

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

Amélioration des performances d'un
détecteur de drone optique

Proposé et dirigé par :

Dr. Cheggaga Nawal

Dr. Messaoudene Kamel

Réalisé par :

Mr Haddaoui Abdelaziz

Soutenue devant le jure composer de :

Dr. Dilmie Smain

MCA

Président

Dr. Krime Mohamed

MCB

Examineur

Promotion: 2022 / 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Nom Prenom, Chair

Institute of Aeronautics and Spaces studies

Dr. Nom Prenom

Institute of Aeronautics and spaces studies

Dr. Nom Prenom

Institute of Aeronautics and Spaces studies

[Click here to enter text.](#)

Add or Delete Committee Member

Add or Delete Department

Approved by:

Dr. FORM 9 THESIS FORM HEAD NAME HERE

Dédicaces

Mon très cher père, l'homme qui m'a donnée le désir d'apprendre et le savoir-vivre.

A ma très chère mère, ma fierté et mon bonheur à qui je souhaite une très longue vie.

Tous mes amis.

Tous ceux qui m'aiment et ceux qu'on aime.

Remerciements

Je remercie Dieu, le Tout-Puissant, de m'avoir donné la bravoure, la patience, la volonté et la force essentielle pour surmonter tous les difficultés et obstacles qui se sont présentes à moi au cours de mes années d'études.

Je voudrais exprimer mes remerciements a « Cheggaga Nawal » pour la confiance qu'elle m'a accordée en acceptant de diriger ce travail et de me guider tout au long de la thèse pour leurs conseils, leurs critiques et leur attitude exemplaire tout au long de la réalisation de ce travail de recherche.

Je tiens à exprimer ma plus profonde gratitude et mes sincères remerciements à tous ceux qui ont contribué de près ou de loin à l'avancement de nos travaux de recherche.

Je remercie également tous les membres du jury qui ont voulu évaluer ce travail.

Table des matières :

Table des matières	6
Table des figures	9
Liste des abréviations	10
RESUME	11
Mots-clés	11
ABSTRACT	12
Keywords.....	12
INTRODUCTION	14
CHAPITRE 1 : RESEAUX DE NEURONES CONVOLUTIFS (CNN)	
1.1. Introduction	17
1.2. Définition	17
1.3. Architecture CNN	18
A. Architecture globale	18
B. Couche de convolution	19
C. Couche de regroupement	22
D. Couches entièrement connectées	22
1.4. Apprentissage par transfert	23
1.4.1. Définition	23
1.4.2. Catégorisation des techniques d'apprentissage par transfert	23
1.4.3. Applications d'apprentissage par transfert	26
1.5. Conclusion	26
CHAPITRE 2: MATERIELS ET METHODES	
2.1. Introduction	29
2.2. Les axes de mouvement d'un aéronef	29
→ L'axe latéral (tangage)	29
→ L'axe longitudinal (Roll)	30
→ L'axe vertical (lacet)	30
2.3. Les mouvements d'un drone	30

2.3.1.	Définition	30
2.3.2.	Les mouvements qu'un drone multi-rotor peut effectuer	31
2.3	The pinhole camera model	32
2.4.	Les capteurs numériques d'image	32
	Glossaire de base	33
2.4.1.	Principe de fonctionnement des capteurs d'image	34
2.4.2.	Choix du capteur	35
2.5.	Format d'image	36
2.5.1.	Format JPG ou JPEG	36
2.5.2.	Format PNG	37
2.5.3.	Format TIFF	37
2.6.	Transformations géométriques	38
2.6.1.	Homographie dite transformation de perspective	39
2.7.	Outils utilisé	41
2.7.1.	Google Colab	41
2.7.2.	Kaggle	42
2.7.3.	Pixels	42
2.7.3.	Photopea	42
2.8.	Conclusion.....	42

CHAPITRE 3: DETECTION DE DRONE SANS TRANSFORMATION DE PERSPECTIVE

3.1.	Introduction	45
3.2.	Description du programme	45
3.3.	Description du modèle	46
3.4.	Entraînement du modèle	48
3.5.	Description du data	49
3.6.	Description des résultats	50
3.7.	Conclusion	50

CHAPITRE 4: DETECTION DE DRONE AVEC TRANSFORMATION DE PERSPECTIVE

4.1.	Introduction	53
------	--------------------	----

4.2.	Description du programme	53
4.3.	Compilation et Entraînement du modèle	54
4.4.	Fonction de transformation de perspective	55
4.5.	Test du modèle	56
4.6.	Description du data	57
4.7.	Description des résultats	57
4.8.	Conclusion	58
	CONCLUSION GENERALE	60
	References	63

Table des figures:

Figure 1-1 une architecture CNN simple, composée de seulement cinq couches [7].....	18
Figure 1-2 activations extraites de la première couche convolutive d'un CNN profond simpliste, après apprentissage sur la base de données manuscrites du MINST.[7].....	20
Figure 1-3 réseau neutre à action directe (FNN) simple à trois couches, composé d'une couche d'entrée, d'une couche cachée et d'une couche de sortie. [7].....	23
Figure 1-4 Un aperçu des différents paramètres de transfert [10]	26
Figure 2-1 les axes de mouvement d'un aéronef [27].....	29
Figure 2-2 Différentes directions de mouvement possible d'un drone [28]	31
Figure 2-3 Comment faire un mouvement de lacet. [29]	32
Figure 2-4 Pinhole camera projection visualisation [3]	32
Figure 2-5 Projection summary [3]	32
Figure 2-6 Les Capteurs Photo CCD et CMOS [30].....	33
Figure 2-7 Schéma de composition capteur d'image. [18]	35
Figure 2-8 Exemple de distorsion perspective : à gauche, l'image originale, à droite, après correction. [20].....	38
Figure 3-1 Résultat de classification sans transformation de perspective.....	50
Figure 4-1 Résultat de classification avec transformation de perspective	58

Liste des abréviations :

ANN	Réseaux de neurones artificiel
CCD	Charge Coupled Device
CMJV	cyan, magenta, jaune, vert (CMYG)
CMOS	Complimentary Metal Oxyde Semiconductor
CNN	Réseaux de neurones convolutifs
CONV	Couche de convolution
Convnets	Réseaux de neurones convolutifs
FNN	réseau de neurones à anticipation
LiDAR	détection et de mesure de la lumière
ReLu	L'unité linéaire rectifiée
RGB	Rouge Vert Bleu
RNN	réseaux de neurones récurrents
TAL	traitement du langage naturel
TL	Transfer Learning
UAS	systèmes aériens sans pilote
UAV	Les véhicules aériens sans pilote

RESUME

Dans cette recherche, nous proposons une nouvelle approche pour améliorer l'efficacité de la détection des drones. Nous introduisons une technique de prétraitement d'image qui utilise des réseaux neuronaux de Transfer Learning (TL), qui se sont révélés efficaces dans diverses tâches de vision par ordinateur. En exploitant les connaissances préalablement acquises à partir de vastes ensembles de données d'images, le TL permet au réseau de s'adapter rapidement et d'apprendre des caractéristiques pertinentes spécifiques à la détection des drones. Cette approche améliore considérablement la précision et la vitesse de détection, la rendant adaptée aux scénarios de surveillance en temps réel. De plus, nous incorporons une séquence de correction de lacet-tangage-roulis dans le pipeline de prétraitement d'image. Les drones peuvent adopter différentes orientations et angles pendant leur vol, ce qui peut affecter leur représentation visuelle dans les images capturées. En appliquant une correction de lacet-tangage-roulis, nous alignons l'orientation du drone, normalisant ainsi l'image et réduisant les erreurs potentielles de détection causées par les perspectives variables. En combinant la puissance des réseaux neuronaux de Transfer Learning avec la séquence de correction de lacet-tangage-roulis, notre méthode proposée vise à atteindre une plus grande efficacité, précision et fiabilité de détection dans les systèmes de détection optique des drones. Les résultats de nos expériences démontrent l'efficacité de cette approche et son potentiel pour améliorer les capacités de surveillance dans des environnements et applications divers.

Mots-clés:

Drones, imagerie, CNN, Transfer Learning, lacet-tangage-roulis

ABSTRACT

In this research, we propose a novel approach to enhance the efficiency of drone detection. We introduce an image pre-processing technique that utilizes Transfer Learning (TL) neural networks, which have proven to be effective in various computer vision tasks. By leveraging the pre-trained knowledge from large-scale image datasets, TL enables the network to quickly adapt and learn relevant features specific to drone detection. This approach significantly improves the accuracy and speed of detection, making it suitable for real-time surveillance scenarios. Additionally, we incorporate a yaw-pitch-roll correction sequence as part of the image pre-processing pipeline. Drones can exhibit various orientations and angles during flight, which can affect their visual representation in captured images. By applying a yaw-pitch-roll correction, we align the drone's orientation, thereby normalizing the image and reducing potential detection errors caused by varying perspectives. By combining the power of Transfer Learning neural networks with the yaw-pitch-roll correction sequence, our proposed method aims to achieve higher detection efficiency, accuracy, and reliability in optical drone detection systems. The results of our experiments demonstrate the effectiveness of this approach and its potential to enhance surveillance capabilities in diverse environments and applications.

Keywords:

Drones, imaging, CNN, Transfer Learning, yaw-pitch-roll

INTRODUCTION

Introduction:

La croissance du nombre de systèmes aériens sans pilote (UAS) de petite taille, ou "drones", est assortie d'une gamme significative d'avantages prometteurs liés à leur utilisation. Cependant, ces avantages ne pourront être pleinement réalisés que s'il est possible d'avoir confiance dans l'exploitation sécurisée de tels UAS. Les rapports faisant état d'une augmentation du nombre d'incidents de sécurité ont légitimement suscité des inquiétudes non seulement au sein de la communauté de l'aviation habitée, mais aussi parmi les fabricants et les exploitants pionniers de drones, ainsi que le grand public [1]. Les véhicules aériens sans pilote sont déjà devenus une partie intégrante des services militaires, de sécurité et de sauvetage. Ils sont couramment utilisés même à des fins de loisirs. Cependant, en raison d'une législation en retard, les UAV n'ont pas encore officiellement rejoint l'espace aérien commun des véhicules pilotés et ne peuvent donc toujours pas être utilisés de manière plus large à des fins commerciales, par exemple par des entreprises logistiques.

Les équipes de recherche scientifique, les étudiants, les professionnels ainsi que les passionnés amateurs font progresser ce domaine de l'aviation. Des groupes d'ingénieurs aéronautiques, de programmeurs, d'opérateurs et d'utilisateurs de UAV qualifiés (ou moins qualifiés) se développent rapidement dans le monde entier. Comme il a été analysé puis déclaré, "Le monde de l'aviation sans pilote est un monde d'innovation très impatient". Cela ouvre de nouvelles possibilités de développement scientifique. En raison des avantages indéniables, les technologies des UAV sont largement utilisées à des fins militaires. Outre une utilisation militaire régulière, elles sont de plus en plus utilisées par des organisations paramilitaires et guérillas (criminelles). Il convient de noter que l'exploitation négative des caractéristiques uniques des UAV ne se limite pas aux groupes criminels. Les personnes qui cherchent à prendre un selfie original, les journalistes à sensation perturbant les célébrités ou les individus capturant des vidéos d'adrénaline extrême peuvent mettre en danger, par exemple, les procédures de décollage des avions. Et même ici, dans le domaine de la protection et de la défense contre de telles activités exploratoires et envahissantes des UAV, il existe un large éventail de disciplines scientifiques qui pourraient être stimulées par ce phénomène [2].

Le nombre de drones disponibles commercialement augmente, tout comme le risque de leur mauvais usage. Les systèmes de lutte contre les drones constituent un besoin émergent pour détecter et éliminer les drones malveillants ou tout type d'UAV qui menace la sécurité

publique ou la vie privée individuelle. Les technologies de détection, de localisation et d'identification des petits UAV comprennent des capteurs infrarouges, des dispositifs laser, des aides et dispositifs de surveillance optique, des dispositifs acoustiques, des capteurs de détection et de mesure de la lumière (LiDAR), des équipements utilisant la technologie de reconnaissance d'image que nous développerons dans cet article, des dispositifs capables de détecter et de localiser les signaux de télécommande des UAV et les observateurs aériens humains. Une fois qu'un drone cible est détecté, des méthodes d'élimination telles que des armes laser, des canons à eau, des oiseaux entraînés à attraper des drones et des brouilleurs peuvent être appliquées [3].

L'approche traditionnelle de la reconnaissance d'images consiste à entraîner un modèle sur un ensemble de données d'images capturées à partir de points de vue et d'orientations spécifiques. Cependant, cela limite la capacité du modèle à généraliser vers de nouvelles perspectives non vues auparavant. Dans des situations réelles, les drones peuvent apparaître sous différentes orientations et perspectives, ce qui peut poser un défi pour une détection précise à l'aide des techniques traditionnelles de reconnaissance d'images. Notre contribution vise à exploiter la transformation de perspective pour combler l'écart entre la diversité des apparences des drones dans le monde réel et les données d'entraînement limitées capturées à partir de points de vue spécifiques. En alignant les images de drones sur une perspective standardisée, nous pouvons améliorer la capacité du modèle à détecter les drones dans différentes orientations et améliorer son applicabilité dans le monde réel.

Le travail est composé de quatre chapitres organisés comme suite :

- Chapitre 1 : Le chapitre présente un bref sur les réseaux de neurone convolutif.
- Chapitre 2 : Ce chapitre est dédié à la description des Matériels et méthodes
- Chapitre 3 : Le troisième chapitre est destiné aux résultats de simulation de détection de drone sans application de transformation géométrique.
- Chapitre 4 : Le quatrième chapitre est destiné aux résultats de simulation de détection de drone avec application de transformation de perspective.

Ce travail est achevé par une conclusion générale.

CHAPITRE 1 : RESEAUX DE NEURONES CONVOLUTIFS

(CNN)

1.1.Introduction :

Depuis leur introduction par LeCun et al. au début des années 1990, Les réseaux neuronaux convolutifs (convnets) ont démontré d'excellentes performances dans des tâches telles que la classification manuscrite des chiffres et la détection des visages. Au cours des dernières années, plusieurs papiers ont montré qu'ils peuvent également offrir des performances exceptionnelles sur des tâches de classification visuelle plus difficiles.

Montrent des performances record sur la classification ImageNet 2012 référence, avec leur modèle convnet atteignant un taux d'erreur de 16,4 %, par rapport à la 2e place de 26,1 %. Suite à ces travaux, Girshick et al. ont montré des performances de détection de premier plan sur l'ensemble de données PASCAL VOC. Plusieurs facteurs sont responsables de cette amélioration spectaculaire des performances : (i) la disponibilité d'ensembles de formation beaucoup plus grands, avec des millions d'exemples étiquetés ; (ii) implémentations GPU puissantes, rendant pratique la formation de très grands modèles et (iii) de meilleures stratégies de régularisation des modèles, telles que Dropout. Malgré ces progrès encourageants, il y a encore peu d'informations sur le fonctionnement interne le fonctionnement et le comportement de ces modèles complexes, ou comment ils obtiennent d'aussi bonnes performances. [4]

1.2.Définition :

Les réseaux neuronaux convolutifs (CNN) sont un type de réseau neuronal artificiel qui nécessite une couche convolutive. [5] qui comprend des calculs de convolution et a une structure profonde. C'est l'un des algorithmes représentatifs de l'apprentissage en profondeur [6] et peut également avoir d'autres types de couches, telles que des couches non-linéaires, de mise en commun et entièrement connectées, pour créer un réseau de neurones à convolution profonde qui s'auto-optimisent par l'apprentissage [5]. Selon l'application, CNN peut être bénéfique. Cependant, il apporte des paramètres supplémentaires pour la formation où les filtres convolutifs sont formés en utilisant la méthode de rétropropagation. Les formes de la structure de filtre dépendent de la tâche donnée par exemple, dans une application telle que la détection de visage, un filtre peut effectuer une extraction de bord, tandis qu'un autre peut effectuer une extraction d'œil. Cependant, nous ne contrôlons pas entièrement ces filtres

dans CNN, et leurs valeurs sont déterminées par apprentissage [5]. Chaque neurone recevra toujours une entrée et effectuera une opération (comme un produit scalaire suivi d'une fonction non-linéaire) - la base d'innombrables ANN [7]. Des vecteurs d'image bruts d'entrée à la sortie finale du score de classe, l'ensemble du réseau exprimera toujours une seule fonction de score perceptif (le poids). La dernière couche contiendra les fonctions de perte associées aux classes, et tous les trucs et astuces réguliers développés pour les ANN traditionnels s'appliquent toujours [7]. La seule différence notable entre les CNN et les ANN traditionnels est que les CNN sont principalement utilisés dans le domaine de la reconnaissance de formes dans les images. Cela nous permet d'encoder des fonctionnalités spécifiques à l'image dans l'architecture, ce qui rend les réseaux de neurones convolutifs plus adaptés aux tâches axées sur l'image [7]. Cette section présente brièvement le CNN.

1.3. Architecture CNN :

A. Architecture globale :

Les CNN sont composés de trois types de couches. Ce sont des couches convolutionnelles, couches de regroupement et couches entièrement connectées. Lorsque ces couches sont empilées, l'architecture CNN a été formée. Une architecture CNN simplifiée pour MNIST classement est illustrée à la figure 1.1.

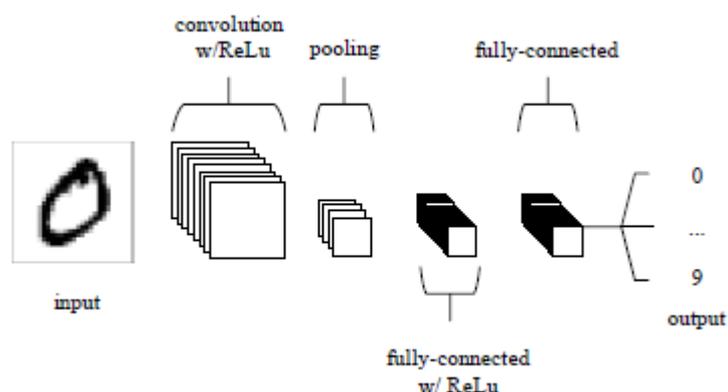


Figure 1-1 une architecture CNN simple, composée de seulement cinq couches [7]

La fonctionnalité de base de l'exemple CNN ci-dessus peut être décomposée en quatre domaines clé.

1. Comme dans d'autres formes d'ANN, la couche d'entrée contiendra les valeurs de pixel de l'image.
2. La couche convolutive déterminera la sortie des neurones dont connecté aux régions locales de l'entrée par le calcul du scalaire produit entre leurs poids et la région reliée au volume d'entrée. L'unité linéaire rectifiée (couramment abrégée en ReLu) vise à appliquer une fonction d'activation "élémentaire" telle que sigmoïde à la sortie de l'activation produite par la couche précédente.
3. La couche de regroupement effectuera alors simplement un sous-échantillonnage le long de la dimensionnalité de l'entrée donnée, réduisant encore le nombre de paramètres au sein de cette activation.
4. Les couches entièrement connectées effectueront alors les mêmes tâches que celles trouvées dans ANN standard et tenter de produire des scores de classe à partir des activations, à utiliser pour le classement. Il est également suggéré que ReLu puisse être utilisé entre ces couches, afin d'améliorer les performances. Grâce à cette simple méthode de transformation, les CNN sont capables de transformer l'entrée d'origine couche par couche à l'aide de techniques de convolution et de sous-échantillonnage pour produire des scores de classe à des fins de classification et de régression. Cependant, il est important de noter que la simple compréhension de l'architecture globale d'une architecture CNN ne suffira pas. La création et l'optimisation de ces modèles peut prendre un certain temps et peut être assez déroutant. Nous allons explorer maintenant en détail les couches individuelles, en détaillant leurs hyperparamètres et les connectivités. [7]

B. Couche de convolution :

La couche CONV est la couche centrale pour la construction d'un réseau neuronal convolutive, qui génère la plupart des calculs dans le réseau. La fonction de la couche convolutive est d'extraire les caractéristiques des données d'entrée [6] où, dans la couche convolutive, plusieurs filtres glissent sur la couche pour les données d'entrée données [5]. Chaque neurone d'une couche convolutive n'est connecté qu'à une petite région du volume d'entrée, cette région est communément appelée la taille du champ récepteur du neurone [7]. Les paramètres des couches se concentrent sur l'utilisation de noyaux 'apprenantes'. Ces noyaux sont généralement petits en dimensionnalité spatiale, mais

s'étendent sur toute la profondeur de l'entrée [7]. Lorsque le noyau de convolution fonctionne, il analyse régulièrement les caractéristiques d'entrée, effectue la multiplication des éléments de la matrice et la sommation des caractéristiques d'entrée dans le champ récepteur et superpose la valeur de l'écart :

$$Z^{l+1}(i, j) = [Z^l \otimes \omega^{l+1}](i, j) + b = \sum_{k=1}^{k_l} \sum_{x=1}^f \sum_{y=1}^f [Z_k^l(s_0 i + s_0 j + y) \omega_k^{l+1}(x, y)] + b \quad (1.1)$$

$(i, j) \in \{0, 1, \dots, L_{l+1}\}$

$$L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \quad (1.2)$$

La formule ci-dessus prend un noyau de convolution bidimensionnel comme exemple. La méthode de travail d'un noyau de convolution unidimensionnel ou tridimensionnel est similaire à la méthode de l'exemple précédant [6]. Chaque noyau aura une carte d'activation correspondante. Ces cartes d'activation peuvent être visualisées, comme le montre la Figure 1.2, qui seront empilées le long de la dimension de profondeur pour former le volume de sortie complet de la couche convolutive. [7].

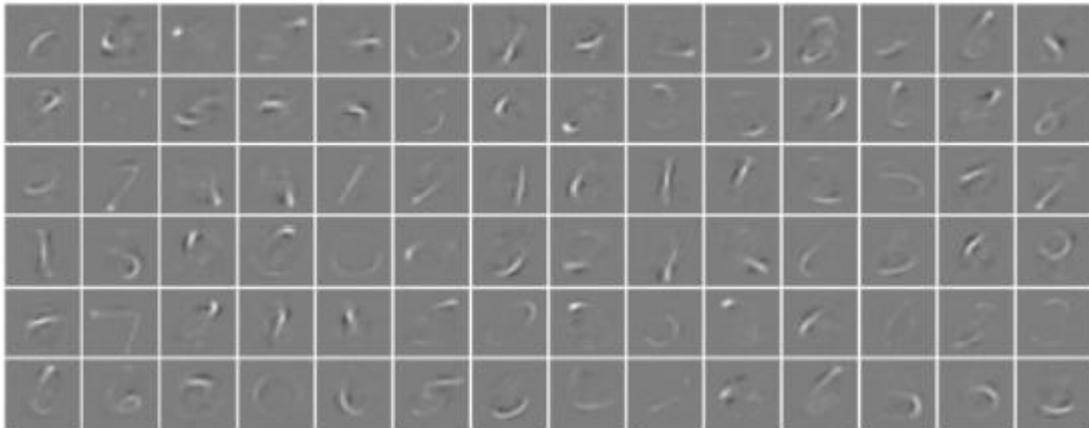


Figure 1-2 activations extraites de la première couche convolutive d'un CNN profond simpliste, après apprentissage sur la base de données manuscrites du MNIST.[7]

À partir de là, le réseau apprendra les noyaux qui "se déclenchent" lorsqu'ils voient une caractéristique spécifique à une position spatiale donnée de l'entrée. Celles-ci sont communément appelées activations [7].

Les paramètres de la couche CONV incluent la taille du noyau de convolution, la taille du pas et le remplissage. Les trois paramètres qui sont les hyperparamètres du CNN déterminent ensemble la taille de la carte des caractéristiques de sortie de la couche convolutive. La taille du noyau de convolution peut être spécifiée comme n'importe quelle valeur inférieure à la taille de l'image d'entrée. Plus le noyau de convolution est grand, plus les caractéristiques d'entrée qui peuvent être extraites sont complexes [6].

Par exemple, si l'entrée du réseau est une image de taille $64 \times 64 \times 3$ (une image de couleur RVB image avec une dimensionnalité de 64×64 et nous définissons le champ réceptif taille comme 6×6 , nous aurions un total de 108 poids sur chaque neurone dans la couche convolutive. $6 \times 6 \times 3$ où 3 est l'ampleur de la connectivité à travers la profondeur du volume). La profondeur du volume de sortie produit par les couches convolutionnelles peut être définie manuellement le nombre de neurones dans la couche à la même région de l'entrée [7].

- La longueur du pas de convolution définit la distance entre la position du noyau de convolution lorsqu'il parcourt deux fois la carte des caractéristiques [6]. Par exemple, le pas est 1 signifie que nous faisons glisser le filtre d'un endroit vers la droite à chaque fois, puis calculons la sortie [5].

- Le rembourrage peut être divisé en quatre catégories selon le nombre de couches et le but :

- Rembourrage valide : aucun rembourrage n'est utilisé et le noyau de convolution n'est autorisé à accéder qu'à la position de la carte de caractéristiques qui contient le champ réceptif complet. Tous les pixels de la sortie sont fonction du même nombre de pixels dans l'entrée.

- Rembourrage identique/demi : ne faites que suffisamment de rembourrage pour conserver la même taille des cartes de caractéristiques de la sortie et de l'entrée, si la valeur du rembourrage est de zéro, cela peut être appelé un rembourrage nul.

- Rembourrage complet : faites suffisamment de rembourrage pour que chaque pixel soit accessible le même nombre de fois dans chaque direction

- Rembourrage arbitraire : entre un rembourrage valide et un rembourrage complet [6]

zéro-padding est le processus simple de rembourrage de la bordure de l'entrée et constitue une méthode efficace pour donner un contrôle supplémentaire sur la dimensionnalité

des volumes de sortie [7]. Le remplissage à zéro vise principalement à inclure les données au bord de la matrice d'entrée. Sans remplissage de zéro, la sortie de convolution est donc plus petite que l'entrée ; la taille du réseau diminue en ayant plusieurs couches de convolutions qui signifient la perte de données précieuses, ce qui limite le nombre de couches convolutives dans un réseau. Cependant, le remplissage zéro empêche le rétrécissement des réseaux et fournit des couches profondes illimitées dans notre architecture réseau [5].

Il est important de comprendre qu'en utilisant ces techniques, nous modifierons la dimensionnalité spatiale de la sortie des couches convolutionnelles. Pour le calculer, vous pouvez utiliser la formule suivante :

$$\frac{(V-R)+2Z}{S+1} \quad (1.3)$$

Où V représente la taille du volume d'entrée (height_width_depth), R représente la taille du champ réceptif, Z est la quantité de jeu de remplissage zéro et S fait référence à la foulée. Si le résultat calculé à partir de cette équation n'est pas égal à un entier, la foulée a été mal définie, car les neurones ne pourront pas s'adapter parfaitement à l'entrée donnée [7].

C. Couche de regroupement :

La couche de regroupement est périodiquement insérée entre des couches convolutionnelles successives. Sa fonction est de réduire progressivement la taille spatiale du volume de données, de sorte que le nombre de paramètres dans le réseau puisse être réduit, par conséquent, il réduit la consommation de ressources informatiques et peut contrôler efficacement le sur-ajustement [6]. La méthode de mise en commun la plus populaire, la mise en commun maximale représente la valeur maximale à l'intérieur du filtre de mise en commun (2×2) en tant que sortie [5]. En raison de la nature destructrice de la couche de mise en commun, il n'y a que deux méthodes généralement observées de mise en commun maximale. Habituellement, la foulée et les filtres des couches de regroupement sont tous deux définis sur 2×2 , ce qui permettra à la couche de s'étendre sur l'intégralité de la dimensionnalité spatiale de l'entrée. En outre, la mise en commun par chevauchement peut être utilisée, où la foulée est définie sur 2 avec une taille de noyau définie sur 3. En raison de

la nature destructrice de la mise en commun, le fait d'avoir une taille de noyau supérieure à 3 réduira généralement les performances du modèle considérablement [7].

D. Couches entièrement connectées :

La couche entièrement connectée contient des neurones qui sont directement connectés aux neurones des deux couches adjacentes, sans être reliés à aucune couche en eux. Ceci est analogue à la façon dont les neurones sont disposés de manière traditionnelle formes d'ANN (figure 1.3) [7].

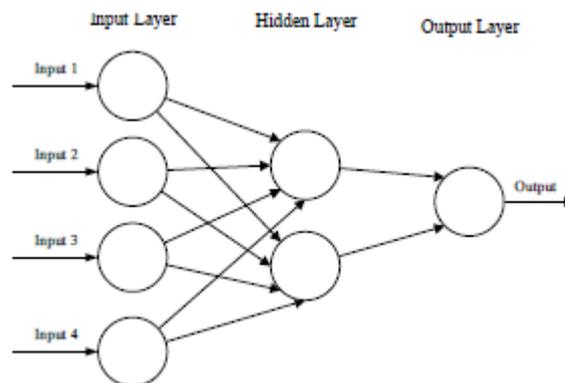


Figure 1-3 réseau neutre à action directe (FNN) simple à trois couches, composé d'une couche d'entrée, d'une couche cachée et d'une couche de sortie. [7]

1.4.Apprentissage par transfert :

1.4.1. Définition :

L'apprentissage par transfert est une technique d'apprentissage automatique dans laquelle un modèle est formé et développé pour une tâche, puis réutilisé pour une deuxième tâche connexe. Il fait référence à la situation dans laquelle ce qui a été appris dans un contexte est exploité pour améliorer l'optimisation dans un autre contexte. L'apprentissage par transfert est généralement appliqué lorsqu'il existe un nouveau ensemble de données plus petit que l'ensemble de données d'origine utilisé pour former le modèle préformé [8].

Dans l'apprentissage en profondeur, le modèle s'entraîne avec un grand volume de données et apprend le poids et le biais du modèle pendant l'entraînement. Ces poids sont transférés à d'autres modèles de réseau pour les tests [9].

1.4.2. Catégorisation des techniques d'apprentissage par transfert:

Dans l'apprentissage par transfert, nous avons les trois principales recherches suivantes problèmes : (1) quoi transférer ; (2) comment transférer ; (3) quand transférer. « Quoi transférer » demande quelle partie de la connaissance peut être transféré entre domaines ou tâches. Après découvrir quelles connaissances peuvent être transférées, apprendre des algorithmes doit être développés pour transférer les connaissances, qui correspondent à la problématique « comment transférer ». « Quand transférer » demande dans quelles situations, transférer les compétences doit être faites. De même, nous sommes intéressés à savoir dans quelles situations, les connaissances ne doivent pas être transférées. Dans certaines situations, lorsque le domaine source et le domaine cible sont non liés les uns aux autres, le transfert par force brute peut échouer. Dans le pire des cas, cela peut même nuire aux performances d'apprentissage dans le domaine cible, une situation souvent appelé transfert négatif. Les travaux les plus récents sur le transfert l'apprentissage se concentre sur "Quoi transférer" et "Comment transférer", en supposant implicitement que les domaines source et cible soient liées les unes aux autres. Cependant, comment éviter le transfert négatif est une question ouverte importante qui attire de plus en plus l'attention à l'avenir.

Nous catégorisons l'apprentissage par transfert dans trois sous-environnements, l'apprentissage par transfert inductif, l'apprentissage par transfert transductif et l'apprentissage non supervisé apprentissage par transfert, basé sur des situations différentes entre les domaines et tâches source et cible. [10]

1) Dans le cadre de l'apprentissage par transfert inductif, la tâche cible est différente de la tâche source, quel que soit le moment où les domaines source et cible sont identiques ou non. Dans ce cas, certaines données étiquetées du domaine cible sont nécessaires pour induire un modèle prédictif objectif $f_T(\cdot)$ pour une utilisation dans le domaine cible. De plus, selon différentes situations de données étiquetées et non étiquetées dans le domaine source, nous pouvons encore catégoriser le domaine inductif transférer le cadre d'apprentissage en deux cas : (1.1) Beaucoup de données étiquetées dans le domaine source sont disponible. Dans ce

cas, l'apprentissage par transfert inductif est similaire au paramètre d'apprentissage multitâche. Cependant, le paramètre d'apprentissage par transfert inductif vise à atteindre des performances élevées dans la tâche cible en transférant les connaissances de la tâche source tout en l'apprentissage multitâches essaie d'apprendre la cible et la source tâche simultanément.

(1.2) Aucune donnée étiquetée dans le domaine source n'est disponible. Dans ce cas, le paramètre d'apprentissage par transfert inductif est similaire au cadre d'apprentissage autodidacte, qui est proposé pour la première fois par Raina et al. En autodidacte paramètre d'apprentissage, les espaces d'étiquette entre la source et les domaines cible peuvent être différents, ce qui implique les informations annexes du domaine source ne peuvent pas être utilisé directement. Ainsi, il est similaire au transfert inductif paramètre d'apprentissage où les données étiquetées dans la source domaine ne sont pas disponibles. [10]

2) Dans le cadre de l'apprentissage par transfert transductif, la source et les tâches cible sont les mêmes, tandis que la source et la cible les domaines sont différents. Dans cette situation, aucune donnée étiquetée dans le domaine cible sont disponible alors que beaucoup de données étiquetées dans la source domaine sont disponibles. De plus, selon différentes situations entre les domaines source et cible, nous peut catégoriser davantage l'apprentissage par transfert transductif mise en deux cases.

(2.1) Les espaces de caractéristiques entre la source et la cible les domaines sont différents $X_S \neq X_T$.

(2.2) Les espaces de caractéristiques entre les domaines sont les mêmes, $X_S = X_T$, mais les distributions de probabilité marginales de les données d'entrée sont différentes, $P(X_S) \neq P(X_T)$.

Le dernier cas de l'apprentissage par transfert transductif le cadre est lié à l'adaptation du domaine pour la connaissance transfert dans la classification de texte et sélection d'échantillons biais ou décalage covariable, dont les hypothèses sont similaire. [10]

3) Enfin, dans le cadre de l'apprentissage par transfert non supervisé, similaire à l'apprentissage par transfert inductif, la cible la tâche est différente mais liée à la tâche source. Cependant, l'apprentissage par transfert non supervisé se concentre sur résoudre des tâches d'apprentissage non supervisé dans le domaine cible, tels que le regroupement,

la réduction de la dimensionnalité et la densité estimation. Dans ce cas, il n'y a pas d'étiquette donnée disponibles dans les domaines source et cible dans entraînement. [10]

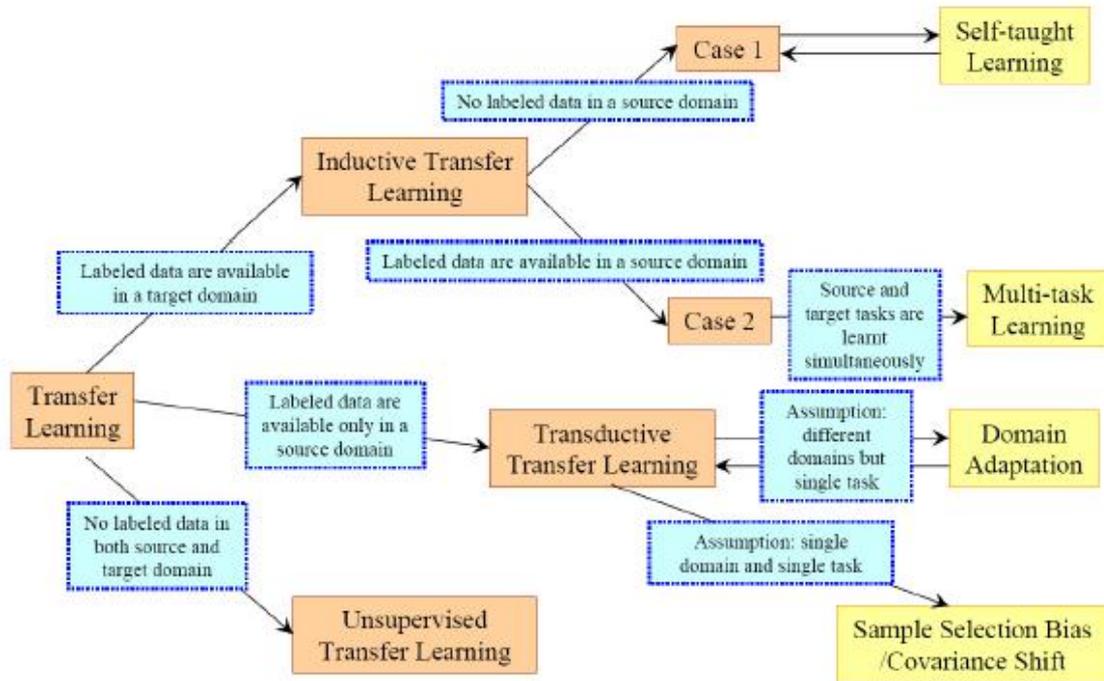


Figure 1-4 Un aperçu des différents paramètres de transfert [10]

1.4.3. Applications d'apprentissage par transfert:

L'examen montre que des structures importantes ont été utilisées pour transférer l'apprentissage. La classification des documents, la catégorisation des textes multilingues, la classification des émotions et la détection des spams ne sont que quelques-unes des nombreuses utilisations du traitement du langage naturel (TAL). La classification des films, photos, papiers et autres artefacts fait partie de ces procédures. La catégorisation de la fatigue musculaire, la catégorisation de l'emplacement Wi-Fi, la catégorisation des actions humaines, la catégorisation de l'efficacité pharmaceutique, la catégorisation des défauts de la machine et la catégorisation de l'arythmie cardiaque sont quelques-unes des applications. [11]

1.5. Conclusion :

Même si la théorie derrière les réseaux de neurones est vieille de près de 80 ans, on a réellement commencé à constater leur performance ces 10 dernières années grâce au concours organisé par ImageNet. Ces performances ont été possibles grâce aux capacités de stockage de plus en plus grandes des serveurs puisque les modèles nécessitent beaucoup de données pour être performants.

Les réseaux de convolution sont de puissantes méthodes de l'apprentissage en profondeur permettant de réaliser du traitement d'images. Ils sont utilisés notamment en médecine pour la détection de cancer sur des IRM et des radios mais également pour la conduite de véhicule autonome et dans notre cas pour la détection de drone.

CHAPITRE 2: MATERIELS ET METHODES

2.1. Introduction :

L'orientation d'un système de coordonnées par rapport à un autre est définie par une séquence de trois rotations successives. Le lacet, tangage et roulis qui définissent une telle séquence de rotation dans un repère lié à l'espace. Il existe plusieurs façons de définir cette séquence.

Dans les résultats d'une analyse de mouvement, les angles de lacet, tangage et roulis représentent la séquence des angles de rotation des axes Z-, Y- négatif et X mesurés selon le système de coordonnées pivotant par rapport au système de coordonnées global.

2.2. Les axes de mouvement d'un aéronef :

Quel que soit le type de l'aéronef soit a voilure fixe comme les avions ou tournant comme les hélicoptères les axes de mouvement sont définie par :

- l'axe de tangage
- l'axe de roulis
- l'axe de lacet

Un avion se déplaçant dans l'espace, sa capacité de mouvement autour de son centre de gravité se décrit autour de ses 3 axes [12].

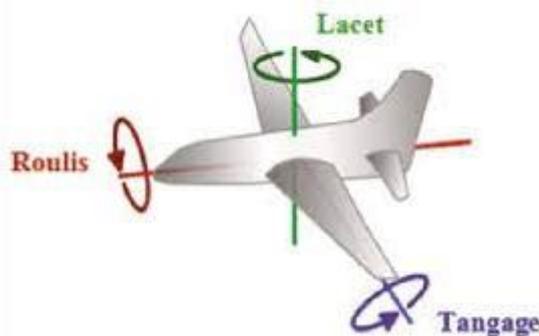


Figure 2-1 les axes de mouvement d'un aéronef [27]

→ **L'axe latéral (tangage)**

L'axe latéral va du bout de l'aile au bout de l'aile. L'avion tangue autour de cet axe. [13] Le tangage est obtenu lors de rotation autour de l'axe de tangage ce qui permet de monter ou de descendre le nez de l'appareil et va ainsi orienter la trajectoire de l'avion vers le haut ou le bas. [12]

→ **L'axe longitudinal (Roll)**

L'axe longitudinal va du nez de l'avion à la queue. C'est l'axe autour duquel s'incline l'aéronef. [13] Le roulis est obtenu lors d'une rotation autour de l'axe de roulis ce qui permet d'incliner les ailes à droite ou à gauche. Cette manœuvre contribue à la mise en virage de l'avion. [12]

→ **L'axe vertical (lacet)**

L'axe vertical est légèrement différent des autres, passant verticalement par le centre de l'avion. L'avion embarde autour de cet axe. [13] Le lacet est obtenu lors d'une rotation autour de cet axe ce qui permet de contrôler l'avion lors de sa course au décollage ou à l'atterrissage et de maintenir un vol symétrique en croisière et en virage. [12]

2.3. Les mouvements d'un drone :

2.3.1. Définition :

Un drone ou Unmanned Aerial Vehicle (UAV) est un aéronef sans passager ni pilote qui peut voler de façon autonome ou être contrôlé à distance depuis le sol. Le mot « drone » est une extrapolation d'un terme anglais qui signifie « faux-bourdon ». En français, le terme est employé pour désigner des véhicules aériens, terrestres, de surface ou sous-marins, alors que la classification anglo-saxonne distingue chaque type d'appareil.

La taille d'un drone aérien peut aller de quelques centimètres pour les modèles miniatures à plusieurs mètres pour les drones spécialisés (surveillance, renseignement, combat, transport, loisirs). L'autonomie en vol va de quelques minutes à plus de 40 heures pour les drones de longue endurance.

Il existe 4 types de drone aérien sont : multi-rotor, à voilure fixe, à rotor unique, VTOL hybride donc on va se focalisée sur les drones multi-rotor. [14]

2.3.2. Les mouvements qu'un drone multi-rotor peut effectuer :

Le drone est un engin extrêmement agile qui peut aller dans huit directions différentes : en avant, en arrière, à gauche, à droite et en diagonale (Avant-Gauche, Avant-Droit, Arrière-Gauche, Arrière-Droit) ; sans oublier les hauteurs. Les mouvements qu'il effectuera se feront via la vitesse des différents rotors. Ainsi on distingue 4 types de mouvements différents :

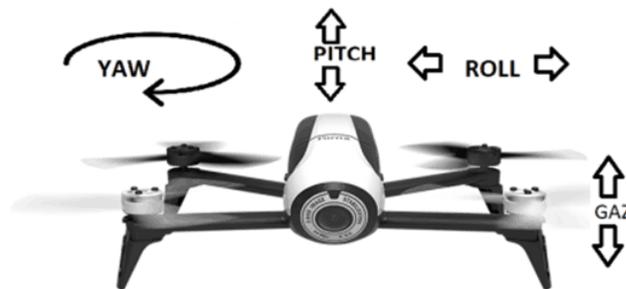


Figure 2-2 Différentes directions de mouvement possible d'un drone [28]

Gaz ou Throttle (monter ou descendre) : ce mouvement s'effectue en augmentant la vitesse des quatre rotors pour monter ou en la diminuant pour descendre.

Tangage ou Pitch (avancer ou reculer) : ce mouvement s'effectue en augmentant la vitesse des rotors A et B de Derrière et en diminuant celle des rotors de Devant, ainsi le drone avancera. Pour le faire reculer, on augmentera donc la vitesse des rotors de Devant et on diminuera celle des rotors de Derrière.

Roulis ou Roll (voler vers la gauche ou vers la droite) : même principe que pour le Pitch mais avec les rotors des côtés. Ainsi on augmente la vitesse des rotors de gauche et on diminue celle des rotors de droite pour aller vers la droite et vice-versa.

Lacet ou Yaw (vol circulaire) : pour faire voler le drone de façon circulaire dans le sens des aiguilles d'une montre on diminuera la vitesse des rotors B (paire de moteurs sur le même axe) et on augmentera la vitesse des rotors A pour rester à la même altitude et on fait le contraire si on souhaite le faire voler dans l'autre sens. [15]

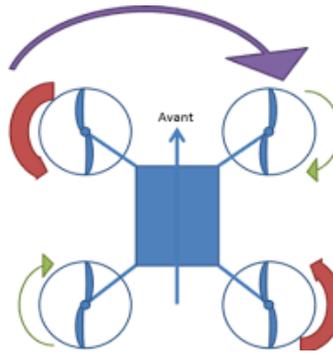


Figure 2-3 Comment faire un mouvement de lacet. [29]

2.3 The pinhole camera model :

Le modèle de caméra sténopé (pinhole camera model) définit la relation géométrique entre un point 3D dans la scène et sa projection 2D correspondante sur le plan de l'image. Cette cartographie géométrique de la 3D à la 2D est une projection en perspective. La figure 2 montre une vue schématique de la projection de la pinhole camera. [3]

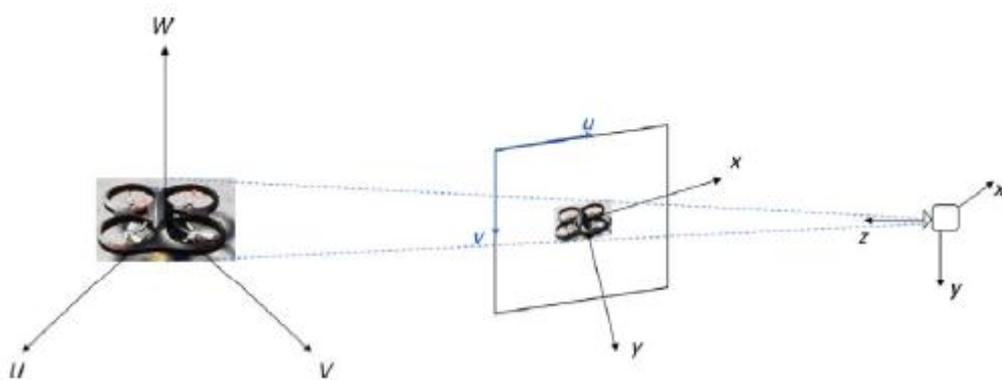


Figure 2-4 Pinhole camera projection visualisation [3]



Figure 2-5 Projection summary [3]

2.4. Les capteurs numériques d'image :

Les capteurs numériques d'image sont omniprésents dans notre vie de tous les jours : appareil photographique, caméras de surveillance, téléphones portables (quoique...). Les capteurs d'images représentent plus de 40% des ventes de composants optoélectroniques, ces ventes étant principalement composées de capteurs CMOS dans les appareils photos basiques des téléphones portables, et de CCD pour les appareils photos numériques, les caméras de surveillance. [16]

Les capteurs CCD sont utilisés dans les caméras de surveillance depuis plus de vingt ans. Ils présentent de nombreux avantages qualitatifs comme une meilleure sensibilité à la lumière que les capteurs CMOS. On obtient ainsi une meilleure qualité d'image en cas de sous-exposition. [17]

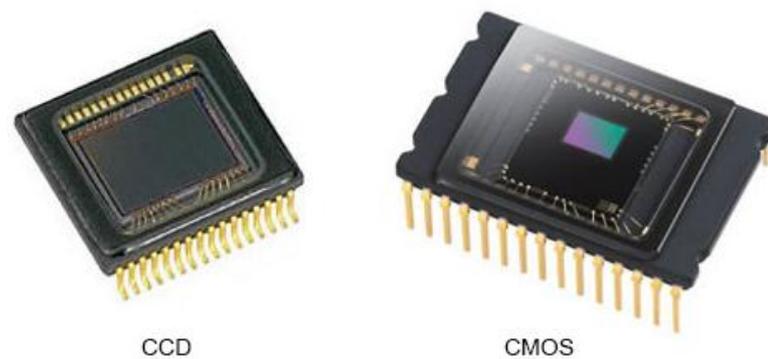


Figure 2-6 Les Capteurs Photo CCD et CMOS [30]

Le but des capteurs d'images est de retranscrire, le plus fidèlement possible, l'image d'un objet éclairé, ou d'une source lumineuse, formée à leur surface par un système optique adéquat. Nous nous attacherons principalement à décrire les deux grandes familles de capteurs optico-électroniques utilisés de nos jours : les capteurs CCD et les capteurs CMOS :

- CCD («Charge Coupled Device»), qui réalise les opérations de Collection, Transfert et Conversion de la charge électrique générée par les photons incidents.

- CMOS («Complimentary Metal Oxyde Semiconductor»), qui réalise les opérations de Collection et Conversion sur le site de collection de la charge générée par les photons incidents. [16]

Glossaire de base : [16]

- Pixel : picture element, taille de quelques μm , jointifs ou non
- Pitch : distance interpixel, varie selon le format du capteur
- Format : line, interline, field, frame, full frame
- Résolution spatiale : liée au nombre de pixels et à la taille du pixel, La résolution spatiale est fonction de la dimension du plus petit élément qu'il est possible de détecter
- Résolution temporelle : liée à la fréquence d'acquisition en image (fr) par seconde
- Résolution sensibilité : liée au semi-conducteur et à la numérisation en niveau de gris
- Binning : groupement de pixel
- Taux de transfert (en pix/s, ou en octets/s): dépend de la résolution et du format vitesse de transfert d'un pixel
- Sensibilité : à la quantité de lumière et à sa couleur, en fonction du semi-conducteur, du type de dopage, et du revêtement
- Bruit : tout signal ne participant pas à la transcription électrique de l'image désirée

2.4.1. **Principe de fonctionnement des capteurs d'image :**

Un capteur d'image est généralement composé de multiples petites photodiodes convertissant la lumière en un signal électrique. Lorsque la lumière frappe la surface du dispositif et est focalisée sur chaque photodiode par une microlentille, le capteur d'image la convertit en un signal électrique avant de former l'image finale. Les capteurs d'image ont la capacité d'acquérir des données relatives à l'intensité lumineuse mais sont incapables de reproduire seuls les couleurs. Afin d'obtenir une image en couleurs, un filtre couleur, généralement RGB (RVB) (red, green, blue : rouge, vert, bleu), est placé devant les photodiodes. Il est également possible d'utiliser un filtre CMYG (CMJV) (cyan, magenta, yellow, green : cyan, magenta, jaune, vert), bien que le filtre RGB (RVB) offre une meilleure reproduction des couleurs et soit plus adapté à la capture numérique. [18]

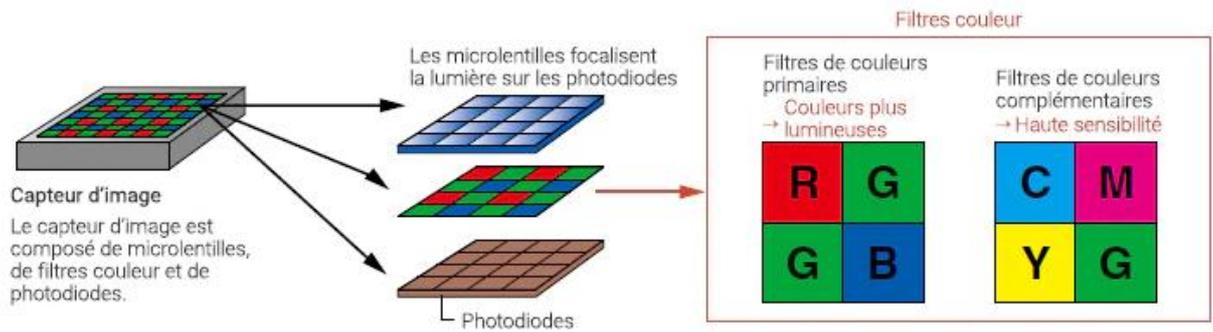


Figure 2-7 Schéma de composition capteur d'image. [18]

2.4.2. Choix du capteur :

Typiquement, outre les considérations de prix pour 1 mégapixels, on s'orientera principalement en fonction de cinq critères :

a. **Résolution spatiale :**

Une résolution accrue permet de mieux distinguer les détails de la scène, mais parfois au détriment de la vitesse de transfert. Une caméra dont on choisit 200 x 100 pixels, donc 20000pixels, doit envoyer 20000 valeurs numériques au système d'acquisition. Si le système opère à 25MHz, cela correspond à 40ns par valeur, soit 0,0008s de temps pour une image complète, ce qui équivaut à 1250fr/s. Si on passe en 640 x 480 pixels, soit environ 15 fois plus de pixels, on réduit la fréquence d'acquisition à environ 80fr/s.

b. **Vitesse et Obturation**

Lors du choix du capteur, il est indispensable de connaître la vitesse de déplacement de l'objet à capturer. Ce déplacement peut être nul ou quasi-stationnaire, ou bien rapide, voire ultra rapide. En fonction de la vitesse de déplacement de l'objet, il faudra peut-être privilégier des temps d'obturation et des vitesses d'acquisition appropriées. Le temps d'obturation permet de figer l'objet en mouvement, par exemple une petite particule en déplacement rapide. Tandis que la vitesse d'acquisition permet d'obtenir plusieurs images figées de la particule dans le même champ de vision. Ceci est particulièrement nécessaire pour des applications de corrélation ou de tracking. On peut également réussir ce genre de capture par des éclairages appropriés (laser pulsé, stroboscope) synchronisés à la caméra. En dernier recours, la longueur floue du trajet

de la particule (très rapide) pendant le temps d'obturation permet également de remonter à sa vitesse en une seule image.

c. Cadence

La cadence d'une caméra définit la capacité d'enregistrer des images complètes à la suite en un temps donné. Elle s'exprime typiquement en image (frame) par seconde : fr/s. Cette cadence dépend non seulement de la taille de la matrice, mais surtout de son architecture. De plus, la cadence dépend également du mode de connexion de la caméra au système d'acquisition du signal, et du fait qu'elle possède ou pas de la mémoire embarquée. On peut, par exemple, à format identique, transférer 640 x 480 pixels à 50 fr/s (mode CCIR) ou à 200fr/s (IEEE1394b), voire plusieurs kfr/s en mémoire embarquée ou en liaison SATA.

d. Réponse spectrale

La réponse spectrale dépend principalement du type de semiconducteur et du traitement du capteur (capteur éclairé de face ou par l'arrière, aminci ou pas aminci). Typiquement, tous les capteurs ont une réponse comprise entre 400nm et un peu moins de 1000nm. Toutefois, il faut toujours tenir compte qu'une caméra à une réponse donnée à un gain donné, et qu'une réponse élevée obtenue à un fort gain peut très bien générer un bruit très élevé, réduire la dynamique et ainsi dégradé la qualité de l'image.

e. Profondeur du puits

Lors de la numérisation du signal de la charge, la capacité à puits plein est discrétisée en niveaux de gris. Cette profondeur du puits peut être codée sur 8 voire sur 16 bits ou plus. Le niveau 0 correspond à 100% de noir, et le niveau 255 correspond à 100% de clair, et équivalent pour 10 bits (1024 niveaux) et 32 bits (4096 niveaux). Ce poids du pixel affecte bien entendu la quantité d'information à transférer (en Mbit/s) et la bande passante du système de transfert devra en tenir compte. [16]

2.5. Format d'image :

Comme toutes les images s'affichent de la même manière sur l'écran de notre ordinateur (ou sur le papier que nous avons imprimé), nous pouvons avoir l'impression qu'elles sont toutes identiques. Mais nous avons en fait plusieurs formats à notre disposition, chacun ayant ses propres forces et faiblesses. Connaître chacun d'entre eux nous aidera à prendre les bonnes décisions et à obtenir un résultat final de qualité [19].

2.5.1. Format JPG ou JPEG :

Ce sont les fichiers d'image les plus courants que nous pouvons trouver. Le format de compression JPG est vraiment efficace, mais doit être utilisé avec précaution, car il s'agit d'un format "avec perte", ce qui signifie que nous serons inévitablement confrontés à une certaine dégradation de l'image pendant le processus de compression. Pour faire simple, l'algorithme divise l'image en petits blocs qui sont soumis à un filtrage à haute fréquence. En conséquence, la taille du fichier image diminue considérablement, mais en entraînant la perte de détails fins. Si l'on a besoin de réduire la taille d'images très grandes, avec une texture riche (notamment des photographies), la compression JPG convient bien. Cependant, l'algorithme peut ne pas fonctionner aussi bien avec des gradients vraiment doux, rendant ainsi les blocs de compression perceptibles.

Il est important de noter que chaque fois que nous exportons une image en tant que fichier JPG, nous pouvons choisir le taux de compression, sur une échelle allant de 0 (taille de fichier plus petite, plus grandes distorsions d'image) à 100 (taille de fichier plus grande, plus petites distorsions d'image).

Indiqué pour : Photographies, images de grande taille, si vous avez besoin de fichiers de petite taille.

A éviter si : Données scientifiques, images avec des détails fins ou du texte [19].

2.5.2. Format PNG :

Ce format offre quelques avantages par rapport à la norme JPG, principalement la possibilité d'utiliser la transparence et la compression sans perte. La transparence est possible grâce à un canal supplémentaire qui contrôle l'opacité des pixels (appelé canal alpha). Bien sûr, cela se fait au prix d'une augmentation de la taille du fichier, mais ce n'est pas un problème pour les images de petite ou moyenne taille.

Indiqué pour : Les images avec transparence (canal alpha), les images de haute qualité.

À éviter si : Images de très grande taille (photographies) [19].

2.5.3. Format TIFF :

Dans la majorité des cas, il s'agit du format de fichier de choix pour la manipulation d'images scientifiques, car il peut fonctionner sans compression (ou compression sans perte), prendre en charge les documents multipages ainsi que toutes les métadonnées intégrées dont vous pouvez avoir besoin. Les fichiers Tiff peuvent également traiter des images de 16 bits (valeurs comprises entre 0 et 65535) au lieu des 8 bits auxquels sont limités les JPG ou PNG (valeurs comprises entre 0 et 255 pour chaque canal). Ceci est essentiel lorsque l'on travaille avec des images de microscopie : les caméras supportent généralement une gamme dynamique supérieure à 8 bits, ce qui signifie que l'on pourrait perdre des informations en utilisant le PNG (malgré une compression sans perte). Le fait que ce format supporte des documents de plusieurs pages permet de stocker facilement des piles Z de microscopie dans un seul fichier (en fait, vous pouvez stocker des données à n dimensions).

Indiqué pour : Les données scientifiques, les cas où vous avez plus de 8 bits par canal.

À éviter si : Images pour les présentations ou l'affichage sur le Web [19].

2.6. Transformations géométriques :

Lorsque l'on prend une image, on peut avoir des défauts de translation, de rotation, d'inclinaison, de perspective, qui sont des déformations simples et faciles à corriger. [20]

Une fois qu'on a construit la scène, on désire généralement de manipuler ses objets, par le changement de leurs attributs qui peuvent être la position, l'orientation, la taille, la forme,... [21]



Figure 0-8 Exemple de distorsion perspective : à gauche, l'image originale, à droite, après correction. [20]

L'idée de la correction est de trouver, pour chaque pixel (x,y) de l'image but, son homologue sur l'image source, de coordonnées (r,s), et de donner au pixel (x,y) de l'image but la valeur du pixel homologue de l'image source [20]:

$$f(x,y) = g(r,s) \quad (2.1)$$

2.6.1. Homographie dite transformation de perspective:

La méthode de transformation par homographie a été utilisée pour rectifier une image en perspective, par exemple pour générer une vue « en plan » d'un bâtiment à partir d'une photo « en perspective ». L'équation de transformation peut être définie Eq (2.2) pour x,y système source à partir de X,Y cible pour dans ce type de processus l'homographie (transformation projective /projectivité / colinéation). Cette transformation est appelée transformation de perspective (ou homographie plan à plan). Une homographie est représentée par **H** une matrice homogène 3*3. Les points sur le plan image, x sont mappés sur des points sur le plan monde, x' , comme $x' = H \cdot x$, où x est une colonne homogène 3-vecteur $x = (x, y, 1)^T$ avec (x, y) position euclidienne sur le plan. La matrice d'homographie a huit degrés de liberté - il y a neuf éléments de matrice, mais l'échelle globale n'est pas significative. Une ligne **l** est également représentée par un 3-vecteur colonne homogène, tel que si un point x se trouve sur **l** alors $l^T x = 0$. [22]

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.2)$$

Mettant $h_{33} = 1$

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11} \cdot x + h_{12} \cdot y + h_{13}}{h_{31} \cdot x + h_{32} \cdot y + 1}, y' = \frac{x'_2}{x'_3} = \frac{h_{21} \cdot x + h_{22} \cdot y + h_{23}}{h_{31} \cdot x + h_{32} \cdot y + 1} \quad (2.3)$$

Si l'éq. (2.3) multiplier par le dénominateur, puis réorganiser comme Eq(2.4) [22]

$$\begin{aligned} h_{11} \cdot x + h_{12} \cdot y + h_{13} &= x' \cdot (h_{31} \cdot x + h_{32} \cdot y + 1) \\ h_{21} \cdot x + h_{22} \cdot y + h_{23} &= y' \cdot (h_{31} \cdot x + h_{32} \cdot y + 1) \end{aligned} \quad (2.4)$$

$$\begin{aligned} h_{11} \cdot x + h_{12} \cdot y + h_{13} - h_{31} \cdot x \cdot x' + h_{32} \cdot y \cdot x' &= x' \\ h_{21} \cdot x + h_{22} \cdot y + h_{23} - h_{31} \cdot x \cdot y' + h_{32} \cdot y \cdot y' &= y' \end{aligned} \quad (2.5)$$

Où x' - y' sont les coordonnées à calculer dans le second référentiel (cible), étant donné les coordonnées x - y dans le premier référentiel (source) en fonction de 8 paramètres de transformation

h_{11} , h_{12} , h_{13} , h_{21} , h_{22} , h_{23} , h_{31} , h_{32} . Donc, ayant ces 8 inconnues (2 échelle, 2 rotation, 2 translation, 2 ligne à l'infini), il faut au moins 4 points connus dans les deux systèmes. Si le nombre de points est supérieur à 4, il peut être réalisé par les moindres carrés. Le système matriciel d'équations de transformation peut être défini comme; [22]

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 \cdot x'_1 & -y_1 \cdot x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 \cdot y'_1 & -y_1 \cdot y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 \cdot x'_2 & -y_2 \cdot x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2 \cdot y'_2 & -y_2 \cdot y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3 \cdot x'_3 & -y_3 \cdot x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3 \cdot y'_3 & -y_3 \cdot y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4 \cdot x'_4 & -y_4 \cdot x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4 \cdot y'_4 & -y_4 \cdot y'_4 \\ \vdots & \vdots \end{pmatrix} \cdot \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \\ \vdots \end{pmatrix} + \begin{pmatrix} v_{x'_1} \\ v_{y'_1} \\ v_{x'_2} \\ v_{y'_2} \\ v_{x'_3} \\ v_{y'_3} \\ v_{x'_4} \\ v_{y'_4} \\ \vdots \end{pmatrix} \quad (2.6)$$

$$A \cdot X = L + v \quad (2.7)$$

Ou :

h_{11} = facteur d'échelle fixe dans la direction X avec l'échelle Y inchangée.

h_{12} = facteur d'échelle dans la direction X proportionnel à la distance Y de l'origine.

h_{13} = translation d'origine dans la direction X.

h_{21} = facteur d'échelle dans la direction Y proportionnel à la distance X de l'origine.

h_{22} = facteur d'échelle fixe dans la direction Y avec l'échelle X inchangée.

h_{23} = translation d'origine dans la direction Y.

h_{31} = facteurs d'échelle proportionnels X et Y en fonction de X.

h_{32} = facteurs d'échelle proportionnels X et Y en fonction de Y.

Les coefficients inconnus des modèles avec leurs informations de covariance sont simplement déterminés selon le principe des moindres carrés qui minimise la somme des carrés des résidus v_i , comme suit avec les poids égaux. [22]

$$X = (A^T \cdot A)^{-1} \cdot (A^T \cdot L), v = A \cdot X - L. \sigma_0^2 = \frac{(v^T \cdot v)}{2n - r} \quad (2.8)$$

L est le vecteur d'observation, qui se compose de coordonnées dans le système de référence cible, X est la valeur estimée du vecteur des coefficients de transformation inconnus

$(h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})$, A est la matrice de conception, v est le vecteur résiduel. Une fois calculés, ces 8 paramètres peuvent facilement être utilisés pour transformer n'importe quel point du premier système de référence au second. σ_0^2 est le poids unitaire de la variance, q est le nombre de points communs utilisés pour la transformation, r est le nombre de paramètre inconnu (nombre de coefficient de transformation). L'homographie induite par un plan est unique à un facteur d'échelle près et est déterminée par 8 paramètres ou degrés de liberté. L'homographie dépend des paramètres intrinsèques et extrinsèques des caméras utilisées pour les deux vues et des paramètres du plan 3D. Pour obtenir la meilleure précision de rectification, utilisez en outre un nombre adéquat (plus de quatre) de points de contrôle avec une précision acceptable (au niveau du centimètre ou mieux). [22]

2.7. Outils utilisé :

2.7.1. Google Colab :

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU. Jupyter est le projet Open Source sur lequel Colab est basé. Colab vous permet d'utiliser et de partager des notebooks Jupyter avec d'autres personnes,

sans avoir besoin de télécharger, d'installer ni d'exécuter quoi que ce soit. Les notebooks Colab sont stockés dans Google Drive. Vous pouvez aussi les charger depuis GitHub. Vous pouvez partager des notebooks Colab comme vous le feriez avec des documents Google Docs ou Sheets. Le code est exécuté sur une machine virtuelle propre à votre compte. Les machines virtuelles sont supprimées lorsqu'elles sont inactives pendant un certain temps et ont une durée de vie maximale imposée par le service Colab. [23]

2.7.2. **Kaggle :**

Filiale de Google, Kaggle est une communauté en ligne de scientifiques des données et d'ingénieurs en apprentissage automatique. Il permet aux utilisateurs de trouver des ensembles de données qu'ils souhaitent utiliser pour construire des modèles d'intelligence artificielle, de publier des ensembles de données, de collaborer avec d'autres scientifiques des données et ingénieurs en apprentissage automatique, et de participer à des compétitions pour résoudre des défis de science des données.

Kaggle a été créé en 2010 en proposant des compétitions d'apprentissage automatique et de science des données, ainsi qu'une plateforme publique de données et d'entreprise basée sur le cloud pour l'éducation en science des données et en IA. Parmi les membres clés de l'équipe, on retrouve Anthony Goldblum et Jeremy Howard.[24]

2.7.3. **Pixels :**

Pixels est une plateforme en ligne qui offre une vaste collection d'images et de photographies de haute qualité, couvrant diverses thématiques et sujets. Elle fournit aux utilisateurs un accès facile à des images de qualité professionnelle pour une utilisation dans des projets créatifs, des sites web, des publications, des présentations et d'autres supports visuels.[25]

2.7.4. **Photopea :**

Photopea est un éditeur d'images avancé qui peut travailler avec à la fois des graphiques matriciels et vectoriels. Vous pouvez l'utiliser pour des tâches simples telles que le redimensionnement d'images, ainsi que pour des tâches complexes telles que la conception de pages web, la création d'illustrations, le traitement de photographies, et bien plus encore. [26]

2.8. Conclusion

En résumé, ce chapitre a exploré l'orientation spatiale, le mouvement et les capteurs d'images. Il a débuté en abordant l'orientation des systèmes de coordonnées à travers des rotations, appliquées notamment au mouvement des avions et spécialement des drones. Le modèle de caméra sténopé a été introduit pour la projection d'images 3D en 2D. Les capteurs d'images numériques, CCD et CMOS, ont été examinés pour leur rôle dans la capture de la lumière.

Le chapitre s'est conclu par une analyse approfondie des transformations de perspective, essentielles pour la rectification d'images. En fin de compte, ce chapitre a démystifié les concepts spatiaux, les capteurs et les transformations, formant ainsi une base solide dans le domaine du traitement d'images et des domaines connexes.

CHAPITRE 3: DETECTION DE DRONE SANS **TRANSFORMATION DE PERSPECTIVE**

3.1. Introduction :

Dans ce chapitre, nous plongeons dans les subtilités d'une simulation centrée autour des Réseaux Neuronaux Convolutifs (CNN en anglais). L'objectif principal de cette simulation est la détection de drones, une technologie aux applications allant de la sécurité à la surveillance. Pour ce faire, nous utilisons un ensemble de données comprenant des images de drones et de non-drones, ce qui permettra l'entraînement d'un modèle CNN.

Ce chapitre vous guide à travers les différentes facettes de notre simulation de détection de drones basée sur CNN, de la préparation initiale de l'ensemble de données à la formation et à l'évaluation du modèle. Nous explorerons l'architecture du modèle, plongerons dans le processus de formation et présenterons les résultats obtenus.

3.2. Description du programme :

Dans ce chapitre, nous avons développé un programme pour la détection de drones en utilisant un modèle de réseau de neurones convolutifs (CNN). Voici une explication des parties essentielles du programme :

- Installation de bibliothèques : Nous commençons par installer la bibliothèque `patool` pour extraire les fichiers compressés contenant les images de drones et sans drones.
- Extraction des fichiers compressés : Les fichiers compressés contenant les images de drones et sans drones sont extraits dans des dossiers spécifiques à l'aide de la bibliothèque `patoolib`.
- Création des répertoires de train, de validation et de test : Les répertoires nécessaires pour les jeux de données de train, de validation et de test sont créés.
- Répartition des images : Les images extraites sont réparties entre les répertoires de train, de validation et de test en utilisant la fonction `split_images`, qui prend en compte les ratios spécifiés.
- Définition de l'augmentation de données : Nous utilisons `ImageDataGenerator` de TensorFlow pour définir les options d'augmentation de données telles que la rotation, le décalage, l'inclinaison, le zoom, etc.
- Préparation des jeux de données : Les générateurs de données d'entraînement et de validation sont créés en utilisant les répertoires de train et de validation, respectivement.

- Définition du modèle CNN : Nous définissons le modèle CNN en utilisant la séquence de couches de Keras. Le modèle comprend des couches de convolution, de normalisation, de max pooling, de dropout, et des couches entièrement connectées.
- Compilation du modèle : Le modèle est compilé avec la fonction de perte binaire (binary_crossentropy), l'optimiseur Adam et la métrique d'exactitude.
- Entraînement du modèle : Le modèle est entraîné en utilisant la méthode fit avec les générateurs de données d'entraînement et de validation. Le nombre d'étapes par époque et le nombre total d'époques sont spécifiés.
- Affichage des courbes d'apprentissage : Les courbes de perte et de précision pour l'entraînement et la validation sont tracées à partir des valeurs enregistrées lors de l'entraînement. La "Classification Accuracy" ou "Précision" est le rapport entre le nombre de prédictions correctes et le nombre total d'échantillons d'entrée.[31]

$$\text{Précision} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions effectuées}} \quad (3.1)$$

- Sauvegarde du modèle : Le modèle entraîné est sauvegardé pour une utilisation ultérieure.
- Chargement du modèle sauvegardé : Le modèle sauvegardé est chargé pour effectuer des prédictions sur de nouvelles images.
- Prédiction d'une image : Une fonction predict_image est définie pour charger une image, la prétraiter et effectuer une prédiction en utilisant le modèle chargé.
- Affichage du résultat de la prédiction : La fonction predict_image est appelée pour prédire si l'image contient un drone ou non. Le résultat de la prédiction est affiché avec un message indiquant la probabilité associée.
- Affichage de l'image : L'image d'origine est affichée pour référence.

Ces étapes essentielles sont expliquées dans le contexte du programme pour la détection de drones. Le programme peut être modifié et adapté en fonction des besoins spécifiques du projet.

3.3. Description du modèle :

Le modèle CNN est défini en utilisant la séquence de couches de Keras. Voici une explication des différentes couches utilisées :

a. Convolutional Layers (Couches de convolution) : Les couches de convolution sont responsables de l'extraction des caractéristiques de l'image en utilisant des filtres convolutifs. Dans notre modèle, nous utilisons quatre couches de convolution avec différentes tailles de noyaux (3x3) et des nombres de filtres croissants (16, 32, 64, 128). La fonction d'activation 'relu' est utilisée pour introduire la non-linéarité dans le modèle.

b. Batch Normalization Layer (Couche de normalisation par lots) : La couche de normalisation par lots est utilisée pour normaliser les activations des couches précédentes, ce qui facilite la convergence du modèle et améliore les performances.

c. Max Pooling Layers (Couches de max pooling) : Les couches de max pooling sont utilisées pour réduire la taille des caractéristiques extraites par les couches de convolution en ne conservant que les valeurs maximales dans chaque région. Cela permet de réduire le nombre de paramètres du modèle et de capturer les informations les plus importantes.

d. Dropout Layers (Couches de désactivation) : Les couches de désactivation sont utilisées pour lutter contre le surapprentissage en désactivant aléatoirement certains neurones lors de l'entraînement. Dans notre modèle, nous utilisons des couches de désactivation avec un taux de désactivation de 0,5.

e. Flatten Layer (Couche d'aplatissement) : La couche d'aplatissement est utilisée pour convertir les caractéristiques en un vecteur unidimensionnel afin de les fournir à la couche entièrement connectée.

f. Fully Connected Layers (Couches entièrement connectées) : Les couches entièrement connectées sont utilisées pour effectuer la classification finale. Dans notre modèle, nous utilisons une couche dense avec 128 neurones et une fonction d'activation 'relu'. Une couche de désactivation avec un taux de désactivation de 0,5 est également utilisée.

g. Output Layer (Couche de sortie) : La couche de sortie est composée d'une seule unité avec une fonction d'activation 'sigmoid'. Elle est utilisée pour prédire la probabilité qu'une image contienne un drone.

En résumé, le modèle CNN que nous avons défini comprend des couches de convolution pour l'extraction des caractéristiques, des couches de normalisation et de désactivation pour améliorer les performances et lutter contre le surapprentissage, et des

couches entièrement connectées pour la classification finale. Le modèle est compilé avec une fonction de perte binaire (`binary_crossentropy`) et un optimiseur Adam.

3.4. Entraînement du modèle :

Après avoir défini le modèle CNN, nous pouvons procéder à son entraînement en utilisant la méthode `fit` de Keras. Voici une explication du processus d'entraînement :

a. Préparation des générateurs de données :

Nous utilisons la classe `ImageDataGenerator` de Keras pour générer des lots d'images à partir des répertoires d'entraînement et de validation.

Les images sont redimensionnées à une taille de (150, 150) et normalisées en divisant les valeurs des pixels par 255.

Nous spécifions également différents paramètres de data augmentation, tels que la rotation, le déplacement horizontal/vertical, le cisaillement, le zoom et le retournement horizontal, pour augmenter la diversité des données d'entraînement.

b. Compilation du modèle :

Avant de commencer l'entraînement, nous devons compiler le modèle en spécifiant la fonction de perte et l'optimiseur.

Dans notre cas, nous utilisons la fonction de perte "`binary_crossentropy`" pour traiter un problème de classification binaire (présence ou absence de drone).

L'optimiseur "`adam`" est utilisé pour ajuster les poids du modèle lors de l'entraînement.

c. Entraînement du modèle :

Nous utilisons la méthode `fit` pour entraîner le modèle en spécifiant les générateurs de données d'entraînement et de validation.

Le paramètre `steps_per_epoch` est défini sur le nombre total d'échantillons d'entraînement divisé par la taille du lot (`batch_size`). Il représente le nombre d'itérations pour chaque époque d'entraînement.

Le paramètre epochs spécifie le nombre total d'époques pour lesquelles le modèle sera entraîné.

Pendant l'entraînement, le modèle ajuste les poids en utilisant la rétropropagation du gradient et minimise la fonction de perte.

d. Suivi des métriques d'entraînement :

Pendant l'entraînement, les métriques d'entraînement telles que la perte (loss) et l'exactitude (accuracy) sont calculées pour chaque époque et sont enregistrées dans l'historique du modèle.

e. Visualisation des courbes d'apprentissage :

Une fois l'entraînement terminé, nous pouvons visualiser les courbes d'apprentissage en traçant les valeurs de perte et d'exactitude pour les époques d'entraînement et de validation.

Cela nous permet d'évaluer les performances du modèle et d'identifier d'éventuels problèmes de surapprentissage ou de sous-apprentissage.

f. Sauvegarde du modèle :

Une fois que le modèle est correctement entraîné, nous pouvons le sauvegarder dans un fichier à l'aide de la méthode save.

Cela nous permet de réutiliser le modèle ultérieurement sans avoir à le réentraîner.

En résumé, l'entraînement du modèle consiste à préparer les générateurs de données, à compiler le modèle avec la fonction de perte et l'optimiseur, à utiliser la méthode fit pour entraîner le modèle sur les données d'entraînement et à enregistrer les métriques d'entraînement. Les courbes d'apprentissage peuvent être utilisées pour évaluer les performances du modèle, et le modèle entraîné peut être sauvegardé pour une utilisation future

3.5. Description du data :

Les données d'entraînement se composent de deux classes : "images de drones" et "images sans drone", qui incluent des images aléatoires d'objets divers tels que des voitures, des motos, des avions, des camions, des animaux, des trains, et plus encore. Les images de

drones représentent spécifiquement différentes vues et orientations de drones, tandis que la classe "sans drone" comprend des images qui ne contiennent pas de drones.

Les images utilisées pour l'entraînement ont été principalement obtenues à partir de base de données Kaggle et de la plateforme Pixels. Ces ensembles de données étaient disponibles dans deux formats : JPEG et JPG. Un total de 400 images ont été utilisées pour l'entraînement, avec une répartition équitable entre les classes de drones et sans drone.

3.6. Description des résultats :

Les résultats obtenus sont les suivants :

Après 30 époques, le modèle a atteint une précision d'entraînement d'environ 87,50 % et une précision de validation de 53,12 %. La perte d'entraînement finale était de 0,3652. La figure 3.1 démontre le déroulement de l'entraînement du modèle.

Ces résultats montrent que le modèle a des difficultés à généraliser aux données de validation, car la précision de validation est relativement faible par rapport à la précision d'entraînement. Cela peut indiquer un surapprentissage du modèle aux données d'entraînement.

En résumé, bien que le modèle ait montré une certaine capacité à apprendre les données d'entraînement, il est nécessaire d'effectuer des ajustements pour améliorer sa capacité à généraliser aux données de validation.

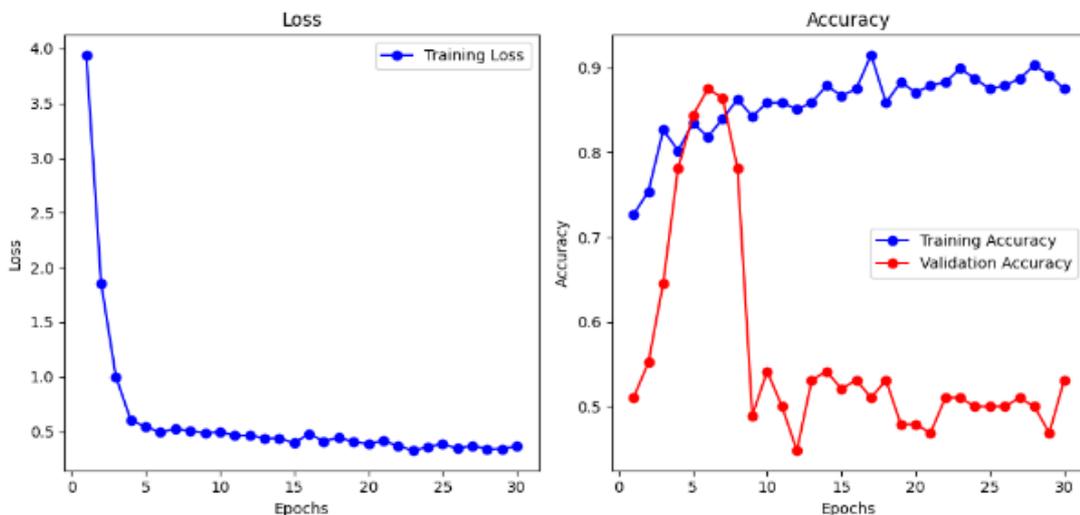


Figure 3-1 Résultat de classification sans transformation de perspective

3.7. Conclusion :

Dans cette étude, nous avons examiné les performances du modèle de simulation sur les données d'entraînement. Bien que le modèle ait montré une certaine capacité à apprendre à partir de ces données, il est essentiel de prendre des mesures pour améliorer sa capacité à généraliser aux données de validation. Les ajustements nécessaires doivent être effectués afin de renforcer l'aptitude du modèle à s'adapter efficacement aux nouvelles données et à fournir des résultats fiables.

En résumé, le modèle de simulation présente un potentiel prometteur, mais des efforts supplémentaires sont nécessaires pour affiner sa performance et garantir sa robustesse.

En se basant sur les résultats et les enseignements tirés de cette phase, des améliorations et des optimisations supplémentaires peuvent être apportées pour obtenir des performances plus solides et une meilleure généralisation. Cela ouvre la voie à des recherches futures visant à affiner et à étendre le modèle

CHAPITRE 4: DETECTION DE DRONE AVEC TRANSFORMATION DE PERSPECTIVE

4.1. Introduction :

Bienvenue dans la deuxième partie de notre voyage de simulation, où nous approfondissons notre exploration dans le monde fascinant des Réseaux Neuronaux Convolutifs (CNN) et de leur application dans la détection de drones. Dans ce chapitre, nous explorons un aspect distinct de la détection de drones : l'effet de la perspective.

Dans notre précédent chapitre, nous nous sommes lancés dans une quête pour détecter les drones en utilisant un ensemble de données diversifié comprenant des images de drones et de non-drones. À présent, notre attention se tourne vers un ensemble de données plus spécialisé. Cette nouvelle collection d'images partage une caractéristique commune : toutes les images de drones possèdent la même perspective.

Vous pourriez vous demander pourquoi cette perspective est si cruciale. La perspective à partir de laquelle une image est capturée peut avoir un impact significatif sur l'efficacité d'un CNN dans la détection de drones. Les scénarios du monde réel impliquent souvent des drones dans des orientations et des angles variés. Comprendre comment un CNN réagit à ces différentes perspectives est essentiel pour faire progresser le domaine de la détection de drones.

À travers cette exploration, nous visons à mettre en lumière les subtilités de la perspective dans le contexte de la détection de drones basée sur CNN. Ce que nous découvrirons ici enrichira non seulement notre compréhension des CNN, mais contribuera également à l'avancement des technologies de détection de drones.

4.2. Description du programme :

Dans ce chapitre, nous avons développé un programme pour la détection de drones en utilisant un modèle de réseau de neurones convolutifs (CNN) en utilisant des techniques d'augmentation de données et une transformation de perspective. Voici les étapes essentielles :

- Extraction des données : Le code commence par extraire les images compressées des drones et des non-drones vers leurs répertoires respectifs en utilisant la bibliothèque `patoolib`. Cette étape prépare le jeu de données pour un traitement ultérieur.

- Préparation du jeu de données : Le code crée des répertoires d'entraînement, de validation et de test pour organiser le jeu de données. Il crée ensuite des sous-répertoires pour les classes de drones et de non-drones dans chaque ensemble. Le jeu de données est ensuite réparti dans ces répertoires en fonction des ratios spécifiés.
- Augmentation des données : ImageDataGenerator du module ImageDataGenerator de TensorFlow est utilisé pour définir les options d'augmentation des données. Ces options comprennent la mise à l'échelle, la rotation, le décalage en largeur et en hauteur, l'inclinaison, le zoom et le retournement horizontal.
- Définition du modèle : Le modèle CNN est défini en utilisant l'API `tf.keras.Sequential`. Il se compose de plusieurs couches de convolution, de couches de normalisation par lots, de couches de max-pooling, de couches de désactivation, d'une couche d'aplatissement et de couches denses avec la couche de sortie finale utilisant la fonction d'activation sigmoïde pour la classification binaire.
- Compilation et entraînement du modèle : Le modèle est compilé avec une perte d'entropie croisée binaire, l'optimiseur Adam avec un taux d'apprentissage de 0,0001 et la métrique d'évaluation de l'exactitude. La méthode `fit` est ensuite utilisée pour entraîner le modèle sur les générateurs de données d'entraînement et de validation pendant un nombre spécifié d'époques.
- Évaluation du modèle : Les valeurs de perte et d'exactitude sont extraites de l'historique d'entraînement et tracées à l'aide de `matplotlib`. Le modèle entraîné est sauvegardé pour une utilisation ultérieure.
- Prédiction d'images : La fonction `predict_image` est définie pour prétraiter une image et obtenir la prédiction du modèle. La fonction `preprocess_image` est utilisée pour redimensionner et normaliser l'image avant de la fournir au modèle.
- Test du modèle : Un chemin d'image à partir de l'ensemble de test est sélectionné, et la fonction `predict_image` est appelée pour obtenir le résultat de prédiction. Si la valeur de prédiction est élevée ($\geq 0,9$), l'image est classée comme contenant un drone. Sinon, la transformation de perspective est appliquée pour générer plusieurs images transformées. Les images transformées sont redimensionnées et soumises au modèle pour obtenir les résultats de prédiction. L'image transformée avec la valeur de prédiction la plus élevée est sélectionnée et affichée, ainsi que son résultat de prédiction correspondant.
- Visualisation des images : L'image d'origine et, le cas échéant, la meilleure image transformée sont affichées à l'aide de `matplotlib`.

4.3. Compilation et Entraînement du modèle :

Le modèle est compilé avec une fonction de perte binaire `binary_crossentropy` et un optimiseur `adam` avec un taux d'apprentissage de 0,0001. Il est ensuite entraîné sur les données d'entraînement et de validation à l'aide de la méthode `fit`. Le nombre d'époques et les paramètres de lot sont spécifiés pour contrôler la durée de l'entraînement.

Après l'entraînement, le modèle est évalué à l'aide des données de test pour mesurer ses performances sur des exemples inconnus. Les résultats d'évaluation, tels que la perte et l'exactitude, sont extraits de l'historique d'entraînement et tracés à l'aide de `matplotlib`. Cela permet de visualiser la progression de l'apprentissage du modèle au fil des époques.

Ensuite, le modèle est sauvegardé pour une utilisation ultérieure. Il peut être chargé à tout moment pour effectuer des prédictions sur de nouvelles images.

4.4. Fonction de transformation de perspective :

La fonction `random_perspective_transform` est une fonction personnalisée de transformation de perspective. Cette fonction effectue une transformation aléatoire de la perspective sur une image donnée.

Voici plus de détails sur le fonctionnement de `random_perspective_transform` :

- La fonction prend une image en entrée, représentée sous forme de tableau `numpy`.
- La hauteur, la largeur et le nombre de canaux de l'image sont extraits à partir de ses dimensions.
- Quatre points de transformation de perspective sont définis de manière aléatoire à l'intérieur de l'image. Chaque point est représenté par une paire de coordonnées (x, y) . Ces points définissent un quadrilatère dans l'image d'origine.
- Quatre points de destination sont définis, formant un quadrilatère de la même taille que l'image de sortie souhaitée. Ces points correspondent à la position souhaitée des quatre coins de l'image transformée.

- Une matrice de transformation de perspective est calculée à l'aide des points de transformation d'origine et de destination à l'aide de la fonction `cv2.getPerspectiveTransform`.
- La fonction `cv2.warpPerspective` est utilisée pour appliquer la transformation de perspective à l'image d'entrée à l'aide de la matrice de transformation calculée.
- L'image transformée est renvoyée en sortie de la fonction.

En utilisant cette fonction de prétraitement dans le pipeline de génération de données, chaque image générée sera soumise à une transformation de perspective aléatoire avant d'être utilisée pour l'entraînement du modèle. Cela peut aider à augmenter la variabilité des données et à améliorer la capacité du modèle à généraliser à de nouvelles perspectives d'images.

4.5. Test du modèle :

Cette partie du programme effectue des tests sur le modèle entraîné en utilisant les données de l'ensemble de test.

Voici plus de détails sur cette partie du programme :

- Un chemin d'image est sélectionné à partir de l'ensemble de test. Cela peut être une image qui n'a pas été utilisée dans l'entraînement ou la validation du modèle.
- La fonction `predict_image` est appelée en passant le chemin de l'image sélectionnée en argument. Cette fonction charge l'image, la prétraite et effectue une prédiction en utilisant le modèle chargé. Elle renvoie le résultat de prédiction.
- La valeur de prédiction est extraite du résultat renvoyé par la fonction `predict_image`.
- Si la valeur de prédiction est supérieure ou égale à 0,9, l'image est classée comme contenant un drone avec une probabilité élevée.
- Si la valeur de prédiction est inférieure à 0,9, une transformation de perspective est appliquée à l'image d'origine pour générer plusieurs images transformées.
- Chaque image transformée est redimensionnée à la taille souhaitée et soumise au modèle pour obtenir les résultats de prédiction.
- Les résultats de prédiction pour chaque image transformée sont stockés dans une liste.

- L'index de l'image transformée ayant la valeur de prédiction la plus élevée est déterminé en utilisant la fonction `np.argmax`.
- L'image transformée correspondant à l'index obtenu est sélectionnée comme la meilleure image transformée avec la prédiction la plus élevée.
- Le résultat de prédiction correspondant à la meilleure image transformée est affiché.
- Si la valeur de prédiction pour la meilleure image transformée est supérieure ou égale à 0,5, elle est classée comme contenant un drone avec une probabilité élevée. Sinon, elle est classée comme ne contenant pas de drone avec une probabilité élevée.
- L'image d'origine et l'image transformée avec la prédiction la plus élevée sont affichées à l'aide de `matplotlib`.

Cette partie du programme permet de tester la capacité du modèle à prédire si une image contient un drone. Si la valeur de prédiction est élevée, le modèle est considéré comme ayant identifié un drone dans l'image. Sinon, une transformation de perspective est appliquée pour générer des perspectives alternatives, et le modèle est utilisé pour sélectionner la meilleure perspective avec la prédiction la plus élevée. Cela permet de tester la robustesse du modèle face à des variations de perspectives.

4.6. Description du data :

Les données d'entraînement sont constituées de deux classes : "images de drones" et "images sans drone", où la classe "sans drone" reste la même. Les images de drones représentent spécifiquement un seul point de vue et une seule orientation des drones. Les images ont été préparées manuellement en utilisant la plateforme Photopea afin d'obtenir une perspective uniforme.

Les images utilisées pour l'entraînement ont été principalement obtenues à partir de bases de données Kaggle et de la plateforme Pixels. Ces ensembles de données étaient disponibles dans les formats JPEG et JPG. Au total, 400 images ont été utilisées pour l'entraînement, avec une répartition équitable entre les classes de drones et sans drone.

4.7. Description des résultats :

Les résultats obtenus sont les suivants : Après 30 époques, le modèle a atteint une précision d'entraînement d'environ 93,95 % et une précision de validation de 48,96 %, la perte d'entraînement finale était de 0,2288. La figure 3.2 démontre le déroulement de l'entraînement du modèle.

Ces résultats montrent que le modèle a une précision d'entraînement élevée, mais il ne parvient pas à généraliser efficacement aux données de validation, car la précision de validation est relativement faible par rapport à la précision d'entraînement. Cela indique un problème de surapprentissage.

En résumé, bien que le modèle ait montré une précision d'entraînement élevée, il est nécessaire d'effectuer des ajustements pour améliorer sa capacité à généraliser aux données de validation et à éviter le surapprentissage.

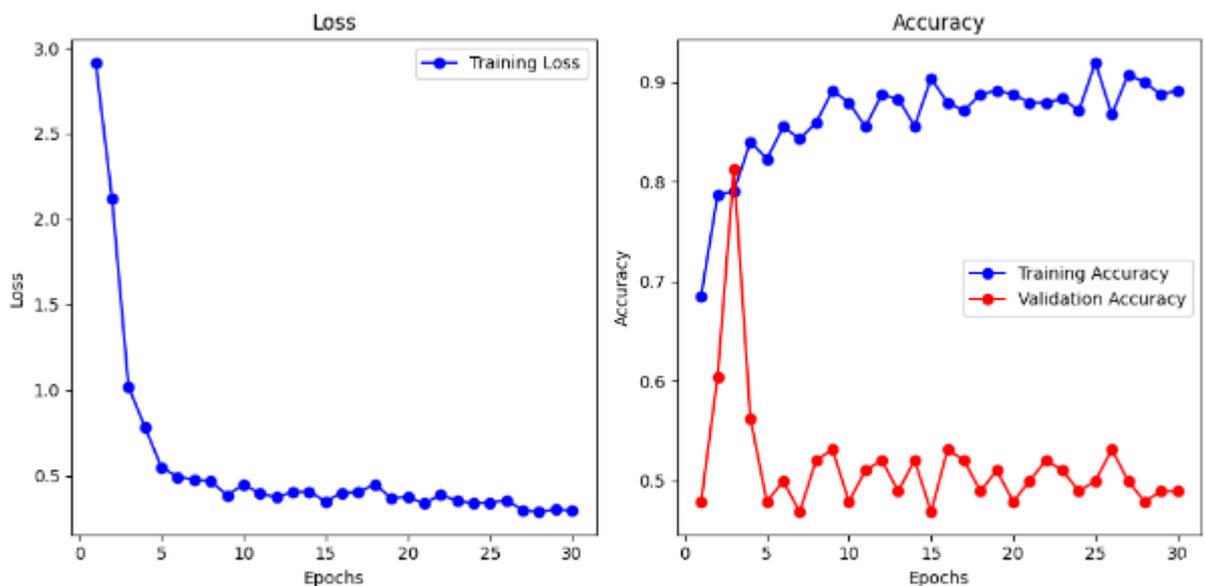


Figure 4-1 Résultat de classification avec transformation de perspective

4.8. Conclusion :

Dans cette étude, nous avons examiné les performances du modèle de simulation sur les données d'entraînement plus personnalisé (un seul perspectif pour les drones). Bien que le modèle ait montré une certaine capacité à apprendre à partir de ces données avec plus de précision, il est essentiel de prendre des mesures pour améliorer sa capacité à généraliser aux données de validation. Les ajustements nécessaires doivent être effectués afin de renforcer l'aptitude du modèle à s'adapter efficacement aux nouvelles données et à fournir des résultats fiables essentiellement en augmentant le nombre d'image utilisé.

En résumé, le modèle de simulation a présenté une amélioration du potentiel, mais des efforts supplémentaires sont nécessaires pour affiner sa performance et garantir sa robustesse.

Les résultats et les enseignements de cette phase fournissent des indications précieuses pour l'amélioration continue du modèle de simulation. En utilisant ces informations, des améliorations et des optimisations supplémentaires peuvent être mises en œuvre afin d'obtenir des performances plus solides et une meilleure capacité de généralisation.

CONCLUSION GENERALE

Conclusion Générale :

En conclusion, la présente étude constitue une synthèse des résultats et des contributions obtenus tout au long de cette recherche. Nous avons exploré différents aspects liés à notre sujet, de l'introduction du réseau de neurones convolutifs (CNN) à l'analyse des performances du modèle de simulation. Cette conclusion générale mettra en évidence les principales conclusions tirées de chaque chapitre et soulignera les perspectives d'amélioration et les directions futures pour renforcer la performance et la généralisation de notre modèle.

Dans le premier chapitre on a introduit le réseau de neurones convolutifs (CNN). Souvent utilisée pour la vision par ordinateur et le traitement d'images. Les CNN sont conçus pour imiter le fonctionnement du cortex visuel humain en utilisant des couches de neurones spécialisées dans la détection de motifs visuels. Ils sont capables d'apprendre automatiquement des caractéristiques à différents niveaux d'abstraction, allant des traits simples (comme des lignes et des contours) aux caractéristiques plus complexes (comme des formes et des objets). Leur capacité à apprendre des caractéristiques pertinentes à partir des données en fait des outils puissants pour l'analyse et la compréhension des images.

Dans le deuxième on a examiné différents aspects liés à notre sujet, notamment les mouvements d'un aéronef, les caractéristiques spécifiques des drones, les capteurs numériques d'image, les formats d'images, les transformations géométriques et les plateformes utilisées. En combinant ces connaissances, nous avons pu approfondir notre compréhension du sujet et réaliser des avancées significatives dans notre recherche.

Dans le troisième chapitre de notre mémoire, nous avons exploré le potentiel prometteur du modèle de classification sans transformation. Cependant, des efforts supplémentaires sont nécessaires pour affiner sa performance et garantir sa robustesse. Les résultats ont montré une précision d'entraînement d'environ 87,50 % et une précision de validation de 53,12 %. La perte d'entraînement finale était de 0,3652.

Dans le chapitre 4, nous avons également examiné la performance du modèle de classification avec transformation de perspective, mettant l'accent sur sa capacité à généraliser aux données de validation et à éviter le surapprentissage. Bien que le modèle ait montré une précision d'entraînement élevée d'environ 93,95 %, des ajustements sont nécessaires pour améliorer sa capacité à généraliser. Les résultats de validation ont montré une précision de 48,96 %, avec une perte d'entraînement finale de 0,2288.

En comparant les résultats des deux simulations, nous constatons que le modèle de classification avec transformation a montré une précision d'entraînement légèrement supérieure, mais une précision de validation plus faible par rapport au modèle de classification sans transformation. Cela indique la présence de surapprentissage dans le modèle de classification avec transformation due au petit nombre d'image utilisé par rapport au nombre nécessaire, ce qui suggère la nécessité d'augmenter le volume de la base de données pour améliorer sa généralisation.

Dans l'ensemble, notre recherche a mis en évidence l'importance de poursuivre les efforts d'optimisation et de validation du modèle de simulation. Les résultats obtenus dans les deux chapitres ont contribué à notre compréhension du sujet, mais des améliorations sont nécessaires pour renforcer la performance du modèle et sa capacité à généraliser aux nouvelles données.

Ces résultats ouvrent la voie à des recherches futures qui visent à affiner, étendre et perfectionner le modèle en ajoutant un moyen de produire les transformations de perspective d'une manière autonome du système.

Ces conclusions soulignent l'importance de l'ajustement des paramètres, de l'exploration de nouvelles méthodes et de l'utilisation de techniques de régularisation pour améliorer la performance des modèles de détection. Les travaux futurs peuvent se concentrer sur ces aspects pour obtenir des résultats plus fiables et robustes.

References:

- [1] EASA Pro. 2016. Study and Recommendations regarding Unmanned Aircraft System Geo-Limitations. <https://www.easa.europa.eu/en/document-library/general-publications/study-and-recommendations-regarding-unmanned-aircraft-system>
- [2] Miroslav Kratky* and Jan Farlik. 2018. Countering UAVs – the Mover of Research in Military Technology. Defence Science Journal. Vol. 68, No. 5. 460-466.
- [3] E. Çetin* C. Barrado and E. Pastor. 2021. Improving real-time drone detection for counter-drone systems. THE AERONAUTICAL JOURNAL. Vol 125 .No 1292. 1871–1896.
- [4] Matthew D. Zeiler and Rob Fergus. s.d. Visualizing and Understanding Convolutional Networks.
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwig_oSvk4mAAxUZWcAKHfCmD9wQFnoECA8QAQ&url=https%3A%2F%2Ffarxiv.org%2Fabs%2F1311.2901&usg=AOvVaw0bGZH7N8K2NZ8upuy-gyju&opi=89978449
- [5] Saad Albawi, Oguz Bayat, Saad Al-Azawi and Osman N. Ucan. 2018. Social Touch Gesture Recognition Using Convolutional Neural Network. Hindawi Computational Intelligence and Neuroscience. Vol 2018. 10 pages.
- [6] Qianji Zhao, Zequn Shang. 2021. Deep learning and Its Development. Journal of Physics: Conference Series. Phys.: Conf. Ser. 1948 012023.14 pages.
- [7] Keiron O’Shea and Ryan Nash. s.d. An Introduction to Convolutional Neural Networks.
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjp7On0lomAAxXRU0EAHTUzC64QFnoECA0QAQ&url=https%3A%2F%2Farxiv.org%2Fabs%2F1511.08458&usg=AOvVaw0kjlYsKvzSvpvilS0bqIf8&opi=89978449>
- [8] Jordan J. Bird et Diego R. Faria .2018. A Study on CNN Transfer Learning for Image Classification.
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjmiqaEmomAAxWGIMAKHR_DAd8QFnoECA4QAQ&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F325803364_A_Study_on_CNN_Transfer_Learning_for_Image_Classification&usg=AOvVaw03N-wa7o-CYGEhjMCOtyos&opi=89978449

- [9] Sajja Tulasi Krishna et Hemantha kumar Kalluri. 2019. Deep Learning and Transfer Learning Approaches for Image Classification. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi8zbbnmomAAxUNRkEAHfvwA9YQFnoECA8QAQ&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F333666150_Deep_Learning_and_Transfer_Learning_Approaches_for_Image_Classification&usg=AOvVaw27P_-29ePvyUkHA81LvU1p&opi=89978449
- [10] Sinno Jialin Pan and Qiang Yang Fellow. s.d. A Survey on Transfer Learning. IEEE.15 pages.
- [11] Jaya Gupta et al. 2022. Deep Learning (CNN) and Transfer Learning: A Review. Journal of Physics. Conf. Ser. 2273 012029. 11 pages.
- [12] Comite Département Aéronautique d'Ille et Vllaine. s.d. CONNAISSANCE DES AÉRONEFS. http://ac.oleron.free.fr/Files/287_cours_bia_ppl_connaissances_acft_2016_12_02.pdf
- [13] National Aeronautics and Space Administration. s.d. Axes / Control Surfaces. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwio8arjyomAAxULVqQEHQQA6C9AQFnoECBYQAQ&url=https%3A%2F%2Fwww.nasa.gov%2Fsites%2Fdefault%2Ffiles%2Fatoms%2Ffiles%2Faxes_control_surfaces_5-8.pdf&usg=AOvVaw1_Mm21sgcjjobawpx6AbS-&opi=89978449
- [14] FUTURA. 27/05/2018. Drone : qu'est-ce que c'est ?. <https://www.futura-sciences.com/sciences/definitions/aeronautique-drone-6174/>
- [15] guide-drone.sd. Comment vole un drone ?. <https://www.guide-drone.com/comment-vole-un-drone/>
- [16] PIERRE SLANGEN ET PASCAL PICART. s.d. Principes de bases des capteurs d'images. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjhmca4r4mAAxV4QkEAHSt7A80QFnoECCwQAQ&url=http%3A%2F%2Fwww.optique-ingenieur.org%2Ffr%2Fcours%2Fpdf%2FOPI_fr_M05_C06.pdf&usg=AOvVaw3Tq9TNY7JtAbTTv3B8o4ip&opi=89978449

- [17] video-surveillance-direct.sd. Lexique des caméras de surveillance. <https://www.video-surveillance-direct.com/content/8-lexique-cameras-video-surveillance>
- [18] keyence.sd. Capteur d'image. https://www.keyence.fr/ss/products/microscope/glossary/cat6/image_sensor/
- [19] PSL Explore.01/02/2022.Format d'images. <https://explore.psl.eu/fr/se-former/tutoriels/formats-dimages>
- [20] Jean-Marc Vézien. s.d. TRAITEMENT DES IMAGES et VISION PAR MACHINE. https://perso.limsi.fr/vezien/pdf_cours_ima_jmv.pdf
- [21] Djihane BABAHENINI.2019. Présentation générale de la synthèse d'image. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjSw4HxuImAAxVOUaQEHWqADdEQFnoECAwQAQ&url=http%3A%2F%2Flearning.univ-biskra.dz%2Fmoodle2019%2Fmod%2Fresource%2Fview.php%3Fid%3D6321&usg=AOvVaw2V6N3tLU5hhhYfdgFo7Eh9&opi=89978449>
- [22] A. Soycan*, M. Soycan. 2019. Perspective correction of building facade images for architectural applications. Engineering Science and Technology, an International Journal. Vol 22. 697–705.
- [23] Google, Colaboratory. [https://research.google.com/colaboratory/faq.html?hl=fr#:~:text=Colaboratory%2C%20soutenu%20raccourci%20en%20"Colab,données%20et%20à%20l%27éducation](https://research.google.com/colaboratory/faq.html?hl=fr#:~:text=Colaboratory%2C%20soutenu%20raccourci%20en%20)
- [24] kaggle, (Consulté le 10 juillet 2023), <https://www.kaggle.com/discussions/general/328265>
- [25] Pixels. (Consulté le 10 juillet 2023). <https://www.pixels.com>
- [26] photopea , (Consulté le 10 juillet 2023), <https://www.photopea.com/learn/>
- [27] researchgate .s.d. les axes de mouvement d'un aéronef. https://www.researchgate.net/figure/Variations-de-latitude-de-la-plateforme-roulis-tangage-et-lacet_fig3_276905733

[28] guide-drone.com. s.d. Différentes directions de mouvement possible d'un drone
<https://www.guide-drone.com/wp-content/uploads/2018/10/force-drone.png>

[29] static.wixstatic. s.d. Comment faire un mouvement de lacet
https://static.wixstatic.com/media/15c516_4621a72e55cf4961bb239904df0a5243.png/v1/fill/w_538,h_423,al_c,q_85,usm_0.66_1.00_0.01,enc_auto/15c516_4621a72e55cf4961bb239904df0a5243.png

[30] Videosurveillance-boutique. s.d. Différence entre un capteur CCD et un capteur CMOS.
<https://www.videosurveillance-boutique.fr/support/difference-entre-capteur-ccd-et-capteur-cmos-3075.html>

[31] kaggle. 2019. How to Choose Right Metric for Evaluating ML Model.
<https://www.kaggle.com/code/vipulgandhi/how-to-choose-right-metric-for-evaluating-ml-model>

- Dasmehdixtr. 2019. Drone Dataset (UAV). [base de données en ligne]. Kaggle.
<https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav>
- Lprdsmil. 2021. Unsplash random images collection. [base de données en ligne]. Kaggle.
<https://www.kaggle.com/datasets/lprdsmil/unsplash-random-images-collection>