

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE Dr. TAHAR MOULAY SAIDA
FACULTE : TECHNOLOGIE
DEPARTEMENT : INFORMATIQUE



MÉMOIRE DE MASTER

Option : Sécurité Informatique et Cryptographie

L'intelligence artificielle (IA) pour manœuvrer la guerre et les opérations militaires

Présenté par :

MEDDAHI Fahima

BOUARARA Imane Djihad

Encadré par :

Dr. BOUARARA Hadj Ahmed

Artificial intelligence (AI) to maneuver war and military operations

Abstract — Artificial Intelligence (AI), also dubbed Industrial Revolution 4.0, is already a part of our lives and, as technology matures, it will play a key role in future wars. The main objective of our work is to validate the following expression : how the convergence of distributed computing and smart grids is leading to the next generation of situational understanding in the battle space. For this we have proposed a system called artificial intelligence for a modern army (AIMA). It is like entering images of battlefields or borders. These images will go through a step of detecting moving objects using the YOLO V4. Each foreground image will be analyzed using our proposed CNN models to detect the type of tank in action to conclude whether the situation captured by the drone cameras is dangerous or not. After various tests and comparisons carried out using evaluation and benchmark measures , the results obtained were satisfactory in terms of accuracy with variation of the various hyper parameters such as the Learning rate, data increase, batch size, number of pockets. . . etc

Keywords :yolov4, tank detection, military operation, object detection, object tracking..

L'intelligence artificielle (IA) pour manœuvrer la guerre et les opérations militaires

Résumé — L'intelligence artificielle (IA), également surnommée la révolution industrielle 4.0 faits déjà partie de nos vie et à mesure que la technologie murira, elle jouera un rôle clé dans les guerres futures. L'objectifs principale de notre travail est de valider l'expression suivante : comment la convergence du calcule distribué et des réseaux intelligents entraine la prochaine génération de compréhension de la situation dans l'espace de bataille. Pour cela nous avons proposé un système appelé l'intelligence artificiel pour un armé moderne (l'IAAM).On ' a comme entrer les images des champs de batailles ou des frontières. Ces images vont passer par une étape de détection des objets en mouvement utilisant le YOLO V4. Chaque image de premier plan sera analysée utilisant nos modèle CNN proposés afin de détecter le type de tank en action pour conclure si la situation filmée par les caméras des drones est dangereuse ou non. Après différents testes et comparaisons réalisés utilisant des mesures d'évaluations et de benchmarks ,

les résultats obtenus ont été satisfaisantes en terme d'accuracy avec variation des différents hyper paramètres comme le Learning rate , data augmentation, batch size, nombre d'époque ...etc.

Mots clés : yolov4, la détection de réservoir ,les opérations militaires, détection des objets, suivie des objets.

الذكاء الاصطناعي في المناورات الحربية و العمليات العسكرية

المخلص --- الذكاء الاصطناعي او ما يطلق عليه اسم الثورة الصناعية الرابعة هي بالفعل جزء من حياتنا ومع التطور التكنولوجي ستلعب دوراً رئيسياً في حروب المستقبل الهدف الرئيسي لعملنا هو التحقق من صحة التعبير التالي: كيف يؤدي التقارب بين الحوسبة الموزعة والشبكات الذكية إلى توليد الوعي الظرفي في ساحة المعركة. لهذا اقترحنا نظاماً يسمى الذكاء الاصطناعي للجيش الحديث. حيث ندخل صور ساحات القتال أو الحدود. هذه الصور ستمر بمرحلة لاكتشاف الأجسام المتحركة باستخدام (يولوفيه) ستكون كل صورة في المقدمة تم تحليلها باستخدام نماذج (سي ان ان) المقترحة الخاصة بنا لاكتشاف نوع الخزان قيد التشغيل لاستنتاج ما إذا كان الموقف الذي تم تصويره بواسطة كاميرات الطائرات بدون طيار خطير أم لا. بعد إجراء اختبارات ومقارنات مختلفة باستخدام القياسات الثلاثة من التقييمات والمعايير ، كانت النتائج التي تم الحصول عليها مرضية من حيث الدقة مع تباين المعلمات الفائقة المختلفة مثل معدل التعلم وزيادة البيانات ، حجم الدفعة ، عدد الجيب ... إلخ.

الكلمات المفتاحية: تالي . ، كشف الدبابات ، كشف النشلملية العسكرية ، الكشف عن الأشياء ، تتبع الكائنات

REMERCIEMENT

Tout d'abord, Nous tenons à remercier notre DIEU qui nous a donné le courage et la volonté pour terminer ce travail. À ceux qui ont contribué de près ou de loin à la réalisation de ce travail en particulier : Monsieur BOUARARA HADJ AHMED pour nous avoir encadré d'assurer le suivi de notre travail. Nous remercions également les professeurs et les enseignants de L'informatique qui nous ont guidés tout en long du chemin de savoir. Nous remercions aussi Monsieur le président de jury et les membres du jury qui ont bien voulu examiner notre travail. Nous tenons à remercier aussi nos amis et nos collègues de la promotion. Enfin, nous remercions tout ce qui nous ont aide et soutenu, de près ou de loin, à réaliser ce mémoire.

TABLE DES SIGLES ET ACRONYMES

Table des sigles et acronymes

[CP-OFDMX]

YOLOv4 *You Look Only Once neural network technology version 4*

IAAM *Intelligence Artificiel pour Armé Moderne*

IA *Intelligence Artificiel*

GPU *Graphics processing unit*

ML *Machine Learning*

AlexNet *Alex Krizhevsky Network*

VGG16 *Visual Geometry Group 16 layers*

DL *Deep Learning*

RELU *REctified Linear Unit*

CPU *Central Processing Unit*

TPU *Tensor Processing Unit*

CNN *Convolution Neural Network*

TABLE DES MATIÈRES

Table des matières	8
	Page
Liste des tableaux	12
Table des figures	13
1 INTRODUCTION GENERALE	1
1.1 Problématique et motivation	1
1.2 Objectifs	2
1.3 Organisation de la thèse	2
2 MANOEUVRER LES OPEARTIONS MILITAIRES	3
2.1 Introduction	3
2.2 Les différentes opérations militaires	3
2.3 L'Intelligence artificielle et les opérations militaires	5
2.3.1 Le soutien en matière d'information et prise décision	6
2.3.2 Les systèmes autonomes robotisés	7
2.4 La vision par ordinateurs pour manoeuvrer les operations militaires	8
2.4.1 Définition	8
2.4.2 Le role de vision par ordinateur dans coté militaire	8
2.5 conclusion	8
3 DEEP LEARNING	9
3.1 Introduction	9
3.2 Définition de l'apprentissage profond (deep learning)	10
3.2.1 Pour quoi le deep learning?	10
3.3 Les applications du Deep Learning	11
3.3.1 La reconnaissance faciale	11
3.3.2 La détection d'objets	11
3.3.3 Le Natural Language Processing	11
3.4 Domaines d'application de l'apprentissage profonde	11

3.4.1	Conduite automatisée	12
3.4.2	Recherche médicale	12
3.4.3	Électronique	12
3.4.4	Recherche en marketing	12
3.5	Histoire de Deep Learning	12
3.6	Comment ça marche?	13
3.7	Réseaux de neurones	13
3.8	Principe de fonctionnement	14
3.9	L'apprentissage	15
3.9.1	Apprentissage supervisé	16
3.9.2	Apprentissage non supervisé	16
3.10	Réseaux de neurones convolutifs	17
3.10.1	Définition	17
3.10.2	L'architecture de CNN	18
3.10.3	La couche pooling (POOL)	19
3.10.4	Fonction d'activation	20
3.10.5	Couche entierement connecté	22
3.10.6	Couche de correction (ReLu)	23
3.11	Les modèles classiques de CNN	23
3.11.1	LeNet-5	23
3.11.2	Alexnet	24
3.11.3	VGGNet	25
3.12	Conclusion	27
4	CONTRIBUTION	29
4.1	Introduction	29
4.2	Architecture de système	29
4.3	Les composants de système	29
4.3.1	Les drones	29
4.3.2	Les capteurs	30
4.3.3	MAP	31
4.4	Classification des images	32
4.4.1	Présentation de notre modèles	33
4.5	conclusion	39
5	RESULTAS EXPERIMENTATION ET COMPARAISON	41
5.1	Introduction	41
5.2	Les outils et les librairies utilisés	41
5.2.1	Tensorflow	41

TABLE DES MATIÈRES

5.2.2	Keras	42
5.2.3	Python	43
5.2.4	Jupyter Notebook	44
5.3	Définition de Datasets	44
5.3.1	Military tanks dataset	44
5.4	Définition des paramètres	44
5.4.1	Batch size	45
5.4.2	Nombre d'Epoch	45
5.4.3	Learning rate	45
5.4.4	Optimizer	45
5.4.5	Loss function	46
5.4.6	Activation function	46
5.5	La définition de mesure d'évaluation	46
5.5.1	Rappel	46
5.5.2	Précision	46
5.5.3	F-mesure	47
5.6	Résultats de détection des objets avec yolo	47
5.6.1	Définition de YOLOv4	47
5.6.2	Fonctionnement de YOLOv4	47
5.6.3	Architecture de YOLOv4	50
5.7	Résultats du tank detection	52
5.8	Resultat alexnet	53
5.8.1	Variation du nombre d'époches	53
5.8.2	Alexnet optimizers comparaison	56
5.9	Résultats du modèle 1	60
5.9.1	Resultats avec variation du nombre d'époches	60
5.9.2	Résultat modèle 1 avec Variation du l'optimizer	63
5.10	Résultats du modèle VGG	67
5.10.1	Résultat du VGG avec variation du nombre d'époches	67
5.10.2	Résultat avec variation de l'optimizer	70
5.10.3	Résultats avec adam	70
5.10.4	Résultats avec admax	71
5.10.5	Résultat avec nadam	72
5.10.6	comparaison variation optimizer	74
5.11	Code source	75
5.12	Conclusion	76
6	Conclusion Général	77

Bibliographie

79

LISTE DES TABLEAUX

TABLE	Page
5.1 Résultats d'accuracy , macro avg et weighted avg pour modele 1	60
5.2 Résultats d 'accuracy et macro avrg en terme d'epoche	67
5.3 Tableau de code source	75

TABLE DES FIGURES

FIGURE	Page
2.1 La place de l'arme aérienne	4
2.2 Manœuvre terrestre	5
2.3 Manoeuvre navale	5
3.1 La relation entre l'IA,ML,Deep learning.	10
3.2 Topologie de réseau de neurones avec une seule sortie	14
3.3 Topologie de réseau de neurones profond.	15
3.4 Les réseaux neurones convolutionnel	17
3.5 Architecture de CNN	18
3.6 Exemple d'une operation de convolution	19
3.7 Max poll et average pool	20
3.8 La fonction sigmoid	21
3.9 La fonction soft max	21
3.10 La fonction relu	21
3.11 La structure de fully connected	22
3.12 Architecture de Lenet5	23
3.13 architecture de Alexnet	24
3.14 architecture de VGGNet	25
4.1 L'architecture générale d'un système militaire par l'intelligence artificielle	32
4.2 Modèle 01	34
4.3 Modèle 02	36
4.4 Modèle 03	38
5.1 Logo Tensorflow	42
5.2 Logo Keras	42
5.3 Logo Python	43
5.4 Logo jupyter	44
5.5 Code généré pour la détection des objets du premier plan avec yolo d'une frame avec la classe et les codonnées de chaque objet	48

TABLE DES FIGURES

5.6	Architcture du yolo v4	49
5.7	Résultats du yolo qui traite le problème de détection des objets dans une video	50
5.8	Architcture du yolo v4	51
5.9	Quelques exemples de TANK du dataset utilisé avec : a,b) alkhalid, b,f) amx, c,g) arjun mk2 d,h) leopard	52
5.10	Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour le modèle alexnet	53
5.11	Configuration et nombre de parametres modèle alexnet. Avec 42442368 paramètres divisé en deux (trainable params = 42440896, non trainable params = 1472)	54
5.12	Precision rappel et f1 score pour chaque classes tank detection utilisant le modèle1	55
5.13	Accuracy du alexnet avec variation du nombre d'époches et optimizer	56
5.14	Validation accuracy du alexnet avec variation du nombre d'époches et optimizer	57
5.15	Validation loss du alexnet avec variation du nombre d'époches et optimizer	58
5.16	Modèle alexnet comparaison entre les optimizers en termes de f-score, rappel et precision	59
5.17	Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour le modèle 1	60
5.18	Precision rappel et f1-score pour chaque classes tank detection utilisant le modèle1	61
5.19	Configuration et nombre de parametres modèle modèle1. Avec 27006786 paramètres divisé en deux (trainable params = 27005406, non trainable params = 1380)	62
5.20	Acuracy du modèle avec variation du nombre d'époches et optimizer	63
5.21	Validation accuracy du modèle1 avec variation du nombre d'époches et optimizer	64
5.22	validation accuracy du modèle1 avec variation du nombre d'époches et optimizer	65
5.23	Modèle1 comparaison entre les optimizers en termes de f-score, rappel et precision	66
5.24	Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour le modèle VGG-16	67
5.25	Evaluation par barchart en terme de precision recall, fmeasure de chaque classe	68
5.26	configuration et nombre de parametres modèle VGG-16. Avec 3345784 paramètres divisé en deux (trainable params = 3345400, non trainable params = 384)	69
5.27	courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour adam	70
5.28	courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour l'optimizer adamax	71
5.29	courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour l'optimizer nadam	72
5.30	Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour l'optimizer MSprop	73
5.31	Modèle VGG comparaison entre les optimizers en termes de f-score, rappel et precision	74

INTRODUCTION GENERALE

1.1 Problématique et motivation

L'intelligence artificielle (IA) est un domaine technologique en croissance rapide avec des implications potentiellement importantes pour la sécurité nationale. En tant que tels, l'Algérie comme d'autres pays développent des applications d'IA pour une gamme de fonctions militaires. Des recherches sur l'IA sont en cours dans les domaines de la collecte et de l'analyse du renseignement, de la logistique, des cybers opérations, des opérations d'information, du commandement et du contrôle, et dans une variété de véhicules semi-autonomes et autonomes. Comme exemple le plus concret, l'IA a été intégrée aux opérations militaires en Irak et en Syrie.

Pour les armées modernes, « l'intelligence artificielle (IA) se présente comme la voie principale de la supériorité tactique » et elle est devenue « un enjeu de défense prioritaire pour les puissances militaires du XXI^e siècle ». En reproduisant les processus cognitifs au moyen d'algorithmes et du traitement automatique du big data, L'IA est désormais capable d'effectuer un nombre grandissant de tâches spécifiques dans lesquelles elle surpasse les performances humaines. Appliqué au domaine militaire, elle permet de gérer et simuler l'environnement opérationnel, de détecter des menaces, de traiter et simplifier les masses de renseignement collectées et d'en livrer une analyse élémentaire.

L'IA, peut avoir des implications intéressantes sur la répartition de la puissance militaire à l'avenir. La possibilité d'avancées inaugurées par l'IA a ouvert le champ d'une course aux armements où les capacités militaires classiques auront beaucoup moins d'importance à mesure que le temps passe. L'exactitude et la précision des armes d'aujourd'hui obligent régulièrement les champs de bataille contemporains à se vider de combattants humains. Alors que de plus en plus de capteurs remplissent l'espace de bataille, renvoyant de grandes quantités de données aux analystes, les humains ont du mal à donner un sens à la montagne d'informations recueillies.

Notre travail est positionner à l'intersection de plusieurs disciplines : bigData, intelligence artificiel, internet des objets (présenter par les capteurs, les caméras de surveillance et les drones).

1.2 Objectifs

1- Traiter le problème de sécurité des frontières à travers l'analyse des personnes qui circule en détectant les activités humaines afin de repérer les personnes ayant un comportement anormal ou une scène à normale.

2- Renforcés la sécurité du payé avec la minimisation des ressources humaines au niveau des frontières.

3- Modernisé les équipements de notre armée algérienne.

4- Détection des objets du premier plan afin de faire la différence entre un objet intéressant à analyser (1er classe) ou non.

5- Régler le problème du manque des données utilisant le transfert Learning et le data augmentation.

6- Proposée des modèles CNN pour la classification des tanks militaire ou les personnes avec activité anormal.

7- Evaluer nos propositions utilisant des mesures de validation comme l'accuracy.

8- Référencier nos travaux en réalisant une comparaison avec d'autre modèles classiques qui existent dans la littérature comme alexnet, VGG-16, lenet5.

9- Réaliser une comparaison en termes de nombre de paramètres pour chaque modèle.

1.3 Organisation de la thèse

Notre thèse est organisée comme suit :

Chapitre 1 : Dans ce chapitre nous avons parlé Dans ce chapitre nous avons parlé des différents opérations militaires et les travaux de l'IA et computer vision dans la manipulation des opérations militaires. et les travaux de l'IA et computer vision dans la manipulation des opérations militaires.

Chapitre 2 : Dans le deuxième chapitre nous avons parlé du Deep Learning et les techniques classiques de CNN comme alexnet, VGG16.

Chapitre 3 : Dans ce chapitre nous avons présenté l'architecture de notre système et l'architecture des différents modèles.

Chapitre 4 : Dans ce chapitre, nous avons défini les data sets et définitions des mesures d'évaluations, et la discussion des résultats obtenus.

MANOEUVRER LES OPEARTIONS MILITAIRES

2.1 Introduction

Aujourd'hui, Intelligence Artificielle pénètre tous les domaines de la vie, en conjonction de cette pénétration le domaine militaire est à l'avant-garde de ces domaines. C'est parce que IA joue un rôle important dans les opérations militaires et même dans le renforcement des capacités militaires traditionnelles et développées, soit au niveau opérationnel soit au niveau tactique. - Au niveau opérationnel : IA améliore les capacités grâce à des capacités (télédétection, connaissance en temps réel des variables, manœuvres, prises de décisions en sous-pression). .

- Au niveau stratégie tactiques : IA prise la décision militaire.

En plus de lui fournir aux opérations militaires des nouveaux techniques qui optimisent ses capacités telles que : la perception visuelle, la reconnaissance vocale. Ainsi que l'utilisation des algorithmes prenaient décision pour mettre en œuvre un groupe d'opération (aérienne, terrestre, navale). Ceci est indépendant de la supervision et de l'intervention humaine. L'intégration de l'IA dans les opérations militaires conduira un changement radical du concept de sécurité, que ce soit la sécurité numérique (hameçonnage ciblé, fabrication phonémique, usurpation d'identité, piratage robotiques, intrusion des données), ou la sécurité physique (les attaques d'essaims de drones).

[37]

2.2 Les différentes opérations militaires

Les opérations militaires : sont des actions de force armées, qui étaient planifiées et qui peuvent entraîner un engagement de forces aériennes, navales ou terrestre, elle s'inscrit dans le

cadre d'une stratégie militaire même soit par un état ou un territoire.

Les opérations militaires catégorisent ont plusieurs types comme les opérations militaires aériennes, les opérations militaires terrestres, les opérations militaires navales. 1- Les opérations militaires aériennes : sont des opérations menées par des moyennes aériennes. on distingue (les opérations tactiques, les opérations aériennes stratégiques).

[23]



FIGURE 2.1: La place de l'arme aérienne

[10]

2- Les opérations militaires terrestres : sont des opérations menées par les troupes au sol. On distingue généralement :

- Les opérations offensives.
- Les opérations défensives.
- Les opérations spéciales.

[20]



FIGURE 2.2: Manœuvre terrestre

3 - Les opérations navales : combats sur mer Les opérations navales : combats sur mers et les océans, la surface de la mer est à la distance de l'horizon.



FIGURE 2.3: Manoeuvre navale

[20]

2.3 L'Intelligence artificielle et les opérations militaires

L'IA est déjà en train de devenir une réalité pour les armées du monde entier. Toutes les forces armées modernes réfléchissent à tout le moins aux enjeux de l'IA, y compris sur les plans éthique et juridique, et un grand nombre d'entre elles se dotent déjà de solutions concrètes intégrant cette technologie. [41]

2.3.1 Le soutien en matière d'information et prise décision

L'être humain évolue généralement dans un environnement où les informations disponibles sont partielles. Il arrive parfois que ces informations soient pléthoriques, mais ce n'est définitivement pas le cas dans le domaine militaire et stratégique. Les responsables politiques et militaires doivent intervenir dans le fameux « brouillard de la guerre ». Le soutien pouvant être apporté par les systèmes d'IA en matière d'information et de décision présente donc un grand intérêt pour les décideurs militaires et stratégiques. Ces systèmes peuvent accroître sensiblement la vitesse et la qualité de traitement, d'exploitation et de diffusion des informations, ainsi que du processus décisionnel des êtres humains et de machines. [25]

Dans le domaine militaire, l'IA peut réduire considérablement le délai d'analyse et d'action des êtres humains et des machines. Les systèmes intégrant l'IA et apportant un soutien en matière d'information et de décision peuvent par exemple :

- Réduire considérablement les temps de réaction des systèmes de défense en cas d'attaque par des systèmes d'armes d'action rapide (missiles hypersoniques, cybers attaques ou armes à énergie dirigée).

- Fournir plus rapidement aux décideurs des informations exploitables, pouvant procurer un avantage déterminant sur les adversaires .

- Repérer rapidement des cybers intrusions en détectant des codes malveillants évasifs ou des types de comportements suspects (plutôt que des codes particuliers).

- Aider à repérer les tentatives de manipulation des citoyens menées via des campagnes de désinformation. [26]

L'IA peut améliorer la qualité de la prise de décision, non seulement par les machines mais aussi c'est peut être le plus important par les humains. La capacité de l'IA à filtrer l'environnement riche en informations d'aujourd'hui et à communiquer des résultats de manière convaincante est à cet égard capitale et deviendra encore plus importante. Les solutions intégrant l'IA peuvent apporter une aide comme : - Fournir une meilleure visualisation et interprétation des données. -Extraire automatiquement des flux de données (par exemple des images satellites ou de vidéosurveillance) des informations présentant de l'intérêt pour les actions de suivi. [25]

- Mettre en évidence les anomalies en vue de recherches ultérieures en comparant les points de données avec des modèles de normalité préétablis ; - Extraire des « signaux faibles » qui ne semblent pas préoccupants en soi mais qui peuvent être très significatifs s'ils sont combinés avec d'autres données.

- Suggérer un ensemble d'options possibles et décrire les effets probables de chacune d'elles.

- Fournir des renseignements sur le comportement de l'adversaire grâce à des informations anticipées.

2.3.2 Les systèmes autonomes robotisés

L'essor de l'IA a coïncidé en grande partie avec la prolifération rapide des systèmes autonomes robotisés qui, à eux seuls, transforment le paysage militaire et stratégique. Cela n'est guère surprenant étant donné que l'IA est la technologie de base utilisée dans ces systèmes. Les systèmes autonomes robotisés sont notamment utilisés pour la destruction d'engins explosifs, les opérations antimines en milieu terrestre ou sous-marin, les missions de sauvetage, l'appui logistique, voire les opérations de combat. L'autonomisation croissante des systèmes militaires pourrait à l'avenir avoir un impact considérable sur les structures de forces. L'intégration de systèmes autonomes robotisés dans les formations de combat permettrait par exemple de réduire sensiblement les effectifs d'une unité. L'autonomisation pourrait également transformer en profondeur les concepts opérationnels. Un très grand nombre de systèmes autonomes robotisés pourraient ainsi être utilisés pour submerger les postures de déni d'accès/interdiction de zone. Au niveau stratégique, le déploiement de tels systèmes pourrait donner aux pays un net avantage militaire et changer la nature de la guerre.

[11]

Les humains et les algorithmes d'IA ont des forces et des faiblesses différentes. Il y a des domaines où les humains sont plus performants que les machines. Dans d'autres, ce sont les machines qui sont plus performantes. Et dans d'autres encore, les humains et les machines fonctionnent tout aussi bien ou tout aussi mal. Le travail en équipe humain-machine vise donc à trouver le juste équilibre entre l'humain et la machine dans l'exécution des tâches. Dans certaines situations, l'équipe humain-machine obtiendra de meilleurs résultats si l'humain est « dans la boucle », lorsqu'une grande partie du contrôle est conservée par le personnel militaire. Les opérateurs doivent conserver un niveau de contrôle humain approprié pour toute décision impliquant un recours à la force. [11]

Cependant, les humains peuvent très bien choisir de conserver ce contrôle aussi dans d'autres situations. Par exemple, en période de crise politique aiguë, les opérateurs voudraient probablement conserver plus de contrôle sur les véhicules aériens sans pilote à proximité de territoires hostiles. Dans d'autres situations, les opérateurs pourraient simplement choisir d'être « dans le circuit ». Par exemple, lorsque des camions télé-pilotes suivent un véhicule de tête habité, l'opérateur peut vouloir surveiller de près le convoi, mais n'intervenir qu'en cas de problème. Dans d'autres circonstances encore, les opérateurs peuvent décider de rester « exclus de la boucle ». Un véhicule sous-marin sans pilote, par exemple, pourrait être programmé pour explorer l'environnement océanique de façon autonome, tout en maintenant le silence radio pour éviter de se faire repérer.

[11]

2.4 La vision par ordinateurs pour manoeuvrer les operations militaires

2.4.1 Définition

La vision par ordinateur est une branche de l'Intelligence Artificiel, son objectif de la vision par ordinateur est de faire en sorte que les ordinateurs voient et interprètent le monde comme des humains et peut-être même mieux que nous. La vision humaine effectue de multiples tâches visuelles sans effort et efficacement.

[29]

2.4.2 Le role de vision par ordinateur dans coté militaire

Les applications militaires sont probablement l'un des domaines les plus importants de la vision par ordinateur.

Les exemples évidents sont la détection de soldats ou de véhicules ennemis et le guidage de missiles. Des systèmes plus avancés de guidage de missile envoient le missile vers une zone plutôt qu'une cible spécifique, et la sélection de la cible est effectuée lorsque le missile atteint la zone grâce à des données d'image acquises localement. [29]

Les concepts militaires modernes, tels que la «conscience du champ de bataille», impliquent que divers capteurs, y compris des capteurs d'image, fournissent un ensemble riche d'informations sur une scène de combat qui peuvent ensuite être utilisées pour prendre des décisions stratégiques. Dans ce cas, le traitement automatique des données est utilisé pour réduire la complexité et pour fusionner les informations de plusieurs capteurs dans le but d'augmenter la fiabilité. [21]

2.5 conclusion

Le développement de technologies dans tous les domaines et l'avancement rapide de l'intelligence artificielle avec ces branches comme la vision par ordinateur donnent un autre sens aux les opérations militaires et les guerres en général. Donc avec tous ces techniques la machine essaie de remplacer le rôle de l'homme et améliore les tactiques des guerres et diminuer le risque et bien sure gagner du temps. En conclure, l'incorporation de l'IA dans les guerres vas provoquer des risques dans le coté de sécurité.

DEEP LEARNING

3.1 Introduction

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage automatique champ d'étude de l'intelligence artificielle. Par conséquent, L'apprentissage profond (en anglais Deep Learning, Deep Structured Learning, Hierarchical Learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond.

Le Deep Learning est un nouveau domaine de recherche du ML, qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

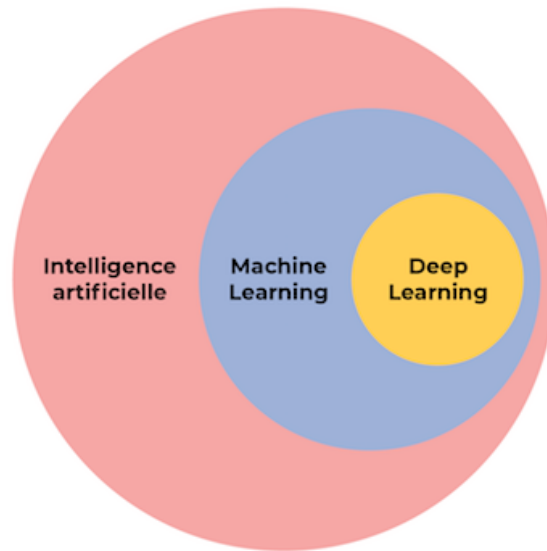


FIGURE 3.1: La relation entre l'IA,ML,Deep learning.

3.2 Définition de l'apprentissage profond (deep learning)

L'apprentissage profond (« Deep Learning ») est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. [35] Par exemple, pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles. . . des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes. . . Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données.

3.2.1 Pour quoi le deep learning ?

Les algorithmes de ML décrits dans la première partie fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoués à résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance vocale et la reconnaissance d'objets. Le développement du Deep Learning fut motivé en partie par l'échec des algorithmes traditionnels dans de telle tâche de l'IA. Mais ce n'est qu'après que de plus grandes quantités de données ne soit disponibles grâce notamment au Big Data et aux objets connectés et que les machines de calcul soient devenues

plus puissantes qu'on a pu comprendre le potentiel réel du Deep Learning. [17]

3.3 Les applications du Deep Learning

C'est une branche du Machine Learning très prometteuse. Que ce soit pour reconnaître des visages sur des images, analyser des textes et les interpréter automatiquement ou encore avoir des voitures qui conduisent toutes seules, les applications du Deep Learning sont nombreuses. Aujourd'hui, nous vous proposons un workshop qui va vous permettre de comprendre ce domaine et voir en quoi nous pourrions concrètement l'utiliser. [52]

3.3.1 La reconnaissance faciale

Les yeux, le nez, la bouche, toutes autant de caractéristiques qu'un algorithme de Deep Learning vont apprendre à détecter sur une photo. Il va s'agir en premier lieu à donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.

3.3.2 La détection d'objets

Sur une image complexe où il y a plusieurs éléments, les algorithmes de détection d'objets vont être maintenant capables d'identifier et de localiser au pixel près un élément ou une personne. 800 millions d'images sont uploadées chaque jour sur Facebook : son algorithme Deep Learning est effectivement capable d'identifier telle ou telle personne sur une photo dès lors qu'elle est uploadée

3.3.3 Le Natural Language Processing

Le Natural Language Processing est une autre application du Deep Learning. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chatbots. La lecture et l'analyse automatique de textes est aussi un des champs d'application du Deep Learning avec le TopicModeling : tel texte aborde tel sujet.

3.4 Domaines d'application de l'apprentissage profonde

L'apprentissage profond à de nombreuses applications en informatique, on cite quelques domaines :

3.4.1 Conduite automatisée

Les chercheurs du secteur automobile ont recours au Deep Learning pour détecter automatiquement des objets tels que les panneaux stop et les feux de circulation. Le Deep Learning est également utilisé pour détecter les piétons, évitant ainsi nombre d'accidents.

3.4.2 Recherche médicale

À l'aide du Deep Learning, les chercheurs en oncologie peuvent dépister automatiquement les cellules cancéreuses. Des équipes de l'Université de Californie à Los Angeles (UCLA) ont conçu un microscope qui génère un ensemble de données de grandes dimensions afin d'entraîner une application de Deep Learning à identifier avec précision des cellules cancéreuses.

3.4.3 Électronique

Le Deep Learning est utilisé pour la reconnaissance audio et vocale. Par exemple, les appareils d'assistance à domicile qui répondent à votre voix et connaissent vos préférences fonctionnent grâce à des applications de Deep Learning.

3.4.4 Recherche en marketing

Recherche en marketing : En plus de rechercher de nouvelles fonctionnalités susceptibles d'améliorer votre application, DL peut également être utile en arrière-plan. La segmentation du marché, l'analyse des campagnes marketing et bien d'autres peuvent être améliorés à l'aide de modèles de régression et de classification DL. Cela vous aidera vraiment beaucoup si vous avez une grande quantité de données. Sinon, vous ferez probablement mieux d'utiliser des algorithmes traditionnels d'apprentissage automatique pour ces tâches plutôt que DL. [36]

3.5 Histoire de Deep Learning

Depuis 2012, les algorithmes à base de deep learning (apprentissage profond) semblent prêts à résoudre bien des problèmes : reconnaître des visages comme le propose DeepFace, vaincre des joueurs de go ou de poker ou bientôt permettre la conduite de voitures autonomes ou encore la recherche de cellules cancéreuses.

Pourtant, les fondements de ces méthodes ne sont pas si récents : le deep learning a été formalisé en 2007 à partir des nouvelles architectures de réseaux de neurones dont les précurseurs sont McCulloch et Pitts en 1943. Suivront de nombreux développements comme le perceptron, les réseaux de neurones convolutifs de Yann Le Cun et Yoshua Bengio en 1998 et les réseaux de neurones profonds qui en découlent en 2012 et ouvrent la voie à de nombreux champs

d'application comme la vision, le traitement du langage ou la reconnaissance de la parole. Pourquoi maintenant? parce que ces nouvelles techniques de machine learning profitent de données massives (big data) que l'on est désormais capables d'analyser ainsi que de capacités de calcul phénoménales notamment grâce aux processeurs graphiques. Preuve que chaque domaine irrigue les autres, c'est pour pouvoir utiliser les immenses promesses du deep learning que Google a mis au point les accélérateurs TPU. [33]

3.6 Comment ça marche ?

Pour comprendre comment fonctionne le Deep Learning, nous allons utiliser un exemple concret de reconnaissance faciale. Imaginons que notre objectif soit de lui faire reconnaître les photos qui comportent une voiture.

Pour pouvoir reconnaître une voiture, l'algorithme doit d'une part savoir distinguer tous les types de voitures existantes, mais aussi savoir identifier une voiture de manière précise et autonome, quel que soit l'angle sous lequel elle se trouve.

Pour y arriver c'est assez simple : le réseau de neurones artificiels est entraîné en analysant des milliers d'images de voitures et apprend à les reconnaître au milieu de photos d'autres objets.

Ces données vont ensuite être assignées à différentes informations permettant à l'algorithme intelligent de déduire si oui ou non se trouve une voiture sur l'image qu'il est en train d'analyser. Le réseau artificiel va également comparer cette réponse aux bonnes réponses indiquées par les humains. Si il a vu juste, l'algorithme de reconnaissance garde cette réussite en mémoire et s'en resserrera plus tard pour reconnaître des voitures. Au contraire, s'il s'est trompé, il en prend note et corrige son erreur de lui-même la fois suivante.

C'est en répétant ce système d'entraînement des milliers de fois que le réseau de neurones finit par être capable de reconnaître une voiture dans toutes circonstances (avec un degré de réussite proportionnel à la durée d'entraînement du réseau et au nombre de couches qu'il possède).

Cette technique d'apprentissage est appelée apprentissage supervisé ou "supervised learning". [49]

3.7 Réseaux de neurones

Avec l'augmentation rapide des capacités de calcul des ordinateurs et l'évolution des techniques d'apprentissage, l'utilisation des réseaux de neurones connaît un vif essor en particulier au sein des communautés du traitement d'image, de la traduction automatique et du traitement de la parole.

Au cours des dernières années, deux types de réseaux de neurones ont marqué une rupture technologique dans le domaine du traitement de la parole : les réseaux dits "profonds" et les réseaux récurrents. Nous détaillons dans cette partie beaucoup plus le fonctionnement de réseaux

de neurones et les travaux que nous avons réalisés sur les modèles eux-mêmes et sur les méthodes d'apprentissage. [27]

3.8 Principe de fonctionnement

Le réseau comporte 3 composants : couche d'entrée couche cachée ou couches et couche de sortie. Le terme « profond » se rapporte généralement au nombre de couches cachées du réseau de neurones. Les réseaux de neurones classiques ne comportent que 2 à 3 couches cachées, tandis que les réseaux profonds peuvent en compter jusqu'à 150

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

Le réseau de neurones le plus connu et le plus simple à comprendre est le réseau de neurones multicouches à anticipation. Il contient un calque d'entrée, un ou plusieurs calques masqués et un seul calque de sortie. Chaque couche peut avoir un nombre différent de neurones et chaque couche est entièrement connectée à la couche adjacente. [9]

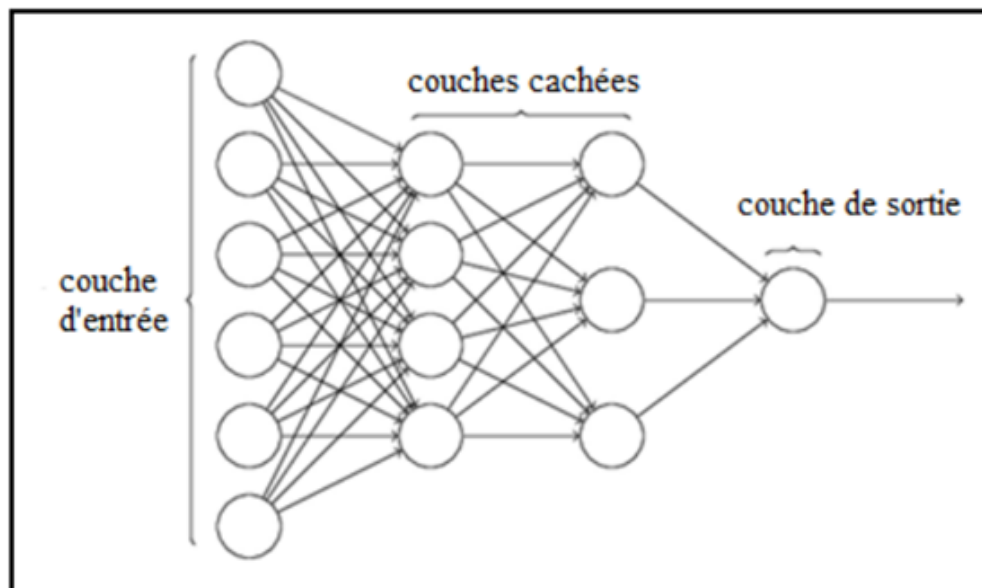


FIGURE 3.2: Topologie de réseau de neurones avec une seule sortie

[16]

Un réseau de neurones est défini comme un ensemble de nœuds (appelés neurones) connectés via des liaisons dirigées (flèche), chaque flèche représente une connexion entre la sortie d'un neurone et l'entrée d'un autre (les flèches entrantes étant les entrées du neurone et les flèches sortantes étant les sorties du neurone), Chaque flèche porte un poids, reflétant son importance, chaque

nœud étant une unité de traitement qui exécute une fonction de nœud statique sur son signal entrant pour générer une sortie de nœud unique

. Les valeurs d'entrée, ou en d'autres termes, nos données sous-jacentes, sont transmises via ce «réseau» de couches masquées jusqu'à ce qu'elles convergent vers la couche de sortie. La couche en sortie correspond à notre prédiction : il peut s'agir d'un nœud si le modèle ne génère qu'un nombre ou de quelques nœuds s'il s'agit d'un problème de classification multi-classe. La forme à l'intérieur des neurones dans les couches centrales représente une fonction d'activation (typiquement un $1 = (1 + e^{-x})$) qui est appliquée à la valeur du neurone avant de le transmettre à la sortie. [16]

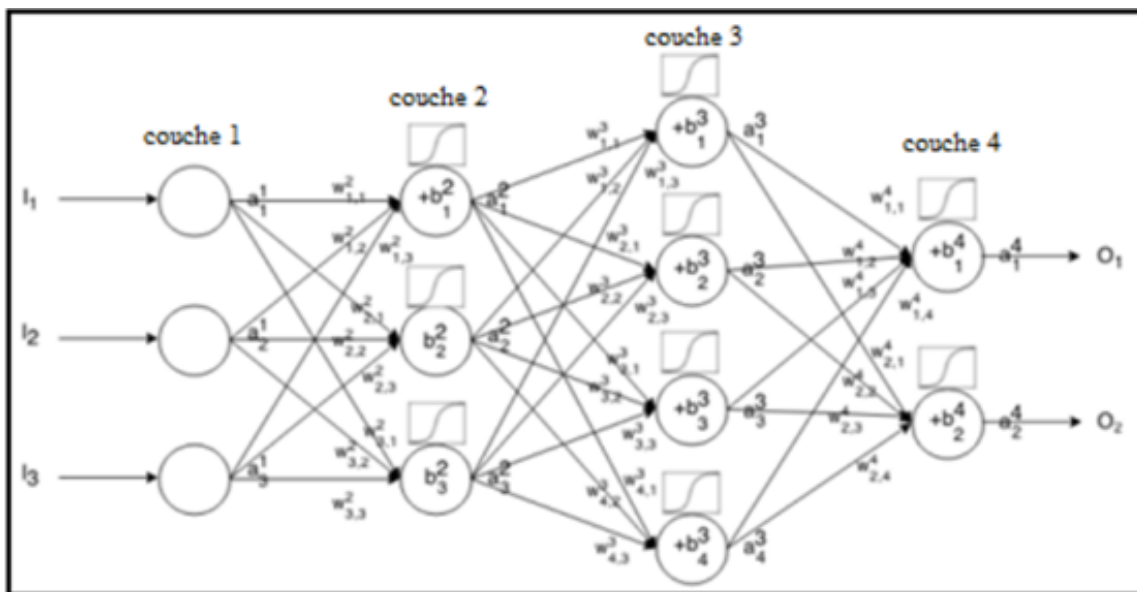


FIGURE 3.3: Topologie de réseau de neurones profond.

[24]

Les couches cachées d'un réseau de neurones apportent des modifications aux données pour éventuellement déterminer quelle est sa relation avec la variable cible. Chaque nœud a un poids et multiplie sa valeur d'entrée par ce poids. Pour déterminer ce que devraient être ces petits poids, nous utilisons généralement un algorithme appelé Back propagation.

3.9 L'apprentissage

En effet, les réseaux de neurones peuvent trouver le lien qui unit des valeurs de sortie à celles en entrée, et ce, même lorsqu'on ne connaît pas cette fonction a priori. Afin de parvenir à ce résultat, il faut « entraîner » le modèle, à l'aide d'un jeu de données initiales.

Celui-ci est alors divisé en deux parties : une pour l'apprentissage, l'autre pour tester le réseau. Dans un premier temps, on va donc soumettre des données d'entraînement au programme,

qui comprennent des valeurs d'entrée, ainsi que les valeurs de sortie attendues. Au début, le réseau de neurones va tenter de calculer les résultats, mais avec peu d'informations, et va donc commettre des erreurs. On va ensuite ajuster ses paramètres, de sorte à réduire ces écarts à chaque itération.

Par exemple, un réseau de neurones peut être utilisé pour prévoir les risques d'apparition d'une maladie chez certains individus. En lui soumettant les caractéristiques de patients, le modèle va d'abord effectuer des prédictions aléatoires. Puis, en apprenant de ses erreurs, il sera de plus en plus pertinent dans son analyse. [48]

3.9.1 Apprentissage supervisé

La majorité des apprentissages automatiques utilisent un apprentissage supervisé (supervised learning).

L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y). Vous utilisez un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie.

$$Y = f(X)$$

Le but est d'appréhender si bien la fonction de mapping que, lorsque vous avez de nouvelles données d'entrée (x), vous pouvez prédire les variables de sortie (Y) pour ces données.

C'est ce qu'on appelle l'apprentissage supervisé, car le processus d'un algorithme tiré de l'ensemble de données de formation (training set) peut être considéré comme un enseignant supervisant le processus d'apprentissage. Nous connaissons les réponses correctes, l'algorithme effectue des prédictions itératives sur les données d'apprentissage et est corrigé par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable. [31]

3.9.2 Apprentissage non supervisé

L'apprentissage non supervisé (Unsupervised Learning) consiste à ne disposer que de données d'entrée (X) et pas de variables de sortie correspondantes.

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données.

On l'appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponse correcte ni d'enseignant. Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données.

L'apprentissage non supervisé comprend deux catégories d'algorithmes : Algorithmes de regroupement et d'association.

[18]

3.10 Réseaux de neurones convolutifs

3.10.1 Définition

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu] [13]

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

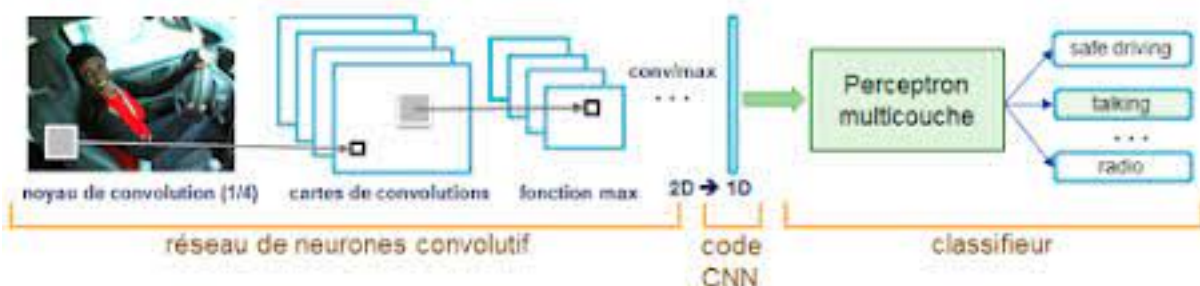


FIGURE 3.4: Les réseaux neurones convolutionnel

[1]

3.10.2 L'architecture de CNN

Les CNN sont composés de trois types de couches : des couches convolutives, des couches de regroupement et des couches entièrement connectées. Lorsque ces couches sont empilées, une architecture CNN a été formée [1]

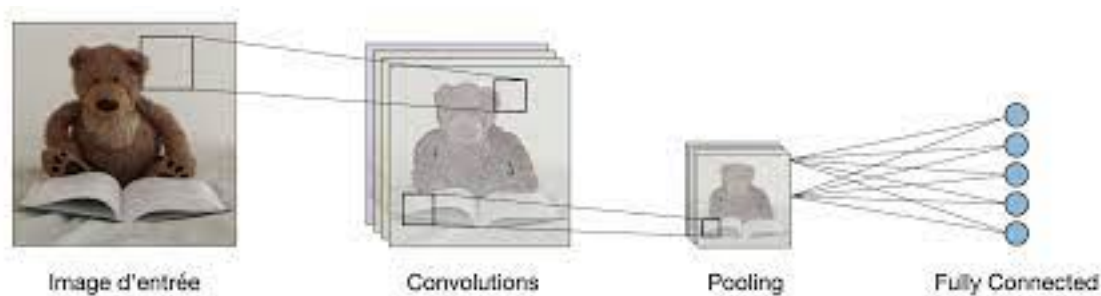


FIGURE 3.5: Architecture de CNN

3.10.2.1 La couche de convolution (CONV)

Quand on lui présente une nouvelle image, le CNN ne sait pas exactement si les caractéristiques seront présentes dans l'image ou où elles pourraient être, il cherche donc à les trouver dans toute l'image et dans n'importe quelle position. En calculant dans toute l'image si une caractéristique est présente, nous faisons un filtrage. Les mathématiques que nous utilisons pour réaliser cette opération sont appelées une convolution, de laquelle les réseaux de neurones à convolution tiennent leur nom. La couche de convolution est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée

Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, On obtient pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature . Finalement, les valeurs des dernières feature maps sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second. Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution :

1. Profondeur de la couche K : nombre de noyaux de convolution
2. Le pas S : contrôle le chevauchement des champs récepteurs
3. La marge (à 0) ou zero padding P : Cette marge permet de contrôler la dimension spatiale du volume de sortie. Pour chaque image de taille $W \times H \times D$ en entrée, où W est sa largeur en pixels, H sa hauteur en pixels et D le nombre de canaux (1 pour une image en noir et blanc, 3 pour une image en couleurs) la couche de convolution renvoie une matrice de dimensions $WC \times HC \times DC$, et

$$WC = \frac{W - f + 2P}{S} + 1$$

$$HC = \frac{H - f + 2P}{S}$$

et $DC=K$

Où F est la taille des filtres et le S c'est le pas avec lequel on fait glisser la fenaitre, le P correspond au zéro padding, et K c'est le nombre des filtres

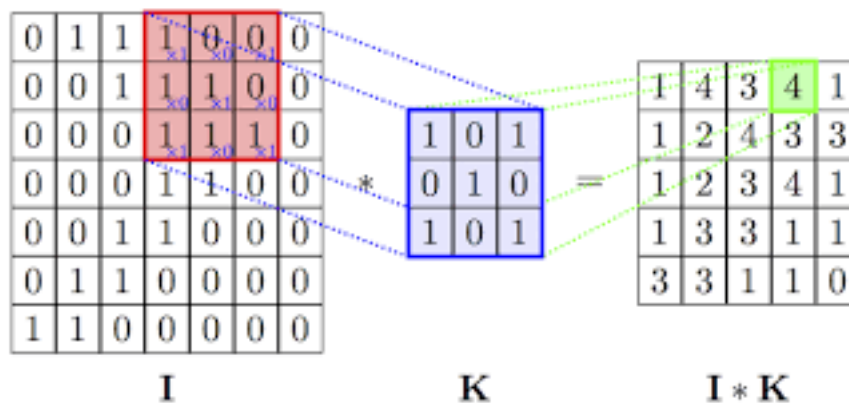


FIGURE 3.6: Exemple d'une operation de convolution

[6]

Choisir

$$P = \frac{F - 1}{2}$$

et $S=1$ permet ainsi d'obtenir des feature maps de même largeur et hauteur que

celles reçues en entrée Ou F est la taille du filtre, chaque filtre est de dimensions $F \times F \times D$.

[43]

3.10.3 La couche pooling (POOL)

Une architecture atypique d'un réseau convolutif se compose de trois types de couches différentes. D'abord une couche convolutive pour générer un ensemble d'activations linéaires ensuite, on les fait passer à travers une couche d'activation non linéaire telle que Rectified Linear Unit (ReLU), enfin on utilise la fonction pooling.

Permet de réduire progressivement la taille des représentations afin de réduire la quantité de paramètres et de calcul dans le réseau et, par conséquent, de contrôler également le surapprentissage.

Permet l'invariance aux petites translations.

Utile lorsqu'on préfère savoir si une caractéristique est présente plutôt que la région de sa présence.

[42]

IL existe plusieurs types de pool :

1 Max pooling : est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) en réduisant sa dimension. De plus, son intérêt est qu'il réduit le coût de calcul en réduisant le nombre de paramètres à apprendre et fournit une invariance par petites translations (si une petite translation ne modifie pas le maximum de la région balayée, le maximum de chaque région restera le même et donc la nouvelle matrice créée restera identique).

2 Average pooling :

L'opération consiste à conserver à chaque pas, la valeur moyenne de la fenêtre de filtre. [42]



FIGURE 3.7: Max poll et average pool

3.10.4 Fonction d'activation

Une fonction d'activation est une fonction mathématique utilisé sur un signal. Elle va reproduire le potentiel d'activation que l'on retrouve dans le domaine de la biologie du cerveau humain. Elle va permettre le passage d'information ou non de l'information si le seuil de stimulation est atteint. Concrètement, elle va avoir pour rôle de décider si on active ou non une réponse du neurone. Un neurone ne va faire qu'appliquer la fonction suivante : $X = (\text{entrée} * \text{poids}) + \text{biais}$

Voici quelque exemples des fonctions d'activation :

1 Sigmoid (logistic) : Fonction la plus populaire depuis des décennies. Mais aujourd'hui, elle devient beaucoup moins efficace par rapport à d'autre pour une utilisation pour les couches cachées. Elle perd de l'information due à une saturation que cela soit pour la phase de feed forward ou de backpropagation, en donnant des effets non linéaires au réseau due à un paramètre unique. Elle a aussi des soucis de gradient 0 avec des entrées étant très large, même si le soucis

est minimisé avec le système utilisant des batch par lots (mini batch). Utilisé en couche de sortie pour de la classification binaire. Intervalle de sortie : 0,1 [6]

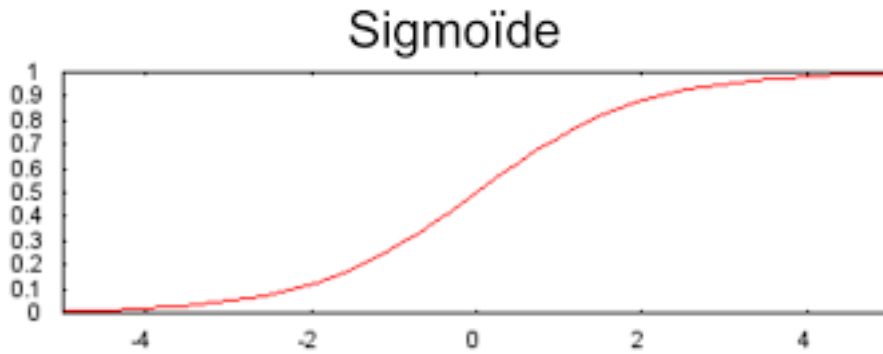


FIGURE 3.8: La fonction sigmoid

2 Softmax : Utilisé pour de la multi classification en couche de sortie. Intervalle de sortie (-;+). [6] 3 ReLU (Rectified Linear Unit) : Ce sont les fonctions les plus populaires de nos jours.

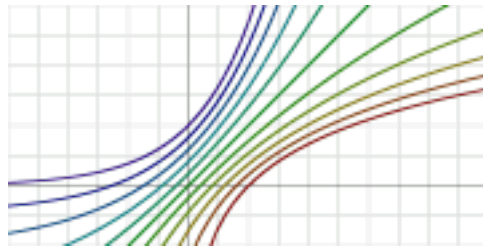


FIGURE 3.9: La fonction soft max

Elles permettent un entraînement plus rapide comparé aux fonctions sigmoid et tanh, étant plus légères. Attention au phénomène de 'Dying ReLU', auquel on préférera les variations de ReLU. Plus d'informations en fin d'article. Très utilisé pour les CNN, RBM, et les réseaux de multi perceptron. Intervalle de sortie (0;+). [51]

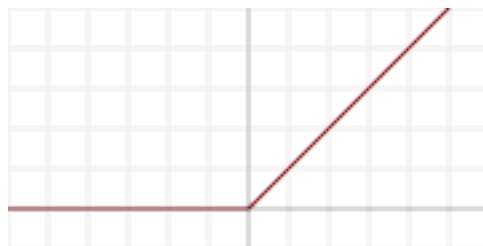


FIGURE 3.10: La fonction relu

3.10.5 Couche entièrement connecté

Enfin, après les couches de convolution et pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches totalement connectées. Dans les réseaux de neurones convolutifs, chaque couche agit comme un filtre de détection pour la présence de caractéristiques spécifiques ou de motifs présents dans les données d'origine. Les premières couches d'un réseau convolutif détectent des caractéristiques qui peuvent être reconnues et interprétées facilement. Les couches ultérieures détectent de plus en plus des caractéristiques plus abstraites. La dernière couche du réseau convolutif est capable de faire une classification ultra-spécifique en combinant toutes les caractéristiques spécifiques détectées par les couches précédentes dans les données d'entrée. Les couches totalement connectées font les mêmes tâches que celles des ANN standard et tenteront de produire des notes de classe à partir des activations, pour les utiliser pour la classification. Il est également suggéré d'utiliser ReLu entre ces couches pour améliorer les performances

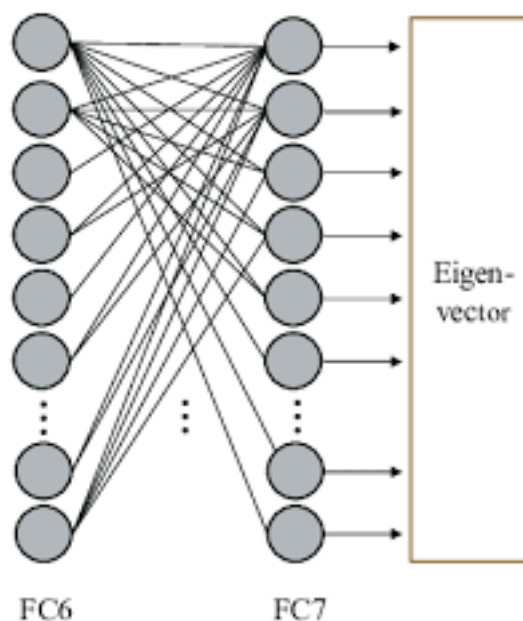


FIGURE 3.11: La structure de fully connected

[6]

3.10.6 Couche de correction (ReLU)

La couche de perte suit la couche entièrement connectée et gère les ajustements de poids sur le réseau. Avant que l'entraînement du réseau ne commence, les poids des couches convolution et entièrement connectées reçoivent des valeurs aléatoires. Ensuite, pendant l'entraînement, la couche de perte vérifie en permanence les suppositions de la couche entièrement connectée par rapport aux valeurs réelles dans le but de minimiser autant que possible la différence entre l'estimation et la valeur réelle. La couche de perte effectue cela en ajustant les poids à la fois dans la couche de convolution et dans les couches entièrement connectées. [50]

3.11 Les modèles classiques de CNN

3.11.1 LeNet-5

3.11.1.1 Définition

LeNet-5, un réseau convolutif pionnier à 7 niveaux par LeCun et al en 1998, qui classe les chiffres, a été appliqué par plusieurs banques pour reconnaître les numéros écrits à la main sur les chèques (chèques) numérisés en images d'entrée 32x32 pixels en niveaux de gris. La capacité de traiter des images de résolution plus élevée nécessite des couches plus grandes et plus convolutives, de sorte que cette technique est limitée par la disponibilité des ressources informatiques. [34]

3.11.1.2 Architecture de Lenet5

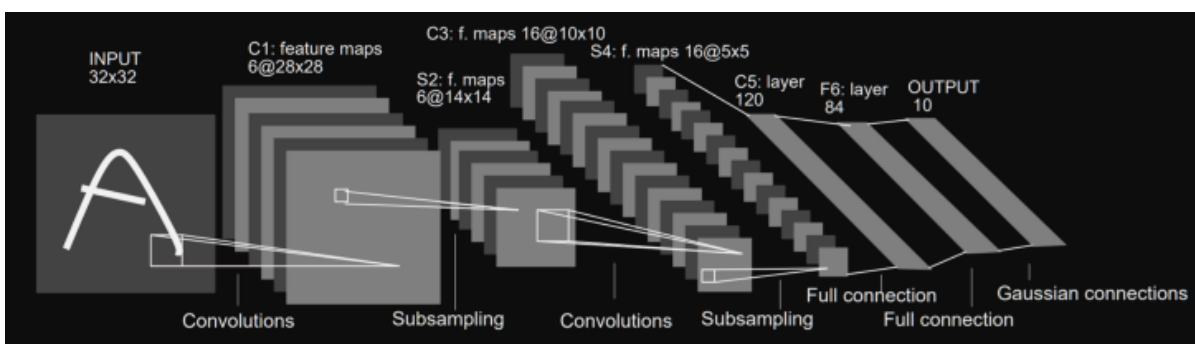


FIGURE 3.12: Architecture de Lenet5

[6] La première couche est la couche d'entrée - elle n'est généralement pas considérée comme une couche du réseau car rien n'est appris dans cette couche. La couche d'entrée est conçue pour prendre en 32x32, et ce sont les dimensions des images qui sont passées dans la couche suivante. Ceux qui connaissent le jeu de données MNIST savent que les images du jeu de données MNIST

ont les dimensions 28x28. Pour que la dimension des images MNIST soit conforme aux exigences de la couche d'entrée, les images 28x28 sont remplies .

Les images en niveaux de gris utilisées dans le document de recherche avaient leurs valeurs de pixels normalisées de 0 à 255, à des valeurs comprises entre -0,1 et 1,175. La raison de la normalisation est de s'assurer que le lot d'images a une moyenne de 0 et un écart type de 1, les avantages de cela se voient dans la réduction du temps d'apprentissage. Dans l'exemple de classification d'images avec LeNet-5 ci-dessous, nous normaliserons les valeurs de pixels des images pour qu'elles prennent des valeurs comprises entre 0 et 1. [34]

3.11.2 Alexnet

3.11.2.1 Définition

AlexNet est le nom de l'architecture d'un réseau profond de neurones à convolutions qui a remporté l'épreuve ImageNet en 2012. Conçu par une équipe de l'Université de Toronto dirigée par Geoffrey Hinton, dont faisaient partie Alex Krizhevsky, architecte principal qui lui a donné son nom, et Ilya Sutskever, AlexNet a marqué un point tournant dans l'emploi des réseaux profonds de neurones. [34]

3.11.2.2 Architecture de AlexNet

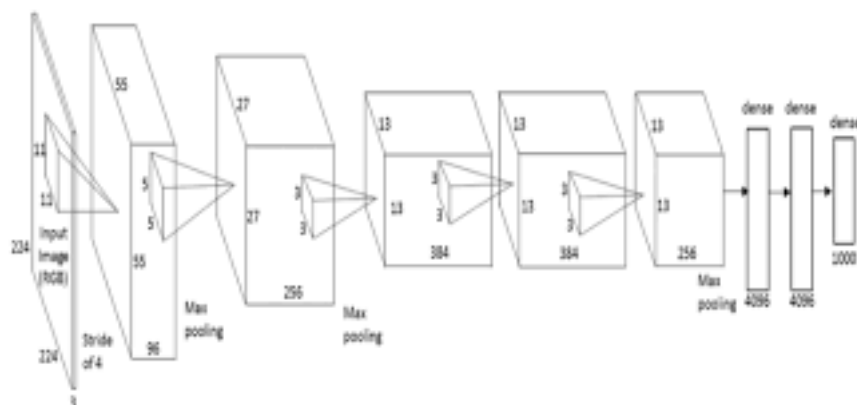


FIGURE 3.13: architecture de Alexnet

[6] 1ère couche convolutive : cette couche filtre l'image d'entrée ($224 * 224 * 3$) avec 96 noyaux de taille $11 * 11 * 3$.

Couche convolutive 2ed : cette couche filtre la sortie de la 1ère couche convolutive qui est normalisée localement en réponse et regroupée avec 256 noyaux de taille $5 * 5 * 96$.

Couche convolutive 3ed : pas de normalisation et de regroupement de la réponse locale. Il existe 384 noyaux de taille $3 * 3 * 256$.

4ème couche convolutive : pas de normalisation et de regroupement de la réponse locale. Il existe

384 noyaux de taille $3 * 3 * 384$.

5ème couche convolutive : dans cette couche, la sortie sera mise en commun. Il existe 256 noyaux de taille $3 * 3 * 384$.

1ère, 2ème couches entièrement connectées : chacune contient 4096 neurones.

3ème couche entièrement connectée : c'est la couche de sortie «alimentée à un softmax 1000 voies qui produit une distribution sur les 1000 étiquettes de classe».

[12]

3.11.3 VGGNet

3.11.3.1 Définition

Le finaliste du concours ILSVRC 2014 est surnommé VGGNet par la communauté et a été développé par Simonyan et Zisserman. VGGNet se compose de 16 couches convolutives et est très attrayant en raison de son architecture très uniforme. Similaire à AlexNet, seulement 3×3 convolutions, mais beaucoup de filtres. Formé sur 4 GPU pendant 2-3 semaines. C'est actuellement le choix le plus préféré de la communauté pour extraire des fonctionnalités à partir d'images. La configuration de poids du VGGNet est accessible au public et a été utilisée dans de nombreuses autres applications et défis en tant qu'extracteur de caractéristiques de base. Cependant, VGGNet se compose de 138 millions de paramètres, ce qui peut être un peu difficile à gérer. [46]

3.11.3.2 Architecture de VGG16

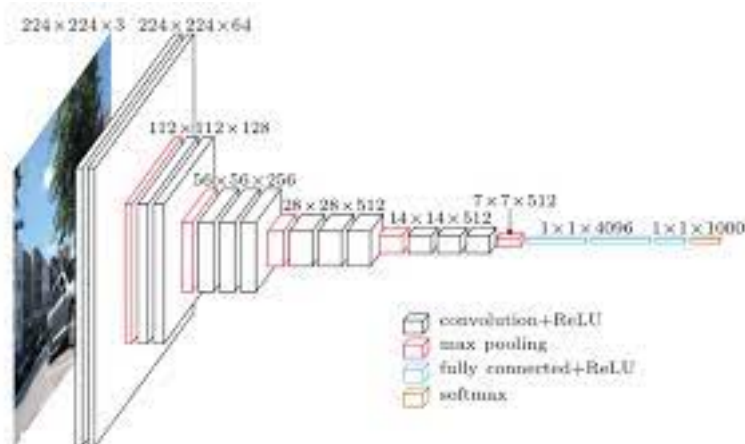


FIGURE 3.14: architecture de VGGNet

[5]

La première couche de convolution est une image RGB de taille 224×224 .

Pour toutes les couches de convolution, le noyau de convolution est de taille 3×3 : la plus petite

dimension pour capturer les notions de haut, bas, gauche/droite et centre.

C'était une spécificité du modèle au moment de sa publication. Jusqu'à VGG16 beaucoup de modèles s'orientaient vers des noyaux de convolution de plus grande dimension (de taille 11 ou bien de taille 5 par exemple).

Rappelons que ces couches ont pour but de filtrer l'image en ne gardant que des informations discriminantes comme des formes géométriques atypiques.

Ces couches de convolution s'accompagnent de couche de Max-Pooling, chacune de taille 2×2, pour réduire la taille des filtres au cours de l'apprentissage.

En sortie des couches de convolution et pooling, nous avons 3 couches de neurones Fully-Connected.

Les deux premières sont composées de 4096 neurones et la dernière de 1000 neurones avec une fonction d'activation softmax pour déterminer la classe de l'image. [46]

3.12 Conclusion

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures. . . etc). Aussi qu'une vision générale sur l'apprentissage profond, toute on donnant en détail la méthode choisie dans notre travail de recherche qui est le CNNs.

CONTRIBUTION

4.1 Introduction

Dans le cadre de notre recherche nous avons traité des images pour la classification des images et la détection des objets. Ce traitement permet de prévenir les attaques avant passe grâce au traitement des images par le Deep Learning. Nous essayons de réaliser un modèle pour classer les images à base des modèles classiques de CNN (VGG, Alexnet). Nous utilisons deux datasets pour tester les modèles.

4.2 Architecture de système

4.3 Les composants de système

Notre système a pour objectif de classer des images associées aux frontières et détecter les personnes anormales. En utilisant des drones des capteurs des caméras de surveillance, qui ont capté l'image puis envoient pour la traiter par notre modèle grâce aux nos datasets et trouve le résultat. Si on détecte l'attaque le système envoyé l'information aux soldats qui ont suivi la position de cette attaque grâce à MAP.

4.3.1 Les drones

4.3.1.1 Définition

Le terme « drone » est issu de la langue anglaise et signifie « faux bourdon ». Il désigne un système pilotable à distance, capable d'emporter une charge utile, présentant différentes formes

selon les besoins et ayant un niveau d'IA (Intelligence Artificielle) plus ou moins évolué suivant les modèles et les applications auxquelles ils sont destinés. Ce niveau d'IA lui permet d'être partiellement ou totalement indépendant d'actions humaines afin de remplir ses tâches.

4.3.1.2 Les catégories des drones :

1 Les drones terrestres : comme leur nom l'indique, ces engins sont conçus pour se déplacer sur la terre ferme. Pour cela, ils utilisent différents types de propulsions, des roues, des chenilles, des pattes à l'instar des insectes, des pieds etc. Leur utilisation est variée comme leur nombre. Ainsi, ces machines sont principalement employées pour effectuer du déminage, de la reconnaissance de terrain, du transport de fret, de la prévention des risques chimiques, de la lutte anti-incendie, du désherbage dans le monde de l'agriculture etc. « Big Dog », conçu par Boston Dynamics (USA), il est destiné à effectuer du transport de fret en tout terrain Page 7 sur 37 « Scarab Vulcain », conçu par la société TECDRON(France) et destinée à la lutte anti incendie .

2 Les drones maritimes : évoluant dans un environnement aquatique, ils sont subdivisés en deux sous-catégories, les drones de surfaces et les drones sous-marins. Ils sont principalement utilisés pour effectuer du déminage, de la recherche sous-marine, de la cartographie des fonds, des interventions sur des installations sous-marines (pipelines, forages, câbles sous-marins ...), de la sécurité côtière. Ils sont souvent totalement autonomes, leurs missions ayant été préprogrammées dans leur mémoire, ou bien filoguidés depuis des navires en surfaces (pour les R.O.Vs en particulier). « Série Alister », développé par ECA robotique (France) et destiné à l'étude des fonds marins.

3 Les drones aériens : Ils sont de loin les plus connus et les plus répandus, que ce soit dans le domaine civil ou militaire. Ces drones sont très largement utilisés pour effectuer des tâches extrêmement variées : de la surveillance, de la destruction de cibles terrestres, des prises de vidéos et de photos, de la reconstitution trois dimensions, de l'inspection, du relevé de terrain etc. Leurs caractéristiques sont extrêmement variées, allant du simple drone de 50 centimètres d'envergure et télé piloté à distance, à des drones furtifs totalement autonomes ayant des envergures supérieur de l'ordre de la dizaine de mètres.

[44]

4.3.2 Les capteurs

4.3.2.1 définition :

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable

4.3.2 Objectifs :

L'objectif de capteurs et de reconnaissance des personnes anormales. Protège les frontières sans l'utilisation des soldats. [15]

4.3.3 MAP

est un service de cartographie en ligne. Le service a été créé par Google à la suite du rachat en octobre 2004 de la start-up australienne Where 2 Technologies. Objectifs : L'objectif de Map dans notre application est de préciser la location et la suivre. [19]

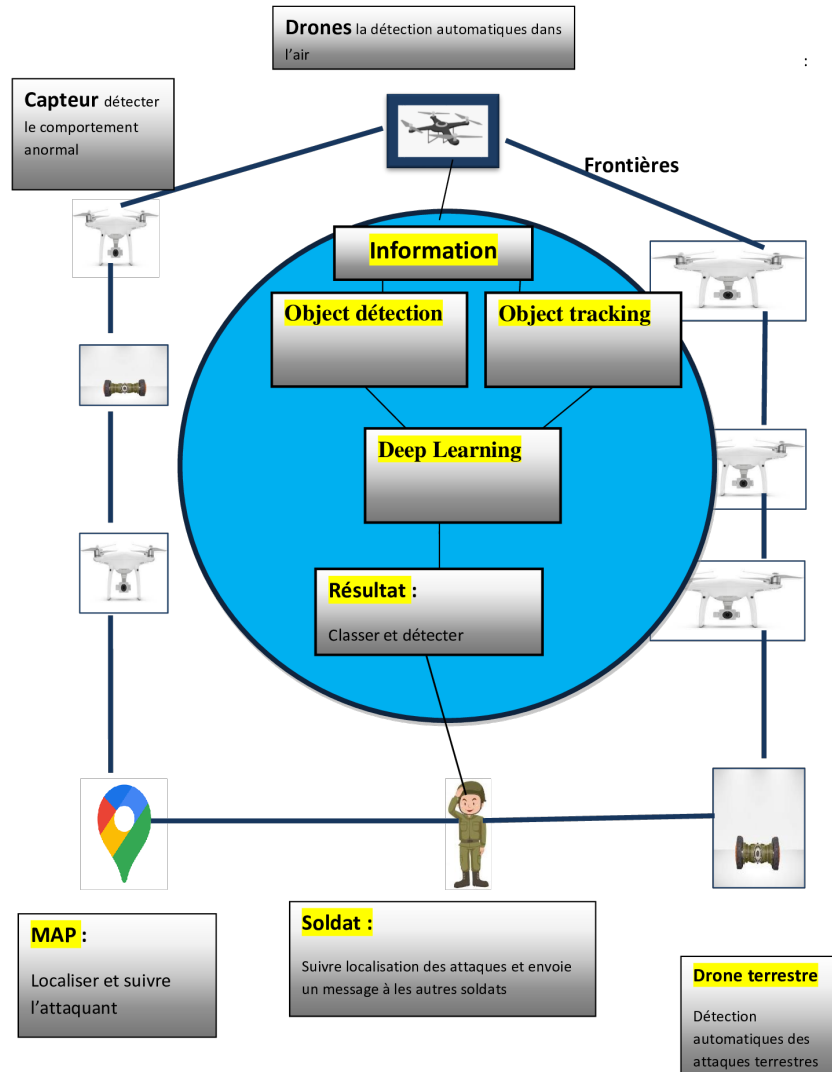


FIGURE 4.1: L'architecture générale d'un système militaire par l'intelligence artificielle

4.4 Classification des images

Pour la classification des images nous proposons des modèles basé sur les modèles classiques de CNN vus dans le chapitre précédent.

4.4.1 Présentation de notre modèles

4.4.1.1 Modèle 1

Notre architecture est constituée de cinq couches de convolution dont la première couche utilise un filtre de taille 6×6 , et nous proposons d'utiliser 50 filtres, toutes les couches convolutives conv1-5 avec une rectification linéaire Relu, cette fonction force les neurones à retourner des valeurs positives. Une couche averagepooling est appliquée après Les trois premières couches de convolution, afin de réduire la taille de l'image et la quantité des paramètres. Dans la deuxième couche utilise un filtre de taille 16×16 , et nous proposons d'utiliser 256 filtres, une couche averagepooling.

Dans la troisième couche utilise un filtre de taille 20×20 , et nous proposons d'utiliser 50 filtres. Dans la quatrième couche utilise un filtre de taille 3×3 , et nous proposons d'utiliser 384 filtres. Dans la cinquième couche utilise un filtre de taille 3×3 , et nous proposons d'utiliser 256 filtres, et une couche de averagepooling. Trois couches entièrement connectées ont été ajoutées, Le nombre des unités pour les deux premières couches égale à 4096, et pour la troisième couche le nombre des unités égalent à 1000. La couche de sortie avec une fonction sigmoïde car notre travail consiste d'une classification binaire. Nous essayons aussi de changer les valeurs de hyper paramètres (batch -size, Learning- rate, nombre des epochs, validation split)

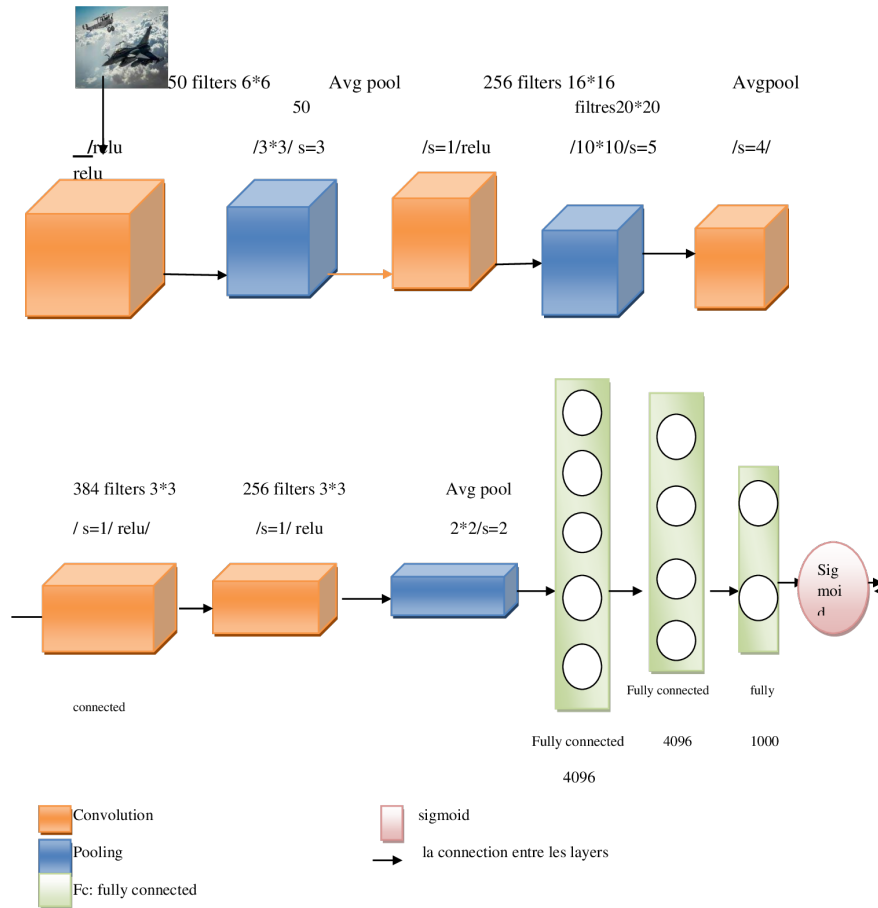


FIGURE 4.2: Modèle 01

4.4.1.2 Modèle 02

Nous extrairons ce modèle de modèle classique alexnet , il constitue de six couches de convolution dont la première couche utilise un filtre de taille 11*11, et nous proposons d'utiliser 96 filtres, toutes les couches convolutives conv1-6 avec une rectification linéaire Relu, cette fonction

force les neurones à retourner des valeurs positive, Une couche averagepooling est appliquée après Les deux premiers couches de convolution, afin de réduire la taille de l'image et la quantité des paramètres. Dans la deuxième couche utilise un filtre de taille 11*11, et nous proposons d'utiliser 256filtres, une couche averagepooling. Dans la troisième et la quatrième couche nous utilisons un filtre de taille3*3, et nous proposons d'utiliser 384 filtres. Dans la cinquième couche utilise un filtre de taille 3x3, et nous proposons d'utiliser 256filtres. Dans la sixième couche utilise un filtre de taille 3x3, et nous proposons d'utiliser 256 filtres, et une couche de averagepooling.Trois couches entièrement connectées ont été ajoutées, Le nombre des unités pour l les deux premières couches égale à 4096, et pour la troisième couche le nombre des unités égalent à 1000. La couche de sortie avec une fonction sigmoïde car notre travail consiste d'une classification binaire.

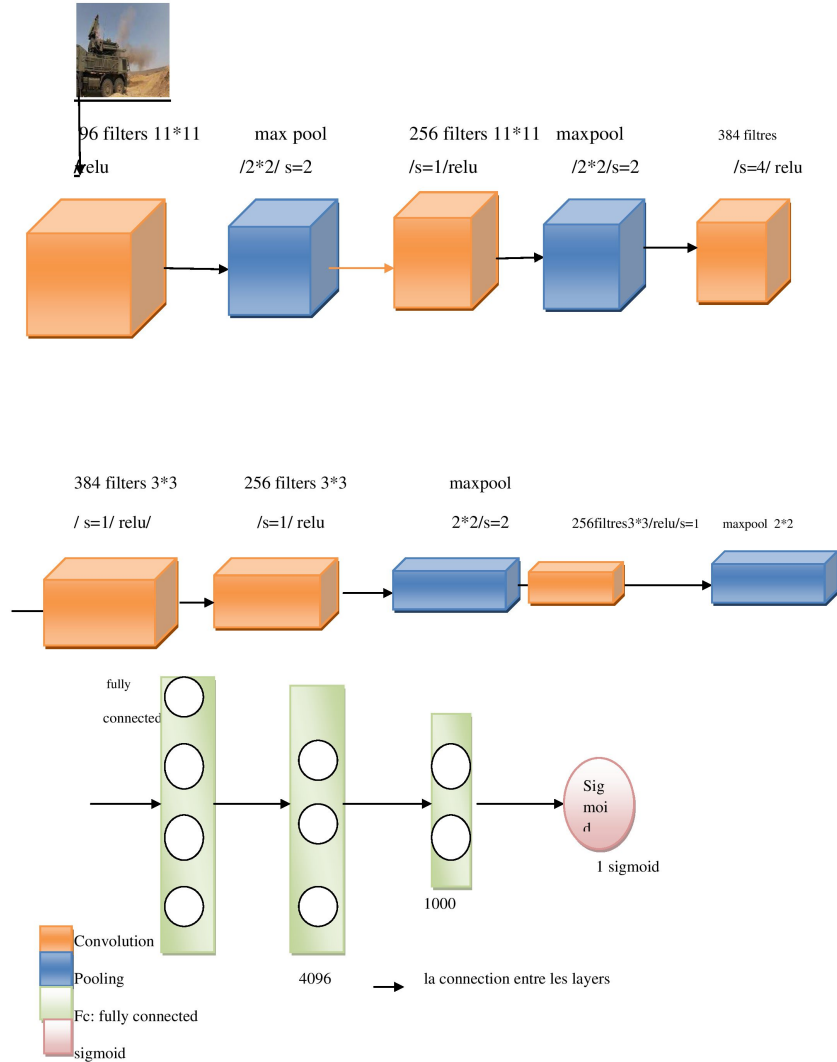


FIGURE 4.3: Modèle 02

4.4.1.3 Modèle 03

Ce modèle est basé sur le modèle classique VGG16, composé de 12 couches de convolution, quatre couches de averagepool et de deux couches de fully connected. Chaque couche de convolution est composée de plusieurs filtres, la taille de chaque filtre est de 3*3, la fonction d'activation

ReLU est utilisée à chaque fois qu'on passe par une couche de

convolution, cette fonction d'activation force les neurones à retourner des valeurs positives. Le Avgpooling est appliqué après pour réduire la taille de l'image. A la fin de la deuxième couche de convolution, Dans les couche, on change quelques paramètres comme le nombre de filtres qui devient 64 à la place de 32, la fonction d'activation reste la même (ReLU) et la même chose pour Avgpooling . Après les couches de convolution et de regroupement, notre partie classification se compose de quelques couches entièrement connectées. Cependant, ces couches ne peuvent accepter que des données à une dimension. Pour convertir nos données 3D en 1D, nous utiliserons la fonction Flatten, cela permettra essentiellement d'arranger notre volume 3D en un vecteur 1D. Les dernières couches d'un réseau neurones convolutionnel sont des couches entièrement connectées. Ces dernières ont des connexions complètes avec toutes les activations de la couche précédente.

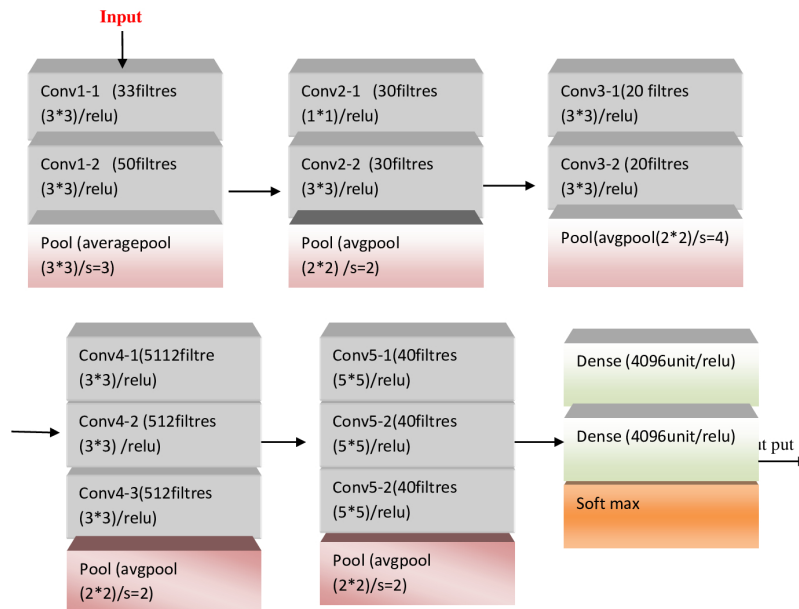


FIGURE 4.4: Modèle 03

4.5 conclusion

Dans ce projet nous avons proposé une approche pour la détection des attaques militaires, qui est basée sur les réseaux de neurones convolutionnel pour classer les images et les menaces, et pour cela nous avons parlé des différents types des algorithmes de la machine Learning et nous avons discuté des notions fondamentales des deep Learning en général et des réseaux de neurones convolutionnels en particulier, en présentant les différents types de couches utilisées dans la classification. et aussi les méthodes de régularisation. =====

RESULTAS EXPRIMENTATION ET COMPARAISON

5.1 Introduction

Ce chapitre est consacré à la présentation des résultats expérimentaux relatifs aux modèles introduits lors du chapitre précédent, à savoir le modèle1, modèle2, VGG, alexnet et lenet5. Les expérimentations seront donc effectuées sur trois problématiques différentes : 1) object detection utilisant le yolo V4 afin d'extraire les objets du premier plan. 2) reconnaissance du tank afin d'aider l'armée à réagir dans les opérations militaire afin d'aider l'armée à contrôler les frontières.

5.2 Les outils et les librairies utilisés

5.2.1 Tensorflow

TensorFlow est une bibliothèque open source de Machine Learning, Créé par l'équipe Google Brain en 2011, sous la forme d'un système propriétaire dédié aux réseaux Learning, est une bibliothèque de logiciels open source pour le calcul numérique utilisant des graphes de flux de données. Les noeuds des graphes représentent des opérations mathématiques, tandis que les arêtes des graphes représentent les tableaux de données multidimensionnels (tenseurs) qui circulent entre eux. Cette architecture flexible vous permet de déployer des calculs sur un ou plusieurs processeurs ou GPU sur un ordinateur de bureau, un serveur ou un périphérique mobile sans réécrire le code. Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification des chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles séquence-to-séquence pour la traduction automatique, ou encore le traitement naturel du langage.

[8]



FIGURE 5.1: Logo Tensorflow

[7]

5.2.2 Keras

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, CNTK ou Theano. Il a été développé pour permettre une expérimentation rapide . -Permet un prototypage simple et rapide (convivialité, modularité et extensibilité). - Prend en charge les réseaux convolutionnels et les réseaux récurrents, ainsi que les combinaisons des deux . - Fonctionne de manière transparente sur CPU et GPU [32]

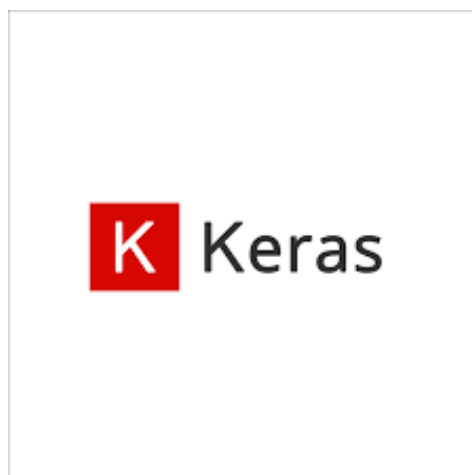


FIGURE 5.2: Logo Keras

[2]

5.2.3 Python

Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par Guido van Rossum et publié pour la première fois en 1991, Python repose sur une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces significatifs. Il fournit des constructions permettant une programmation claire à petite et grande échelle. En juillet 2018, Van Rossum a démissionné en tant que leader de la communauté après 30 ans.

Python propose un système de typage dynamique et une gestion automatique de la mémoire. Il prend en charge plusieurs paradigmes de programmation, notamment orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard étendue et complète.

Les interpréteurs Python sont disponibles pour de nombreux systèmes d'exploitation. CPython, l'implémentation de référence de Python, est un logiciel open source et dispose d'un modèle de développement basé sur la communauté, comme le font presque toutes les autres implémentations de Python. Python et CPython sont gérés par l'association à but non lucratif Python Software Foundation.

[53]



FIGURE 5.3: Logo Python

[3]

5.2.4 Jupyter Notebook

Jupyter Notebook est un outil open source permettant d'écrire du code informatique et de le partager pour collaborer. Grâce à ses nombreux avantages, ce " bloc-note " de calcul est devenu une référence incontournable pour les Data Scientists

Jupyter a été créé pour faciliter la présentation du travail de programmation d'un développeur et permettre à d'autres d'y participer. Il permet de mélanger du code, des commentaires et des visualisations dans un document interactif appelé notebook qui peut être partagé, réutilisé et retravaillé. Et comme Jupyter Notebook s'exécute dans un navigateur web, le « cahier » lui-même peut être hébergé au choix sur l'ordinateur du développeur ou sur un serveur distant.

[40]



FIGURE 5.4: Logo jupyter

5.3 Définition de Datasets

5.3.1 Military tanks dataset

L'ensemble de données contient une variété de chars militaires et leurs images. L'ensemble de données peut encore être étendu en ajoutant plus d'images pour des classes spécifiques. [4]

5.4 Définition des paramètres

Pour le protocole d'expérimentation nous avons à chaque étape varié un paramètre et fixé les autres.

5.4.1 Batch size

La taille du lot est un terme utilisé dans l'apprentissage automatique et fait référence au nombre d'exemples de formation utilisés dans une itération. La taille du lot peut être l'une des trois options suivantes : mode batch : où la taille du lot est égale à l'ensemble de données total, ce qui rend les valeurs d'itération et d'époque équivalentes.

Mode mini-lot : où la taille du lot est supérieure à un mais inférieure à la taille totale de l'ensemble de données. Généralement, un nombre qui peut être divisé par la taille totale de l'ensemble de données.

Mode stochastique : où la taille du lot est égale à un. Par conséquent, le gradient et les paramètres du réseau de neurones sont mis à jour après chaque échantillon. [14]

5.4.2 Nombre d'Epoch

Une époque est un terme utilisé dans l'apprentissage automatique et indique le nombre de passes de l'ensemble de données d'entraînement complet que l'algorithme d'apprentissage automatique a effectué. Les ensembles de données sont généralement regroupés en lots (en particulier lorsque la quantité de données est très importante). Certaines personnes utilisent le terme itération de manière vague et se réfèrent au passage d'un lot dans le modèle en tant qu'itération. [22]

5.4.3 Learning rate

Le taux d'apprentissage est un hyper paramètre qui contrôle le degré de modification du modèle en réponse à l'erreur estimée à chaque fois que les poids du modèle sont mis à jour. Le choix du taux d'apprentissage est difficile car une valeur trop petite peut entraîner un long processus d'entraînement qui pourrait rester bloqué, tandis qu'une valeur trop grande peut entraîner un apprentissage trop rapide d'un ensemble de poids sous-optimal ou un processus d'entraînement instable.

[14]

5.4.4 Optimizer

Les algorithmes d'optimisation sont utilisés dans l'apprentissage automatique pour réduire une fonction appelée fonction de perte ou fonction d'erreur. En minimisant la fonction de perte, les algorithmes d'optimisation peuvent atteindre une différence minimale ou nulle entre la sortie réelle et prévue, ce qui rend notre modèle plus précis pour une tâche. Il existe différents types d'algorithmes d'optimisation utilisés dans les réseaux de neurones : variantes de descente en pente (gradient descent) descente de gradient par lots descente de gradient stochastique descente de gradient mini-lot Gradient accéléré Nesterov (moment Nesterov) algorithmes avec des taux d'apprentissage adaptatifs AdaGrad AdaDelta RMSprop Adam [47]

5.4.5 Loss function

Une fonction de perte est une fonction mathématique couramment utilisée en statistique. Les fonctions de perte sont fréquemment utilisées pour créer des algorithmes d'apprentissage automatique. La fonction de perte calcule l'erreur pour un seul exemple d'apprentissage contrairement à une fonction de coût, qui est la moyenne des fonctions de perte de chaque exemple dans l'ensemble de données d'apprentissage². Les fonctions de perte sont utilisées dans l'apprentissage automatique pour mesurer la distance mathématique entre les valeurs prédites et les valeurs réelles [38]

5.4.6 Activation function

les fonctions d'activation effectuent une transformation sur une somme pondérée d'entrées plus des biais sur un neurone afin de calculer sa sortie. En utilisant une analogie biologique, la fonction d'activation détermine la « cadence de tir » d'un neurone en réponse à une entrée ou à un stimulus. Ces fonctions introduisent des non-linéarités dans les réseaux de neurones leur permettant d'effectuer des tâches complexes telles que la reconnaissance d'images et le traitement du langage. Sans fonctions d'activation non linéaires, les réseaux de neurones artificiels se comportent comme de simples modèles de régression linéaire. Ces fonctions comprennent : fonction sigmoïde fonction d'unité linéaire rectifiée (ReLU) fonction tangente hyperbolique (Tanh). [38]

5.5 La définition de mesure d'évaluation

Les résultats obtenus seront présentés dans les parties suivantes avec les mesures d'évaluation : [28]

5.5.1 Rappel

Le rappel est calculé comme le rapport entre le nombre d'échantillons positifs correctement classés comme positifs et le nombre total d'échantillons positifs. Le rappel mesure la capacité du modèle à détecter des échantillons positifs. Plus le rappel est élevé, plus les échantillons détectés sont positifs.

$$Recall = \frac{Tp}{Tp + Fn}$$

5.5.2 Précision

La précision est calculée comme le rapport entre le nombre d'échantillons positifs correctement classés et le nombre total d'échantillons classés comme positifs (correctement ou incorrectement). La précision mesure l'exactitude du modèle en classant un échantillon comme positif.

$$Précision = \frac{Tp}{Tp + Fp}$$

5.5.3 F-mesure

Le score F, également appelé score F1, est une mesure de la précision d'un modèle sur un ensemble de données. Il est utilisé pour évaluer les systèmes de classification binaire, qui classent les exemples en « positif » ou « négatif ». Le score F est un moyen de combiner la précision et le rappel du modèle, et il est défini comme la moyenne harmonique de la précision et du rappel du modèle. Le F-score est couramment utilisé pour évaluer les systèmes de recherche d'informations tels que les moteurs de recherche, ainsi que pour de nombreux types de modèles d'apprentissage automatique, en particulier dans le traitement du langage naturel. Il est possible d'ajuster le score F pour donner plus d'importance à la précision par rapport au rappel, ou vice-versa. Les scores F ajustés courants sont le score F0,5 et le score F2, ainsi que le score F1 standard. [28]

$$F_1 = \frac{2}{\frac{1}{recall} \times \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall}$$

$$= \frac{tp}{tp + \frac{1}{2}(fp + fn)}$$

5.6 Résultats de détection des objets avec yolo

5.6.1 Définition de YOLOv4

YOLO est un algorithme de réseau neuronal convolutif avec sélection de régions, similaire à ceux présentés à la section précédente. À la base, YOLO utilise un modèle de réseau basé sur GoogLeNet (Szegedy et al., 2015), qu'il nomme Darknet, avec modifications sur les couches en entrée afin que effectuent les phases de détection et de classification à l'intérieur même du réseau., sans nécessité l'utilisation d'un second réseau.

[39]

5.6.2 Fonctionnement de YOLOv4

Plutôt que d'entraîner son modèle et ses fonctions d'activation sur des régions spécifique d'une image. YOLO les entraîne sur l'image d'entrée dans son entièreté, avant d'effectuer une phase de localisation de région probable. Cette approche permet d'assurer que les vecteurs de caractéristiques pour chaque couche du réseau gardent leur donnée de contexte spatiale pertinentes, afin d'éviter des faux négatifs si les régions proposées ne détiennent pas la totalité de l'image. Pour ce faire, YOLO divise l'image en entrée en cellules de taille 32 x 32. Pour chaque cellule, YOLO traite 5 cadres d'ancrage possibles aussi appelés des clusters de dimension. Ces cadres ont

utilisé lors de l'entraînement et de la classification pour calculer pour chaque cellule le niveau de confiance qu'il détient d'un objet le taux d'erreur entre la taille de l'objet trouvé comparée à celle modélisé par le réseau et le taux de confiance que l'objet est l'une des classes du connues par le réseau. Grâce à ces données, le réseau est capable, par réduction de caractéristique dans ses dernières couches de convolution, de transmettre une grille qui divise l'image en zones de taille 32 x 32, où chaque zone contient les cadres de limitations avec les taux de confiance de contenir un objet le plus élevé ainsi que les niveaux de confiance que l'objet appartient à une des k classes connue par le système. Cette approche permet d'offrir un système tout- en-un, capable de faire la localisation et la classification sur des caractéristiques riches sans toutefois causer un goulot d'étranglement pour la phase de détection d'objet. pour cette phase yolo en plus de détection de la classe de l'objet il donne une probabilité pour chaque boxes capturer d'une frame avec un lot de coordonnées comme le montre l'image suivante.

[30]

```

Frame #: 4
Tracker ID: 1, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (1220, 470,
1513, 683)
Tracker ID: 2, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (789, 572,
1085, 810)
Tracker ID: 3, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (910, 300,
1118, 464)
Tracker ID: 4, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (861, 889,
1220, 1077)
Tracker ID: 5, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (609, 409, 921,
603)
Tracker ID: 6, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (577, 341, 799,
524)
Tracker ID: 7, Class: person, BBox
Coords (xmin, ymin, xmax, ymax): (1288,
499, 1345, 542)
Tracker ID: 8, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (500, 279, 683,
450)
Tracker ID: 9, Class: person, BBox
Coords (xmin, ymin, xmax, ymax): (917,
934, 1014, 993)
Tracker ID: 10, Class: car, BBox Coords
(xmin, ymin, xmax, ymax): (285, 286, 402,
365)
    
```

FIGURE 5.5: Code généré pour la détection des objets du premier plan avec yolo d'une frame avec la classe et les codonnées de chaque objet

our détecter objets dans une même image l'approche originale fut de lancer des classifications en faisant glisser des plus petites fenêtres sur toute l'image ... mais cette approche était longue et surtout imposait de relire plusieurs fois l'image source. L'idée fondamentale derrière YOLO est de ne faire qu'une seule passe (lecture) de l'image (YOLO = You Only Look Once).

Résultat la détection est vraiment beaucoup plus rapide Les prédictions d'objectivité sont générées en calculant l'IoU (Intersection over Union) de la boîte de vérité terrain et de la boîte englobante prédite, qui mesure la précision d'une prédiction compte tenu de la vérité terrain. Pour générer des prédictions finales, YOLOv3 utilise une suppression non maximale en triant d'abord les prédictions par scores de confiance et en produisant des boîtes de prédiction qui atteignent ou dépassent un seuil de confiance spécifié.

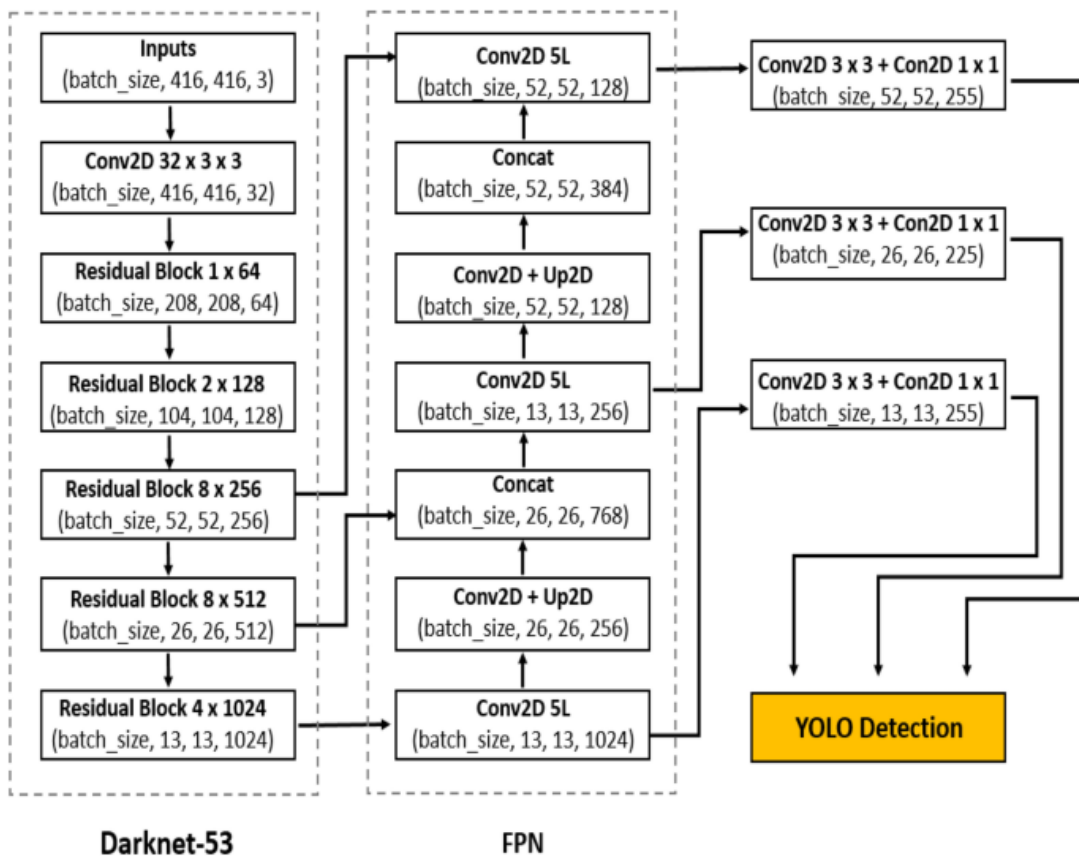


FIGURE 5.6: Architecture du yolo v4

5.6.3 Architecture de YOLOv4



FIGURE 5.7: Résultats du yolo qui traite le problème de détection des objets dans une video

L'architecture de Yolov4 est découpée en trois parties, fonctionnellement différentes, appelées backbone (la colonne vertébrale), neck (le cou) et head (tête). [39]

5.6.3.1 La backbone

Le “corps” du réseau, ce qui va permettre l'ensemble des décisions prises par celui-ci. Pour simplifier, il peut être vu comme un “convertisseur” qui converti l'image en entrée, un format de données tel quel difficile à traiter par de l'IA, en un ensemble d'informations qui caractérisent son contenu appelées “features” (comme la présence de formes, couleurs, textures, ...) à partir desquelles il est facile de reconnaître des objets. Il est ainsi composé d'une suite de couches successives, et extrait des cartes de features, i.e. des cartes indiquant quelles features sont présentes à quels endroits de l'image. La backbone est généralement entraînée séparément sur des concours de classification d'image comme le challenge ImageNet qui recense des centaines de milliers d'images au contenu très divers comme des animaux, des véhicules, des plantes, etc. Cette diversité de contenu force la backbone à apprendre des caractéristiques très variées par la taille, la couleur, la forme des éléments qu'elle observe et ainsi d'être plus robuste, capable d'extraire des features utiles quelle que soit l'image que l'on présente à la backbone. [39]

5.6.3.2 Le neck

A pour rôle d'extraire les features pertinentes de l'ensemble des couches de la backbone, et de les combiner en features utiles à notre tâche de détection. En effet, toutes les couches constituant

la backbone n'apprennent pas la même information : les premières couches, généralement de plus haute résolution spatiale, vont détecter des features souvent plus simples (la présence de lignes, de couleurs) et des éléments petits. Les dernières couches, de plus faibles résolutions, vont au contraire avoir tendance à fournir des features plus complexes (par exemple la combinaison de formes et couleurs spécifiques comme un cercle métallique à trou pour une jante de voiture) et des objets de grande taille. La neck permet ainsi d'intégrer et combiner des features de différentes résolutions et complexités, pour permettre une détection d'éléments petits comme grands, simples comme complexes.

5.6.3.3 Dense prédiction

La tête est responsable de la décision finale du réseau. A partir des informations fournies par le cou, celle-ci décide où sont les éléments à détecter, en traçant des bounding box autour, et ce qu'ils sont, en donnant la nature de l'objet présent dans chaque bounding box. Cette partie est en fait celle de son prédécesseur, Yolov3, et fonctionne en donnant un modèle, un "canvas" de boîtes possibles. Yolov3 et Yolov4 apprennent en fait simplement à sélectionner parmi ces boîtes quelles sont les plus proches des objets détectés, et à ajuster légèrement leurs dimensions pour coller au mieux aux objets détectés. [45]

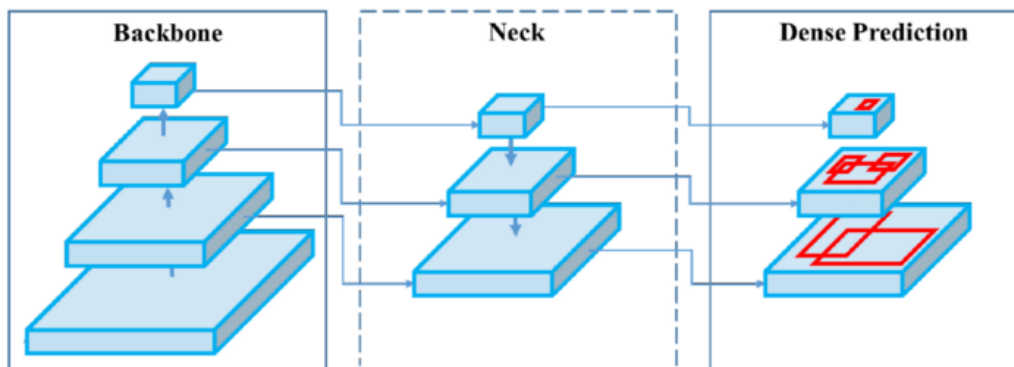


FIGURE 5.8: Architecture du yolo v4

[45]

5.7 Résultats du tank detection

Les résultats dans cette problématique ont été obtenus sur le dataset . . . Des exemples de cet dataset sont présenté dans la figure suivante. Nous allons discuté par la suite les résultats obtenus par chaque modèle détaillé dans le chapitre précédent :

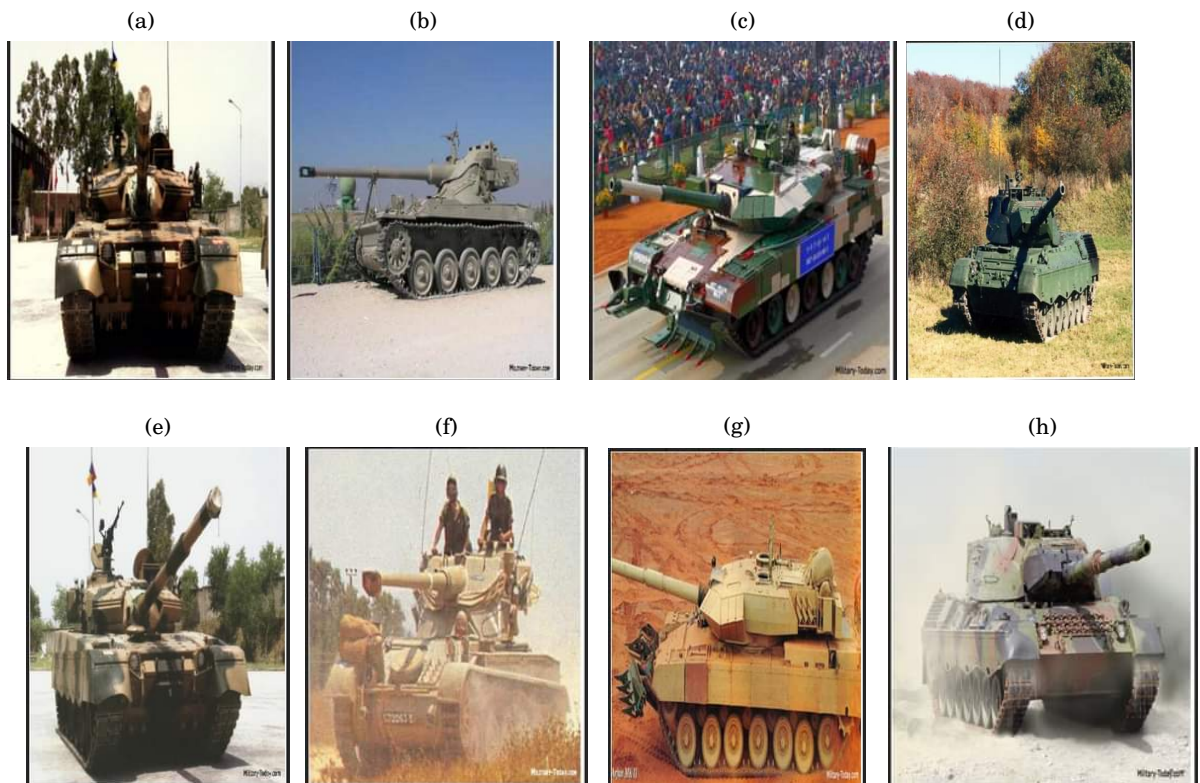


FIGURE 5.9: Quelques exemples de TANK du dataset utilisé avec : a,b) alkhalid, b,f) amx, c,g) arjun mk2 d,h) leopard

[4]

5.8 Resultat alexnet

5.8.1 Variation du nombre d'époches

5.8.1.1 Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches

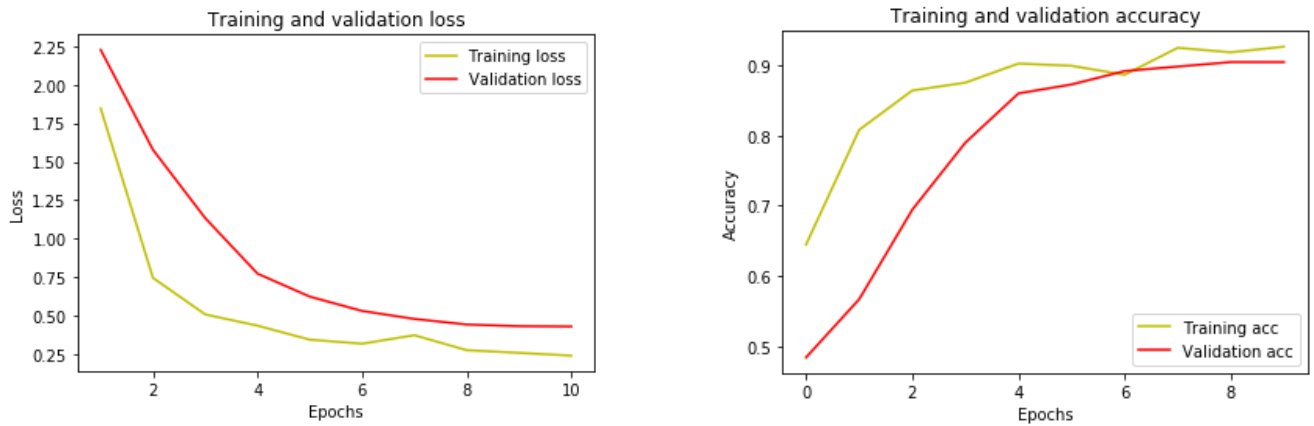
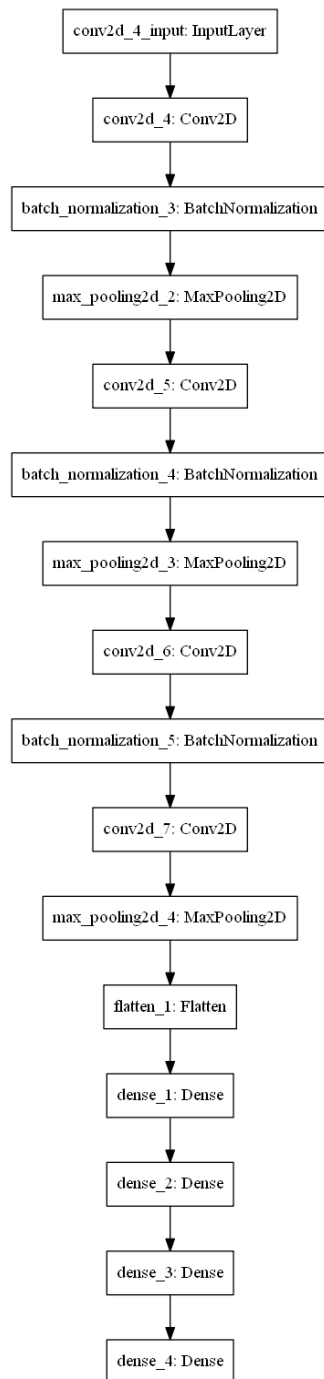


FIGURE 5.10: Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour le modèle alexnet

Le nombre d'époche jouent un rôle très important d'après les résultats de la figure précédente on remarque qu'il y a du overfitting au début de la courbe avec une différence entre training accuracy et validation accuracy.

5.8.1.2 Configuration et nombre de parametres modèle modéle1



Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 70, 70, 96)	11712
batch_normalization_3 (Batch Normalization)	(None, 70, 70, 96)	384
max_pooling2d_2 (MaxPooling2D)	(None, 35, 35, 96)	0
conv2d_5 (Conv2D)	(None, 25, 25, 256)	2973952
batch_normalization_4 (Batch Normalization)	(None, 25, 25, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 256)	0
conv2d_6 (Conv2D)	(None, 10, 10, 384)	885120
batch_normalization_5 (Batch Normalization)	(None, 10, 10, 384)	1536
conv2d_7 (Conv2D)	(None, 8, 8, 256)	884992
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 256)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 1000)	4097000
dense_4 (Dense)	(None, 24)	24024
Total params: 42,442,368		
Trainable params: 42,440,896		
Non-trainable params: 1,472		

FIGURE 5.11: Configuration et nombre de parametres modèle alexnet. Avec 42442368 paramétres divisé en deux (trainable params = 42440896, non trainable params = 1472)

5.8.1.3 Comparaison entre les classes

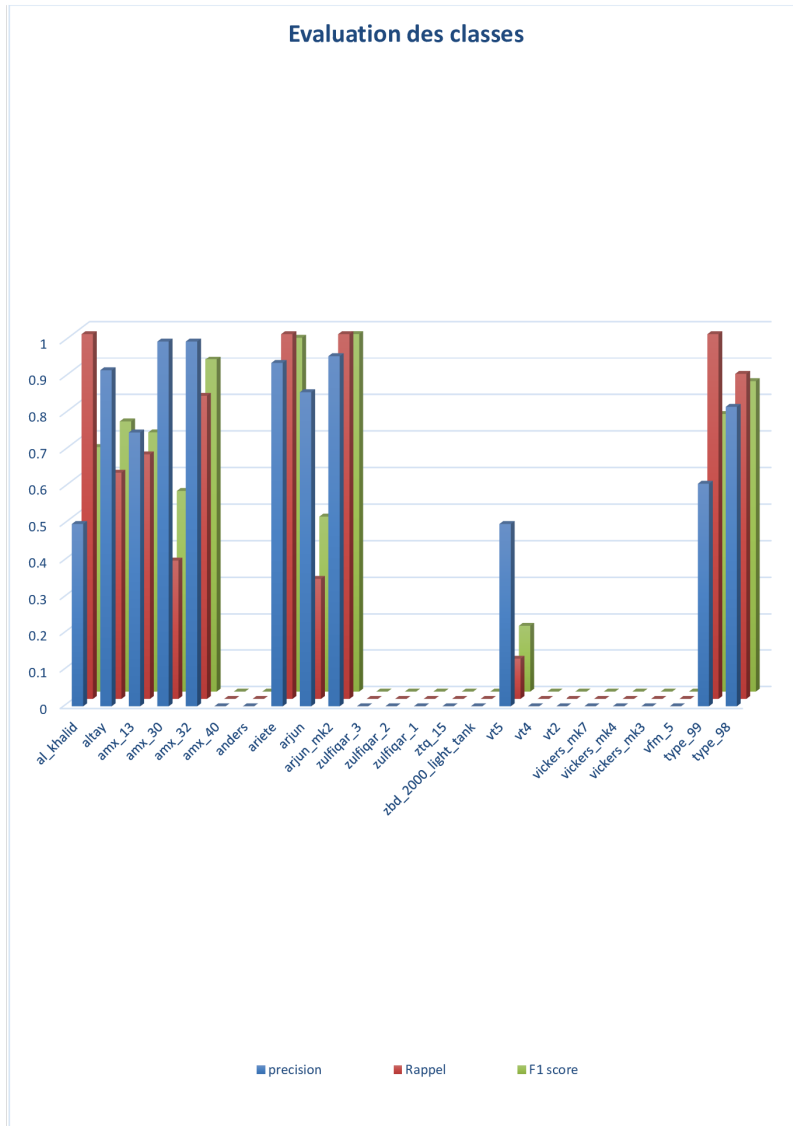


FIGURE 5.12: Precision rappel et f1 score pour chaque classes tank detection utilisant le modèle1

On remarque que alexnet a du mal a mieux détecter les différents type de tank expliqué par les données non équilibré.

5.8.2 Alexnet optimizers comparaison

5.8.2.1 Accuracy

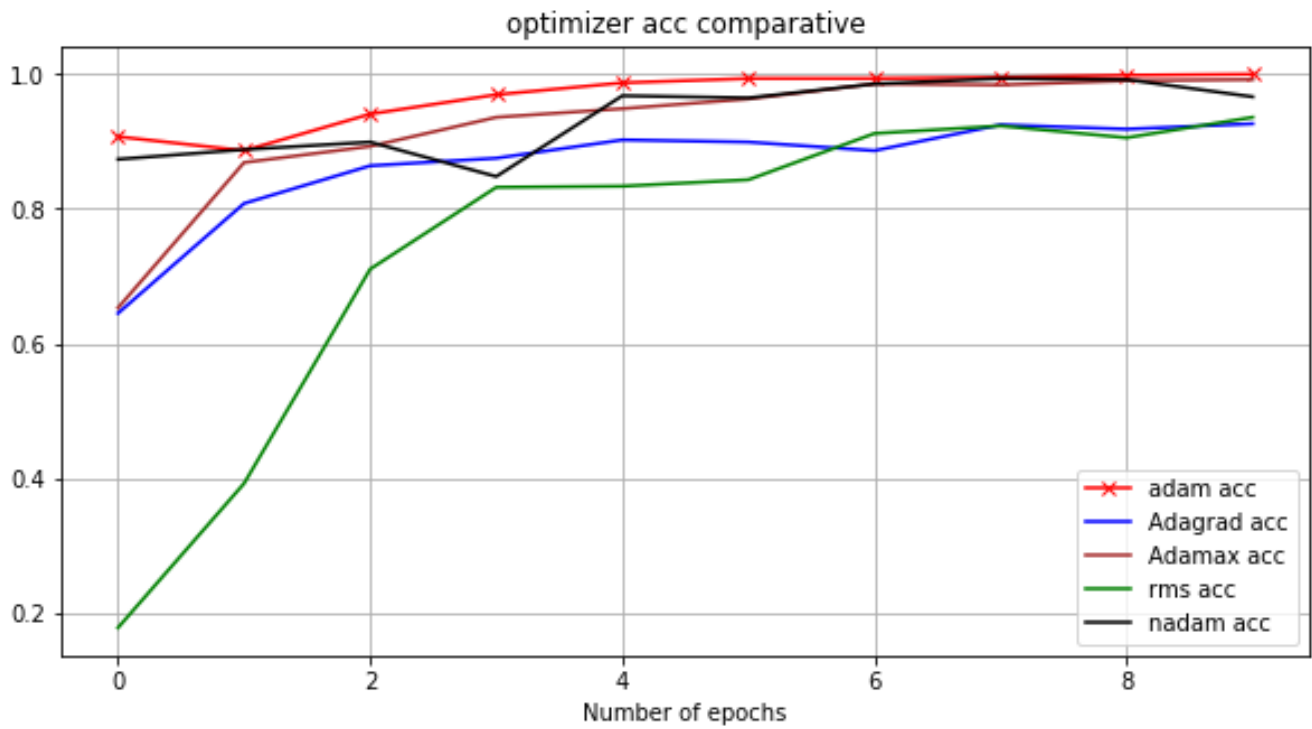


FIGURE 5.13: Accuracy du alexnet avec variation du nombre d'epoches et optimizer

On remarque que les optimizers adam et adamax donnent les meilleurs résultats en terme accuracy.

5.8.2.2 Validation accuracy

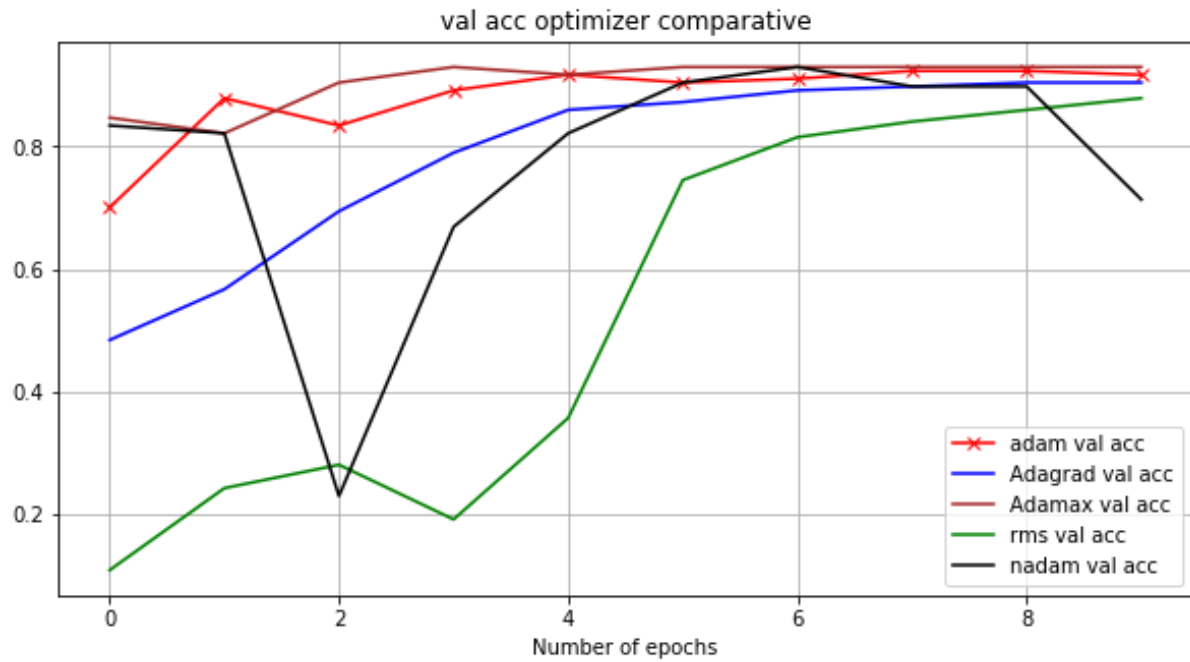


FIGURE 5.14: Validation accuracy du alexnet avec variation du nombre d'epoches et optimizer

On remarque que optimizer adamax donne des résultats optimaux par rapport aux autres optimizers en terme de validation accuracy .

sui par optimizer adam qui donne des résultats proche de résultats d' adamax.

5.8.2.3 validation loss

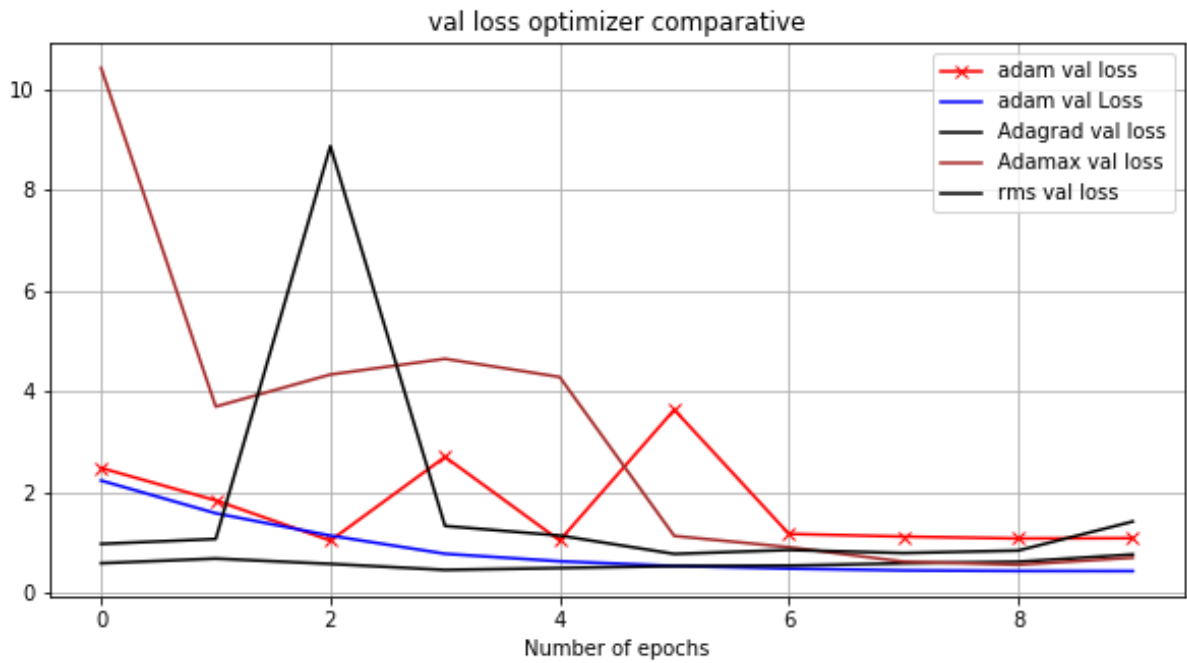


FIGURE 5.15: Validation loss du alexnet avec variation du nombre d'époques et optimizer

On remarque que la validation loss est élevée pour les deux optimizers adamax et adagrad pour le nombre d'époques inférieur à 5 et pour le nombre d'époque supérieur à 5 elle est élevée pour l'optimizer adam.

5.8.2.4 Comparaison

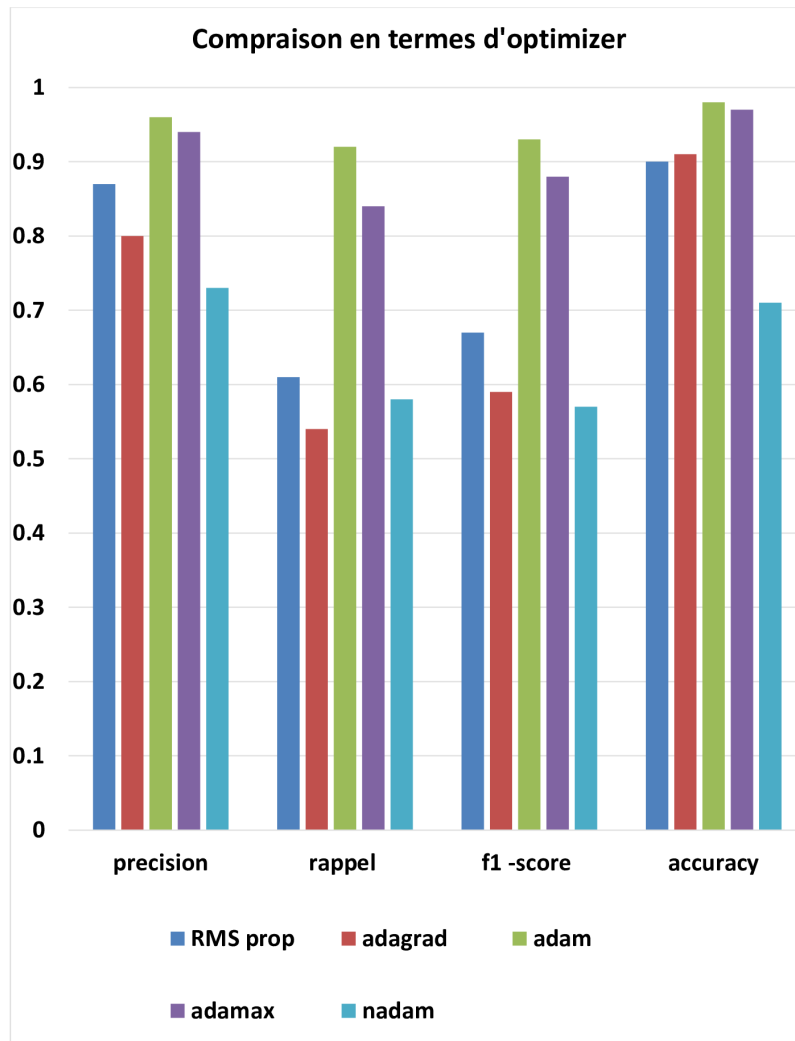


FIGURE 5.16: Modèle alexnet comparaison entre les optimizers en termes de f-score, rappel et precision

D'après les résultats on remarque clairement que les optimizer adam et adamax donnent les meilleurs résultats que soit en terme de precision, rappel et f-mesure ou en terme d'accuracy ce qui valide les résultats présenté dans la littérature.

5.9 Résultats du modèle 1

5.9.1 Resultats avec variation du nombre d'époches

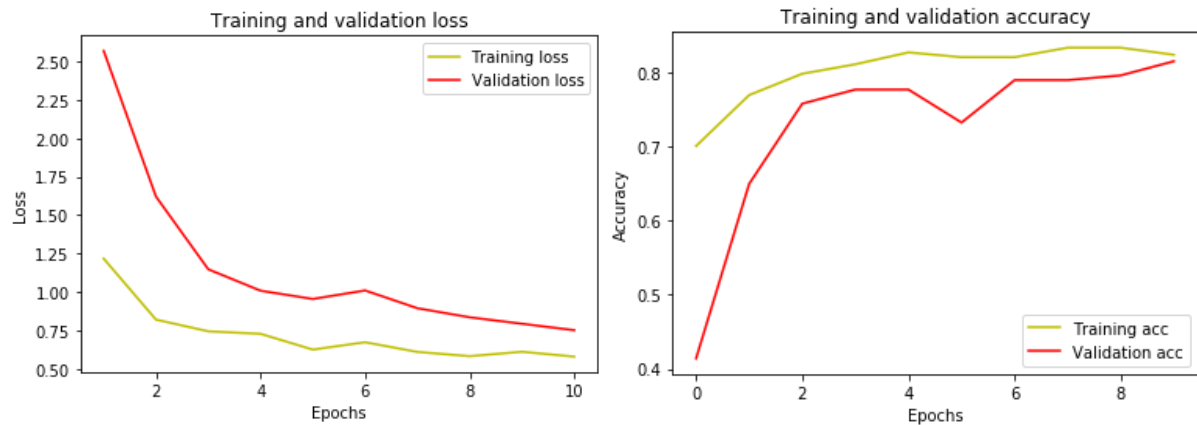


FIGURE 5.17: Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époches pour le modèle 1

accuracy			0.94	782
macro avg	0.67	0.60	0.62	782
weighted avg	0.92	0.94	0.93	782

TABLE 5.1: Résultats d'accuracy , macro avg et weighted avg pour modele 1

on remarque clairement que les résultats s'améliorent après nombre d'époche = 3 jusqu'à obtenir une accuracy = 0.9 avec nombre d'époche = 9. Cette situation est expliquée par le fait que l'optimizer commencent a générer moins d'erreur après nombre d'époche = 3. Au début il y'a un surapprentissage causé par le mal réglage du paramètres nombre d'époche.

5.9.1.1 precision rappel et f1-score

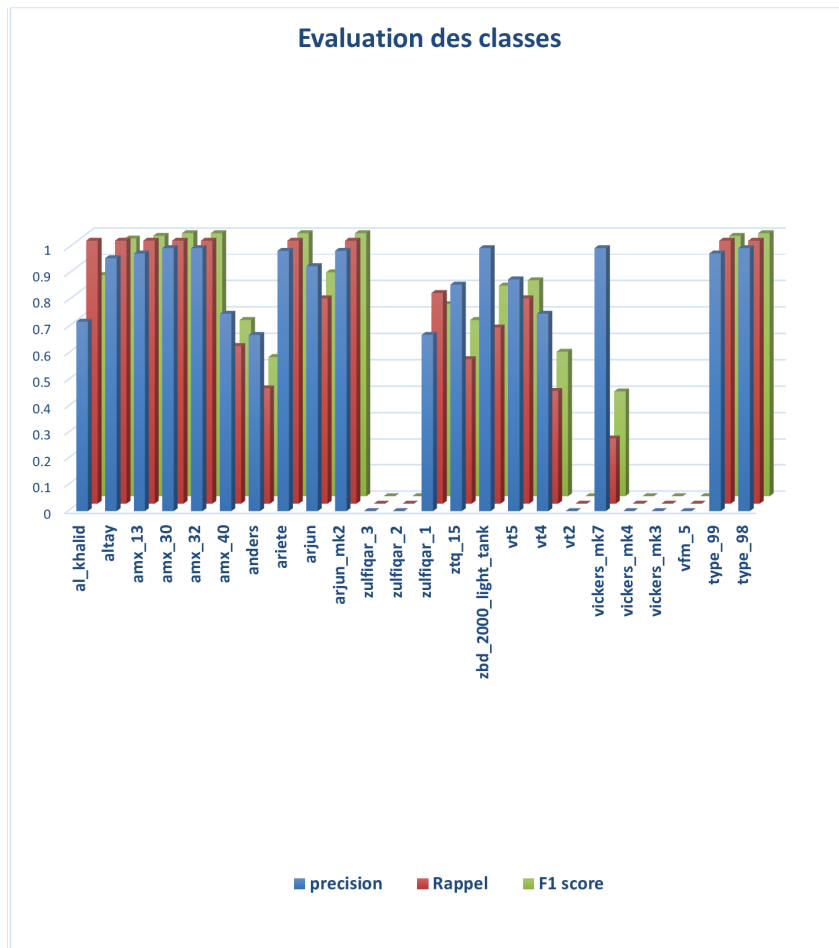
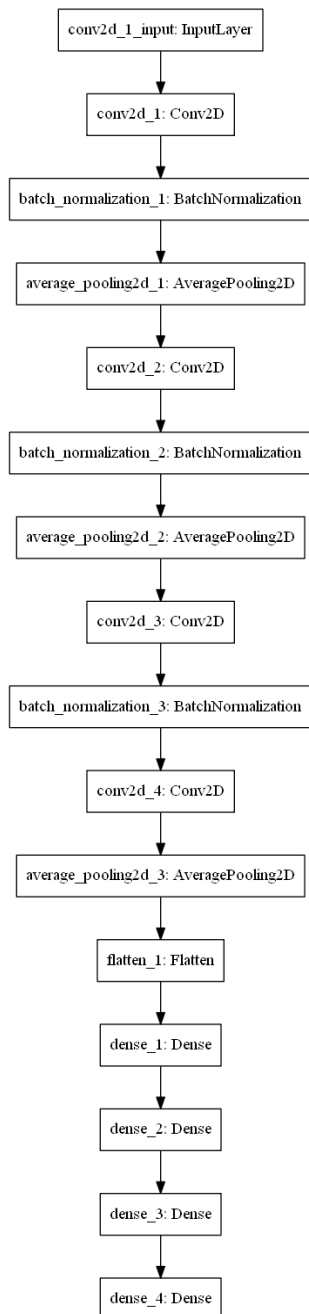


FIGURE 5.18: Precision rappel et f1-score pour chaque classes tank detection utilisant le modèle1

D'après les résultats on remarque clairement que précision et rappel et f1 score ont des meilleurs résultats pour les classes altay , armax 13 , armax 30 , armax 32 , ariete.

5.9.1.2 configuration et nombre de parametres modèle modélé1



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 75, 75, 50)	1850
batch_normalization_1 (Batch Normalization)	(None, 75, 75, 50)	200
average_pooling2d_1 (Average Pooling)	(None, 25, 25, 50)	0
conv2d_2 (Conv2D)	(None, 10, 10, 256)	3277056
batch_normalization_2 (Batch Normalization)	(None, 10, 10, 256)	1024
average_pooling2d_2 (Average Pooling)	(None, 2, 2, 256)	0
conv2d_3 (Conv2D)	(None, 2, 2, 384)	885120
batch_normalization_3 (Batch Normalization)	(None, 2, 2, 384)	1536
conv2d_4 (Conv2D)	(None, 2, 2, 256)	884992
average_pooling2d_3 (Average Pooling)	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 4096)	1052672
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 1000)	4097000
dense_4 (Dense)	(None, 24)	24024
Total params: 27,006,786		
Trainable params: 27,005,406		
Non-trainable params: 1,380		

FIGURE 5.19: Configuration et nombre de parametres modèle modélé1. Avec 27006786 parametres divisé en deux (trainable params = 27005406, non trainable params = 1380)

Le nombre de parametres joue un role important dans la qualite des resultats mais il est lie directement avec la quantite de donnees pour cette raison nous avons utilise le data augmentation afin d'obtenir des resultats efficaces.

5.9.2 Résultat modèle 1 avec Variation du l'optimizer

5.9.2.1 accuracy

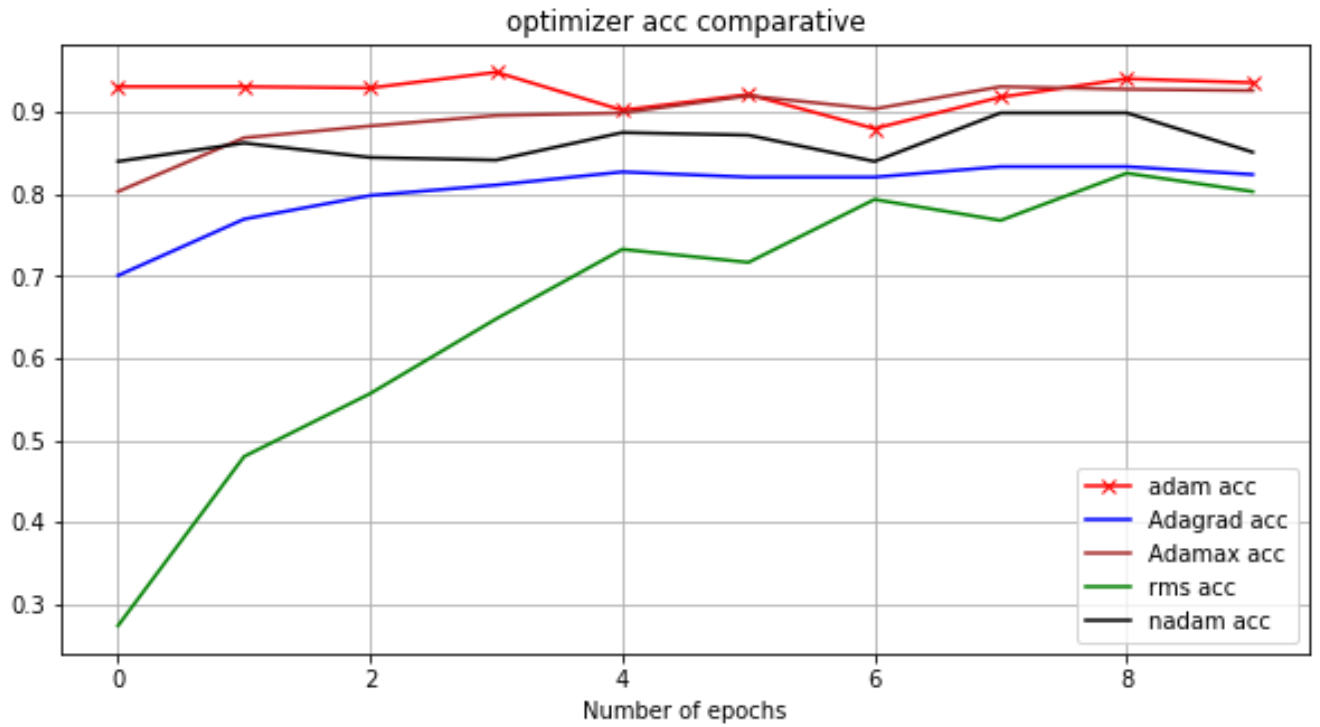


FIGURE 5.20: Acuracy du modèle avec variation du nombre d'epoches et optimizer

On remarque que l' accuracy a des valeurs élevé pour les optimizers adam et adamax .

5.9.2.2 validation accuracy

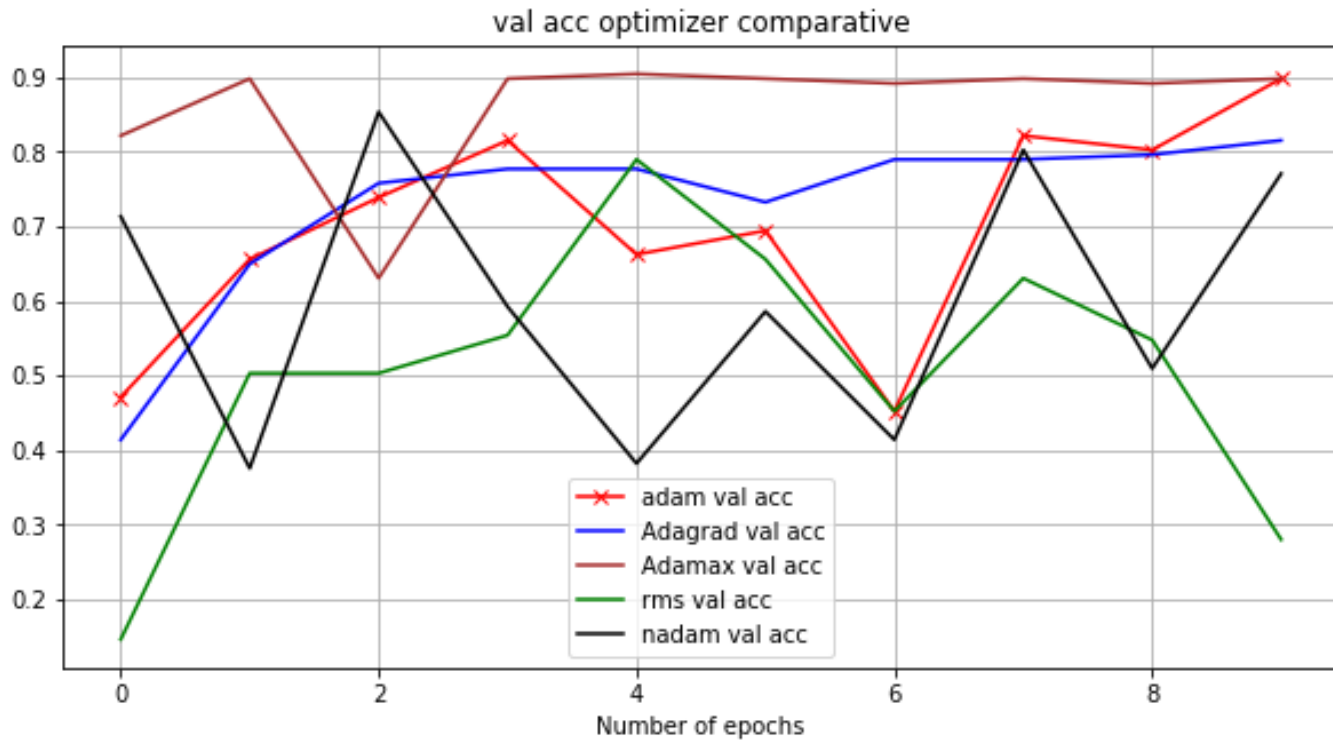


FIGURE 5.21: Validation accuracy du modèle1 avec variation du nombre d'époques et optimizer

L'optimizer adamax donne les meilleurs résultats pour le nombre d'époques supérieur à 3 en terme de validation accuracy.

5.9.2.3 validation loss

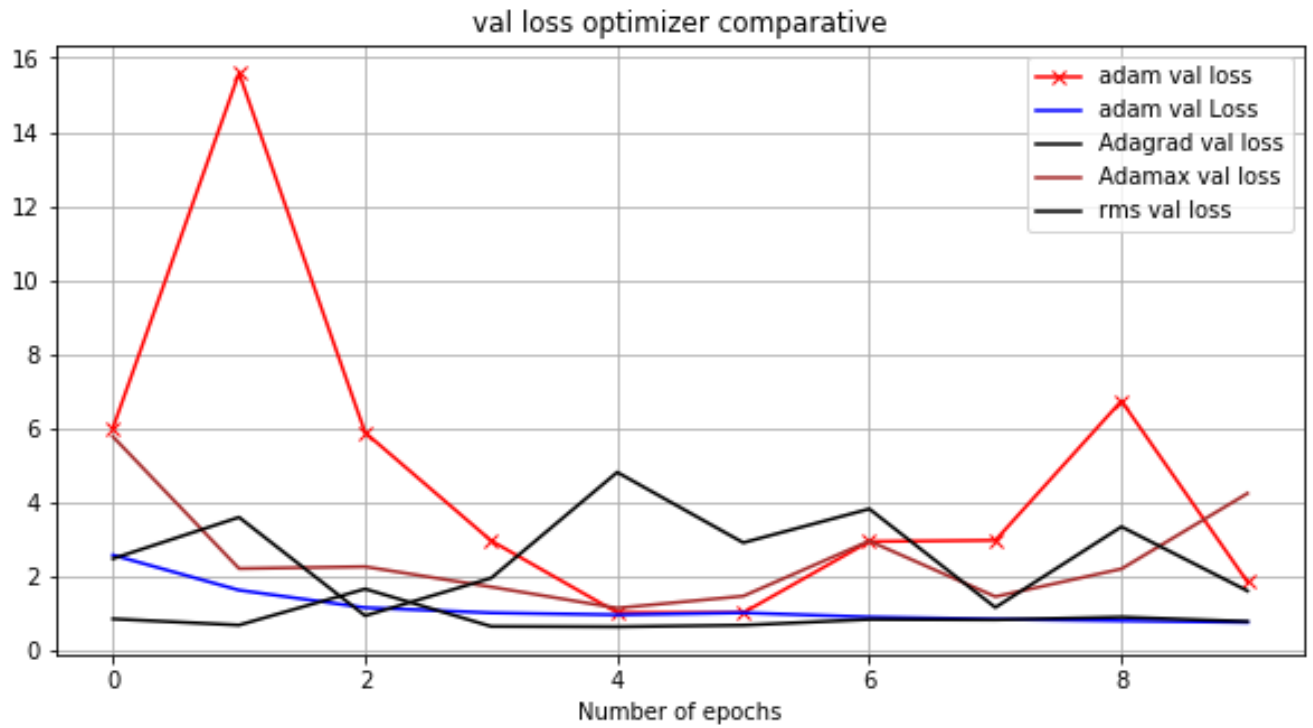


FIGURE 5.22: validation accuracy du modèle1 avec variation du nombre d'époques et optimizer

L'optimizer adam donne les meilleurs résultats que soit en terme de accuracy val accuracy ou val loss ce qui confirme les décisions trouvées dans la littérature.

5.9.2.4 Comparaison

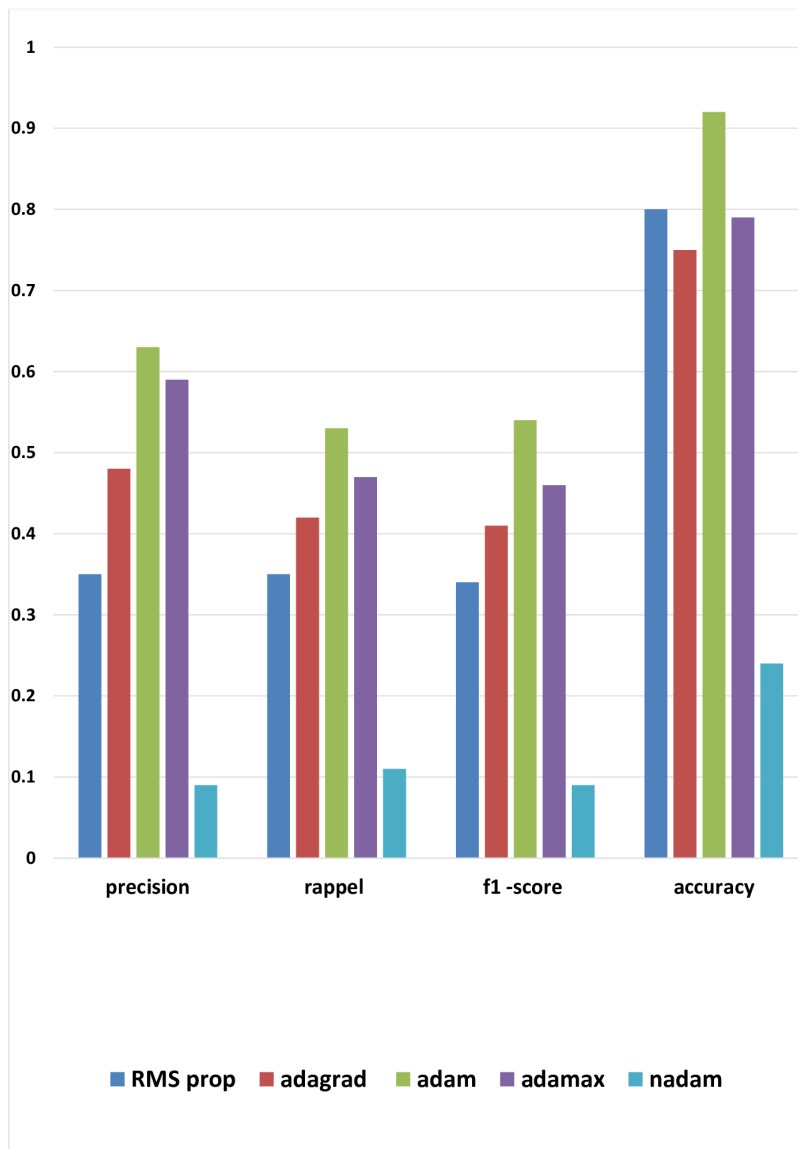


FIGURE 5.23: Modèle1 comparaison entre les optimizers en termes de f-score, rappel et precision

En terme de rappel precision et f-measure il est claire que la meilleur classification est donné par l'optimizer adam comme le montre la figure précédente.

5.10 Résultats du modèle VGG

5.10.1 Résultat du VGG avec variation du nombre d'époques

5.10.1.1 courbe accuracy

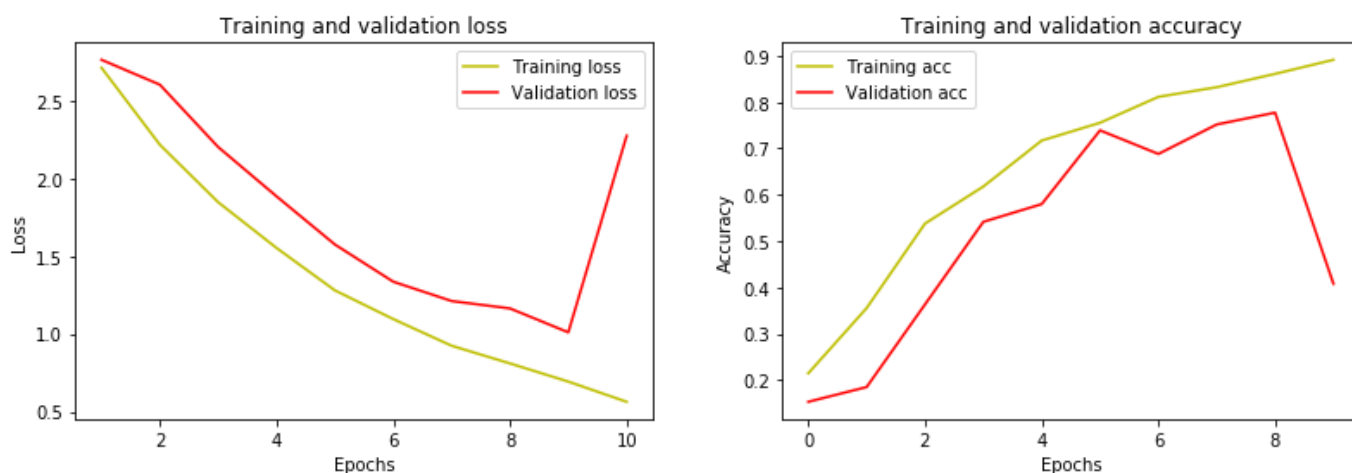


FIGURE 5.24: Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époques pour le modèle VGG-16

accuracy			0.92	782
macro avg	0.81	0.56	0.62	782

TABLE 5.2: Résultats d'accuracy et macro avrg en terme d'époque

On remarque clairement que les résultats s'améliorent après nombre d'époque = 3 jusqu'à obtenir une accuracy = 0.9 avec nombre d'époque = 9 pour training mais pour la validation les résultats augmentent jusqu'à atteindre la valeur d'époque = 3, puis diminuent immédiatement.

Cette situation est expliquée par le fait que l'optimizer commence à générer moins d'erreur après nombre d'époque = 3. pour la partie training. Et pour la validation la valeur d'erreur diminue jusqu'à nombre d'époque = 9 elle augmente.

5.10.1.2 Evaluation par barchart

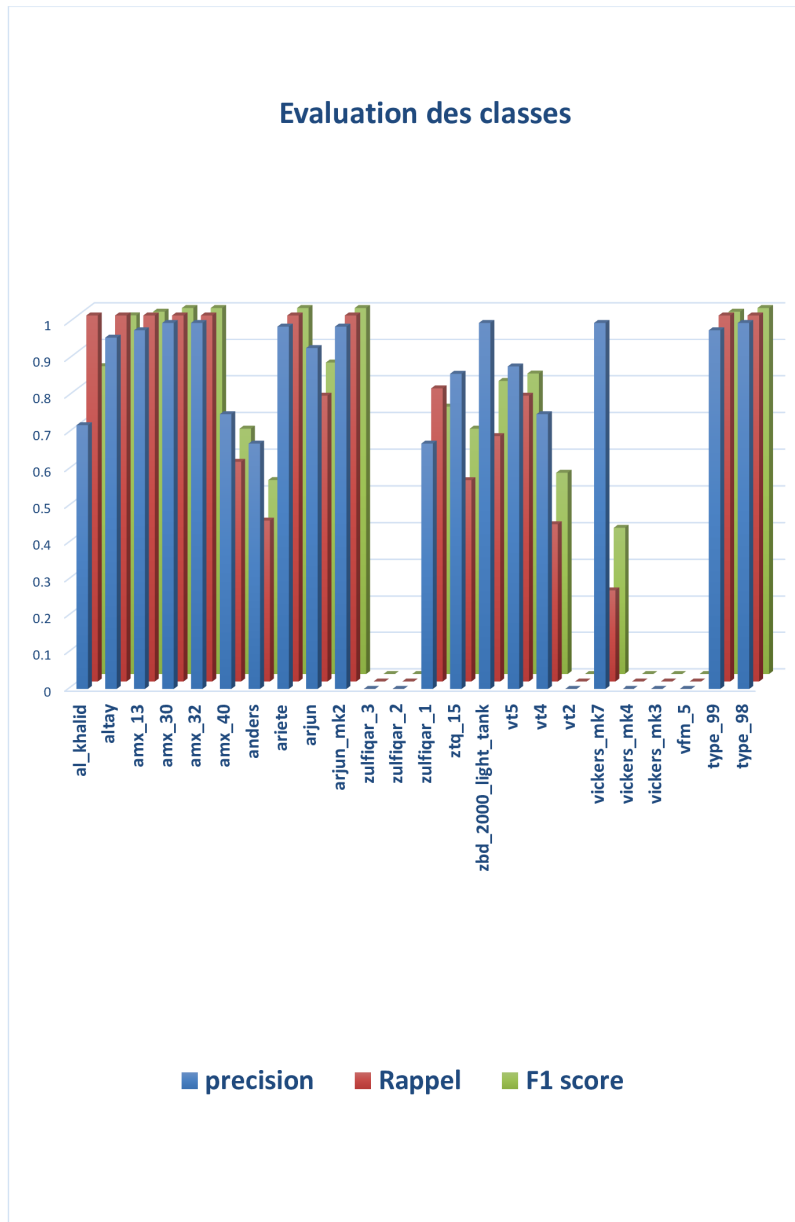
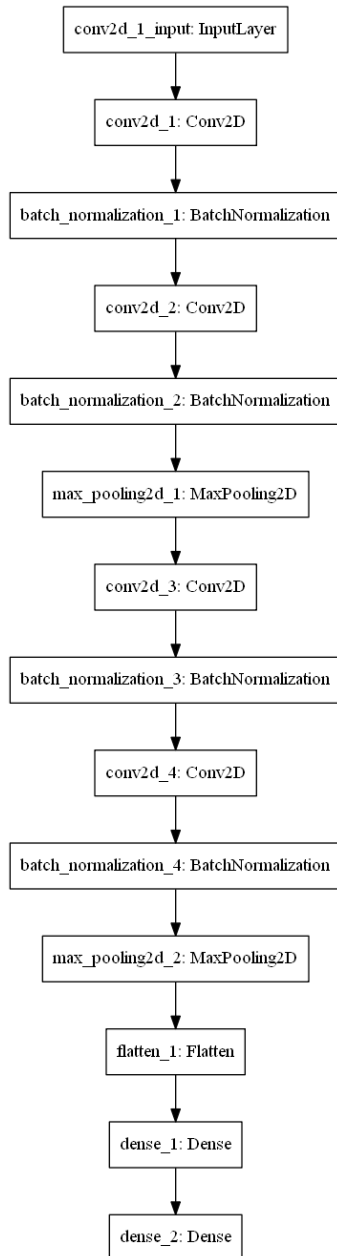


FIGURE 5.25: Evaluation par barchart en terme de precision recall, fmeasure de chaque classe

D'après les résultats on remarque clairement que précision et rappel et f1 score ont des meilleurs résultats pour les classes altay , armax 13 , armax 30 , armax 32 , ariete.

5.10.1.3 Configuration



Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 80, 80, 32)	320
batch_normalization_1 (Batch Normalization)	(None, 80, 80, 32)	128
conv2d_2 (Conv2D)	(None, 80, 80, 32)	9248
batch_normalization_2 (Batch Normalization)	(None, 80, 80, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 40, 40, 32)	0
conv2d_3 (Conv2D)	(None, 40, 40, 64)	18496
batch_normalization_3 (Batch Normalization)	(None, 40, 40, 64)	256
conv2d_4 (Conv2D)	(None, 40, 40, 64)	36928
batch_normalization_4 (Batch Normalization)	(None, 40, 40, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 20, 20, 64)	0
flatten_1 (Flatten)	(None, 25600)	0
dense_1 (Dense)	(None, 128)	3276928
dense_2 (Dense)	(None, 24)	3096
Total params: 3,345,784		
Trainable params: 3,345,400		
Non-trainable params: 384		

FIGURE 5.26: configuration et nombre de parametres modèle VGG-16. Avec 3345784 paramètres divisé en deux (trainable params = 3345400, non trainable params = 384)

5.10.2 Résultat avec variation de l'optimizer

5.10.3 Résultats avec adam

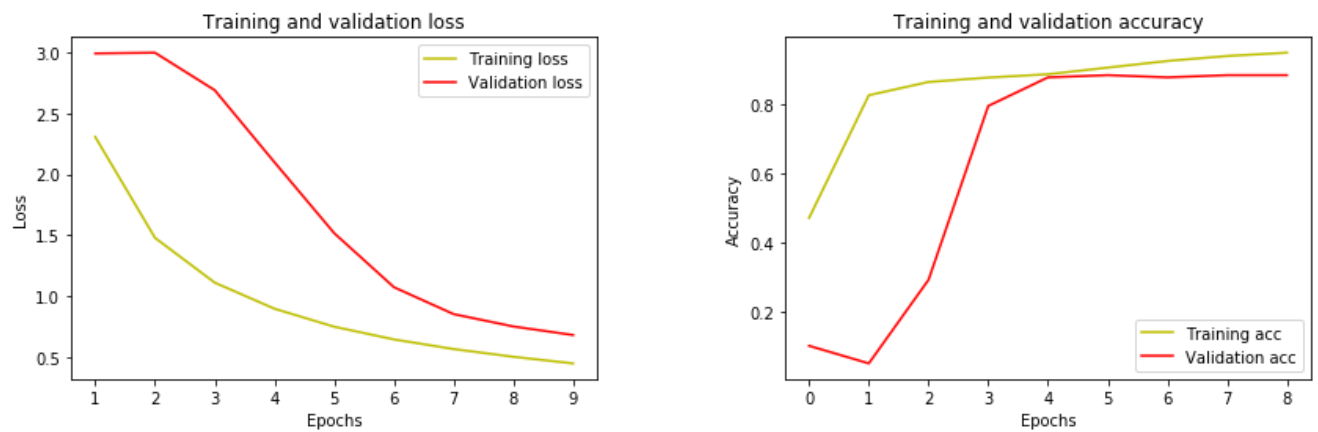


FIGURE 5.27: courbes accuracy, val accuracy, loss, val loss et variation du nombre d'epoches pour adam

On remarque que l'optimizer adam commencent a générer moins d'erreur après nombre d'epoche = 6. Au début il y'a un surapprentissage causé par le mal réglage du paramètres nombre d'poche.

5.10.4 Résultats avec admax

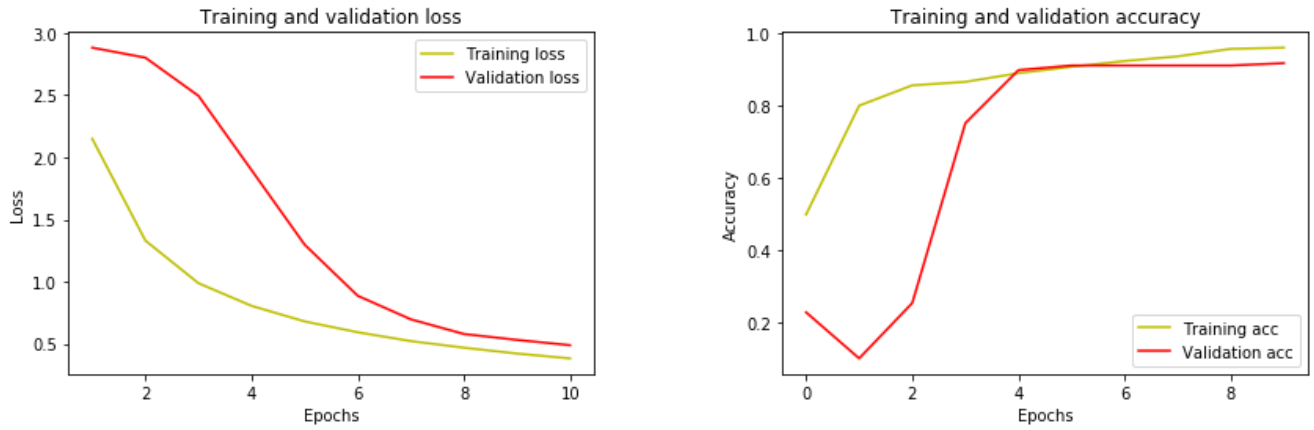


FIGURE 5.28: courbes accuracy, val accuracy, loss, val loss et variation du nombre d'epoches pour l'optimizer adamax

On remarque que l'optimizer adam commencent a générer moins d'erreur après nombre d'epoche = 5. Au début il y'a un surapprentissage causé par le mal réglage du paramètres nombre d'epoche.

5.10.5 Résultat avec nadam

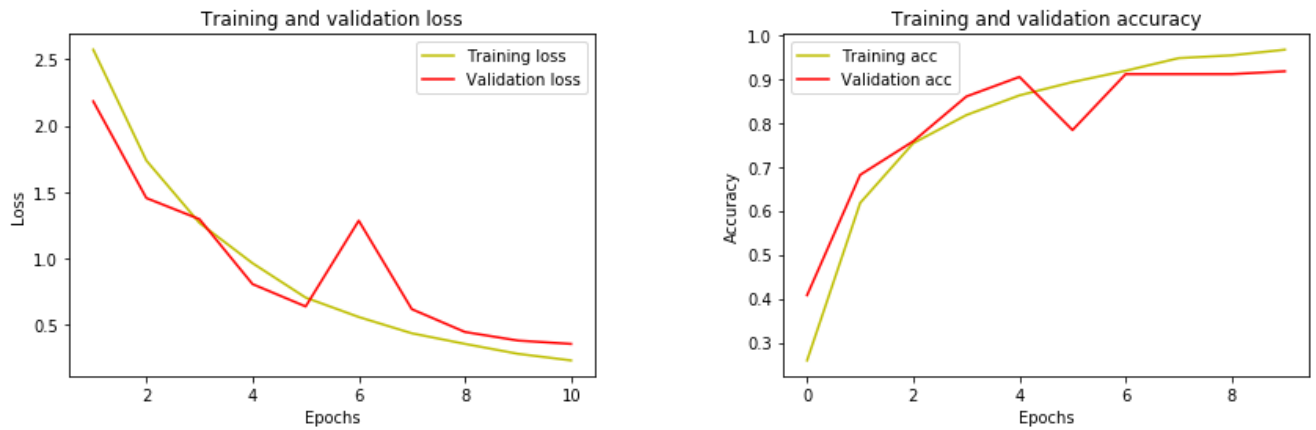


FIGURE 5.29: courbes accuracy, val accuracy, loss, val loss et variation du nombre d'epoches pour l'optimizer nadam

On remarque que l'optimizer nadam commencent a générer moins d'erreur après nombre d'epoche inférieur à 4 . Au début il y'a un sousapprentissage causé par le mal réglage du paramètres nombre d'epoche.

5.10.5.1 Résultats avec MSprop

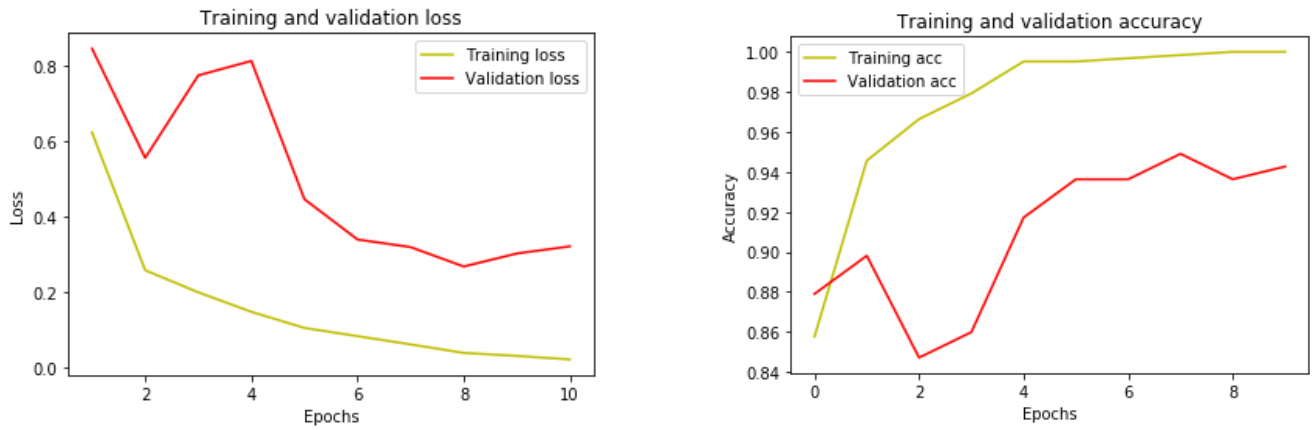


FIGURE 5.30: Courbes accuracy, val accuracy, loss, val loss et variation du nombre d'époques pour l'optimizer MSprop

Pour l'optimizer MSprop on remarque qu'il ya overfitting avec une grande différence entre training accuracy et validation accuracy.

5.10.6 comparison variation optimizer

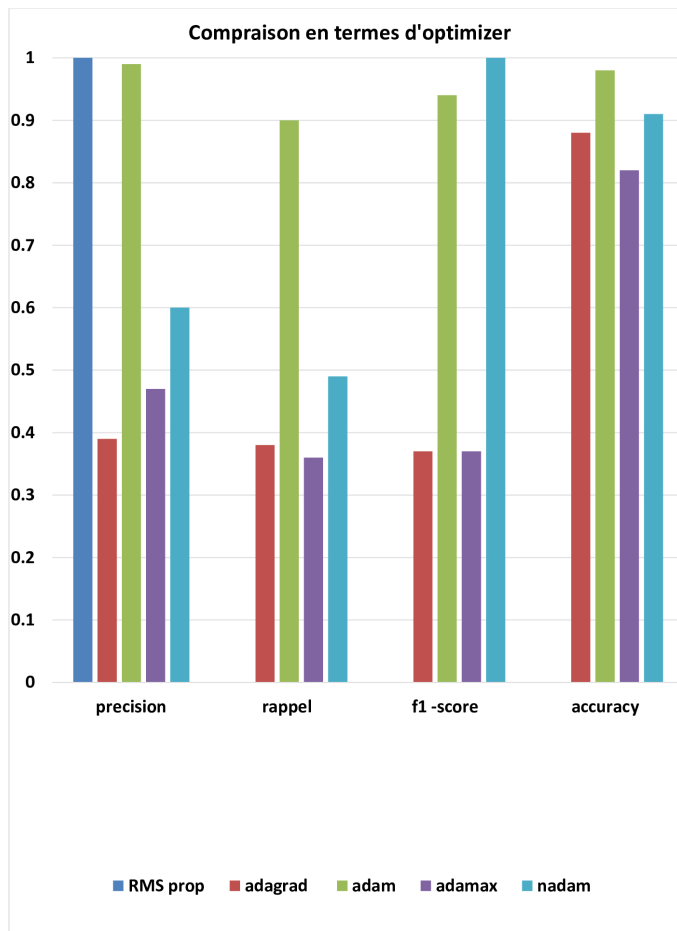


FIGURE 5.31: Modèle VGG comparaison entre les optimizers en termes de f-score, rappel et precision .

On remarque que les optimizers RMS prop et adam donnent les meilleurs résultats en terme de précision et rappel. ET en terme de f mesure et accuracy les optimizers nadam ,adam donnent les meilleurs résultats Donc l'optimizer adam donne les meilleurs résultats pour ce modèle .

5.11 Code source

Requet python	Role de requet
Input shape (54, 32,3)	la taille de l'image en entrée
pool size=(2,2)	taille de pool
Batch size	définit le nombre d'échantillons qui seront propagés sur le réseau
Image Data Generator	classe qui permet aux utilisateurs d'effectuer une augmentation d'image d'une manière très facile
Epoche	c'est le nombre d'itérations d'entraînement
Model=séquentiel ()	La fonction séquentielle initialise une pile linéaire de couches
model.add(Conv2D(input shape=X.shape[1:],filters=50, kernel size=(6,6), padding="same",activation="relu"))	Cette commande permet de créer 50 features maps en utilisant un filtre de taille 6*6 et un padding égal à la taille de l'image précédente, avec une dilatation standard=1, et une fonction d'activation ReLu
classifier.add(MaxPooling2D(pool size=(2,2)))	Cette commande permet de réduire la taille de l'image tout en conservant les caractéristiques importantes, en utilisant un filtre de taille 2*2
model.add(Flatten())	Un flatten qui va créer le vecteur final à envoyer au réseau de neurones artificiels (dense)
model.add(Dense(units=1, activation="sigmoid"))	Cette commande permet de créer une couche de sortie avec une fonction d'activation sigmoid car le travail consiste d'une

TABLE 5.3: Tableau de code source

5.12 Conclusion

Dnas ce chapitre nous avons vu les modeles proposer avec les differents variation de paramètres et en deduit le meilleur modele parmi les modeles proposer.

CONCLUSION GÉNÉRAL

Notre société devient de plus en plus intelligente a l'heure actuelle, et comme tous les autres domaines de l'intelligence artificielle, le domaine de la classification d'image a connu une évolution majeure depuis l'apparition du Deep Learning. Dans ce projet nous avons appliqué l'IA dans les guerres et vu son impact et pour cela nous avons proposé une approche, qui est basée sur les réseaux de neurones convolutionnels pour classer les attaques à détecter, donc nous avons parlé de deux architectures de CNN, et on l'a implémenté avec changement de nombre de couches et optimiser. Afin de valider notre approche, nous avons utilisé une Dataset Military tanks dataset, et on a testé notre modèle avec ces données afin d'obtenir des meilleurs résultats en termes de précision et d'erreur.

on peut ajouter notre programme dans un drone pour le tester réellement. Ajouter la carte pour suivre les localisations des tanks. Améliorer les résultats obtenus en testant d'autres configurations. Configurer un environnement IOT pour donner plus d'impact à notre idée.

BIBLIOGRAPHIE

- [1] <https://stanford.edu/~shervine/1/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>.
- [2] <https://keras.io/>.
- [3] <https://www.python.org>.
- [4] <https://www.kaggle.com/antoreepjana/military-tanks-dataset-images>.
- [5] <https://www.techleer.com/articles/305-vgg-16-an-advanced-approach-towards-accurate-large>
- [6] https://www.researchgate.net/figure/AlexNet-CNN-architecture-layers_fig1.
- [7] <https://www.tensorflow.org/>.
- [8] M. ABADI, M. ISARD, AND D. G. MURRAY, *A computational model for tensorflow : an introduction*, in Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, 2017, pp. 1–7.
- [9] H. ABDELAZIZ, I. OMAR, A. OUAHAB, ET AL., *La classification des images satellitaires par l'apprentissage profonde (deep learning)*, PhD thesis, Université Ahmed Draïa-Adrar, 2019.
- [10] C. D. ABZAC-EPEZY, *L'armée de l'air de Vichy : 1940-1944*, PhD thesis, Paris 1, 1996.
- [11] G. ALLEN AND T. CHAN, *Artificial intelligence and national security*, Belfer Center for Science and International Affairs Cambridge, MA, 2017.
- [12] M. Z. ALOM, T. M. TAHA, C. YAKOPCIC, S. WESTBERG, P. SIDIKE, M. S. NASRIN, B. C. VAN ESESN, A. A. S. AWWAL, AND V. K. ASARI, *The history began from alexnet : A comprehensive survey on deep learning approaches*, arXiv preprint arXiv :1803.01164, (2018).
- [13] R. ARANDJELOVIC, P. GRONAT, A. TORII, T. PAJDLA, AND J. SIVIC, *Netvlad : Cnn architecture for weakly supervised place recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.

BIBLIOGRAPHIE

- [14] L. BALLE, J. ROMERO, AND P. HENNIG, *Coupling adaptive batch sizes with learning rates*, arXiv preprint arXiv :1612.05086, (2016).
- [15] N. BENDIMERAD, *Système de surveillance d'infrastructures publiques à l'aide des réseaux de capteurs vidéo sans fil*, PhD thesis, Université d'Oran1-Ahmed Ben Bella, 2015.
- [16] S. BOUBAYA, D. BERBIT, ET AL., *DEEP LEARNING POUR LA SEGMENTATION D'IMAGES*, PhD thesis, university of M'sila, 2021.
- [17] É. BRUNET AND L. VANNI, *Deep learning et authentification des textes*, *Texte! Textes et Cultures*, 24 (2019), pp. 1–34.
- [18] A. BUHOT, *Etude de propriétés d'apprentissage supervisé et non supervisé par des méthodes de physique statistique*, PhD thesis, Université Joseph-Fourier-Grenoble I, 1999.
- [19] L. CAO AND J. KRUMM, *From gps traces to a routable road map*, in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 3–12.
- [20] J. CHAS, *Tableau historique et politique des opérations militaires et civiles de Bonaparte, premier consul de la République française*, de l'imprimerie de Guilleminet, 1801.
- [21] R. DERICHE AND O. FAUGERAS, *Les EDP en traitement des images et vision par ordinateur*, PhD thesis, INRIA, 1995.
- [22] I. DJELLOUL AND K. LATRECHE, *Optimisation et simulation du fonctionnement d'un système de production sous labview pour réaliser la technique de surveillance par approche neuro-flou*, in *International Conférence on Systems and Processing Information*, 2013.
- [23] R. A. DOUGHTY, *Pyrrhic victory : French strategy and operations in the great war*, *Politique étrangère*, (2009), pp. 678–681.
- [24] B. DYHIA AND I. THANINA, *Classification d'images agricoles avec le Deep Learning.*, PhD thesis, Université Mouloud Mammeri, 2019.
- [25] B. FALLERY, *Intelligence artificielle : à qui profitent les ambiguïtés entre l'informatique, l'humain et la fiction ?*, in *Journées IP&M sur L'Intelligence artificielle*, 2019.
- [26] S.-C. FISCHER, *Intelligence artificielle : les ambitions de la chine*, *Politique de sécurité : analyses du CSS*, 220 (2018).
- [27] A. GRIDI AND T. GUERRAM, *Un outil de deep learning pour les données textuelles*, (2020).
- [28] C. GROUIN, J.-B. BERTHELIN, S. EL AYARI, T. HEITZ, M. HURAUULT-PLANTET, M. JARDINO, Z. KHALIS, AND M. LASTES, *Présentation de deft'07 (défi fouille de textes)*, *Actes du troisième DEfi Fouille de Textes*, (2007), p. 3.

- [29] R. HORAUD AND O. MONGA, *Vision par ordinateur : outils fondamentaux*, Editions Hermès, 1995.
- [30] Z. JIANG, L. ZHAO, S. LI, AND Y. JIA, *Real-time object detection method based on improved yolov4-tiny*, 2020.
- [31] R. KESSLER, J. M. TORRES-MORENO, AND M. EL-BÈZE, *Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé*, les actes de VSST, (2004), pp. 493–504.
- [32] N. KETKAR, *Introduction to keras*, in *Deep learning with Python*, Springer, 2017, pp. 97–111.
- [33] H. LAROCHELLE AND Q. SHERBROOKE, *Historique de travail*, Études, 3, p. 2Y5.
- [34] L. LAZEBNIK, *Convolutional neural network architectures : from lenet to resnet*, Presentation, University of Illinois, accessed, 20 (2018).
- [35] D. LEARNING, *Deep learning*, High-Dimensional Fuzzy Clustering, (2020).
- [36] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, nature, 521 (2015), pp. 436–444.
- [37] J. LEMAIRE, *La nouvelle vague de l'intelligence artificielle et le monde de la défense*, Revue Defense Nationale, (2018), pp. 83–88.
- [38] S. S. LIEW, M. KHALIL-HANI, AND R. BAKHTERI, *Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems*, Neurocomputing, 216 (2016), pp. 718–734.
- [39] P. MAHTO, P. GARG, P. SETH, AND J. PANDA, *Refining yolov4 for vehicle detection*, International Journal of Advanced Research in Engineering and Technology (IJARET), 11 (2020).
- [40] M. NAUGE, *Executable papers : vers une science plus ouverte, transparente et réutilisable*, in Progedo : Semaine DATA SHS, 2020.
- [41] J.-C. NOËL, *Intelligence artificielle : vers une nouvelle révolution militaire ?*, Focus stratégique, 84 (2018).
- [42] H. NOURA AND K. NADIA, *Classification des textures par les réseaux de neurones convolutifs*, PhD thesis, Université Mouloud Mammeri, 2018.
- [43] K. O'SHEA AND R. NASH, *An introduction to convolutional neural networks*, arXiv preprint arXiv :1511.08458, (2015).
- [44] H. H. PERRITT JR AND E. O. SPRAGUE, *Drones*, Vand. J. Ent. & Tech. L., 17 (2014), p. 673.

- [45] J. H. SEJR, P. SCHNEIDER-KAMP, AND N. AYOUB, *Surrogate object detection explainer (sodex) with yolov4 and lime*, Machine Learning and Knowledge Extraction, 3 (2021), pp. 662–671.
- [46] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv :1409.1556, (2014).
- [47] A. TATO AND R. NKAMBOU, *Improving adam optimizer*, (2018).
- [48] F. TORRE, *Globo : un algorithme stochastique pour l'apprentissage supervisé et non-supervisé*, in Actes de la Première Conférence d'Apprentissage, Citeseer, 1999, pp. 161–168.
- [49] C. TOUZET, *les réseaux de neurones artificiels, introduction au connexionnisme*, Ec2, 1992.
- [50] P. WANG, J. XU, B. XU, C. LIU, H. ZHANG, F. WANG, AND H. HAO, *Semantic clustering and convolutional neural network for short text categorization*, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers), 2015, pp. 352–357.
- [51] J. WU, *Introduction to convolutional neural networks*, National Key Lab for Novel Software Technology. Nanjing University. China, 5 (2017), p. 495.
- [52] P. YAO, H. HUANG, W. TANG, Q. SHI, R. WU, AND C. ZHANG, *Fuzzing smt solvers via two-dimensional input space exploration*, in Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2021, pp. 322–335.
- [53] V. ZOCCA, G. SPACAGNA, D. SLATER, AND P. ROELANTS, *Python deep learning*, Packt Publishing Ltd, 2017.