréo). ....	42
Figure 3. 4: Résultats de visualisation des sons de drones dans le domaine temporel (mono). ....	43
Figure 3. 5: Résultats de visualisation des sons de bruit dans le domaine temporel (mono). ....	43
Figure 3. 6: Résultats de visualisation des sons de drone dans le domaine fréquentielle. ....	45
Figure 3. 7: Résultats de visualisation des sons de bruit dans le domaine fréquentielle. ....	46
Figure 3. 8: Résultats de visualisation des sons des drones dans le domaine spectrale ....	48
Figure 3. 9: Résultats de visualisation des sons des bruits dans le domaine spectrale.	48
Figure 3. 10: L'architecture des réseaux siamois.....	49
Figure 3. 11: Résultats de visualisation des images paires positive. ....	50
Figure 3. 12: Résultats de visualisation des images paires négative.....	51
Figure 3. 13: graphe de Accuracy et Loss en fonction d'epochs .....	53
Figure 3. 14: graphe de Accuracy et Loss en fonction d'epochs .....	53
Figure 3. 15 : graphe de Accuracy et Loss en fonction d'epochs.....	60

## Liste des Tableaux :

Chapitre 01 : Généralité sur les drones

Tableau 1.1 : Calcification de UAV.....[5]

Tableau 1.2 : Les avantages et les inconvénients des différents systèmes de détection.....[22]

Chapitre 03 : Détection acoustique de drone

Tableau 3.1 : Modèle de CNN .....[5]

## Abréviation /Acronymes

CNN	Convolution Neural Network.
DFT	Discrete Fourier Transform.
ESC	Electronic Speed Controller.
EO/IR	Electro-optique Infrarouge.
ESC	Electronic Speed Controller.
EUROUVS	European Organization of Unmanned Aerial Vehicles.
FFT	Fast Fourier Transform.
IA	Intelligence Artificiels.
LSTM	Long Short Term Memory
RF	Radio Frequency.
RNN	Recurrent Neural Network.
SDDD	Système de Défense Des Drones.
UAV	Unmanned Aerial Vehicles (Drone).
UVS	Unmanned Vehicles Systems.
2D	Deux Dimension.

# **Introduction**

## **Générale**

## **Introduction générale :**

Actuellement, l'innovation rapide de la technologie des drones a conduit à une augmentation significative de leur utilisation dans de nombreux domaines tels que l'agriculture, la surveillance, la photographie aérienne. Cependant, due à cette popularité croissante les drones peuvent tout de même présenter des risques potentiels sur la sécurité générale et la vie privée. Les drones peuvent être utilisés de manière illégale pour espionner des individus ou bien pour le transport de substances frauduleuses.

Dans ce cadre, la détection des drones est devenue une préoccupation majeure pour de nombreux secteurs. Pour les acteurs de ces domaines il devient impératif de développer des systèmes de détection avancée capable de détecter et de poursuivre les drones non autorisés. Les chercheurs ont déjà créé des procédés d'identification tels que la détection optique, détection par Radar et par de radio fréquence. Chaque une de ces méthodes a ses points positifs et ses limitations.

Dans notre projet, nous allons développer un détecteur acoustique de drone. La détection acoustique fait référence à la capacité d'identifier et de localiser des signaux sonores. Elle est basée sur l'analyse des signaux sonores capté par des microphones ou des capteurs acoustique afin d'extraire des informations sur les sources sonores. Les drones produisent un son nome bourdonnement. Ce dernier sera capté et traités afin de détecter le drone à partir de son bourdonnement.

Donc il est possible de reconnaître ses signatures sonores spécifiques et de les distinguer des autres sources de bruit ambiant. Ce système peut être utilisé dans de vastes domaines, tel que la lutte contre les utilisations illégales des drones, notamment la sécurité et la vie privée. D'où la nécessité d'utiliser ce système dans les endroits fermés comme les banques, les bureaux de poste et les cellules de prisons, mais aussi dans les bureaux accoises à des postes sensibles à l'image de bureau de ministre, chefs de l'armée et la présidence des états.

Pour rester en parallèle avec le progrès technologique qui se passe dans le monde entier. On a intégrés un modèle d'intelligence artificielle dans notre travail. L'IA se réfère à la capacité des machines à effectuer des taches qui nécessitent une intelligence similaire à celle humaine, telles que la prise de décision et la reconnaissance vocale. Grâce à l'utilisation des techniques telles que l'apprentissage automatique (Machine Learning) et le de réseaux neurones. L'IA utilise pour des algorithmes capables d'analyser et de reconnaître l'empreinte sonore des UAV. Nous avons cherché à améliorer la précision de la détection acoustique en utilisant les réseaux de neurones. Afin de classer les enregistrements sonores de l'ambiance sonore dans un endroit fermé.

Ce mémoire est divisé en trois(03) chapitres. Nous avons essayé de collecter le maximum d'information sur notre thème.

- Le premier chapitre a traité le domaine des drones d'une façon générale. Historique des drones, leurs caractéristiques, les différents types. Afin de débiter le terrain pour la réalisation de notre travail.
- Dans le deuxième chapitre nous avons abordé le thème de l'intelligence artificielle spécifiquement les réseaux de neurones. On a aussi pu expliquer le fonctionnement des différentes architectures dont RNN, CNN et les Siamois. En plus de déclarer ses avantages et ses points faibles.
- Dans le dernier chapitre on a étudié les caractéristiques des sons dans le domaine temporelle, fréquentielle et spectrale. C'est sur cette dernière modalité qu'on a basé notre travail pour classifier les différents sons (drones ou bruits) à l'aide de réseaux de neurones Siamois et Convolutifs. Tout ce travail est réalisé par l'utilisation du langage de programmation Python.

**Chapitre 01 :**  
**Généralité sur les**  
**drones**

## 1.1 Introduction

Les drones ou UAV (Unmanned Aerial Vehicles) sont des véhicules aériens sans pilote capables d'effectuer des tâches autonomes ou semi-autonomes. Leur utilisation principale est les missions militaires de reconnaissance ou de surveillance sans risque de pertes.

En effet, ils sont idéaux pour les missions qui pourraient mettre en danger l'équipage, ou qui nécessitent des séjours prolongés dans des zones fastidieuses pour l'équipage à bord. Leur utilisation commence par tout ce qui concerne l'observation et s'étend ensuite à l'acquisition d'objectifs ainsi qu'à la guerre électronique et à la destruction d'objectifs.

Les applications civiles émergentes dans un flux sans fin, telles que la surveillance du trafic routier, la prévention des incendies de forêt, la collecte de données météorologiques et même l'inspection des ouvrages d'art. Les drones peuvent varier en taille de quelques centimètres à quelques mètres, et leurs méthodes de propulsion varieront selon les besoins. On distingue deux types de voilures : les voilures fixes pour le vol en mode d'avance et les voilures tournantes pour le vol stationnaire [1].

## 1.2 Historique

Les drones dans l'histoire, ont vu leur intérêt augmenter au courant de la seconde guerre mondiale au vu de la nécessité d'avoir des appareils sans pilotes capables de réaliser des missions de reconnaissances et d'espionnage.[2]



**Figure 1. 1** : les drones dans l'histoire. [2]

## Chapitre 01: Généralité sur les drones

---

Les premiers drones apparurent en France dans les années 1960, tel le R 20 Nord-Aviation, dérivé de l'avion cible CT 20. Mais les exemples significatifs d'une utilisation opérationnelle des drones sont encore peu nombreux.

Pendant la guerre du Vietnam, les Américains ont utilisé des drones (Firebee) pour localiser la rampe de lancement du missile sol-air soviétique "SAM-2": 3500, plus tard, pendant la guerre du Golfe de 1991, ils ont fait appel au drone (Pioneer) pour la surveillance jour/nuit, acquisition de cible et paramètres d'artillerie. Dans le même conflit, la Grande-Bretagne et la France commencent à servir des drones.

De leur côté, les Israéliens ont saturé les systèmes de défense aérienne le long du canal suez pendant la guerre du Yom Kippour (1973) avait beaucoup de bons drones marché. Plus tard, ils ont détecté et leurré par le même moyen les batteries syriennes anti-aériennes.

En général, les experts conviennent que les drones peuvent vraiment démontrer leurs capacités d'observation aérienne de combat (renseignement), concernant les trois théâtres les plus proches de l'ancien conflit Yougoslavie, Irak et Afghanistan.

C'est encore le drone (Hunter), qui a successivement assuré la Réunion du G8 d'Evian 2003 et célébration du 60e anniversaire du débarquement allié en Normandie en 2004.

Mais les drones sont encore peu connus, et donc préoccupants, un demande de test de surveillance du trafic routier sur la rocade grande ville française rejetée par un minuscule drone (environ 6 kg) année 2004 par les autorités pour raisons de sécurité.

### 1.3 Classification des drones

Les drones peuvent être divisés en différentes catégories selon leur taille, leur capacité de charge utile, leur autonomie et leur objectif, une classification intéressante a été proposée par l'Association européenne des systèmes de véhicules sans pilote (EUROUVS), devenue plus tard l'International Unmanned Vehicle Systems (UVS International).

#### 1.3.1 Classification selon P.A.E

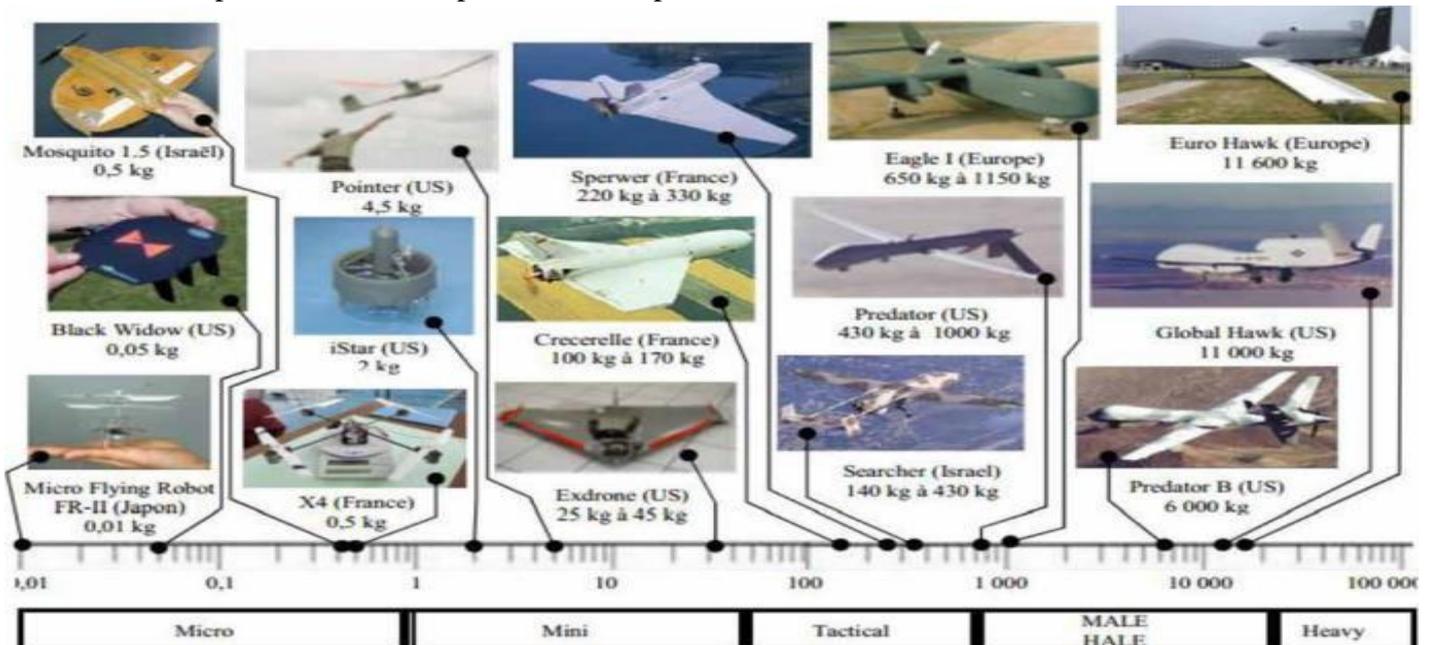
La classification est basée sur différents paramètres tels que la masse maximale au décollage, l'altitude maximale et le temps de vol maximal. Dans le tableau 1.1, nous fournissons un résumé complet des quatre principales catégories de drones et de leurs acronymes. [3]

## Chapitre 01: Généralité sur les drones

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	<i>Hunter B</i>
	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>

**Tableau 1.1** : calculcation des UAVs. [3]

Nous pouvons le définir par le schéma spectral suivant :



**Figure 1. 2** : Spectre de masse des différents drones selon le classement américain (en Kg) [4].

### 1.3.2 Classification selon la configuration aérodynamique

Une autre possibilité de classement est permise par la configuration aérodynamique. Les drones peuvent être divisés en trois catégories :

#### a) Aile battante :

Le drone est guidé et piloté en battant des ailes. Ils imitent les insectes et peuvent planer à basse vitesse et effectuer des tâches dans des espaces très restreints. [5]



**Figure 1. 3:** Drone à aile battante. [5]

#### b) Drone à voilure fixe

Ce genre de drone est constitué d'une paire d'ailes, un fuselage, un empennage et un ensemble d'hélices. Les configurations de ce genre donnent plus d'autonomie au drone.



**Figure 1. 4:** Drone à voilure fixe. . [5]

#### c) Drones à voilure tournante

Ce type de drone a des capacités de décollage et d'atterrissage verticaux, utilise des rotors ou des multi-rotors pour planer et voler, n'a pas besoin de piste de décollage et d'atterrissage et peut atteindre une très grande diversité. Mission impossible par rapport aux drones à voilure fixe. [5]

Nous pouvons alors classer les drones à voilures tournantes en quatre catégories :

- Les mono-rotors.
- Les birotors.
- Les tri-rotors.
- les rotors multiples.

## Chapitre 01: Généralité sur les drones

### ➤ Les mono-rotors

Les mono-rotors se composent d'un rotor principal qui permet le décollage et l'atterrissage. Ils se divisent en trois sous-catégories selon la configuration utilisée pour permettre au drone de se déplacer dans ses différents axes (en translation et en rotation), ainsi que pour l'empêcher de tourner sur lui-même sous l'action du couple de réaction du rotor principal.



Figure 1. 5: Drones mono-rotors. [5]

### ➤ Les birotors

Les birotors contrarotatifs se composent de deux rotors tournants en sens opposé et à la même vitesse de rotation. Ils sont divisés en 3 sous catégories selon la configuration utilisée pour permettre au drone de se déplacer suivant ses différents axes.

- Les birotors à pas cyclique et collectif : ils utilisent la variation cyclique et collective du pas pour se positionner.
- les birotors augmentés d'une ou plusieurs hélices.
- les birotors utilisant des volets.



Figure 1. 6: Les différents types de drone birotors. [5]

## ➤ Les tri-rotos

Ils sont constitués de deux rotors à l'avant qui tournent dans les sens opposés pour modifier le tangage et d'un rotor en arrière pour régler le roulis. Le fonctionnement est similaire à celui d'un quadricoptère mais la performance en vol n'est pas aussi satisfaisante. [5]



**Figure 1. 7:** Drone tri-rotor. [5]

## ➤ Les rotors multiples

Ces robots volants sont généralement équipés de quatre rotors, mais ils existent avec six ou même huit rotors



**Figure 1. 8:** Exemples des drones à rotors multiples. [5]

## 1.4 Architecture et Composants (Partie Mécanique & Electronique)

Dans la présente partie, nous allons parler des différents composants qui entrent dans la réalisation d'un quadricoptère.

### 1.4.1 Composants

La fabrication d'un quadricoptère 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,

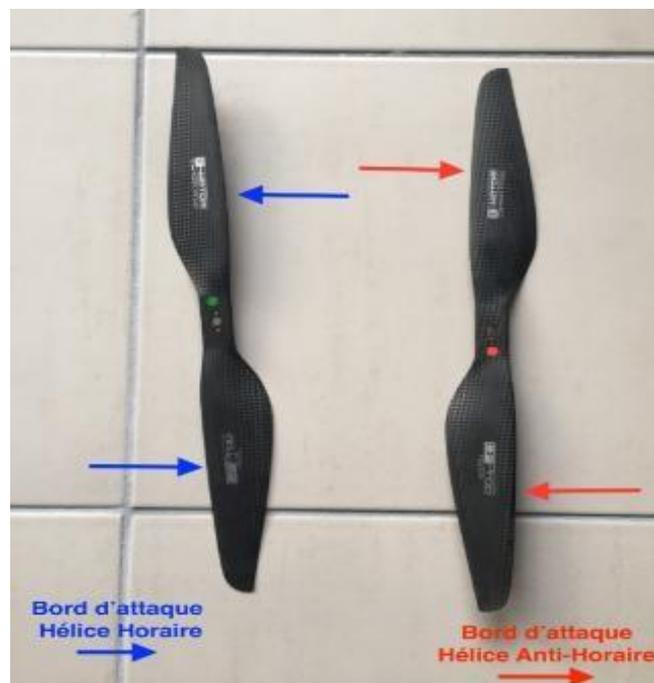
## 1.4.2 Hélices

Un quadricoptère est constitué de deux hélices standard (sens horaire) et deux hélices pusher (sens anti horaire) rotation contraire [6].



**Figure 1. 9:** Exemples d'hélices pour drone. [6]

La désignation des hélices est une double valeur du type 8\*5. La première valeur indique la longueur de l'hélice en pouces (inches en anglais qui correspond à 2.54 cm). Le pas d'une hélice est défini comme étant la distance théorique d'avancement de l'hélice en un tour complet. Plus l'hélice est grande, plus elle va apporter de portance et de stabilité. Une grande hélice sera moins agile, vive et réactive. La seconde valeur indique le « pas de l'hélice » 4 – 4,5 – 5 – 5,5 -...etc. Plus le pas est important, plus l'hélice va brasser d'air en 1 tour, et plus elle va nécessiter d'énergie.



**Figure 1. 10:** Les deux types d'hélices pour drone. [6]

## 1.4.3 Moteurs électriques

Quatre moteurs sans balai (brushless) sont nécessaires pour le quadrirotor. Il en existe de différentes tailles et puissances. Ils sont caractérisés par le diamètre de leur cage tournante et par le nombre de *tours/volt* ou *KV*. Le *KV* est la division du nombre de tours par minute par volt, par exemple un moteur ayant un *KV* de 1000 *tours/V* fonctionnera à 12000 *tours/min* s'il est alimenté en 12 V. Sur les moteurs sans balai utilisés en modélisme, les bobinages en cuivre sont montés sur le stator et les aimants sur le rotor, à l'inverse des moteurs électriques conventionnels [6].



**Figure 1. 11:** Exemple d'un moteur Brushless. [6]

## 1.4.4 Contrôleurs de vitesse (ESC)

Les ESCs (Electronics Speed Controller) convertissent les DC en AC pour alimenter les moteurs synchrones sans balai (brushless). Chaque moteur a besoin d'un ESC.



**Figure 1. 12:** Electronic Speed Controller (ESC). [6]

## 1.4.5 Contrôleur de vol

C'est une carte électronique ou plutôt un ordinateur embarqué, équipé de capteurs très précis (ses entrées), qui va traiter les consignes du pilote envoyées à l'émetteur ainsi que les informations envoyées par ses capteurs et va transmettre à travers ses sorties des impulsions électriques aux contrôleurs des moteurs pour faire varier leur vitesse. Ces cartes sont équipées de gyroscopes et d'accéléromètres pour mesurer et compenser les déplacements. Certaines cartes évoluées sont équipées de *GPS* et d'altimètre afin de maintenir un point fixe ou une altitude ou même de retourner au point de départ du drone en cas de perte de signal radio [6]. Il existe différents types de contrôleur de vol avec des spécificités et des options différentes. En effet, Certains contrôleurs de vol peuvent être équipés de diverses options comme par exemple

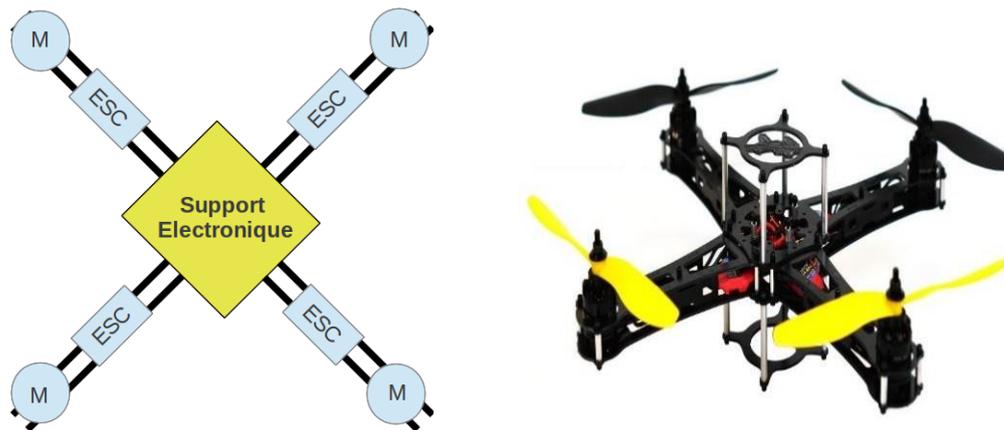
- gyroscope,
- accéléromètre,
- navigation,
- baromètre,
- GPS,
- sonar
- OSD pour On Screen Display (affichage sur l'écran)
- bluetooth
- etc.

D'autres peuvent avoir des logiciels (nécessaires à la configuration et à la programmation) en "open source" ou bien en "closed source" :

- **Exemples de contrôleur de vol open source**
  - APM
  - PX4
  - Rushduino FC V1.1 USB version
  - Flip32 Acro Controller
  - ...etc
- **Exemples de contrôleur de vol closed source**
  - NAZA MV2
  - NAZA light
  - NAZE 32
  - ...etc

## 1.4.6 Le châssis

Sur les quadricoptères, le châssis comporte quatre bras qui forment une croix. Suivant les règles de la construction mécanique. Il existe plusieurs configurations possibles : en « X », en « + » et en « H ».



**Figure 1. 13:** Types de châssis (Configuration en « X » et en configuration en « + »). [6]

Le châssis va déterminer l'envergure totale du quadrirotor. Le type du châssis utilisé affectera les caractéristiques de l'ensemble des composants (type et taille, c'est-à-dire poids des hélices, des moteurs, de la batterie, des fixations...).

### 1.4.7 Les Batteries

Les batteries utilisées sur un quadricoptère sont essentiellement des « Lithium Polymère » (Li-Po). Elles sont issues d'une technologie qui permet d'avoir un très bon rapport *poids/puissance*. Un élément Li-Po (1S) fournit une tension de 3,7v. Sur un drone, on utilise en général des batteries à 3 ou 4 éléments (3S ou 4S). L'intensité est aussi un critère de choix. Un chargeur spécifique est à prévoir ainsi que des règles de sécurité très strictes car ces batteries peuvent exploser en cas de mauvaise manipulation. A noter que ces batteries doivent être stockées, en cas de non utilisation, à des températures relativement basses de l'ordre de 5°C [7].



**Figure 1. 14:** Batterie LiPo. [7]

### 1.4.8 La Caméra embarquée :

Généralement ce sont des caméras à faible résolution ou haute résolution et qui envoient des images à des stations terrestres via des ondes radio. [7]



Figure 1. 15: Exemple de caméra à embarquer sur quadricoptère. [7]

### 1.4.9 La Radio-commande (RC)

Pour piloter le quadricoptère, il faut un émetteur radio pour le pilote et un récepteur dans le drone. Il existe plusieurs technologies pour les radio-commandes : les radios FM en 41MHz de moins en moins utilisées et les radios en 2,4 GHz, souvent programmables pour s'adapter à chaque appareil radio-commandé. Une radio-commande doit idéalement comporter 6 voies pour piloter un quadricoptère. Ces 6 voies se répartissent comme suit : 4 voies sur les manettes et 2 voies sur des interrupteurs pour actionner certaines fonctions. Il existe deux modes de configuration des manettes, le premier mode dans lequel les gaz sont à droite et le deuxième mode où les gaz sont à gauche. Certaines radios fonctionnent dans les deux sens, c'est-à-dire qu'elles peuvent envoyer des ordres à l'émetteur mais aussi recevoir des informations. Sur la plus part des récepteurs il est possible de choisir la répartition des commandes (lacet, roulis et tangage) sur les sticks de droite et de gauche. Il existe en tout 4 modes, les plus utilisés sont le 1 et le 2. Pour le pilotage de quadricoptère, le mode 2 est le plus adapté car toute l'assiette est contrôlée par une même manche (ce qui se rapproche le plus d'un vrai hélico). [7]



Figure 1. 16: Principe de fonctionnement d'une radiocommande. [7]

### 1.5 Domaine d'application

L'objectif principal des drones est de remplir une mission qui pourrait être de nature militaire, scientifique, économique ou même commerciale.

L'intérêt pour le contrôle et la navigation des drones est dû à leur utilisation dans des environnements dangereux. [8]

#### 1.5.1 Applications militaires (marine, armée et force aérienne)

- Intelligence électronique,
- Brouillage et destruction du système radar,
- Relais de signaux radio,
- Observation des flottes ennemies,
- Désignation et suivi des cibles,
- Élimination des bombes qui non pas explosées,
- Leurre de missiles par émission de signatures artificielles.

#### 1.5.2 Applications civil

- Topographie aérienne pour les recherches géographiques,
- Pulvérisation et surveillance agricoles,
- Chercher et sauver,
- Lutte contre l'incendie et détection d'incendie forestier,
- Surveillance des importations illégales,
- Études de pollution et surveillance des terres,
- Inspection des pipelines et des lignes électriques,
- Recherche de pétrole et de gaz,
- Livraison de colis,
- Détection de véhicules mobiles au sol.

#### 1.5.3 Les utilisations négatives des drones

Malheureusement, les drones peuvent également avoir des utilisations négatives et poser des risques potentiels pour la vie privée, la sécurité et la sécurité publique:

**-Espionnage et surveillance illégale:**

Les drones peuvent être utilisés pour espionner ou surveiller illégalement des personnes, des entreprises ou des gouvernements. Cela peut inclure la collecte de données personnelles ou sensibles, la surveillance de mouvements ou de comportements, ou la collecte d'informations stratégiques.

### - **Violation de la vie privée:**

Les drones peuvent être utilisés pour filmer ou photographier des personnes dans leur vie privée, par exemple, dans leur jardin ou sur leur balcon, ce qui peut constituer une violation de la vie privée.

### - **Attaques terroristes:**

Les drones peuvent être utilisés pour mener des attaques terroristes en larguant des explosifs ou en créant des perturbations dans les transports publics ou les infrastructures.

### - **Attaques de piratage informatique:**

Les drones peuvent être utilisés pour mener des attaques de piratage informatique, par exemple, en piratant des réseaux Wi-Fi ou des systèmes de sécurité.

### - **Pollution sonore:**

Les drones peuvent être très bruyants et causer une pollution sonore, ce qui peut causer des nuisances pour les personnes vivant ou travaillant à proximité.

### -**Collision:**

Les drones peuvent causer des accidents s'ils entrent en collision avec d'autres drones, des avions, des hélicoptères ou des objets sur le sol.

### -**Utilisation illégale de drogues et la livraison:**

Les drones peuvent être utilisés pour transporter illégalement des drogues ou d'autres produits dangereux, contournant les contrôles de sécurité et les lois. Il est donc important que les gouvernements, les entreprises et les particuliers prennent des mesures pour réglementer et contrôler l'utilisation des drones, afin de minimiser les risques potentiels pour la vie privée, la sécurité et la sécurité publique.

## 1.6 Les systèmes de détection

Au cours des dernières années, la mise en œuvre des drones est en forte augmentation, ce qui en fait une arme très dangereuse entre les mains de l'ennemi. La détection fiable des drones est très difficile et exige une tâche et la grande majorité des solutions analysées dans ce document sont réalisées au cours des dernières années.

Il existe plusieurs techniques de détection de drones. Chacun d'eux a ses avantages et ses inconvénients, c'est-à-dire des situations où ils sont adaptés à la mise en œuvre et où leurs résultats de détection ne sont pas satisfaisants. C'est pourquoi habituellement plusieurs techniques différentes sont mises en œuvre ensemble.

Les techniques les plus utilisées sont le radar, la radiofréquence (RF), la détection acoustique, optique et thermique (infrarouge).

### 1.6.1 Détection radar

Les systèmes radar sont largement utilisés dans la vie quotidienne et dans les applications militaires. Ces systèmes radar conventionnels sont ajustés pour détecter des objets relativement grands. Ils ne conviennent pas à la détection de petits drones, surtout lorsque la vitesse du drone est faible, lorsqu'ils volent à basse altitude. La solution radar pour les systèmes de défense des drones (SDDD) représente une méthode active d'identification et localiser une menace potentielle d'UAV. Afin de déterminer la portée, l'angle ou la vitesse d'un UAV.



**Figure 1. 17 :** Système de détection de drones avec Radar [9]

### 1.6.2 Détection Radio Fréquence:

La détection UAV RF est une technique qui implique l'interception et l'analyse des signaux transmis (Tx, Rx) entre l'UAV et la station au sol. Habituellement, ces signaux consistent en des signaux de contrôle en liaison montante (depuis la station au sol) et en liaison descendante (depuis le drone) signaux de données (données de position et vidéo) [10].

De plus, le déploiement de techniques s'appuyant sur les machines pour ce type de données est nouveau et la littérature ne dispose pas d'un ensemble de données public complet pour les signaux RF qui pourraient être utilisés pour la validation et la comparaison. Donc toutes les méthodes existantes ont des performances limitées pour de faibles rapports signal/bruit.



**Figure 1. 18:** Système de détection de drones Radio Fréquence [11].

### 1.6.3 Détection optique

Cette technologie implique l'utilisation de caméras qui prennent des images d'une zone désignée afin de déterminer la présence d'un drone cible.

#### 1.6.3.1 Caméras électro-optiques (EO)

Certains SDDD utilisent des capteurs d'imagerie (EO/IR), qui pourraient être dirigés par d'autres capteurs (tels que comme radar et RF) afin d'obtenir des images du drone et de ses principales caractéristiques (par exemple : charge utile).

Ces images peuvent être enregistrées et analysées par des spécialistes afin de déterminer le niveau de menace [12].le principal inconvénient de cette méthode est sa faible performance dans l'obscurité et le brouillard conditions. En outre, la qualité des images dépend de la qualité des objectifs et l'angle de la photographie (LoS obligatoire).



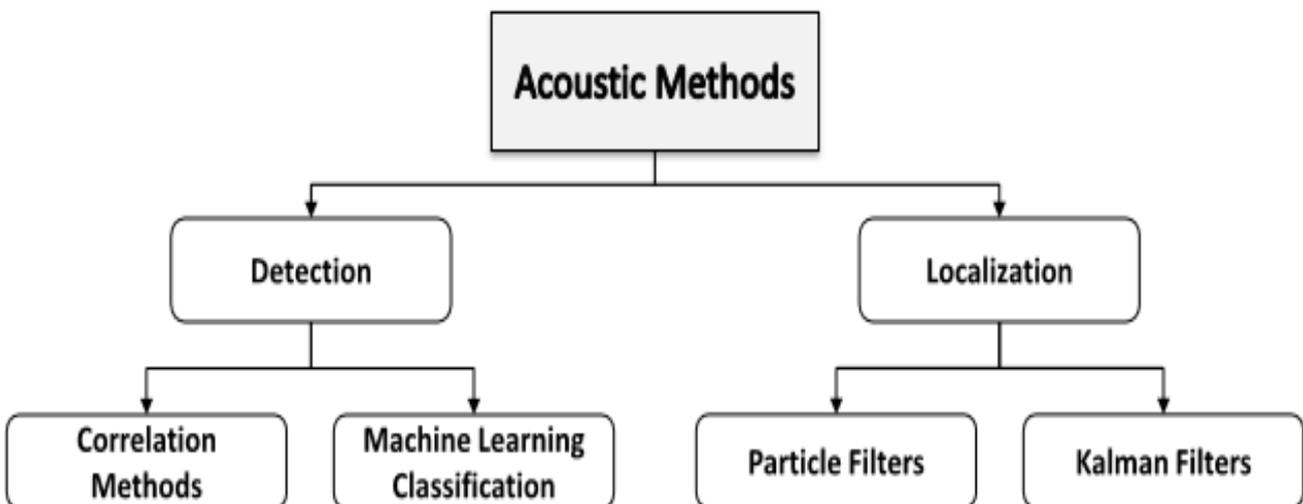
**Figure 1. 19:** Système de détection de drones par Caméras électro-optiques (EO) [14].

### 1.6.3.2 Caméras infrarouges/thermiques

Cette méthode utilise des caméras IR thermiques capables de détecter la chaleur produite par les composants matériels d'un UAV, tels que les moteurs, les batteries et les processeurs.

Cette méthode de détection présente des inconvénients liés à la plage de détection et à la sensibilité des capteurs qui mesurent la différence thermique entre le drone et l'arrière-plan. En conséquence, la détection de la présence de drones dépend sur la température du moteur du drone, l'angle (LoS est obligatoire), la distance et la température des capteurs IR [13].

### 1.6.4 Détection acoustique



**Figure 1. 20:** Schéma synoptique de la méthode de détection acoustique [15].

Un drone volant produit un bruit de ronflement qui peut être capturé par des capteurs acoustiques (des microphones) et analysé en utilisant différentes méthodes pour identifier l'empreinte digitale audiophone spécifique.

Le résultat idéal serait de déterminer le type de drone ou même le drone individuel par son empreinte audio. En général, la détection acoustique des drones repose soit sur des méthodes de corrélation/autocorrélation ou sur la classification d'apprentissage automatique. [15]

## Chapitre 01: Généralité sur les drones

---

### 1.7 Les avantages et les inconvénients des systèmes de détection :

Système de détection	Les avantages	Les inconvénients
<b>Détection Acoustique</b>	<ul style="list-style-type: none"> <li>• Couvre le spectre de 20 Hz à 20 kHz.</li> <li>• Bibliothèque de signatures acoustiques mise à jour facilement d'un vol à l'autre.</li> <li>• Léger et facile à utiliser</li> <li>• associés à d'autres types de capteurs.</li> </ul>	<ul style="list-style-type: none"> <li>• Portée limitée.</li> <li>• sont vulnérable au bruit ambiant.</li> <li>• Sensible aux leurres.</li> <li>• Nécessite une base de données de signature acoustique pour différents drones pour la formation et les tests.</li> </ul>
<b>Imagerie optique</b>	<ul style="list-style-type: none"> <li>• Couvre tout le visible et l'IR spectre (3 MHz à 300 GHz).</li> <li>• Les caméras infrarouges pourraient fonctionner dans des conditions nuageuses, la météo et le jour ou la nuit.</li> <li>• Pourrait être aidé par technologies de l'informatique visuel.</li> </ul>	<ul style="list-style-type: none"> <li>• Fournit des images 2D</li> <li>• Performances limitées en fonction des conditions météorologiques conditions et contexte Température.</li> <li>• Dépend des données de géo référence.</li> </ul>
<b>Détection Radar</b>	<ul style="list-style-type: none"> <li>• Largeur de bande utilisée : 3 MHz à 300 GHz.</li> <li>• Peut fonctionner par tous les temps et les conditions de jour et de nuit.</li> <li>• Offre des renseignements sur vitesse de la cible.</li> <li>• Peut reconnaître le micro-doppler signatures (MDS)</li> </ul>	<ul style="list-style-type: none"> <li>• Grande section radar souhaitée.</li> <li>• Difficile de différencier les UAV oiseaux.</li> <li>• Performance limitée pour faible altitudes et vitesses (cône de mort).</li> <li>• Pourrait facilement nuire aux petites les objets, en particulier les oiseaux.</li> </ul>

## Chapitre 01: Généralité sur les drones

---

	<ul style="list-style-type: none"> <li>• offre une couverture élevée.</li> <li>• Bonne précision.</li> <li>• Compact et mobile, nécessaire pour les applications tactiques.</li> <li>• Grande fiabilité.</li> </ul>	<ul style="list-style-type: none"> <li>• Une marge de crédit est requise.</li> <li>• Coût élevé.</li> </ul>
<b>Détection RF</b>	<ul style="list-style-type: none"> <li>• Saisir la communication spectre et signaux UAV et opérateurs.</li> <li>• Faible complexité et facile à mettre en œuvre.</li> <li>• Peut fonctionner par tous les temps et les conditions de jour et de nuit.</li> <li>• Amélioration plus facile grâce à la modularité mise en œuvre des récepteurs et unités de traitement du signal numérique utilisées dans la mise en œuvre.</li> </ul>	<ul style="list-style-type: none"> <li>• Sont vulnérables aux actes malveillants ou illégaux RF modifiée qui dépassera capacités de réception.</li> <li>• Utilisation difficile dans les zones urbaines</li> </ul>

**Tableau.1.2** : Les avantages et les inconvénients des différents systèmes de détection. [16]

### 1.8 Conclusion

Dans ce chapitre nous avons découvert que Les drones est un domaine très vaste se base sur la discipline de la robotique, et aussi nous avons présenté en détailles les composants d'un drone, ses caractéristiques techniques et son domaine d'application,

Les drones ont des applications très bénéfiques pour humanité mais peuvent être utilisées pour des fins néfastes telque le transport illégal, espionnage, attaques terroristes,..... .Ce qui rend la présence de drones indésirable et doit être contrôlée, donc dans ce chapitre, nous avons abordé les façons les plus importantes de capturer et détecter les drones qui sont utilisés dans plusieurs zones illégales .Nous avons également mis l'accent sur les avantages et les inconvénients les plus Importants de chaque technique.

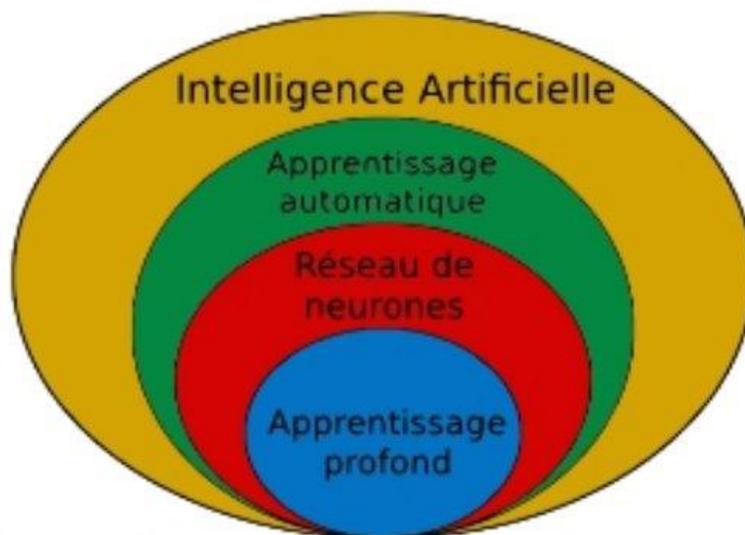
# **Chapitre 02 :**

# **Les Réseaux De**

# **Neurone Artificiels**

### 2.1 Introduction

Le domaine de l'intelligence artificielle est divisé en plusieurs sous-domaines imbriqués (comme le montre la figure 2.1). L'apprentissage automatique est un sous-domaine de l'intelligence artificielle qui consiste à apprendre à répondre à un problème donné soit par l'expérience, soit par une base de données de règles implicites. Ce domaine est spécialisé dans l'analyse statistique des données de formation. Dans le cadre de l'apprentissage automatique, nous verrons qu'il existe de nombreux algorithmes qui utilisent divers modèles mathématiques. Les réseaux de neurones sont l'un de ces modèles, certainement le plus répandu et celui utilisé dans les domaines les plus divers. L'apprentissage en profondeur est un ensemble de techniques qui utilisent des réseaux de neurones pour résoudre des problèmes complexes. Ces techniques sont largement utilisées, notamment dans les domaines du traitement d'images, du traitement de séries temporelles (reconnaissance de la parole, etc.).[17]



**Figure 2. 1:** Schéma de décomposition du domaine de l'intelligence artificielle et de ces sous-domaines. [17]

### 2.2 L'apprentissage en profondeur (Deep learning)

L'apprentissage en profondeur est un sous-ensemble de l'apprentissage automatique dans lequel un modèle (généralement un réseau neuronal artificiel) utilise plusieurs couches pour créer des représentations plus riches des données.

Alors que les algorithmes d'apprentissage automatique traditionnels tels que les machines à vecteurs de support nécessitent des fonctionnalités artisanales pour obtenir les meilleurs résultats, les algorithmes d'apprentissage en profondeur sont capables de créer leurs propres fonctionnalités à partir de données plus brutes basées sur un contenu pertinent appris pendant la formation.

## Chapitre 02 : Les Réseaux De Neurone Artificiels

---

L'apprentissage en profondeur permet aux modèles informatiques constitués de plusieurs couches de traitement d'apprendre des représentations de données avec plusieurs niveaux d'abstraction.

Ces méthodes ont considérablement fait progresser l'état de l'art en matière de reconnaissance vocale, de reconnaissance visuelle d'objets, de détection d'objets et de nombreux autres domaines tels que la découverte de médicaments et la génomique. L'apprentissage en profondeur révèle des structures complexes dans de grands ensembles de données en utilisant l'algorithme de rétro propagation pour indiquer à la machine comment modifier ses paramètres internes, qui sont utilisés pour calculer la représentation de chaque couche en fonction de la représentation de la couche précédente [18].

### 2.3 Réseaux de neurones et son principe de fonctionnement

Avec l'augmentation rapide de la puissance de calcul des ordinateurs et le développement des techniques d'apprentissage, l'utilisation des réseaux de neurones est en plein essor, notamment dans les domaines du traitement d'images, de la traduction automatique et de l'informatique, du traitement de la parole, etc. Deux types de réseaux de neurones ont marqué ces dernières années des ruptures technologiques dans le traitement de la parole : les réseaux dits « profonds » et les réseaux récurrents.

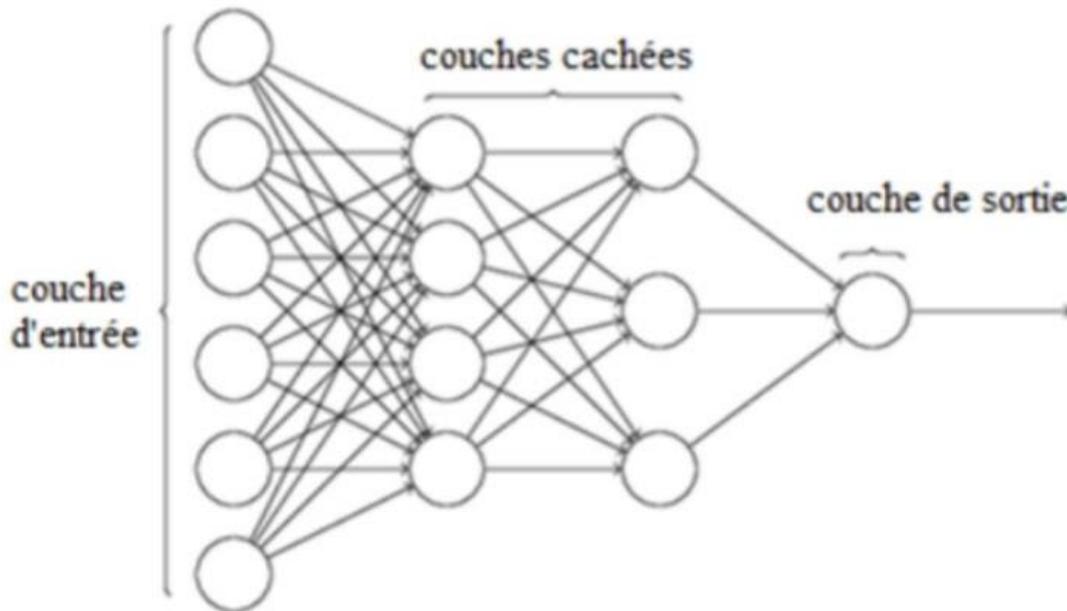
Dans cette section, nous détaillons le fonctionnement du réseau de neurones et ce que nous avons fait avec le modèle lui-même et la méthode d'apprentissage. [19]

Le réseau comporte 3 composants : couche d'entrée couche cachée et couche de sortie. Le terme « profond » se rapporte généralement au nombre de couches cachées du réseau de neurones.

Les réseaux de neurones classiques ne comportent que 2 à 3 couches cachées, tandis que les réseaux profonds peuvent en compter jusqu'à 150. [20]

L'idée est d'utiliser la structure de couche de réseau neuronal en empilant plusieurs couches les unes sur les autres, de manière à faciliter le mécanisme de décomposition. Par conséquent, chaque couche d'un réseau de neurones profonds (Deep Neural Networks DNN) fonctionne comme une seule transformation pour extraire davantage les données [21]

Le réseau de neurones le plus connu et le plus simple à comprendre est le réseau de neurones multicouches à anticipation. Il contient un calque d'entrée, un ou plusieurs calques masqués et un seul calque de sortie. Chaque couche peut avoir un nombre différent de neurones et chaque couche est entièrement connectée à la couche adjacente



**Figure 2. 2:** Topologie de réseau de neurones avec une seule sortie. [21]

Un réseau de neurones est défini comme un ensemble de nœuds (appelés neurones) connectés via des liaisons dirigées (flèche), chaque flèche représente une connexion entre la sortie d'un neurone et l'entrée d'un autre (les flèches entrantes étant les entrées du neurone et les flèches sortantes étant les sorties du neurone), Chaque flèche porte un poids, reflétant son importance, chaque nœud étant une unité de traitement qui exécute une fonction de nœud statique sur son signal entrant pour générer une sortie de nœud unique [22]

### 2.4 L'apprentissage :

En effet, les réseaux de neurones peuvent trouver le lien qui unit des valeurs de sortie à celles en entrée, et ce, même lorsqu'on ne connaît pas cette fonction a priori. Afin de parvenir à ce résultat, il faut « entraîner » le modèle, à l'aide d'un jeu de données initial. Celui-ci est alors divisé en deux parties : une pour l'apprentissage, l'autre pour tester le réseau. Dans un premier temps, on va donc soumettre des données d'entraînement au programme, qui comprennent des valeurs d'entrée, ainsi que les valeurs de sortie attendues. Au début, le réseau de neurones va tenter de calculer les résultats, mais avec peu d'informations, et va donc commettre des erreurs. On va ensuite ajuster ses paramètres, de sorte à réduire ces écarts à chaque itération. Par exemple, un réseau de neurones peut être utilisé pour prévoir les risques d'apparition d'une maladie chez certains individus. En lui soumettant les caractéristiques de patients, le modèle va d'abord effectuer des prédictions aléatoires. Puis, en apprenant de ses erreurs, il sera de plus en plus pertinent dans son analyse. [23]

## Chapitre 02 : Les Réseaux De Neurone Artificiels

---

### 2.4.1 Apprentissage supervisé

La plupart des apprentissages automatiques utilisent l'apprentissage supervisé.

L'apprentissage supervisé se compose de variables d'entrée ( $x$ ) et de variables de sortie ( $Y$ ). Vous utilisez un algorithme pour apprendre une fonction de mappage de l'entrée à la sortie.

$$Y=f(X)$$

L'objectif est de bien comprendre la fonction de mappage afin que lorsque vous avez de nouvelles données d'entrée ( $x$ ), vous puissiez prédire la variable de sortie ( $Y$ ) pour ces données. C'est ce qu'on appelle l'apprentissage supervisé car le processus d'extraction de l'algorithme de l'ensemble d'apprentissage peut être considéré comme l'enseignant qui supervise le processus d'apprentissage. Nous connaissons la bonne réponse, et l'algorithme fait des prédictions itératives sur les données d'apprentissage, corrigées par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable.

### 2.4.2 Apprentissage non supervisé

L'apprentissage non supervisé implique de n'avoir que des données d'entrée ( $X$ ) sans variables de sortie correspondantes.

Le but de l'apprentissage non supervisé est de modéliser la structure sous-jacente ou la distribution dans les données afin d'en savoir plus sur les données.

C'est ce qu'on appelle l'apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'a pas de bonnes réponses ou d'enseignants. Ces algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter des structures intéressantes dans les données.

L'apprentissage non supervisé comprend deux classes d'algorithmes : les algorithmes de clustering et les algorithmes d'association.

### 2.4.3 Apprentissage semi-supervisé

Un problème où vous avez beaucoup de données d'entrée ( $X$ ) et seulement une partie d'entre elles est étiquetée ( $Y$ ) est appelé un problème d'apprentissage semi-supervisé. Donc, Ces problèmes se situent quelque part entre l'apprentissage supervisé et non supervisé. Exemple : Une archive de photos où seules certaines images sont étiquetées (par exemple, des chiens, des chats, des personnes) et la plupart ne le sont pas. De nombreux problèmes d'apprentissage automatique dans le monde réel relèvent de ce domaine. En fait, l'étiquetage des données peut prendre beaucoup de temps ou être coûteux, car il peut nécessiter l'accès à des experts du domaine. Les données non étiquetées sont bon marché et faciles à collecter et à stocker. Vous pouvez utiliser des techniques d'apprentissage non supervisé pour découvrir et apprendre la structure des variables d'entrée.

## Chapitre 02 : Les Réseaux De Neurone Artificiels

Vous pouvez également utiliser des techniques d'apprentissage supervisé pour faire des prédictions optimales sur des données non étiquetées, les alimenter en tant que données de formation dans un algorithme d'apprentissage supervisé et utiliser le modèle pour faire des prédictions. [24]

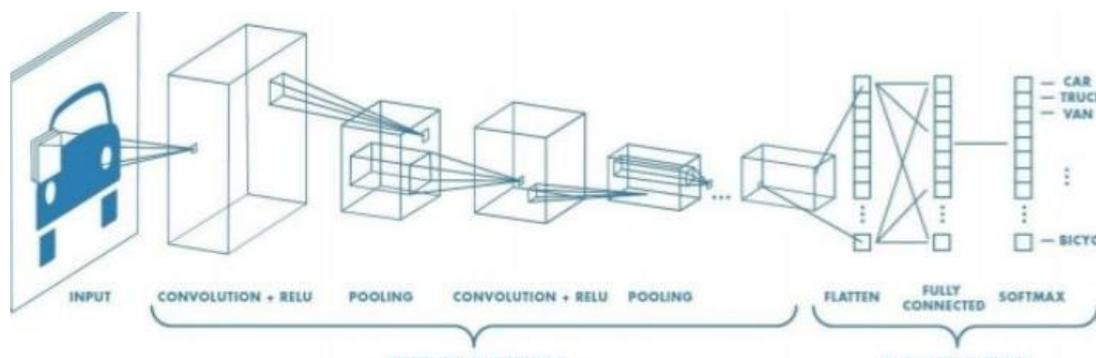
### 2.5 Réseau de neurones à convolution (CNN)

Le nom « réseau neuronal convolutif » suggère que le réseau emploie une opération mathématique appelée convolution. Un réseau convolutif est un type spécial de réseau neuronal qui utilise des convolutions au lieu de la multiplication matricielle générale dans au moins une de ses couches. Les CNN sont l'un des meilleurs algorithmes d'apprentissage pour effectuer des opérations de convolution, qui aident à extraire des caractéristiques utiles à partir de points de données corrélés localement.

La sortie du noyau de convolution est ensuite affectée à une unité de traitement non linéaire (fonction d'activation), qui non seulement facilite l'apprentissage de l'abstraction, mais intègre également la non-linéarité dans l'espace des caractéristiques. Cette non-linéarité génère différents modèles d'activation pour différentes réponses, aidant ainsi à apprendre les différences sémantiques dans les.

[25]

La Figure 2.3 montre la structure générale d'un réseau CNN.



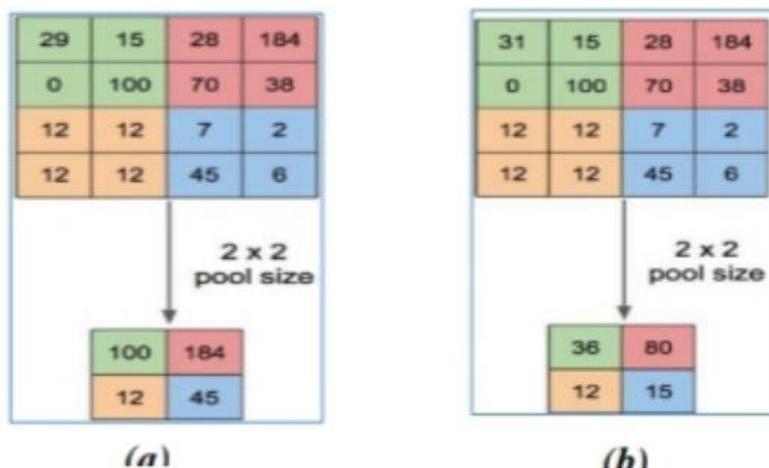
**Figure 2.3:** Réseau de neurones avec de nombreuses couches convolutives. [25]

**Couche de convolution :** la convolution est la première couche qui extrait les caractéristiques de l'image d'entrée. Les convolutions préservent la relation entre les pixels en apprenant les caractéristiques d'une image à l'aide de petits carrés de données d'entrée. Il s'agit d'une opération mathématique qui nécessite deux entrées, telles qu'une matrice d'image et un filtre ou un noyau (Liu et al., 2013) (Indolia et al., 2018). La Figure 2.4 monte une opération simple d'un filtre pour une étape de convolution.



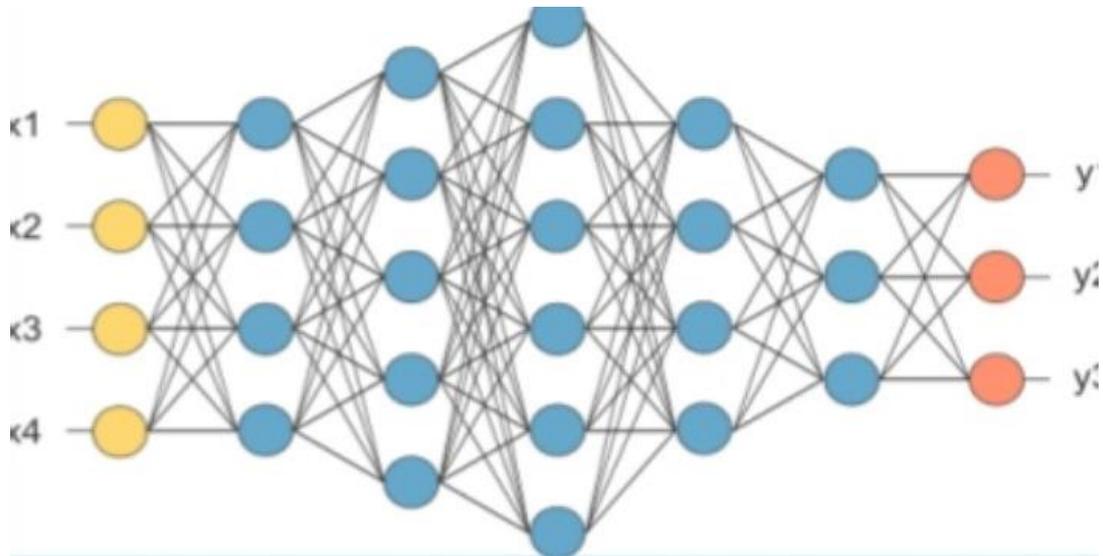
**Figure 2. 4:** Exemple explicative sur l’opération de convolution. [25]

**Couche de pooling :** une couche de regroupement (pooling) est généralement placée entre deux couches convolutionnelles, et le regroupement est un processus de discrétisation basé sur des échantillons. Le but est de lever la représentation d'entrée (image, matrice de sortie de la couche cachée, etc.), en réduisant sa dimensionnalité et en tenant compte des hypothèses à être rendues des caractéristiques contenues dans les sous-régions regroupées. Il existe différents types de pooling : (1) Pooling moyen qui prend la moyenne de tous les pixels de la sélection Figure 2.5. (a), (2) Pooling maximal qui prend le pixel qui a la valeur maximale entre tous les pixels de la sélection. Voir Figure 2.5. (b).



**Figure 2. 5:** (a) pooling maximal, (b) pooling moyen. [25]

**Couche fully-connected** : Les couches entièrement connectées sont similaires aux réseaux entièrement connectés dans les modèles convolutifs. La sortie de la première étape (y compris la convolution et la mise en commun répétée) est introduite dans la couche entièrement connectée, et le produit scalaire du vecteur de poids et du vecteur d'entrée est calculé pour obtenir la sortie finale. [25]

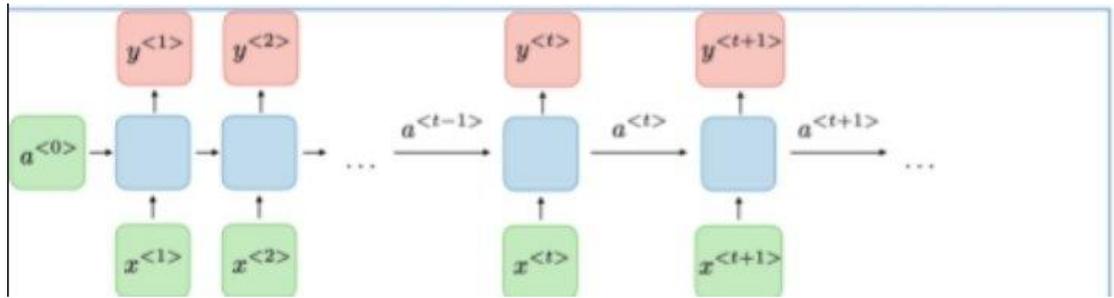


**Figure 2. 6:** Couche fully-connected. [25]

### 2.6 Réseaux Neuronaux Récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une variante très importante des réseaux de neurones qui sont largement utilisés dans le traitement du langage naturel. Elles sont appelées boucles (Récurrents) car elles effectuent la même tâche sur chaque élément de la séquence et la sortie dépend des calculs précédents. Une autre façon de penser aux RNN est qu'ils ont une "mémoire" qui capture des informations sur ce qui a été calculé jusqu'à présent. En théorie, les RNN peuvent utiliser des informations provenant de séquences arbitrairement longues, mais en pratique, ils se limitent à examiner quelques étapes en arrière.

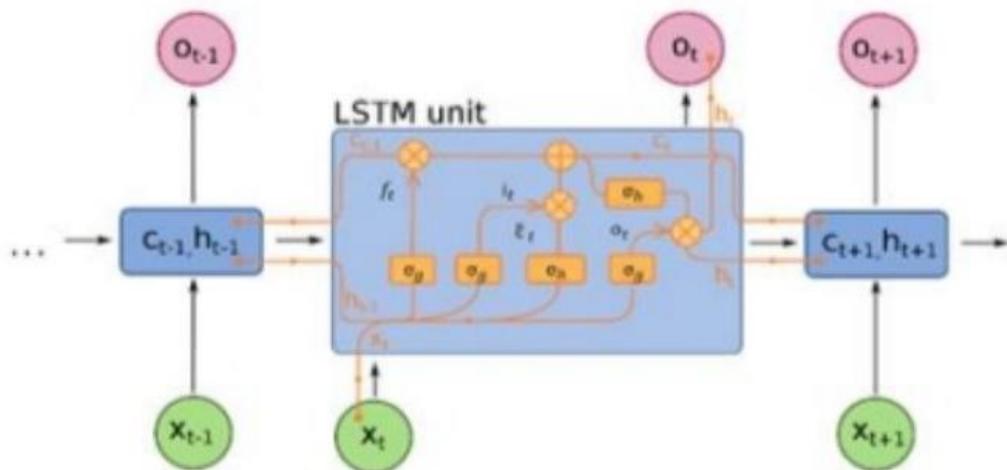
Les RNN sont une classe de réseaux de neurones qui permettent d'utiliser les prédictions passées comme entrée via des états cachés. Leur forme est illustrée à la figure 2.7.



**Figure 2. 7:** Architecture de RNN. [25]

### 2.7 Réseaux Long Short-Term Memory (LSTM)

En théorie, les RNN classiques peuvent suivre des dépendances arbitraires à long terme dans les données d'entrée. Le problème avec les RNN classiques est pratique : lorsqu'un RNN classique est entraîné par rétro propagation, le gradient à long terme de rétro propagation peut "disparaître" (c'est-à-dire tendre vers zéro) ou "exploser" (c'est-à-dire tendre vers l'infini), en raison des calculs impliqués utilisent des nombres à précision finie. Pour éviter ces problèmes de gradient, un type de réseau de neurones appelé Long Short-Term Memory (LSTM) a été développé. [26]



**Figure 2. 8:** Architecture de Long-Short Term Memory Unit. [26]

Contrairement aux réseaux de neurones à anticipation standard, les LSTM ont des connexions de rétroaction. De tels réseaux de neurones récurrents peuvent non seulement traiter des points de données statiques, mais également des données dynamiques. Une cellule LSTM commune se compose d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli, comme illustré à la figure 3.8. La cellule mémorise des valeurs sur un intervalle de temps arbitraire, et trois portes régulent le flux d'informations entrant et sortant de la cellule, ce qui a pour effet de limiter le

## Chapitre 02 : Les Réseaux De Neurone Artificiels

---

problème de gradient évanescent. La forme compacte de l'équation de transfert direct pour une cellule LSTM avec une porte d'oubli est :

$$f_t = (W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = (W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = (W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = (W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

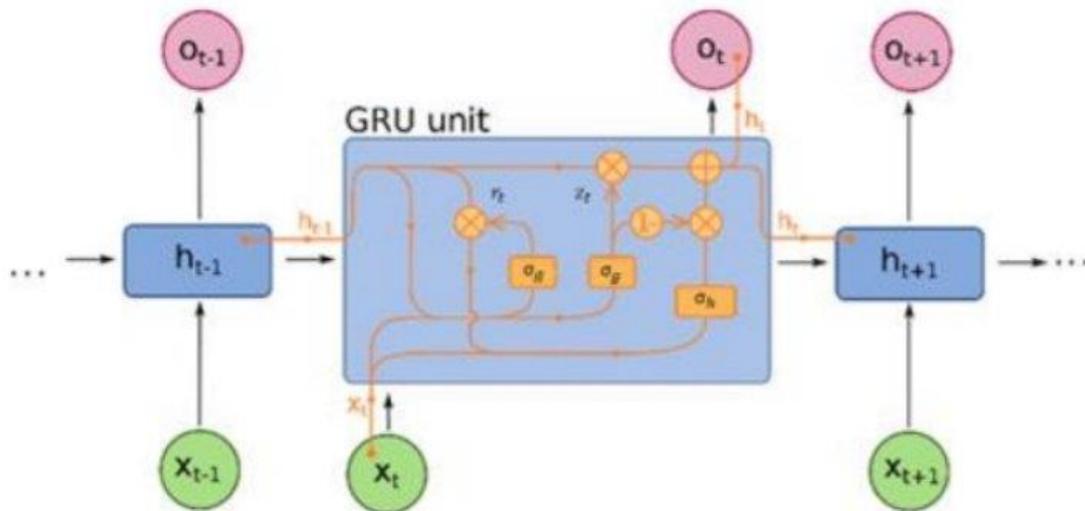
$$h_t = o_t \circ \sigma h(c_t)$$

Valeur initiale  $c_0=0$ ,  $h_0=0$ . L'opérateur  $\circ$  représente le produit d'Hadamard (produit élément par élément). L'indice  $t$  représente le pas de temps. Les autres variables sont les suivantes :

- $x_t \in R^d$  : vecteur d'entrée
- $f_t \in (0,1)^h$  : vecteur d'activation de la porte d'oubli
- $i_t \in (0,1)^h$  : vecteur d'activation de la porte d'entrée/mise à jour
- $o_t \in (0,1)^h$  : vecteur d'activation de la porte de sortie
- $h_t \in (-1,1)^h$  : vecteur de sortie de l'unité LSTM.
- $\tilde{c}_t \in (-1,1)^h$  : vecteur d'activation de l'entrée de la cellule
- $c_t \in R^h$  : vecteur d'état des cellules
- $W \in R^{h \times d}$ ,  $U \in R^{h \times h}$  et  $b \in R^h$  : les matrices de poids et les paramètres des vecteurs de biais qui doivent être appris pendant la formation

Les RNN utilisant des unités LSTM résolvent donc partiellement le problème du gradient évanescent.

Cependant, les réseaux LSTM peuvent toujours souffrir du problème du gradient explosif. Quant au GRU, il est comme un LSTM avec une porte d'oubli, mais a moins de paramètres que le LSTM, car il manque une porte de sortie comme illustré dans la figure 2.9.



**Figure 2. 9:** Gated Recurrent Unit. [26]

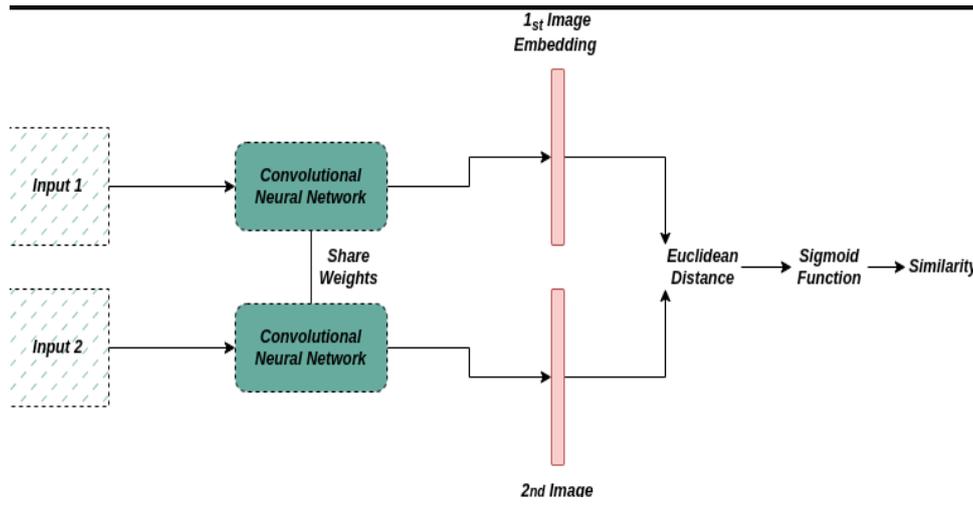
Malgré cela, il a été démontré que les GRU ont une meilleure performance sur certains ensembles de données plus petits et moins fréquents. [26]

## 2.8 Les Réseaux Siamois

Les réseaux siamois ont été introduits par Gregory Koch en 2015. Le nom « siamois » vient du fait que le réseau est conçu avec deux sous-réseaux identiques, chacun traitant un échantillon d'entrée différent avec les mêmes poids. Les sorties de ces deux sous-réseaux sont ensuite comparées dans la couche finale afin de générer une prédiction. Ces réseaux sont donc utilisés dans des tâches de reconnaissance d'images afin de décider si deux images sont différentes ou non. Les réseaux siamois se sont avérés particulièrement utiles pour les situations avec peu de données, car ils peuvent être formés sur de petites quantités de données étiquetées, puis être affinés sur des ensembles de données plus volumineux.

### 2.8.1 L'architecture des réseaux siamois

Tout d'abord, les deux branches du réseau neuronal ont les mêmes poids de réseau neuronal, et elles apprennent toutes les deux une représentation commune pour les deux entrées mais sont entraînées séparément sur des entrées différentes. Ce fait permet au réseau d'apprendre à comparer des images plutôt qu'à les classer. L'architecture d'un réseau siamois se compose des éléments suivants :



**Figure 2. 10 :** L'architecture des réseaux siamois. [27]

Architecture du réseau siamois Les couches de convolution et de mise en commun sont responsables de l'extraction de caractéristiques significatives à partir des échantillons d'images de chaque réseau. La dernière couche de chaque branche est la couche de comparaison, qui génère une intégration, une représentation compacte de l'échantillon de données. La couche de comparaison est généralement conçue de plusieurs manières, qui dépendent de la tâche spécifique. Des exemples typiques sont la distance euclidienne (eucliden distance) ou la similarité de corrélation. L'intégration est ensuite pilotée par une fonction de comparaison afin de produire une prédiction qui décide si les deux échantillons de données sont identiques. Par conséquent, la sortie finale du réseau siamois est un score de similarité, qui indique à quel point les deux images d'entrée sont similaires ou différentes. Ce score peut être utilisé pour faire une prédiction, par exemple si deux images de visage appartiennent à la même personne ou non. [27]

### 2.8.2 Formation des Réseaux Siamois(Training)

La formation des Réseaux Siamois (Training of Siamiese Network) peut être encadrée ou non. Dans l'apprentissage supervisé, le réseau est formé sur des données étiquetées avec des étiquettes de vérité terrain connues. Dans l'apprentissage non supervisé, le réseau est formé sur des données non étiquetées et doit apprendre à générer ses propres étiquettes en fonction des images d'entrée. La fonction de perte pour les réseaux siamois est généralement basée sur le score de similarité généré par la fonction de comparaison. Habituellement, le triplet ou la perte contrastive est utilisé pour l'apprentissage. Dans la reconnaissance faciale, la fonction de perte pourrait être la perte d'entropie croisée entre le score de similarité prédit et la véritable étiquette. La formation des réseaux jumeaux doit être effectuée avec des ensembles de données équilibrés afin que les deux réseaux soient également formés. Le réseau siamois doit apprendre à faire la distinction entre

## Chapitre 02 : Les Réseaux De Neurone Artificiels

---

des images similaires et différentes, et un ensemble de données équilibré fournit un échantillon plus représentatif des deux types d'images. [27]

### 2.9 L'utilisation des réseaux de neurones

Les réseaux de neurones ont la capacité extraordinaire de comprendre et d'extraire des règles et des tendances à partir de données complexes, bruyantes et imprécises. Ils peuvent être utilisés pour extraire des modèles et détecter des tendances basées sur des fonctions mathématiques complexes qui sont difficiles (voire impossibles) à modéliser à l'aide de techniques d'analyse ou de paramétrage traditionnelles.

L'un des avantages des réseaux de neurones est la capacité de prédire avec précision des données qui ne sont pas des données d'entraînement, un processus appelé généralisation. Au vu de ces caractéristiques et de sa large gamme d'applications, les réseaux de neurones sont particulièrement adaptés à l'application de problèmes spécifiques dans les domaines de la recherche scientifique, commerciale et industrielle [28].

Voici de nombreux domaines dans lesquels les réseaux de neurones sont appliqués avec succès :

- Traitement du signal ;
- Prédiction ;
- Maîtrise des processus ;
- Robotique ;
- Classification ;
- Prétraitement des données ;
- Reconnaissance de formes ;
- Analyse d'image et synthèse vocale ;
- Diagnostiques et suivi médical ;
- Marché boursier et prévisions ;
- Demande de crédits ou de prêts immobiliers.

### 2.10 Les avantages et les inconvénients des réseaux de neurone

Le réseau neuronal peut exécuter n'importe quelle fonction non linéaire jusqu'à ce qu'il atteigne un certain degré de fiabilité. Il peut également implémenter des fonctions dynamiques et avoir n'importe quel nombre d'entrées et de sorties. Nous énumérerons les avantages et les inconvénients des réseaux de neurones artificiels.

#### 2.10.1 Les Avantages des réseaux de neurones

- La capacité d'exprimer n'importe quelle fonction, qu'elle soit linéaire ou non, simple ou complexe.
- Une faculté d'apprendre à partir d'exemples représentatifs par rétro propagation d'erreurs. L'apprentissage (ou la construction de modèles) se fait automatiquement.
- Insensible aux données non fiables ou au bruit.
- Facile à utiliser, le travail personnel exigeait beaucoup moins qu'une analyse statistique classique. Aucune compétence en mathématiques et calculs statistiques requis.
- Moins de mauvais comportements en cas de petite quantité de données.
- Pour un utilisateur novice, l'apprentissage est plus facile à comprendre que la complexité des statistiques multi variées.

#### 2.10.2 Les Inconvénients des réseaux de neurones

- Il n'y a pas de méthode systématique de définir la topologie optimale du réseau et le nombre de neurones à placer dans la couche cachée.
- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage jouent un rôle important dans la vitesse de convergence.
- Le problème du sur-apprentissage qui ne se généralisera pas.
- Les connaissances acquises par le réseau neuronal sont encodées par la valeur du poids synaptique, le réseau neuronal est donc une boîte noire et les utilisateurs ne peuvent pas comprendre ces connaissances [28].

### **2.11 Conclusion**

Dans le présent travail nous allons appliquer les Réseaux de neurones artificiels, inspirés du comportement du cerveau humain, permettent de créer de l'intelligence artificielle. Notamment appliqué en datamining principalement, donc dans ce chapitre nous avons vu un aperçu général sur les types de réseaux de neurones et ses architecteurs et citer les domaines d'utilisation des réseaux de neurones, enfin nous nous sommes intéressés aux réseaux de neurones artificiels Siamois.

**Chapitre 03 :**  
**Détection acoustique**  
**de drone**

### 3.1 Introduction

Dans cette partie du mémoire, on va présenter l'analyse et le traitement du son de la bases de données pour ainsi la classifier et cela en utilisant le langage de programmation python, mais aussi nous avons opté pour l'application d'un réseau neuronal qui sera donc en charge de faire la différence entre un bruit quelconque et celui émis par un drone, cette opération a été réalisé grâce à une analyse des différences entre les spectrogrammes des différents enregistrements.

### 3.2 Logiciel de travail

Le choix du langage de programmation s'est fait sur les bases suivantes : car il propose une large série de bibliothèques qui sont très utiles dans le traitement des signaux acoustiques, parmi ces dernières on pourrait citer :

- Wave : package utilise pour le traitement des ondes sonores
- Matplotlib : pour procéder au plotting des différents graphs mais aussi créer des visualisations statiques, animées et interactives en Python.
- Numpy : on a utilisé cette outil pour la programmation scientifique et notamment la data science.
- Scipy : pour obtenir de bonnes performances d'exécution.

**Python** est un langage de programmation « open source » qui s'exécutent sur tous les systèmes d'exploitation, y compris Linux, Mac OS et Windows. Il est largement utilisé dans le traitement big data. Certaines caractéristiques notables de ce langage qui nous a encouragés à l'utiliser sont :

- la facilité à trouver de l'aide car la communauté python est active.
- la disponibilité des ressources.
- La syntaxe est propre et met l'accent sur la production d'un code bien structuré et lisible.

### 3.3 La collecte des données (dataset)

On a premièrement procéder à l'étude des sonorités provenant d'un environnement clos pour ainsi avoir une bonne précision, on a donc eu la chance d'obtenir des extraits audio de drones mais aussi de plusieurs sources de bruits pour simuler l'environnement d'un bureau :

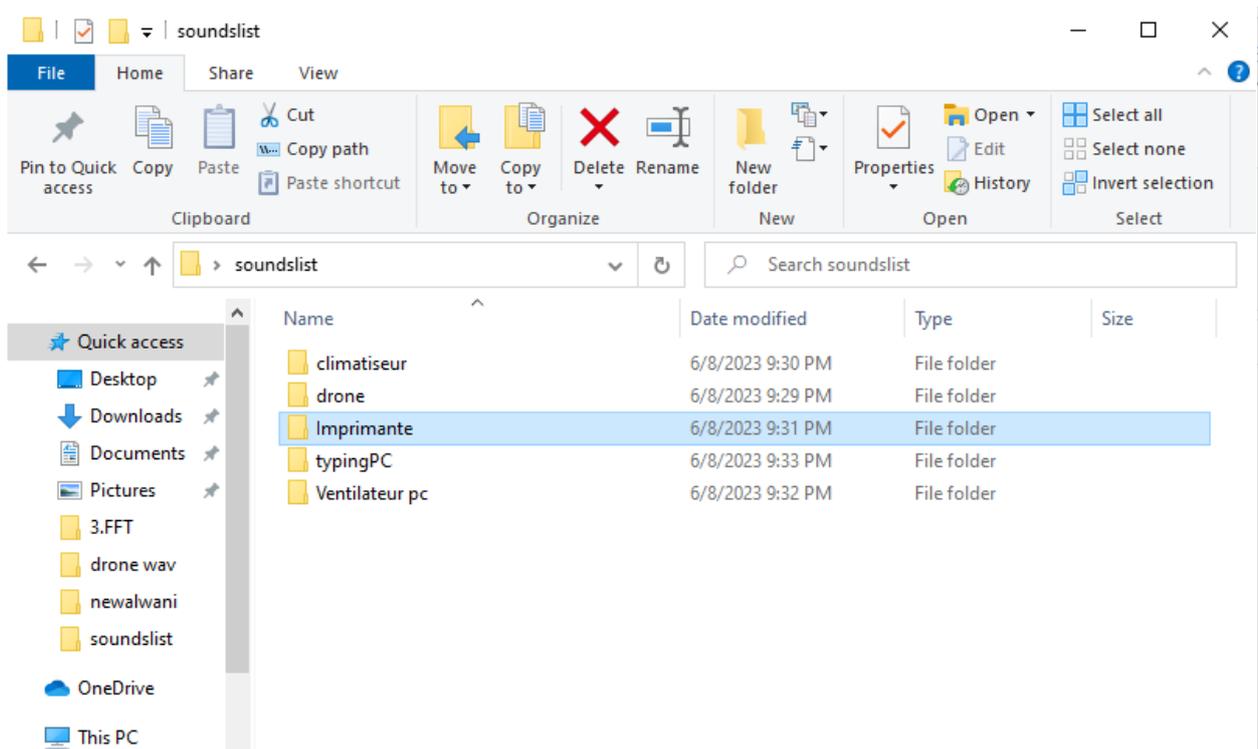
Imprimante

Ventilateur pc

Climatiseur....etc.

NB : en ce qui concerne le bruit des drones nous avons pris plusieurs types de drones mais aussi de configurations, a noter un drone s'éloignant ou se rapprochant, et aussi en vol stationnaire.

## Chapitre 03 : Détection acoustique de drone



**Figure 3. 1** : Liste des sons collectés.

Les sons de la figure.3.1 sont téléchargés gratuitement sur zaplsplat [29]. Certains parmi ces audio on les trouve sous format d'une onde (.wav) mais aussi il existe des sons sous format (.mp3). Ces derniers on les convertit vers des ondes à l'aide d'un convertisseur en ligne [30]. Après la conversation des sons nous avons préparés un document texte (.Txt) pour faciliter la lecture de nos données

Ces fichiers sont implémente dans le python en utilisant la commande “*from scipy.io import wavfile*”. Après l’importation des bibliothèque, nous avons fait une boucle pour ouvrir Le document texte comme un dossier afin que lire plusieurs sons à la fois “*with open('tr.txt') as file*”

### 3.4. La visualisation des sons

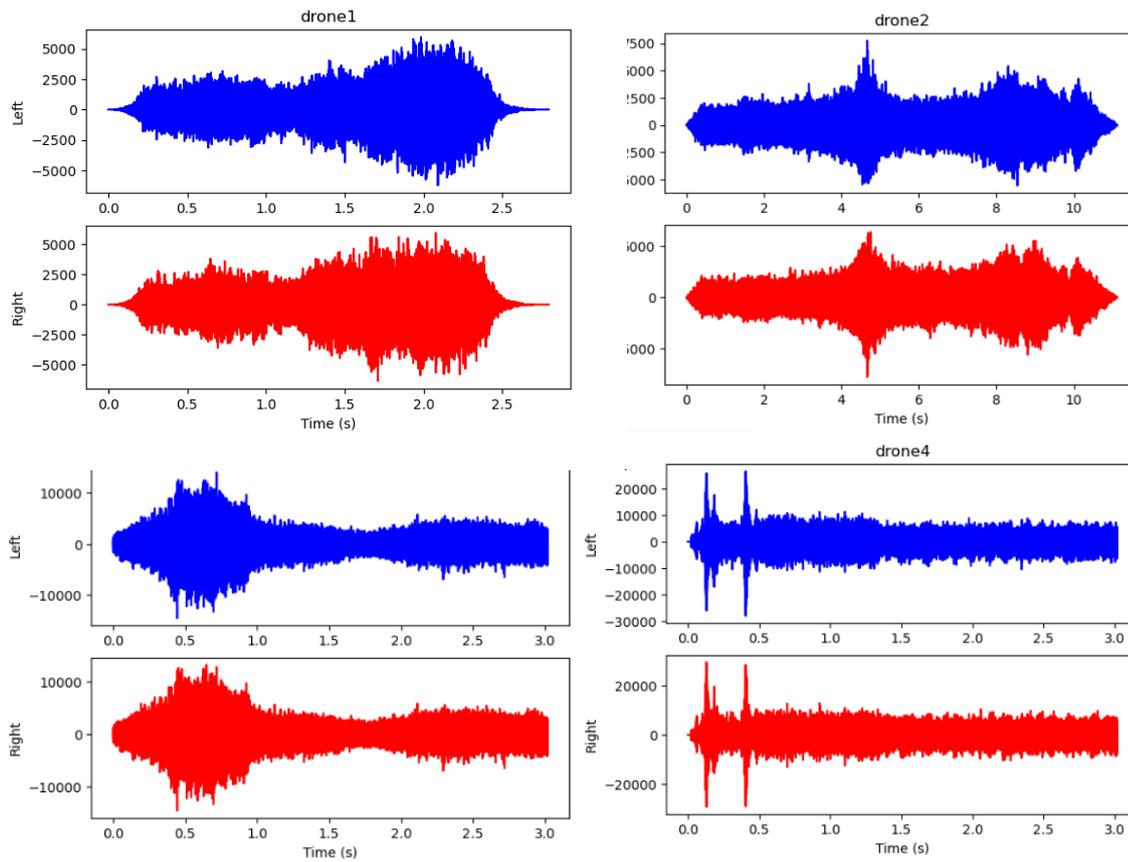
Pour détecter le son il faut connaitre quelques informations sur le signal : la représentation temporelle, transformé rapide de fourrier, spectrogramme.

#### 3.4.1. La représentation temporelle

Cette représentation montre l'évolution de l'intensité du signal sonore dans le temps avec une amplitude normalisé et ne renseigne pas sur le contenu fréquentiel.

Premièrement, nous étudions l'évolution de signal (ondes) dans le domaine temporelle. Pour effectuer ce travail nous avons tracé le graphe de l'amplitude de signal en fonction de temps(s) et comme un résultat on a observé que le signal est ou mode **Stéréo**.

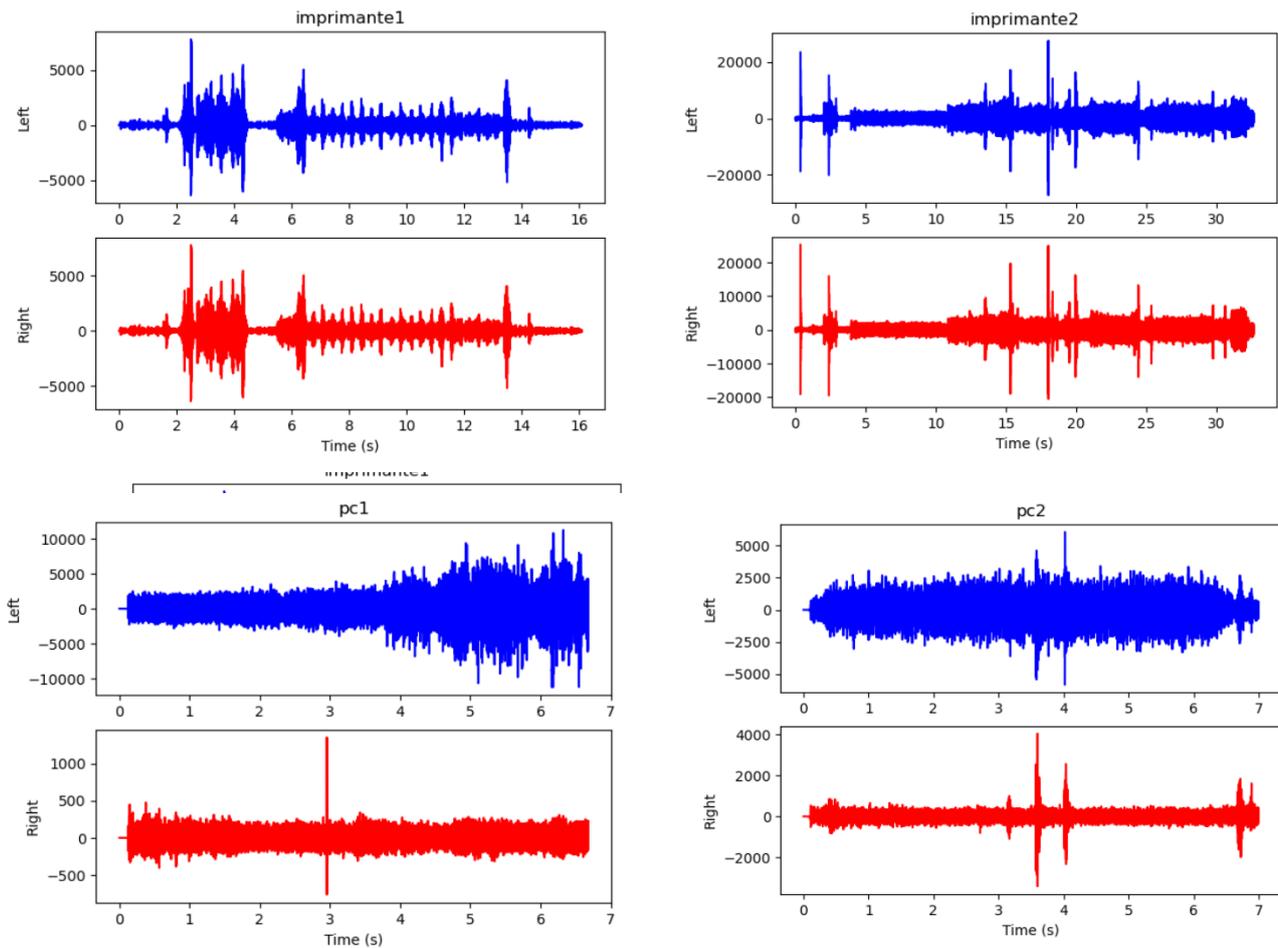
## Chapitre 03 : Détection acoustique de drone



**Figure 3. 2:** Résultats de visualisation des sons de drones dans le domaine temporel.

- Les résultats pour d'autres sons :

## Chapitre 03 : Détection acoustique de drone



**Figure 3.3 :** Résultats visualisation des sons de Bruit dans le domaine temporel (stéréo).

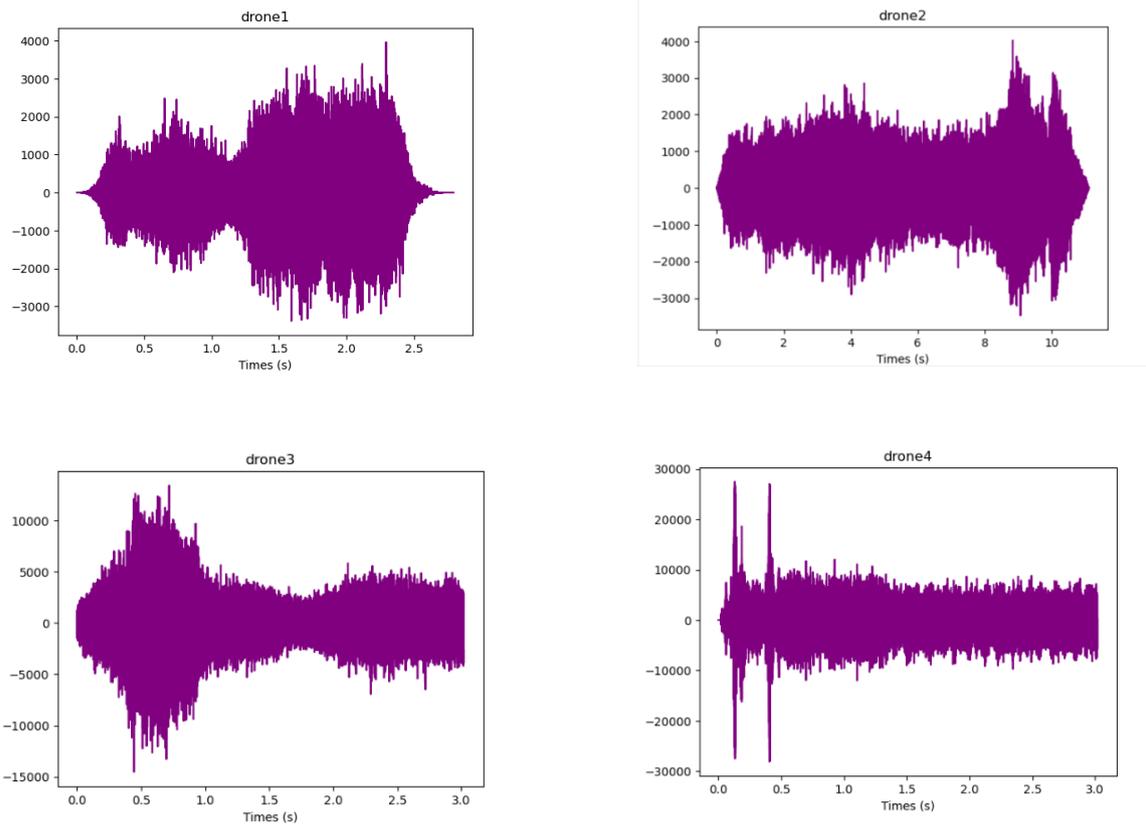
Après le traçage des graphes, nous avons converti les sons de format Stéréo vers format **mono** a but de faciliter l'étude de nos données. Cette étape accomplir par la bibliothèque de python **pydub** et plus précisément la commande **AudioSegment**. Le code utilise est :

```
« sound = AudioSegment.from_wav(f) »
```

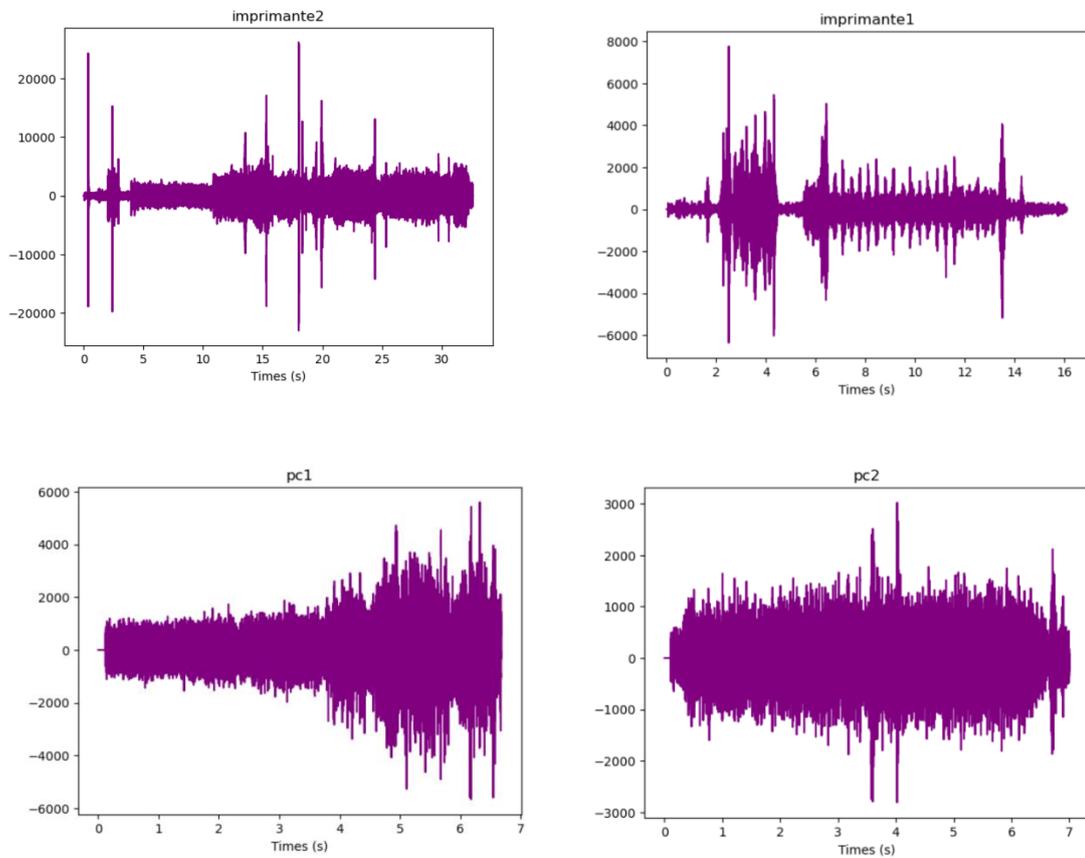
```
« sound = sound.set_channels(1) »
```

Nous avons finalisé le programme par traçage des ondes mono et nous avons obtenu :

## Chapitre 03 : Détection acoustique de drone



**Figure 3. 4:** Résultats de visualisation des sons de drones dans le domaine temporel (mono).



**Figure 3. 5:** Résultats de visualisation des sons de bruit dans le domaine temporel (mono).

### 3.4.2. La représentation fréquentielle

Comme dans de nombreux domaines de la physique (acoustique, traitement du signal...) il est très courant de considérer un signal par sa représentation fréquentielle.

Pour cela on fait subir au signal  $S(t)$  une transformée de Fourier qui fait apparaître les fréquences caractéristiques qu'il contient.

#### 3.4.2.1. Transformée de Fourier rapide

La Transformée de Fourier Rapide, appelée **FFT** [*Fast Fourier Transform*], est un algorithme qui permet de calculer des Transformées de Fourier Discrètes **DFT** [*Discrete Fourier Transform*].

Parce que la DFT permet de déterminer la pondération entre différentes fréquences discrètes, elle a un grand nombre d'applications en traitement du signal, par exemple Il peut être utile pour lire des choses comme les ondes sonores ou pour toute technologie de traitement d'image. Par conséquent, les données discrètes qu'elle prend en entrée sont souvent appelées *signal* et dans ce cas on considère qu'elles sont définies dans le domaine temporel. Les valeurs de sortie sont alors appelées le *spectre* et sont définies dans le domaine des fréquences. Toutefois, ce n'est pas toujours le cas et cela dépend des données à traiter. [31]

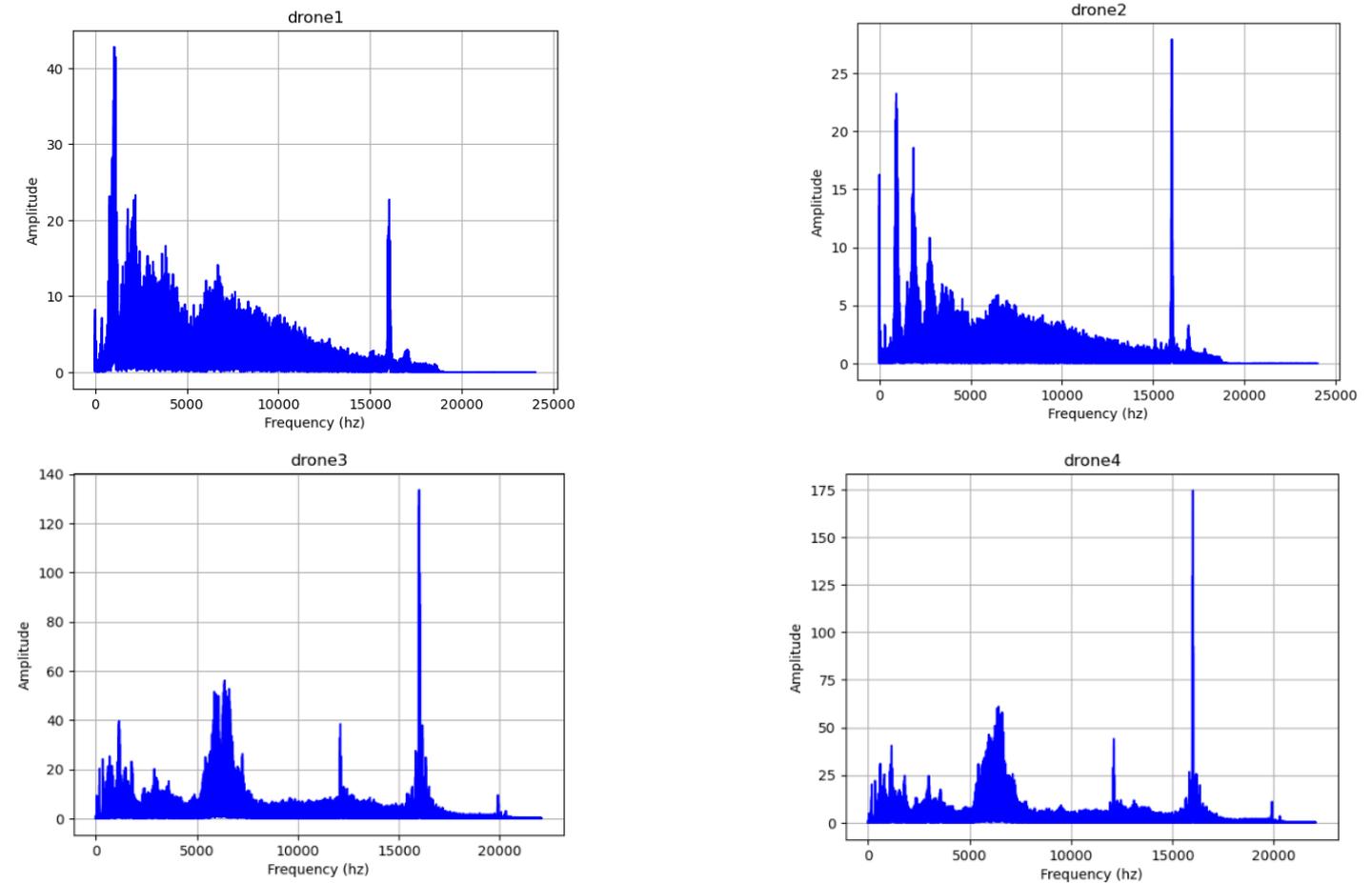
$$A_k = \sum_{m=1}^{n-1} a_m + e^{-2\pi i \frac{mk}{n}} \quad \text{Avec } k = 1, 2, \dots, n-1, \dots \dots \dots (1)$$

La DFT inverse est donnée par :

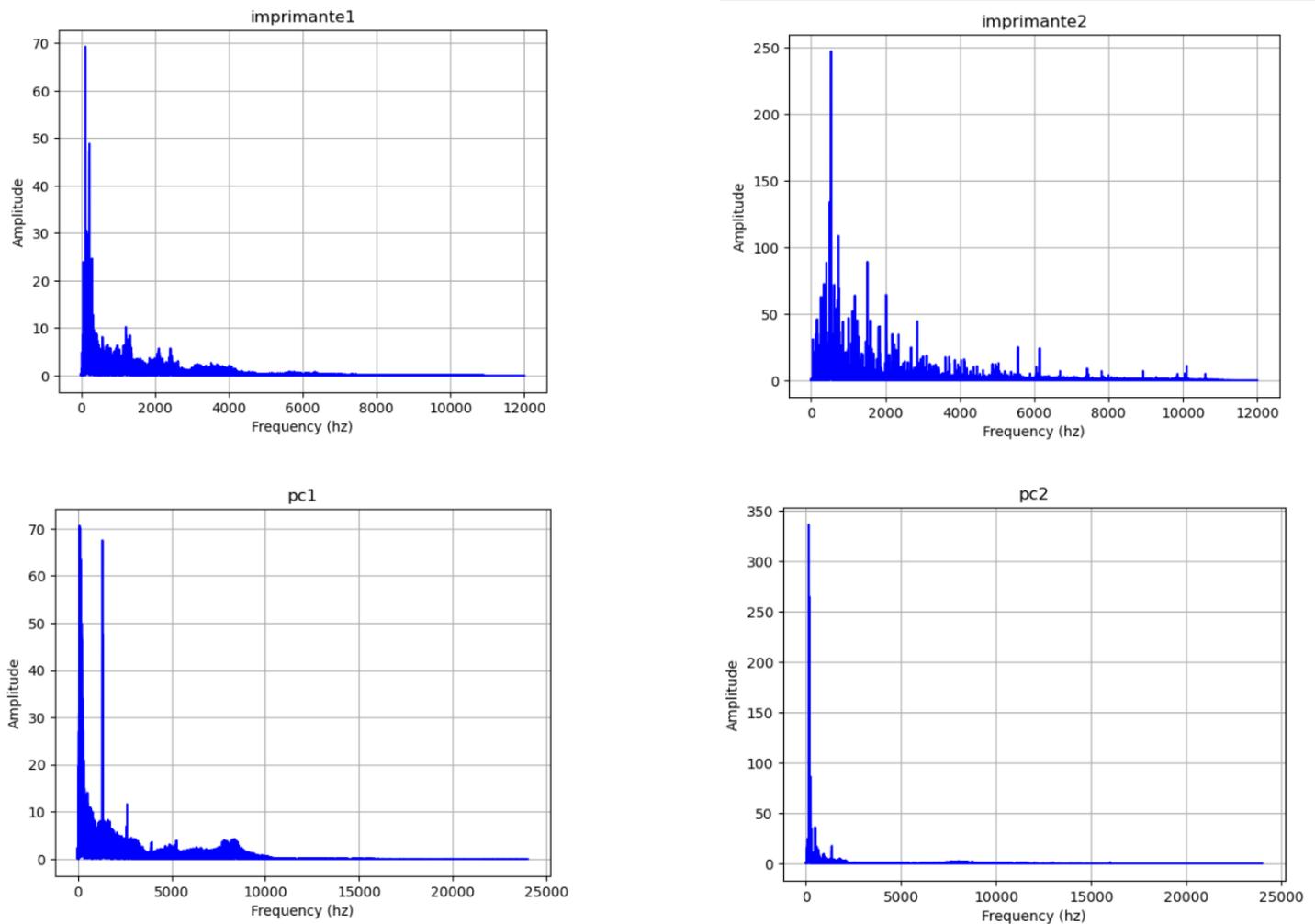
$$a_m = \frac{1}{n} \sum_{k=0}^{n-1} A_k * e^{-2\pi i \frac{mk}{n}} \quad \text{Avec } m = 1, 2, \dots, n-1, \dots \dots \dots (2)$$

Dans le présent programme, nous avons choisi le modèle Numpy (np) pour calculer le FFT de signal «  $a\_k = np.fft.fft(a)[0:int(na/2)]/na$  ». Après nous avons enlevé la partie imaginaire de signal parce que la plupart des informations importantes sont contenues dans la partie réelle du spectre obtenu par la FFT. La partie réelle représente l'amplitude des différentes fréquences présentes dans le signal audio. La partie imaginaire, en revanche est symétrique et ne contient pas d'informations significatives dans les contextes de l'audio. En plus la suppression de la partie imaginaire réduit la dimension des données à traiter. «  $Pxx = np.abs(a\_k)$  ». Finalement nous avons trace les graphes de l'amplitude en fonction de fréquence :

## Chapitre 03 : Détection acoustique de drone



**Figure 3. 6:** Résultats de visualisation des sons de drone dans le domaine fréquentielle.



**Figure 3. 7:** Résultats de visualisation des sons de bruit dans le domaine fréquentielle.

### 3.4.3. La représentation spectrale

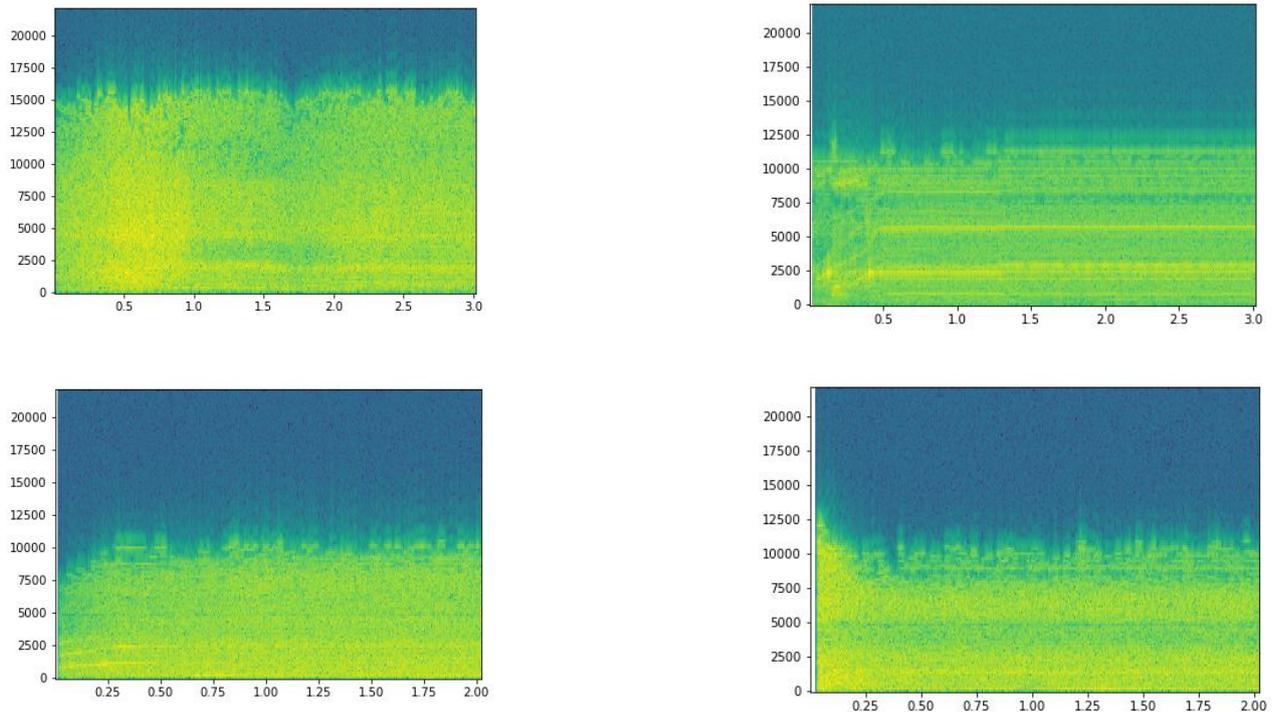
Le spectrogramme est une représentation visuelle qui permet de visualiser le contenu d'un audio dans le domaine temps-fréquence. Il permet de visualiser l'évaluation des différentes présentes dans le signal audio. Cette représentation est largement utilisée dans le traitement de son, la musique...etc. elle permet d'analyser les composantes harmoniques, les formats vocaux, les modulations fréquentielles. En plus, elle peut être utilisée pour détecter des événements spécifiques, classifier et reconnaître des sons.

Revenons à notre programme Python. Pour représenter le spectrogramme des extraits sonores, nous avons utilisé une commande spéciale **pcolormesh**: "`plt.pcolormesh(fm[i], tr[i], lspg[i], shading='auto')`" tel que "`fr, tm, spgram = signal.spectrogram(a, s)`", "`lspg = np.log(spgram)`"; et nous avons obtenu :

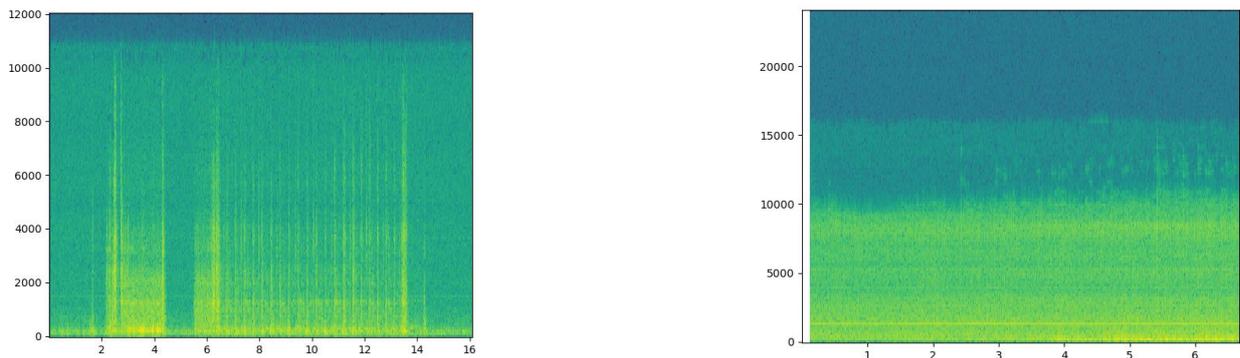
Après le traçage des spectrogrammes, nous avons les enregistrements comme des photos sous format **png** dans un dossier "`plt.savefig(f'sppng/spect{i}.png')`".



## Chapitre 03 : Détection acoustique de drone



**Figure 3. 9:** Résultats de visualisation des sons des bruits dans le domaine spectrale.



**Figure 3. 8:** Résultats de visualisation des sons des drones dans le domaine spectrale

- **Explication des couleurs**

Les couleurs utilisées dans un spectrogramme sont essentielles pour aider à comprendre les caractéristiques temporelles et fréquentielles du signal sonore.

➤ **Jaune:** Le jaune est souvent utilisé pour représenter des parties du signal sonore qui ont une intensité relativement élevée ou une énergie acoustique élevée. Dans un spectrogramme, cela pourrait correspondre aux parties du signal où le son est plus fort ou plus intense. Par exemple, si tu avais un son fort comme un son de drone avec des hélices, ces parties seraient probablement représentées en jaune.

➤ **Bleu:** Le bleu est souvent utilisé pour représenter des parties du signal avec une intensité modérée à faible. Cela peut inclure des parties du signal où l'intensité sonore

## Chapitre 03 : Détection acoustique de drone

est plus douce, calme ou moins énergétique. Les parties du signal qui contiennent moins d'énergie ou des fréquences moins dominantes pourraient être affichées en vert.

Globalement, l'utilisation de jaune et de vert dans un spectrogramme permet d'identifier rapidement les variations d'intensité du signal sonore. Les couleurs aident les auditeurs à visualiser les différents composants fréquentiels et temporels du son, ce qui est particulièrement utile pour analyser des signaux sonores complexes tels que de la musique, de la parole ou d'autres phénomènes acoustiques.

### 3.5. Type et structure de réseau de neuronal

Dans notre travail, nous avons choisi réseau neuronal siamois(Siamese) pour la classification des spectrogrammes (drone ou non-drone).

Le Réseau Neural Siamois est un type particulier de réseau neuronal dans un premier temps nous formons une image avec une séquence de couches convolutionnelles, des couches de mise en commun et des couches entièrement connectées nous finissons par un vecteur  $f(x1)$ . Puis nous formons une autre image dans la même séquence pour obtenir un autre vecteur de fonction  $f(x2)$ . Maintenant nous calculons  $d$  qui sera la distance entre chacun des points dans le vecteur de fonction  $f(x1)$  avec le vecteur de fonction  $f(x2)$ . Si  $d$  est petit on peut dire que les deux images sont identiques sinon si  $d$  est grand c'est l'inverse. [32]

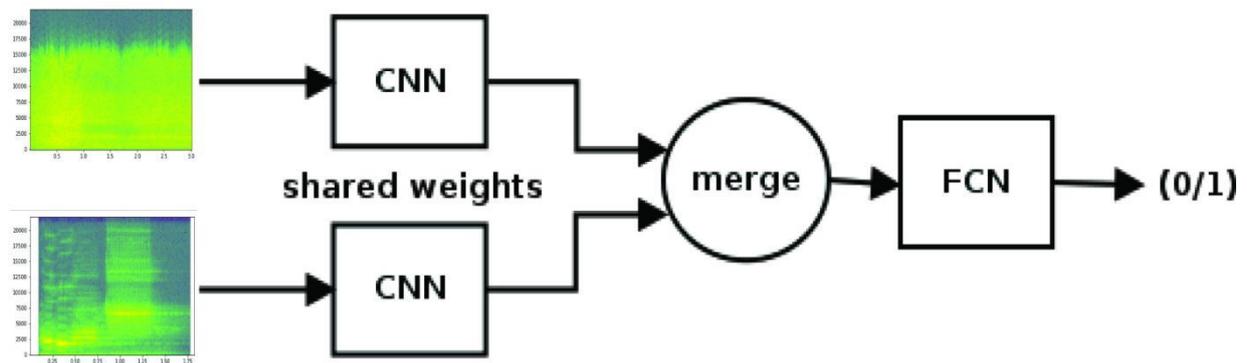


Figure 3. 10: L'architecture des réseaux siamois. [32]

#### 3.5.1 Bibliothèque utilisé

TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissages expérimentaux et de les transformer en logiciels. Cette bibliothèque permet notamment d'entraîner et d'exécuter des **réseaux de neurones** pour la classification de chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents. [33]

Cette bibliothèque est contient plusieurs commandes, parmi ces derniers nous avons utilisé les commandes dont nous avons besoin par exemple : Model, layers, applications

```
“ from tensorflow.keras import Model ”
```

```
“ from tensorflow.keras import layers ”
```

```
« from tensorflow.keras import applications »
```

### 3.6 Algorithme de manipulation

#### 3.6.1 Lecture des données

Maintenant, nous définissons une fonction pour lire nos images d'entrée et les générer comme des paires. Nous avons utilisé l'entrée comme des images et les transforme vers une matrice (numpy array) grâce à la commandes Image : `« from PIL import Image »`, après la conversation nous avons enregistré le résultat comme un fichier `« spectrograms.npy »`. ce fichier nous le divise en deux classes drone et bruit et les enchaîne :

```
« drone_labels=np.full (20, 0) »
```

```
« bruit_labels=np.full (14, 1) »
```

```
« spect_labels=np.concatenate ((drone_labels,bruit_labels) »
```

#### 3.6.2 Visualisation des images paires

Ensuite, on va visualiser les spectrogrammes qui existent dans le fichier `spectrograms.npy` à l'aide de la fonction qui génère les paires des images en utilisant une autre fonction appelée 'visualize' : `« def visualize (image_pairs, labels, n = 5, title = "Image Pair Examples"): """ Visualize a few pairs """`, cette fonction crée une figure contenant des sous-trames organisées en paires d'image. Chaque paire est extraite du paramètre 'image\_pairs' et affichée dans une sous-trame distincte. Le nombre des paires affichées est déterminée par le paramètre (n). Dans notre programme nous avons affiché les résultats (paires des images) en deux différents affichages Positifs et Négatifs.

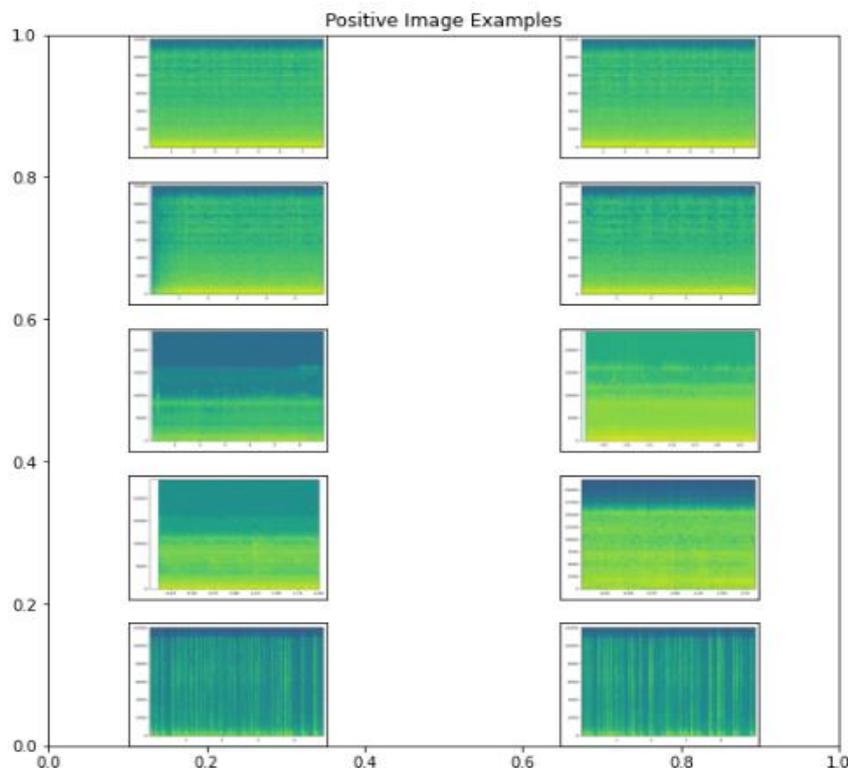
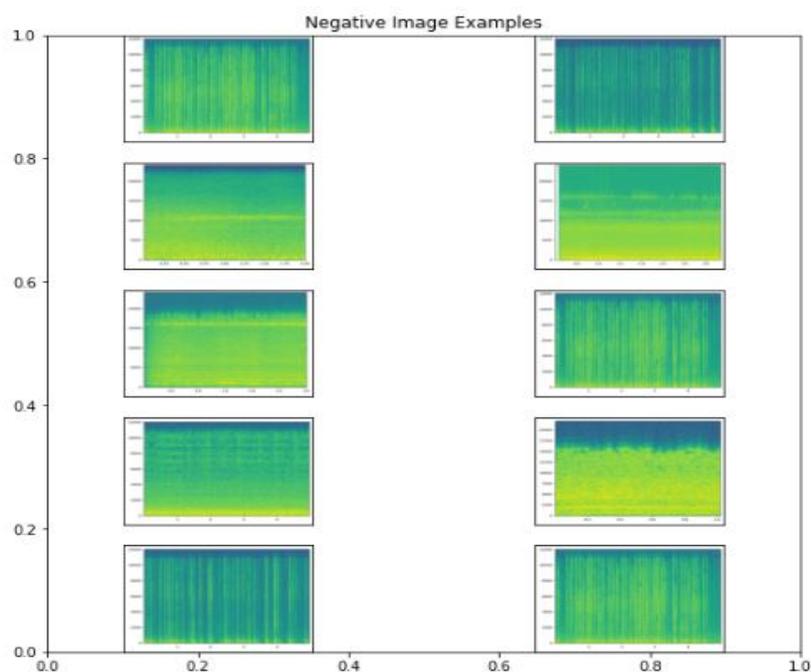


Figure 3. 11: Résultats de visualisation des images paires positive.



**Figure 3. 12:** Résultats de visualisation des images paires négative.

### 3.6.3. Création de réseau de neurone

Après l'affichage des paires nous allons créer notre réseau de neuronal convolutionnelles :

Layer Type	Kernels	Kernels Size	Kernels stride size	Padding	Rate	Activation
2D Convolutional	16	10×10	-	Same	-	Relu
2D Max pooling	-	2×2	2	-	-	-
Dropout	-	-	-	-	0.3	-
2D Convolutional	32	7×7	-	Same	-	Relu
2D Max pooling	-	2×2	2	-	-	-
Dropout	-	-	-	-	0.3	-
2D Convolutional	32	4×4	-	Same	-	Relu
2D Max pooling	-	2×2	2	-	-	-
Dropout	-	-	-	-	0.3	-
2D Convolutional	64	4×4	-	Same	-	Relu
Flatten	-	-	-	-	-	-
Dense	-	-	-	-	-	Relu
Dense	-	-	-	-	-	Sigmoid

**Tableau.3.1 :** Modèle de CNN

### 3.6.4 L'entraînement de réseau

Pour l'apprentissage de notre programme nous avons rédigé ce code :

```
siamese_model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
history = siamese_model.fit([images_dataset[:, 0, :], images_dataset[:, 1, :]], labels_dataset,
                             epochs=50, validation_split = 0.2, batch_size = 64)
```

Ce code concerne l'entraînement d'un modèle siamois utilisant les images d'un dataset pour effectuer une tâche de classification binaire. Voici une explication étape par étape :

#### a) Compilation du modèle Siamois :

Cette ligne compile le modèle siamois en spécifiant la fonction de perte utilisée (binary\_crossentropy), l'optimiseur (adam) et les métriques d'évaluation (accuracy). Le modèle est prêt à être entraîné, avec :

**Accuracy (Précision) :** L'accuracy est une mesure de performance utilisée dans le domaine de l'apprentissage automatique pour évaluer la précision d'un modèle de classification. Elle est calculée en comparant les prédictions du modèle aux véritables Étiquettes de classe des données d'évaluation. L'accuracy est exprimée sous forme de pourcentage, où une valeur de 100 % représente une classification parfaitement précise. Equation mathématique de l'Accuracy est :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total d'échantillons de données}} \dots \dots \dots (3)$$

**Loss (Perte) :** Le loss, également appelé fonction de perte ou fonction d'erreur, est une mesure qui évalue à quel point les prédictions d'un modèle sont éloignées des véritables valeurs cibles lors de l'entraînement. L'objectif est de minimiser cette perte afin que le modèle puisse produire des prédictions précises. Equation de la perte est :

$$\text{loss} = \frac{1}{N} \cdot \sum_1^N (y_{\text{true}} - y_{\text{pred}})^2 \dots \dots \dots (4)$$

Où N est le nombre total d'échantillons, y\_true représente les vraies valeurs cibles et y\_pred représente les prédictions du modèle.

#### b) Entraînement du modèle Siamois :

Cette ligne de code entraîne le modèle **Siamois** en utilisant les données d'entraînement images\_dataset et les étiquettes correspondantes labels\_dataset. Les données d'entraînement sont fournies sous forme de deux ensembles d'images, où images\_dataset[:, 0, :] représente le premier ensemble d'images et images\_dataset[:, 1, :] représente le deuxième ensemble d'images.

Les autres paramètres de l'entraînement sont les suivants :

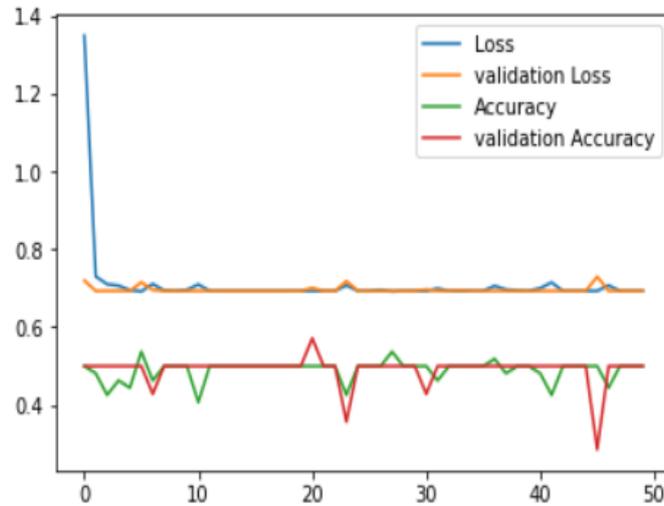
**Epochs=50 :** Nombre d'époques d'entraînement, c'est-à-dire le nombre de fois où le modèle parcourt l'ensemble de données complet.

## Chapitre 03 : Détection acoustique de drone

**validation\_split=0.2** : Fraction de données à utiliser comme ensemble de validation. Dans ce cas, 20% des données sont réservées à la validation pendant l'entraînement.

**batch\_size=64** : Taille du lot, c'est-à-dire le nombre d'exemples d'entraînement utilisés à la fois pour mettre à jour les poids du modèle.

c) **Illustration des différents graphes:** et comme de résultat nous avons trouvé :



**Figure 3. 13:** graphe de l'Accuraciy et Loss en fonction d'epochs

Pour régler les résultats et améliorer la précision de système. Nous avons décidé d'ajouter des données et les divise en plusieurs classes :

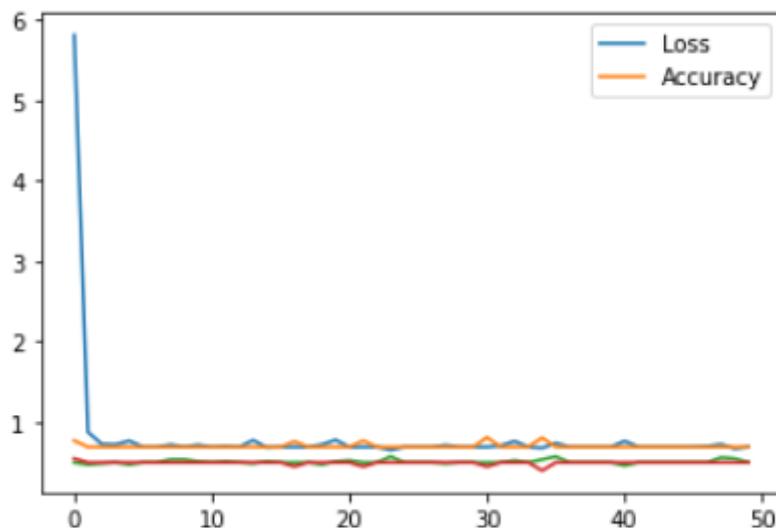
```
« drone_labels=np.full (10, 0) »
```

```
bruit1_labels=np.full (10, 1) »
```

```
« bruit2_labels=np.full (10, 2) »
```

```
« bruit3_labels=np.full (10, 3) »
```

Nous avons obtenu donc:



**Figure 3. 14:** graphe de l'Accuraciy et Loss en fonction d'epochs

## Chapitre 03 : Détection acoustique de drone

---

Malheureusement, malgré tous les efforts (Augmentation des classes, augmentation de la data, augmentation des epochs de l'apprentissage et le changement de l'architecture des Siamois), l'accuracy ne dépasse pas les 53% et le Loss ne diminue pas au-dessous de 72%.

A partir des résultats obtenus on a jugé de changer l'apprentissage et le type de structure de réseau neuronal.

### 3.6.5 Renforcement de la dataset

Nous avons donc décidé de réécrire un autre programme et de renforcer notre base de données de plus en plus. Dans cette fois-ci nous avons utilisé la bibliothèque PyTorch de Python et ses différentes commandes. En plus nous avons utilisé le modèle de CNN.

Un réseau de convolution neuronale (CNN) est un type de modèle de réseau neuronal utilisé principalement pour le traitement des données visuelles, tels que des images. Il est basé sur des opérations de convolution qui permettent de détecter des motifs et des caractéristiques spécifiques dans les données d'entrée. Les CNN ont été largement utilisés avec succès dans des domaines tels que la reconnaissance d'images, la détection d'objets et la segmentation sémantique.

#### Définition du PyTorch

PyTorch est une bibliothèque d'IA, développée par Meta (ex-Facebook), écrite en Python pour se lancer dans le deep learning (ou apprentissage profond) et le développement de réseaux de neurones artificiels. À partir de plusieurs variables, elle peut servir à réaliser des calculs de gradients ou à utiliser des tableaux

#### Comparaison PyTorch vs TensorFlow

PyTorch utilise un réseau de neurones dynamique tandis que TensorFlow s'adosse à une architecture de graph statique. C'est là l'une des principales différences entre les deux bibliothèques de deep learning. Résultat : il est beaucoup plus facile avec PyTorch de faire évoluer nativement les structures de réseau de neurones au fur et à mesure de la phase d'apprentissage. [34]

Parmi les commandes de PyTorch, nous avons utilisé :

```
"import torch.nn as nn"
"import torch.optim as optim "
"import torchaudio.transforms as transforms"
```

## Chapitre 03 : Détection acoustique de drone

**torch.optim:** Cette bibliothèque fait partie du framework de calcul tensoriel PyTorch et fournit des algorithmes d'optimisation pour entraîner des modèles d'apprentissage automatique.

**torchaudio:** Cette bibliothèque est spécifiquement conçue pour le traitement audio avec PyTorch. Elle fournit des fonctionnalités pour lire, écrire et manipuler des données audio, ainsi que des transformations audio courantes utilisées dans les tâches d'apprentissage automatique.

L'importation `import torchaudio.transforms as transforms` permet d'accéder aux transformations audio prédéfinies du module `transforms` de la bibliothèque **torchaudio**, offrant ainsi des fonctionnalités supplémentaires pour le traitement du signal audio dans le programme.

### 3.6.6 Création de réseau CNN

Une fois que nous avons importé les commandes de Pytorch, nous avons créé le modèle de CNN.

Partie du programme qui permet de créer le modèle CNN avec 11 couches cachées de 2 dimensions et 224 neurones (32 neurones dans la 1<sup>ère</sup> couche, 64 neurones dans la 4<sup>ème</sup> couche et 128 neurones dans la 9<sup>ème</sup> couche).

```
] # Define the CNN model
class AudioClassifier(nn.Module):
    def __init__(self, num_classes):
        super(AudioClassifier, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=(2, 4))
```

- **def \_\_init\_\_(self, num\_classes):** Il s'agit de la méthode d'initialisation de la classe `AudioClassifier`. Elle prend en argument `num_classes`, qui représente le nombre de classes que le modèle doit être capable de classifier.
- **Super (AudioClassifier, self).\_\_init\_\_ ():** Cette ligne appelle la méthode d'initialisation de la classe parente `nn.Module` pour s'assurer que l'initialisation de la classe de base est effectuée correctement.
- **self.conv1 = nn.Conv2d (1, 32, kernel\_size = (3, 3), stride = (1, 1)) :** Cela crée une couche de convolution 2D avec 1 canal d'entrée, 32 canaux de sortie, un noyau de taille (3, 3) et un pas de (1, 1). Cette couche sera utilisée pour extraire des caractéristiques à partir des données audio.
- **self.relu1 = nn.ReLU () :** Cela crée une couche d'activation ReLU qui sera appliquée après la couche de convolution pour introduire des non-linéarités dans les activations.

## Chapitre 03 : Détection acoustique de drone

- **self.maxpool1 = nn.MaxPool2d (kernel\_size= (2, 4))** : Cela crée une couche de pooling maximale 2D avec une fenêtre de pooling de taille (2, 4) Cette couche réduira la dimensionnalité des caractéristiques extraites par la couche de convolution.

Et pour la lecture des données nous avons rédigé ce code :

```
# Define a custom dataset
class AudioDataset(Dataset):
    def __init__(self, file_paths, labels, transform):
        self.file_paths = file_paths
        self.labels = labels
        self.transform = transform
```

Le code que nous avons fourni définit une classe personnalisée AudioDataset qui hérite de la classe Dataset du module torch.utils.data de PyTorch. Cette classe est utilisée pour créer un jeu de données personnalisé pour l'entraînement d'un modèle d'apprentissage automatique audio.

- **class AudioDataset(Dataset)** : Cela définit une nouvelle classe appelée AudioDataset qui hérite de la classe parente Dataset. Cette classe sera utilisée pour créer un jeu de données personnalisé.
- **def \_\_init\_\_(self, file\_paths, labels, transform)** : Il s'agit de la méthode d'initialisation de la classe AudioDataset. Elle prend en argument file\_paths, labels et transform. file\_paths est une liste de chemins de fichiers audio, labels est une liste d'étiquettes correspondant aux fichiers audio, et transform est une transformation audio que vous souhaitez appliquer aux données.
- **self.file\_paths = file\_paths** : Cela assigne la liste file\_paths à une variable d'instance self.file\_paths, qui sera utilisée pour accéder aux chemins des fichiers audio.
- **self.labels = labels** : Cela assigne la liste labels à une variable d'instance self.labels, qui sera utilisée pour accéder aux étiquettes des fichiers audio.
- **self.transform = transform** : Cela assigne la transformation audio transform à une variable d'instance self.transform, qui sera utilisée pour appliquer la transformation aux données audio.

### 3.6.7 Apprentissage de modèle

Nous sommes passé vers la partie d'entraînement, pour cela nous avons rédigé ce code et nous avons ajouté aussi l'optimisateur pour réduire les erreurs:

```
# Training loop
total_steps = len(dataloader)
loss_values = [] # List to store the loss values
accuracy_values = [] # List to store the accuracy values

for epoch in range(num_epochs):
    running_loss = 0.0
    correct_predictions = 0
    total_predictions = 0

    for i, (waveforms, labels) in enumerate(dataloader):
        waveforms = waveforms.to(device)
        labels = labels.to(device)

        # Forward pass
        outputs = model(waveforms)
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Le résultat après apprentissage est :

## Chapitre 03 : Détection acoustique de drone

---

```
Epoch [10/10], Step [10/366], Loss: 0.024958787485957146
Epoch [10/10], Step [20/366], Loss: 0.020714107900857925
Epoch [10/10], Step [30/366], Loss: 0.008376717567443848
Epoch [10/10], Step [40/366], Loss: 0.0691433921456337
Epoch [10/10], Step [50/366], Loss: 0.08597790449857712
Epoch [10/10], Step [60/366], Loss: 0.15894073247909546
Epoch [10/10], Step [70/366], Loss: 0.01135026291012764
Epoch [10/10], Step [80/366], Loss: 0.06570021063089371
Epoch [10/10], Step [90/366], Loss: 0.05721180513501167
Epoch [10/10], Step [100/366], Loss: 0.21843628585338593
Epoch [10/10], Step [110/366], Loss: 0.05482328310608864
Epoch [10/10], Step [120/366], Loss: 0.12477762252092361
Epoch [10/10], Step [130/366], Loss: 0.01714419387280941
Epoch [10/10], Step [140/366], Loss: 0.03680560737848282
Epoch [10/10], Step [150/366], Loss: 0.053912453353405
Epoch [10/10], Step [160/366], Loss: 0.020918194204568863
Epoch [10/10], Step [170/366], Loss: 0.054526057094335556
Epoch [10/10], Step [180/366], Loss: 0.15416045486927032
Epoch [10/10], Step [190/366], Loss: 0.06331051141023636
Epoch [10/10], Step [200/366], Loss: 0.05163770914077759
Epoch [10/10], Step [210/366], Loss: 0.031123140826821327
Epoch [10/10], Step [220/366], Loss: 0.04568178206682205
Epoch [10/10], Step [230/366], Loss: 0.012887794524431229
Epoch [10/10], Step [240/366], Loss: 0.03181494027376175
Epoch [10/10], Step [250/366], Loss: 0.006819005124270916
Epoch [10/10], Step [260/366], Loss: 0.13909204304218292
Epoch [10/10], Step [270/366], Loss: 0.0823800340294838
Epoch [10/10], Step [280/366], Loss: 0.05331781879067421
Epoch [10/10], Step [290/366], Loss: 0.02449415810406208
Epoch [10/10], Step [300/366], Loss: 0.1094408705830574
Epoch [10/10], Step [310/366], Loss: 0.08480871468782425
Epoch [10/10], Step [320/366], Loss: 0.024339133873581886
Epoch [10/10], Step [330/366], Loss: 0.011833331547677517
Epoch [10/10], Step [340/366], Loss: 0.026635171845555305
Epoch [10/10], Step [350/366], Loss: 0.002728909719735384
Epoch [10/10], Step [360/366], Loss: 0.029321158304810524
Epoch [10/10], Loss: 0.06905667186597657, Accuracy: 0.978298017771702
```

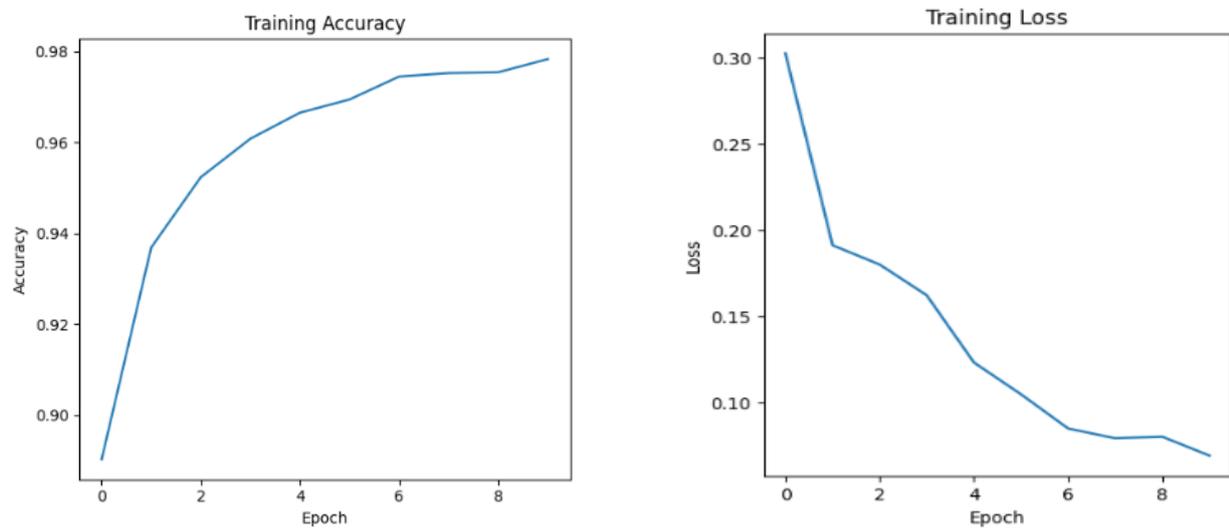
Une fois l'apprentissage est terminé, nous avons tracé l'Accuracy et loss en fonction des epochs:

```
# Plot the loss and accuracy curves
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(loss_values)
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("Training Loss")
plt.subplot(1, 2, 2)
plt.plot(accuracy_values)
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.title("Training Accuracy")

plt.tight_layout()
plt.show()
```

Donc nous avons obtenu :

**Figure 3. 15:** graphe de l'Accuracy et Loss en fonction d'epochs



## Chapitre 03 : Détection acoustique de drone

### 3.6.8 Partie de validation

La dernière étape est le test. Pour cela, nous avons fait deux différents tests:

- **Drone** : nous avons pris un audio quelconque d'un drone dans un dossier.

```
# Load and preprocess the audio file
waveform, sample_rate = torchaudio.load("/content/DroneAudioDataset/Multiclass_Drone_Audio/membo_1/Membo_0_000-membo_001.wav")
```

Le résultat obtenu est :

```
# Get the predicted class
_, predicted_class = torch.max(output.data, 1)

classes = ['drone' , 'unknown']
print(classes[predicted_class.item()])

drone
```

- **Bruit** : nous avons pris un audio d'une source sonore n'est pas celui d'un drone

```
# Load and preprocess the audio file
waveform, sample_rate = torchaudio.load("/content/DroneAudioDataset/Multiclass_Drone_Audio/unknown/1-100032-A-03.wav")
```

Le résultat est :

```
# Get the predicted class
_, predicted_class = torch.max(output.data, 1)

classes = ['drone' , 'unknown']
print(classes[predicted_class.item()])

unknown
```

Donc nous sommes arrivés aux résultats escompte, et avant constater que le programme fonctionne bel et bien selon nos espérances.

### 3.7 Conclusion

Ce chapitre a décrit les différentes étapes de simulation réalisées dans ce projet de fin d'étude en utilisant des réseaux de neurones convolutifs. Nous avons obtenu des résultats attendus avec une précision de 98%. Ces résultats démontrent l'efficacité de l'intelligence artificielle dans le domaine de détection acoustique.

En résumé, ce chapitre met en évidence l'importance des étapes de simulation et l'utilisation des réseaux de neurones convolutifs pour obtenir des résultats prometteurs. L'accuracy de 98% que nous avons atteints renforce notre confiance dans la validité de nos résultats pour les étapes ultérieures de notre recherche.

# **Conclusion Générale**

## **Conclusion générale :**

En conclusion, Le présent travail a apporté une contribution significative à la compréhension et l'application des drones, de l'intelligence artificielle et le traitement des signaux sonores.

Dans le premier chapitre, nous avons examiné en détail le domaine des drones, en retraçant leur évaluation historique, en mettant en évidence leurs caractéristiques et les différents types des drones qui excite sur le marché. De plus nous avons déclaré les différentes techniques qui sont déjà utilisé dans le cadre de notre travail, ce qui a permis d'acquérir une perceptive globale sur ce domaine en constante évolution.

Ensuite, nous sommes concentre sur l'intelligence artificielle, avec un accent particulier sur les réseaux de neurones. Nous avons approfondi notre connaissance sur les différentes architectures des réseaux de neurones, notamment les Convolutions Neural Network (CNN) et les réseaux de Siamois (Siemese Network). Nous avons examiné leurs fonctionnements, fiabilité, leurs capacités et leurs limitations. Cette étude approfondie nous a permis de choisir l'architecture appropriée dans notre travail de détection acoustique des drones.

Le dernier chapitre de ce mémoire a été concentre sur l'étude des caractéristiques de sons dans les domaines temporelle, fréquentiel et spectrale. En utilisant la modalité spectrale comme une base de notre travail, nous avons développé un système de classification des sons de drones et des bruits environnants en utilisant les réseaux de neurones convolutifs. Nous avons mis en œuvre ce système en utilisant le langage de programmation Python, en exploitant ses différentes bibliothèques spécialisées pour le traitement des signaux sonores et l'entraînement des réseaux de neurones.

Les résultats obtenus au cours de cette recherche démontrent clairement l'efficacité de réseaux de neurones spécifiquement CNN dans la classification de sons de drones transformés sous forme d'image grâce traçage du spectrogramme. Cette simulation ouvre la voie à des applications pratiques de surveillance et de sécurité tels que la détection autonome et la lutte contre l'utilisation illégale des drones. En permettant une détection plus rapide et plus précise des drones, l'intégration de l'intelligence artificielle dans les systèmes de drones peut améliorer la sécurité des opérations, prévenir les intrusions non autorisées et faciliter la prise de décisions en temps réel.

Cependant, il convient de souligner que cette recherche malgré les résultats encourageants il reste encore des opportunités de réalisation et d'amélioration. Par exemple, l'exploration d'autres architectures de réseaux de neurones et l'affinement des techniques de traitement du signal pourraient permettre d'atteindre des performances encore meilleures. De plus, l'expansion de cette recherche pour inclure un plus large éventail de sons de drones et de situations environnementales contribuerait à renforcer la robustesse et l'applicabilité du système de classification proposé.

**Référence**  
**Bibliographique**

## Chapitre 03 : Détection acoustique de drone

---

### **Bibliographie**

- [1] ALLOUACHE Mohamed Amine, «Commande par PID-flou d'un simulateur d'hélicoptère». Mémoire de Master option Systèmes Electroniques Complexes, Université de Bouira, 2017.
- [2] Conférence «Mieux connaitre les drones». <https://www.onera.fr/>
- [3] MARIA DE FATIMA BENTO, « Unmanned Aerial Vehicles: An overview ». 2010. <https://www.researchgate.net/>
- [4] «Drone civils grand public et commerciaux, Sylvian Labbe, IRSTEA, IP address : 78.192.112.105 HAL, archives-ouvertes.fr, consulte le 02/04/2022.
- [5] MOHAMMEDI & IRBAH, «Conception d'un auto pilote pour un drone léger type quadri-rotor». Mémoire de Master option Avionique, Université de Blida, 2019
- [6] DIARRA ZEGUE DIT MOUSSA, «Etude et réalisation d'un système de Commande embarqué pour le contrôle d'un UAV». Mémoire de Master option Electronique des systèmes embarqués, Université de Annaba, 2018
- [7] John K. Borchardt. «Unmanned aerial vehicles spur composites use». Reinforced plastics, 48(4). 28-31 Consulté le 2004 Elsevier.
- [8] Anibal OLLERO et Iván MAZA. «*Multiple heterogeneous unmanned aerial vehicles*». T. 37. Springer, 2007.
- [9] Site web <https://images.app.goo.gl/SiEz9po3AaPqkkN7>
- [10] Jawad.Y; Huma Zia «Drone and controller detection and localization: trends and challenges». Applied sciences, 2022, 12(24), 12612.
- [11] Site web <https://www.fhr.fraunhofer.de/en/businessunits/security/Detection-of-small-drones-with-millimeter-wave-radar.html>
- [12] Rozantsev, A.; Lepetit, V.; Fua, P. « Detecting flying objects using a single moving camera». IEEE Trans. 2017, 39, 879–892.
- [13] Müller, T. Robust. « Drone detection for day/night counter-UAV with static VIS and SWIR cameras». Proc. SPIE 2017, 10190, 302–313.
- [14] Site web <https://www.aerocontact.com/salon-aeronautique-virtuel/produit/370-systeme-doberservation-journuit-shapo-air>
- [15] BILAL TAHA ET ABDULHADI SHOUFAN, « Machine Learning-Based Drone Detection and Classification: State-Of-The-Art in Research», Septembre 23/2019 IEEE Access. <https://www.researchgate.net/>
- [16] Citation: Chiper, F.-L.; Martian, A.; Vladeanu, C.; Marghescu, I.; Craciunescu, R.; Fratu, O. « Drone Detection and Defense Systems: Survey and a Software-Defiend Radio-Based Solution». Sensors 2022, 22, 1453. <https://doi.org/10.3390/s22041453>.

## Chapitre 03 : Détection acoustique de drone

---

- [17] Steven Janny.Solal Nathan. «Introduction à l'apprentissage automatique». Université paris-Saclay. Consulté le 24/05/2022
- [18] A.Krizhevsky, I.Sutskever, G.E. Hinton. « Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems» Consulté en 2012. <https://www.proceeding.neurips/>
- [19]Gregory Gelly. «Réseaux de neurones récurrents pour le traitement automatique de la parole». Thèse de doctorat de l'université paris-Saclay option Informatique préparée à l'université paris-sud, 12/10/2017.
- [20] «Deep Learning 3 choses a savoir». <https://fr.mathworks.com/discovery/deep-learning.html>.
- [21] W. G. W CHRISTOPHER. SGERASIMOS. «flexible deep neural network structure with application to natural language processing». Department of knowledge engineering, Maastricht University, the Netherlands.
- [22] SITE WEB, <https://link.springer.com/article/>.
- [23] «Introduction aux réseaux de neurones – 3/3 : Apprentissage des réseaux de neurones». (15 novembre 2019). <https://blog.clevy.io/fr/>.
- [24] Z. ISMAILI. « Machine learning ou apprentissage automatique».
- [25]O.ZEDADRA. «système de prédiction de la consommation d'énergie basé deep learning». Mémoire de Master option Systèmes et technologie de l'informatique, université de guelma.2021.
- [26]Valentin .noel. «Séries temporelles et réseaux de neurones récurrents». Université paris-Saclay, Domaine Energie et information. Consulté le 09/2022. <https://eduscol.education.fr/>.
- [27]Georgios Nanos. «How Siamese Network Work in image Recognition» Consulté le 02/05/2023 <https://baeldung.com/>.
- [28] HABIBOULLAH, Beha Dine et LADJEDEL, Billal. «Utilisation des réseaux de neurones artificiels pour la prédiction de la vitesse de vent». Mémoire de Master, Electronique : Université Mohamed Boudiaf M'SILA, 2017.
- [29] Alan MCKINNEY et Grant HARRIS une bibliothèque d'effets sonores et de musiquelibres de droits <https://www.zapsplat.com/>
- [30] Site web Free WAV to MP3 Converter <https://www.Convertio.com/>
- [31] David Cassagne, «Transformation de Fourier, FFT et DFT». <https://www.courspython.com/> , Consulté le 22/10/2022.
- [32] Georgio Nanos, «How do Siamese networks work in image Recognition?».<https://www.Baeldung.com/> , Consulté le 2/05/2023.
- [33] Bastien L, «TensorFlow: tout savoir sur la bibliothèque Machine Learning open source». <https://www.lebigdata.fr/> , Consulté le 19/10/2018.

## Chapitre 03 : Détection acoustique de drone

---

[34]Antoine Crochet-Damais, «Pytorch: tout savoir sur la bibliothèque de deep learning». <https://www.journaldunet.fr/>, Consulté le 21/10/2022.