

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Algiers 1 Benyoucef BENKHEDDA



Faculté des Sciences
Département d'informatique
Mémoire de fin d'étude pour l'obtention du diplôme de
Master en Informatique
Spécialité : Ingénierie de systèmes informatique intelligents

Présenté par :

Melle BOUCHELALEG Houria

Melle KOURI Ferial

Thème

Conception et Réalisation d'un système de
contrôle d'état des rails ferroviaires.

Devant le jury composé de :

Mme. GUENDOUZI.W	Pr. Université Alger 1	jury
Mme. BEGHLI	Pr. Université Alger 1	jury
Mme. SOUAMI.F	Pr. Université Alger 1	Encadrante

Année universitaire : 2021-2022

Remerciements

*Nous tenons à remercier en premier lieu **ALLAH** de nous avoir donné la force, la patience, la volonté afin d'accomplir ce modeste travail qui présente le fruit de plusieurs années de sacrifices.*

*Nous avons l'honneur et le plaisir de présenter notre profonde gratitude et nos sincères remerciements à notre promotrice : **Mme SOUAMI** et notre Co-promoteurs **Mr SAMADI Karim, Mr BERCHICHE Khaled et Mr GUENOUNE Kheireddine** pour leur précieuse aide, leurs orientations et le temps qu'ils ont accordé à notre encadrement.*

*Nous tenons à remercier également l'ensemble de nos professeurs chacun avec son nom pour la qualité de l'enseignement qu'ils nous ont prodigué au cours des cinq années passées à l'université d'Alger 1 BENYOUCEF
BENKHEDDA.*

*Nous remercions par ailleurs vivement les membres du jury **Mme SOUAMI, Mme GUENDOUZI et Mme BEGHLI** de nous avoir fait l'honneur d'examiner notre travail.*

*Sans oublier de remercier **Mr LOUNIS Amar et Mr HAMA Soltani.***

Arrivés au terme de la rédaction de ce mémoire, nous exprimons notre gratitude pour toutes les personnes qui ont participé de près ou de loin à notre réussite.

Dédicaces

Je commence par remercier Dieu, qui m'a donné la force, la volonté et la perspicacité pour réaliser ce travail.

*Je dédie ce modeste travail : A cet homme, si précieux pour moi à qui je dois durant toute ma vie, tout mon respect et ma gratitude, lui qui a toujours été à mes côtés et qui a énormément contribué à la réalisation de ce travail : mon valeureux et cher Père **AZZEDINE**.*

*A cette femme qui a souffert sans me laisser souffrir, qui n'as jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse et qui a énormément contribué à la réalisation de ce travail : mon adorable mère **SAFIA**.*

*A mes merveilleuses sœurs **FAHIMA, RADIA, ASMA** et **SARA**, et mes chers frères **ISSAM** et **BADRIDDINE**.*

*A mon neveu **AYOUB** et mes nièces **AYA, ASSIL** et **RACHA**.*

*A mes chers grands parents **ABDELKADER** et **YAMINA**.*

*A mes très chers grands parents **ALI** et **KIDIRA** qui nous ont quitté que Dieu les garde dans son vaste paradis.*

*A mon cher oncle **OMAR** et ma tante **AICHA, HOURIA**.*

*A mon Binôme **FERIEL** qui a partagé ce travail avec moi ainsi qu'à sa famille.*

*A mes chères amies **FARAH, NOUARA, NERMINE, LOUIZA, NESRINE, MALIKA** et **RANIA**.*

A mes très chères cousines et leurs enfants. Merci pour votre amour.

BOUCHELALEG Houria

Dédicaces

*Je dédie ce travail à l'âme pure de **Mon Père** qui me manque*

trop 

*À ma très **chère mère**, à mon cher frère **Khireddine** et mes chères sœurs, que dieu les gardes et les donne de la bonne santé.*

*À ma sœur, ma meilleure amie, mon binôme **HOURIA** et à toute sa chère famille.*

*À Mon patron au travail oncle N. **BOUDJEMAA** et sa famille*

À ceux qui me donnent de l'amour et de la vivacité et à ceux qui m'ont aidé et supporter dans les moments difficiles.

À tous ceux qui m'aime.

À la forte Moi...

KOURI Ferial

Résumé

Ces dernières années, de plus en plus d'entreprise veulent investir dans des applications de Deep Learning et de vision par ordinateur. En effet, la détection d'objets est un domaine très actif de la recherche qui vise à localiser des régions ou zones d'une image ou d'un flux vidéo. Dans ce projet, nous voulons détecter à partir d'une image, un rail ferroviaire et y extraire ses défaillances (usures) afin que l'entreprise puisse réparer les corrosions trouvées au temps voulu, pour ne pas perdre du temps ni de l'argent et donc offrir un meilleur service aux citoyens.

Dans le but d'atteindre ces objectifs, l'entreprise SETRAM souhaite automatiser ce processus de détection d'usures dans les rails ferroviaires. Une telle transition implique le développement d'une plateforme pour répondre aux besoins de l'entreprise.

Notre système comporte une application web destinée aux employés afin qu'il puisse insérer une image d'un rail, détecter et extraire les types d'usures qui s'y trouvent et enfin envoyer les experts au terrain pour réparer les dégâts trouvés.

Mots clés: Rail ferroviaire, usure, apprentissage profond, détection d'objets, VGG16, YOLOV4, application Web.

Sommaire

Table des matières	i
Liste des figures	v
Liste des tableaux	ix
Liste des abréviations	x
Introduction générale	1
1. Cadre du projet	3
1.1.Introduction	4
1.2.Historique	4
1.3.Présentation de SETRAM	4
1.4.Mission de l'entreprise	5
1.5.Activité	5
1.6.Unité opérationnelle d'Alger	9
1.7.Pôles existants au niveau de la SETRAM	9
1.7.1.Pôles maintenances	10
1.8.Objectif	10
1.9.Problématique	11
1.9.1.Inspection des appareils de voies	11
1.9.2.Inspection des rails	11
1.10. Conclusion	12
2. Etat de l'art.....	13
2.1. Introduction	13
2.2. L'apprentissage profond	13
2.2.1.Relation entre l'intelligence artificielle, ML et le Deep Learning.....	13
2.2.2.L'intelligence artificielle	14
2.2.3. Machine Learning.....	15
2.2.4. Deep Learning.....	16
2.3. CNN : Réseau de neurone à convolution.....	17
2.3.1. Couche de convolution.....	18

2.3.2. Couche de mise en commun (Pooling).....	18
2.3.3. Couche d'aplanissement (Flattering).....	19
2.3.4. Couche dense (full connexion).....	20
2.4. Mask R-CNN.....	20
2.5. VGG 16	21
2.5.1. Architecture.....	22
2.6. YOL.....	23
2.6.1. Blocs résiduels.....	23
2.6.2. Régression de la boîte englobante.....	24
2.6.3. Intersection sur union (IOU).....	24
2.6.4. Combinaison des trois techniques.....	24
2.7. Description de l'architecture	25
2.8. Etude de l'existant	26
2.8.1. Vossloh	26
2.8.1.1. Service d'inspection	26
2.8.1.2. Service de mesure	27
2.8.2. Speno	27
2.8.2.1. Solution proposée par Speno	28
2.8.2.3. BLET instruments et contrôles	28
2.9. Conclusion	30
3. Conception.....	30
3.1. Introduction	30
3.2. Définition du rail	30
3.2.1. Type de rails	31
3.3. Définition de l'usure	32
3.3.1. Les types d'usures	32
3.4. Architecture générale du système	33
3.4.1. Collection de données	36

3.4.2. Segmentation	36
3.4.3. Annotation	36
3.4.3.1. Annotation pour Mask RCNN	37
3.4.3.2. Configurations pour le MaskR-CNN	38
3.4.3.3. Annotation de la détection de l'usure	39
3.4.4. Augmentation des données	41
3.5. Les modèles qualifiés	42
3.5.1. Configuration de VGG16	42
3.5.2. Configuration de YOLOV4	43
3.6. Conclusion	44
4. Implémentation	45
4.1. Introduction	45
4.2. Langages de programmation et environnement de travail	45
4.2.1. Python	45
4.2.2. TensorFlow	45
4.2.3. Keras	46
4.2.4. Google Colab	47
4.2.5. Flask	47
4.2.6. Visual studio code	48
4.2.7. Wampserver	48
4.2.8. MySql	48
4.3. Configuration du matériel utilisée	49
4.4. Résultats et discussions	49
4.4.1. VGG16	50
4.4.2. YOLOv4	51
4.5. Comparaison des deux modèles	51
4.6. Application Web	54

4.7Conclusion	57
Conclusion générale.....	58
Bibliographie	60
A. VGG16 pour deux classes	61
B. Application Web pour le VGG16	63
Abstract	56

Liste des figures

Figure 1: logo SETRAM.	4
Figure 2 : Réseau tramways Algérie.	5
Figure 03 : Tramway d'Alger.	6
Figure 04 : Tramway d'Oran.	6
Figure 05: Tramway de Constantine.	7
Figure 06 : Tramway de SidiBelAbbas.	7
Figure 07: Tramway de Sétif.	8
Figure 08 : Tramway d'Ouargla.	8
Figure 09 : Ligne de tramway d'Alger.	9
Figure 10: Schéma général du système proposé.	12
Figure 11 : Schéma de la relation entre le IA, ML et DL.	14
Figure 12 : Processus du Machine Learning.	15
Figure 13 : Processus du Deep Learning.	16
Figure 14 : Exemple d'un réseau de neurone.	17
Figure 15 : Architecture du CNN.	17
Figure 16 : Application du filtre sur une image.	18
Figure 17 : Explication du Pooling.	19
Figure 18 : Représentation de la couche full connexion.	19
Figure 19 : Application du Mask R-CNN.	17
Figure 20 : Architecture Algorithme VGG16.	21
Figure 21 : Structure Algorithme VGG16.	21
Figure 22 : Illustration des blocs résiduels.	21
Figure 23 : Illustration d'une boite englobante.	24
Figure 24 : Illustration d'une détection YOLO.	24
Figure 25 : Architecture de YOLO.	25
Figure 26 : Détails du couple rail-traverse.	31
Figure 27 : Les types de rails.	31
Figure 28 : Usure ondulatoire.	32

Figure 29 : Usure latérale de la bavette.	33
Figure 30 : Usure de flanc de champignon.	33
Figure 31: Architecture du modèle proposé.	35
Figure 32 : Exemple d'annotation dans LABELME.	37
Figure 33 : Fichier JSON généré dans l'annotation.	38
Figure 34 : Images avant et après la segmentation.	39
Figure 35 : Annotation de la détection de l'usure.	40
Figure 36 : Fichiers .txt généré dans l'annotation.	40
Figure 37 : Floutage	41
Figure 38 : Rotation à droite	41
Figure 39 : Rotation à gauche	41
Figure 40 : Rotation 180°	41
Figure 41 : illustration du fichier classes.names	43
Figure 42 : illustration du fichier obj.data	43
Figure 43 : illustration du fichier yolo_custom_test.cfg.	44
Figure 44 : Logo de Python	45
Figure 45 : Logo TensorFlow.	46
Figure 46 : Logo de Keras.	46
Figure 47 : Logo de Google Colab.	47
Figure 48 : Logo de Flask.	47
Figure 49 : Logo de Visual Studio.	48
Figure 50 : Logo de WampServer.	48
Figure 51 : Logo de MySql.	49
Figure 52 : courbe graphique de la précision et la perte pour 4 classes.	50
Figure 53 : résultats détaillé de la précision et la perte pour 4 classes.	50
Figure 54 : résultat du test pour le modèle VGG16.	51
Figure 55 : Exemple de test du modèle VGG16	51
Figure 56 : Schéma d'entraînement de YOLOv4.	52
Figure 57 : précision d'entraînement.	52

Figure 58 : précision de chaque classe.	53
Figure 59 : prédiction de la classe non usure.	53
Figure 60 : L'interface graphique principale.	54
Figure 61 : Interface graphique 2.	55
Figure 62 : prédiction de la classe usure Ondulatoire.	55
Figure 63 : prédiction de la classe usure de Champignon.	56
Figure 64 : Prédiction de la classe usure de la Bavette.	56
Figure 65 : Prédiction de la classe non usure.	57
Figure 66 : courbe graphique de la précision et la perte VGG16 pour 2 classes.....	61
Figure 67 : résultats détaillé de la précision et la perte pour 2 classes.	61
Figure 68 : Interface graphique pour VGG16.	63
Figure 69 : affichage de l'image à prédire.	64
Figure 70 : Affichage du résultat de la prédiction.....	64

Liste des tableaux

Tableau 1 : Tableau de la base de données.....	36
Tableau 2 : tableau de configuration du MaskR-CNN	38
Tableau 3 : Base de données avant et après l'augmentation de données.....	42
Tableau 4 : Tableau de configuration de VGG16.	42
Tableau 05 : Tableau comparatif entre YOLOv4 et VGG16.	53

Liste des abréviations

VPO : Vision Par Ordinateur.....	1
IA : Intelligence Artificielle.....	1
SETRAM : Société d'Exploitation des Tramways.....	5
ML : Machine Learning.....	14
DL : Deep Learning.....	14
CNN : Convolutional Neural Network.....	17
R-CNN : Region-CNN	20
YOLO : You Only Look Once.....	22

Introduction générale

Le monde a explosé ces dernières années par une révolution technologique qui transforme nos vies, qui est l'intelligence artificielle cette dernière est devenu une partie importante de la vie. Le monde se tourne vers l'intelligence artificielle car cette technologie a pu répondre aux plusieurs questions et résoudre plusieurs problèmes complexes, chose que les autres technologies n'ont pas pu réussir à faire.

L'un des domaines les plus anciens et les plus importants de l'IA est la vision par ordinateur, elle permet d'analyser des images captées par un équipement tel qu'une caméra. La VPO est capable de reconnaître une image, de la comprendre et de traiter les informations qui en découlent.

Les applications modernes de la vision par ordinateur s'appuient sur ce qui est connu le Deep Learning. Avec ce dernier, une application de vision par ordinateur s'exécute sur un type d'algorithme appelé réseau de neurone, ce qui lui permet de fournir des analyses encore plus précises des images. Par ailleurs, le Deep Learning permet à un programme de VPO de conserver les informations de chaque images qu'il analyse, de sorte que plus il est utilisé plus sa précision augmente.

Contexte générale

L'impact de l'intelligence artificielle et ses domaines sur le monde de travail et notre quotidien redistribue le rôle de chacun. De nombreuses entreprises utilisent l'intelligence artificielle pour augmenter l'efficacité, augmenter les revenus, et fournir un service à forte valeur ajoutée dans un univers professionnel concurrentiel.

Nous allons aborder dans ce mémoire l'utilisation de l'IA dans le domaine de la sécurité ferroviaire.

C'est pour cela la société d'Exploitation des Tramways 'SETRAM' à besoin d'automatiser le processus de la sécurité ferroviaire qui est maintenant fait manuellement.

L'objectif étant de réduire un maximum d'erreurs que le processus humain peut causer comme la négligence d'un morceau de la voie ferrée et ne pas l'inspecter, réduire le temps de

l’inspection et donc gagner plus du temps et de l’argent (remis rapidement les tramways et les métros en service et donc ne pas perturber les intérêts des gens).

Problématique et motivation

La sécurité ferroviaire est faite maintenant manuellement par des experts des domaines en utilisant des équipements spéciaux comme le chariot d’inspection. Ce dernier doit être placé sur les deux extrémités du chemin de fer, les experts le déplacer le long de la voie ferrée ou juste la distance qu’ils veulent vérifier et puis la réparer.

Cette tâche prendra sûrement une longue durée de temps, et ça va être une grande perte du temps et d’argent si les experts trouvent que le rail contient des usures, car la réparation de la corrosion ferroviaire provoque une suspension temporaire du mouvement des Tramways.

Afin de résoudre ce problème nous faisons appel à une solution basée sur l’intelligence artificielle, qui est un système de contrôle d’état des rails ferroviaires. Le système comprendra un ensemble d’algorithmes de l’IA qui vont traiter l’état des rails grâce à ses images prises par des caméras spéciales. Le rôle principal de ce système est de prédire les types d’usures.

Cette solution peut éviter les grands dégâts et réduire l’arrêt du trafic des Tramways dû aux dysfonctionnements ferroviaires en réparant la panne en dehors des heures de travail.

Contribution

Au cours de notre stage, notre objectif principal était d’introduire l’intelligence artificielle au processus de contrôle d’état de rails par l’utilisation d’un modèle de Machine Learning. Pour ce faire, nous avons suggéré de concevoir :

1. Un premier modèle qui va faire la classification des images selon leurs classes d’usures.
2. Un deuxième modèle qui va déterminer la localisation de l’usure dans l’image en plus de la classification.
3. Une application web qui affiche les résultats su traitement du modèle performant.

Plan du mémoire

Pour cela nous avons structuré notre travail en quatre grands chapitres qui sont les suivants :

- 1.** Le premier chapitre est consacré à l'étude préalable dans laquelle nous présentons le cadre de travail après avoir procédé à une présentation de l'organisme d'accueil.
- 2.** Le deuxième chapitre est divisé en deux parties, la première présente la description de l'aspect théorique de l'Intelligence Artificielle et du Deep Learning avec ses domaines d'applications, ses modèles de segmentation et détection d'objets et extraction d'informations avec leurs architectures. Ensuite une deuxième partie consacrée à l'étude de l'existant à travers quelques articles expliquant la solution au problème d'usures des rails ferroviaires.
- 3.** Le troisième chapitre, décrit la phase de conception de la solution, en présentant l'architecture du système et expliquant chacune de ses étapes, ainsi que les modèles utilisés pour son implémentation.
- 4.** Le dernier chapitre présente les résultats expérimentaux, leurs interprétations et discussions des précisions et performances pour finir avec la présentation d'une application web afin de concrétiser le modèle.
- 5.** Nous clôturons ce mémoire par une conclusion générale résumant notre travail et présentant nos perspectives pour une meilleure amélioration de notre système.

1.Cadre du projet

1.1.Introduction

De nos jours, L'automatisation informatique est devenue essentiel pour une meilleure optimisation des tâches et une efficience dans la réalisation des prestations. Notre projet consiste à mettre en place un système automatisé qui permet de contrôler l'état des rails en temps réel et qui permet de maîtriser le risque de l'usure en donnant des alertes préventives.

1.2.Historique

Créée en septembre 2012, la SETRAM est une co-entreprise algéro-française, filiale des sociétés algériennes ETUSA (36 %) et EMA (15 %), filiale du groupe TRANSTEV, et de la société française RATP Dev (49 %), filiale du groupe RATP. Initialement, la co-entreprise SETRAM devait être détenue à 49 % par RATP El Djazaïr, à 30 % par l'EMA et à 21 % par l'ETUSA.

1.3.Présentation de SETRAM

Dans le cadre d'un partenariat entre l'Entreprise du Métro d'Alger (EMA), la Régie Autonome des Transports Parisiens (RATP Développement) et RATP El Djazaïr ainsi que l'Entreprise de Transports Urbains et Suburbains d'Alger (ETUSA), il est créé une société de droit algérien chargée de la maintenance et de l'exploitation des Tramways en Algérie.



Figure 1:logo SETRAM.

La société est dénommée "Société d'Exploitation des Tramways" par abréviation SETRAM.

L'EMA participe de manière active au développement de SETRAM. RATP Développement actionnaire à raison de 41% apporte à SETRAM son savoir-faire.

En 2013, les travaux de trois autres tramways ont été lancés, celui de Mostaganem et Sidi Bel Abbes dans l'ouest du pays et celui d'Ouargla au sud. D'autres grandes villes de l'Algérie telles que Sétif, Tlemcen, Annaba etc. sont également concernées par le projet de développement des transports en Algérie initié par le Ministère des Transport



Figure 2 : Réseau tramways Algérie.

1.4.Mission de l'entreprise

SETRAM est en charge de la mise en service, l'exploitation commerciale et la maintenance des Tramways Algériens. Elle s'est fixée comme objectifs :

1. D'offrir un service de transport de haute qualité en intégrant la sécurité, le confort, la régularité et la propreté aux voyageurs.
2. D'accompagner les algériens pendant la période d'adaptation à l'utilisation du nouveau mode de transport qu'est le tramway et l'ancrer dans leurs habitudes de déplacements.
3. D'assurer le transfert de savoir-faire des experts du groupe RATP vers l'ensemble des salariés de la SETRAM.
4. De se positionner comme référence en Afrique et dans le monde.
5. De satisfaire leurs client EMA et voyageurs.
6. De répondre à toutes les exigences contractuelles et atteindre les objectifs fixés.
7. D'adapter une culture qualité et protection de l'environnement.
8. De s'engager dans le développement durable.

1.5. Activité

Elle exploite actuellement :

1. Les Tramways d'Alger depuis 10-2012 avec la couleur bleue.



Figure 03 : Tramway d'Algier.

2. Les Tramways d'Oran depuis 06-2013 avec la couleur rouge.



Figure 04 : Tramway d'Oran.

3. Les Tramways de Constantine depuis 07-2013 avec la couleur verte.



Figure 05: Tramway de Constantine.

4. Les Tramways de Sidi bel Abbes depuis 07-2017 avec la couleur jaune.



Figure 06 : Tramway de SidiBelAbbas.

5. Les Tramways de Sétif depuis 05-2018 avec la couleur rouge.



Figure 07: Tramway de Sétif.

6. Les Tramways d’Ouargla depuis 03-2018 avec la couleur grise.



Figure 08 : Tramway d’Ouargla.

- 7.** Projet court terme :
 - 8.** Les tramways de Mostaganem
 - 9.** Projet long terme :
 - 10.** Les tramways d'Annaba et de Tlemcen ...

1.6. Unité opérationnelle d'Alger

1. Unité Opérationnelle d'Alger, Rue du cap Domaine Mimouni Hamoud, Bordj El Kiffan - Alger - ALGERIE.
 2. Ligne de tramway d'Alger : Ruisseau- Dergana Centre : qui s'étend sur 23,2 km avec trente-sept stations et cinq agences commerciales.

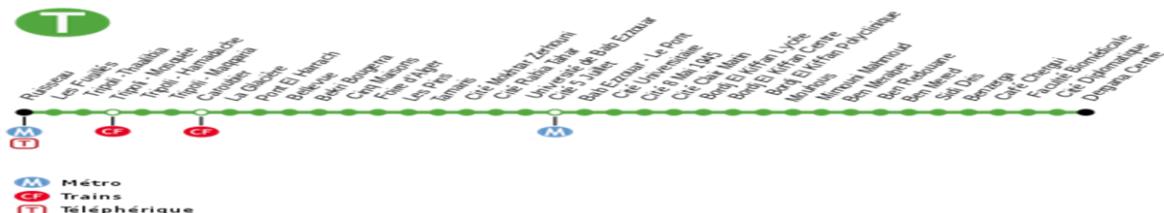


Figure 09 : Ligne de tramway d'Alger.

1.7.Pôles existants au niveau de la SETRAM

1. Management de direction
 2. Qualité, sécurité, environnement
 3. Exploitation.
 4. Client.
 5. Comptabilité & trésorerie
 6. Contrôle de gestion.
 7. Ressources humaines.
 8. Juridique.
 9. Sûreté interne.
 10. Système d'information.
 11. Maintenance.

1.7.1. Pôles maintenances

La SETRAM est en charge de la maintenance des installations fixe :

- 1.** Billettique.
- 2.** Voie.
- 3.** Bâtiments et ouvrages.
- 4.** CFA.
- 5.** CFO.
- 6.** LAC.
- 7.** Etc...

Le Pôle Maintenance est chargé de l'établissement et la mise en œuvre de la politique de maintenance et d'apporter des propositions d'ordre organisationnel et/ou technique visant à améliorer la qualité de service et/ou à réduire les coûts de maintenance dans le respect de la politique qualité de l'entreprise et de la réglementation en vigueur.

1.8. Objectif

- 1.** Maintenir le système de transport du niveau 01 jusqu'au niveau 04 tel que défini par la norme NF X60-000.
- 2.** Garantir dans le temps, les niveaux nominaux de sécurité, fiabilité et disponibilité
Maintenance de la voie et plateforme du tramway.
- 3.** SETRAM est chargée de la maintenance préventive et corrective des équipements liés au sous-système « Voie » qui comprend :
 - La plate-forme et ses abords (revêtement, regards, drainage,...)
 - La voie et appareils de voie.
- 4.** Garantir, dans le respect du référentiel de maintenance, la réalisation et le suivi de la production de maintenance des équipements et installation VOIE.
- 5.** Garantir le respect de la réglementation, des règles de hygiène et de sécurité, des procédures qualité et développement durable.

1.9. Problématique

Suite à notre visite sur terrain, nous avons constaté que le service maintenance fait des inspections sur la voie et le rail d'une façon manuelle, présenté comme suit :

1.9.1. Inspection des appareils de voies:

Tous les deux mois pour les appareils de voie de la ligne et tous les trois mois pour ceux du dépôt, une inspection des composants de base des APV sera effectuée par deux agents chargés de la maintenance. Cette inspection permettra de s'assurer que les appareils de voie se trouvent dans un état de fonctionnement approprié.

1.9.2. Inspection des rails:

En complément des inspections de la voie, une inspection approfondie des rails sera réalisée tous les six mois. Pour des raisons de commodité, cette inspection pourra être segmentée en plusieurs tronçons dont la longueur sera définie par le mainteneur. Cette opération sera réalisée par deux agents chargés de la maintenance de la voie.

Cette inspection se compose des opérations suivantes :

1. Un contrôle visuel de l'aspect du rail (usure ondulatoire, défaut ponctuel, dépôt de calamine, etc.)
2. Un contrôle visuel des joints de rail
3. Un contrôle approfondi de l'usure des rails dans les courbes : cette opération peut être réalisée, soit visuellement, soit par un gabarit de profil de rail ou soit par un instrument de mesure de profil de rail (le choix de la méthode utilisée sera fait par le mainteneur).

La réalisation de cette opération consomme du temps, d'énergie, des ressources humaines et financières avec un coefficient d'erreur non négligeable.

L'automatisation de ce processus récurrent représente une piste d'amélioration importante pour la SETRAM et ces employés, elle leur permet d'obtenir des données instantanées avec exactitude afin de maîtriser le risque en question ainsi d'économiser dans plusieurs volets économique, énergétique et humains. De ce fait, nous avons conçu un système automatisé qui permet d'identifier le risque tel que défini en temps réel et avec précision.

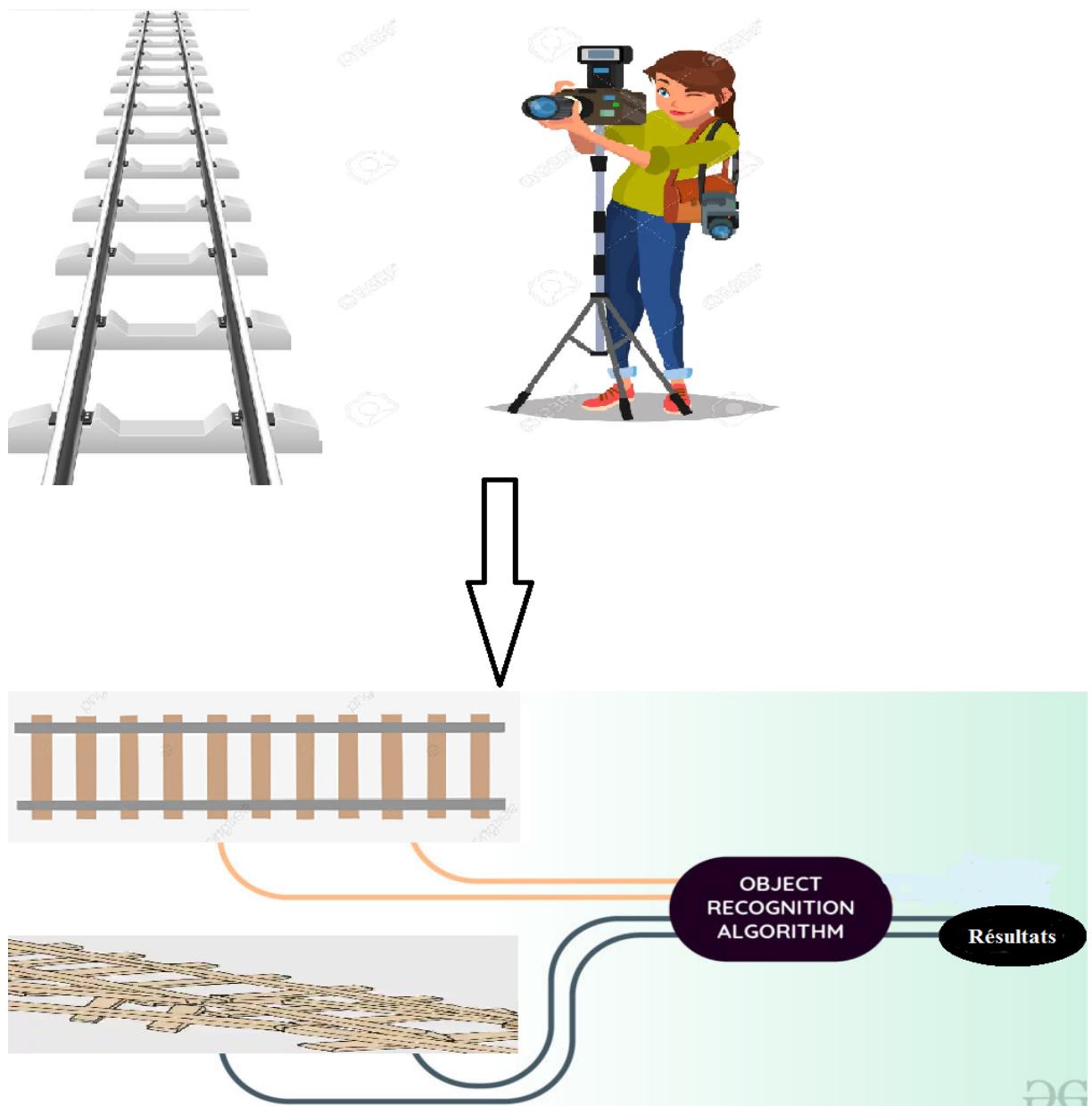


Figure 10: Schéma général du système proposé.

1.10. Conclusion

Après avoir présenté l'environnement de stage, et exposé une problématique, le chapitre suivant sera consacré à l'état de l'art, c'est-à-dire définir le contexte technologique de notre étude.

Chapitre 2

2. Etat de l'art

2.1. Introduction

Avec l'arrivée de l'internet, le monde est devenu hyper connecté dans lequel chaque objet manipulé (les véhicules, les animaux, les réseaux sociaux...) génèrent des millions de données supplémentaires chaque jour, qui s'ajoutent à un océan de données sans fin.

La question qui se pose est comment peut-on transformer un tel océan de données sans fond en flux régulier d'informations pertinentes pour répondre à ces attentes ? La réponse est l'intelligence artificielle (IA).

2.2. L'apprentissage profond

Avant d'expliquer les notions de l'apprentissage profond, il important de définir l'intelligence artificielle, l'apprentissage automatique et la relation entre ces trois domaines :

2.2.1. Relation entre l'intelligence artificielle, Machine Learning et le Deep Learning

L'apprentissage automatique et l'apprentissage profond font partie de l'intelligence artificielle. Ces approches ont toutes deux pour résultat de donner aux ordinateurs la capacité de prendre des décisions intelligentes. Cependant, l'apprentissage profond est une sous-catégorie de l'apprentissage automatique, car il s'appuie sur un apprentissage sans surveillance. Dans les deux cas, l'intelligence se limite à des utilisations spécifiques. On parle d'intelligence artificielle faible, par opposition à une intelligence artificielle forte qui serait à même, dans de nombreux domaines et circonstances, de prendre des décisions intelligentes semblables à celles de l'humain. Les deux technologies ont impérativement besoin de disposer de grandes quantités de données, qui leur servent de base d'apprentissage. (IONOS, 2020).

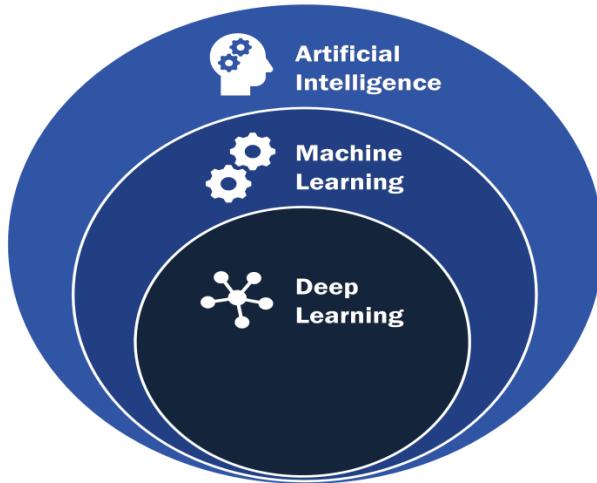


Figure 11 : Schéma de la relation entre le IA, ML et DL.

2.2.2. L'intelligence artificielle

L'intelligence artificielle est née dans les années 50, quand une poignée de pionniers du domaine naissant de l'informatique, ont commencé à se demander si les ordinateurs pouvaient être amenés à « penser », une question dont nous explorons encore aujourd'hui.

Une définition concise du champ serait la suivante : l'effort d'automatiser les tâches intellectuelles normalement effectuées par les humains. En tant que tel, l'IA est un domaine général qui englobe l'apprentissage automatique et l'apprentissage en profondeur, mais qui comprend également beaucoup plus d'approches qui n'impliquent aucun apprentissage.

Les programmes d'échecs initiaux, par exemple, ne concernaient que des règles codées en dur élaborées par des programmeurs et ne se qualifiaient pas comme apprentissage automatique. Pendant un temps assez long, de nombreux experts ont estimé que l'intelligence artificielle au niveau humain, pouvait être obtenue en faisant en sorte que les programmeurs fabriquent à la main un ensemble suffisamment large de règles explicites pour manipuler les connaissances.

Cette approche est connue sous le nom d'IA symbolique et elle était le paradigme dominant de l'IA des années 50 et à la fin des années 80. Elle a atteint son pic de popularité durant le boom des systèmes experts des années 80. Bien que l'IA symbolique se soit révélée appropriée pour résoudre des problèmes logiques bien définis, comme jouer aux échecs, il était difficile de trouver des règles explicites pour résoudre des problèmes flous plus complexes, tels que la classification des images, la reconnaissance de la parole et la

traduction. Une nouvelle approche est apparue pour prendre la place de l'IA symbolique : l'apprentissage automatique. [1]

2.2.3. Machine Learning

L'apprentissage automatique (Machine Learning) est un domaine de recherche en informatique qui traite des méthodes d'identification et de mise en œuvre de systèmes et algorithmes par lesquels un ordinateur peut apprendre, ce domaine a souvent été associé à l'intelligence artificielle et plus spécifiquement l'intelligence computationnelle.

L'intelligence computationnelle est une méthode d'analyse de données qui pointe vers la création automatique de modèles analytiques. Autrement dit, permettant à un ordinateur d'élaborer des concepts, d'évaluer, prendre des décisions et prévoir les options futures. [2]

L'ensemble du processus d'apprentissage nécessite un ensemble de données comme suit:

- 1. Ensemble de données pour l'entraînement :** c'est la base de connaissance utilisée pour entraîner, notre algorithme d'apprentissage, pendant cette phase, les paramètres du modèle peuvent être réglés (ajustés) en fonction des performances obtenues.
- 2. Ensemble de données pour le test :** cela est utilisé juste pour évaluer les performances du modèle sur les données non-vues.

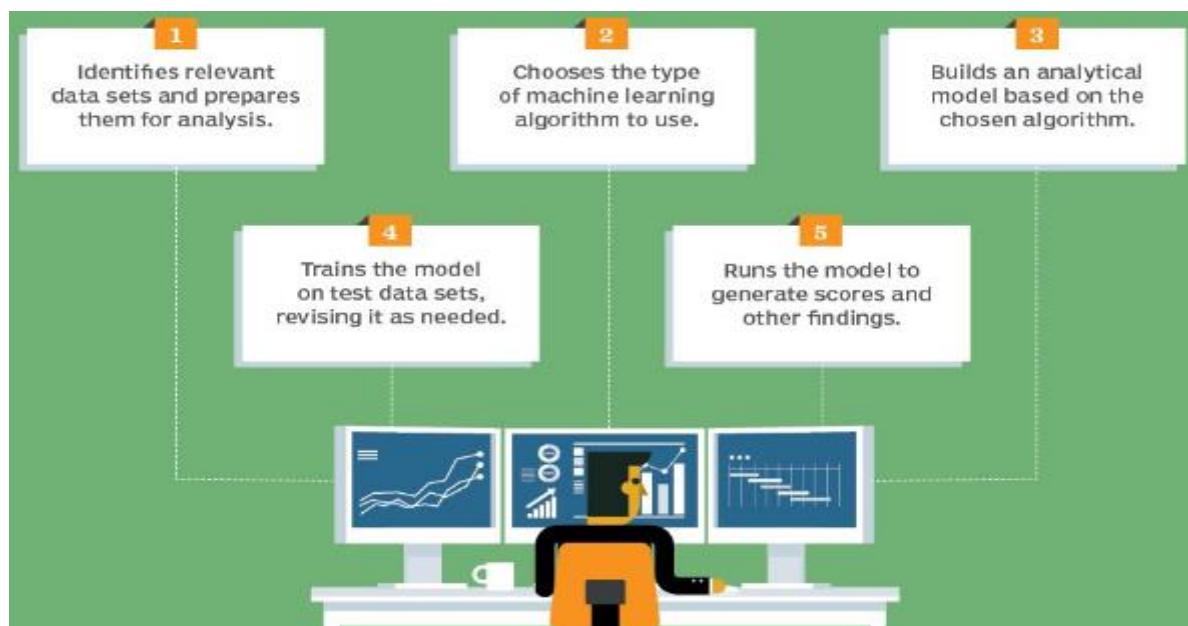


Figure 12 : Processus du Machine Learning.

2.2.4. Deep Learning

L'apprentissage en profondeur (Deep Learning) est un domaine de recherche sur l'apprentissage automatique basé sur un type particulier de mécanisme d'apprentissage. Il est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d'abstraction.

Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral. [2]

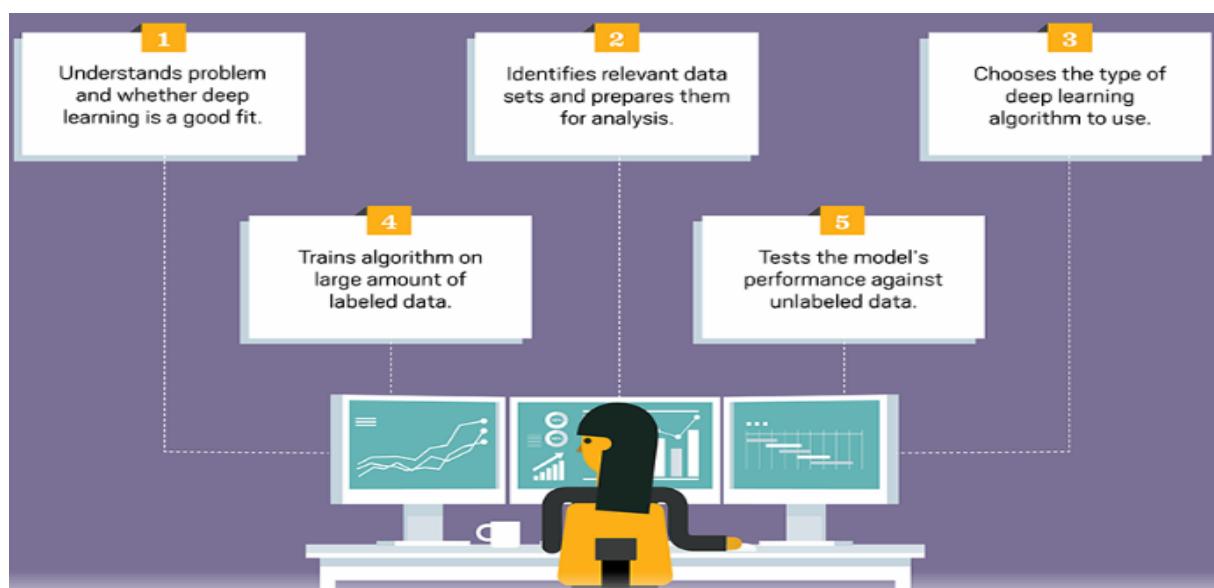


Figure 13 : Processus du Deep Learning.

Le Deep Learning se base sur un réseau de neurones artificiels imitant le cerveau humain. Cette structure est disposée en plusieurs couches, interconnectées entre elles.

La première couche correspond aux neurones d'entrée et la dernière transmet les résultats de sortie. Entre les deux se trouvent plusieurs couches intermédiaires par lesquelles l'information est traitée. Cette architecture est propre au Deep Learning et permet que chaque couche analyse de manière plus précise les données d'entrée.

Ainsi, plus le réseau de neurones artificiels est profond et donc contient plusieurs couches, plus le système peut effectuer des tâches complexes. Il est capable de déterminer par lui-même une représentation de ce qu'il reçoit, que ce soit une image ou un texte.

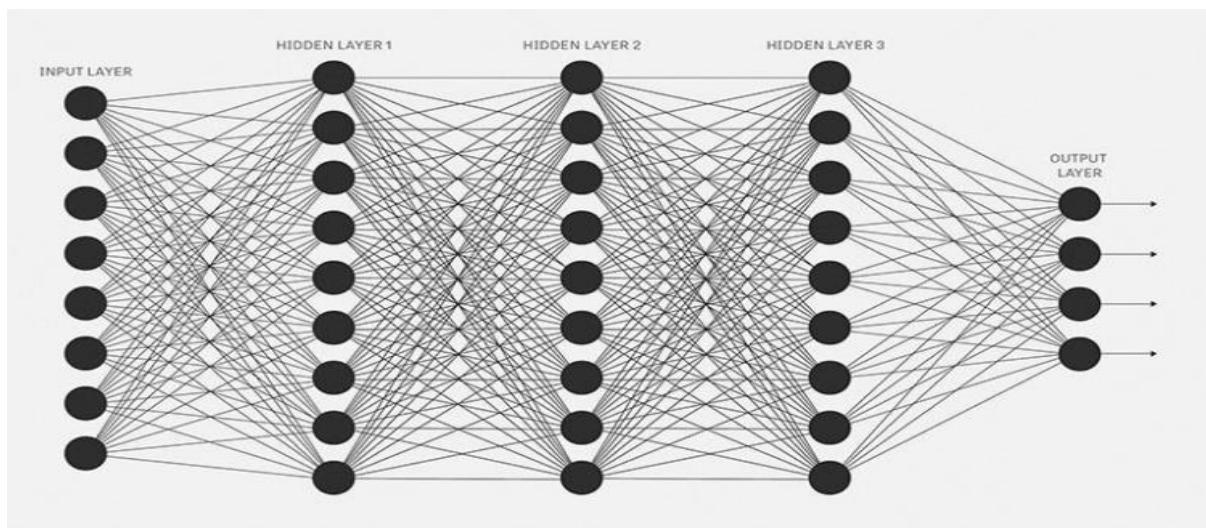


Figure 14 : Exemple d'un réseau de neurone.

2.3.CNN : Réseau de neurone à convolution

Revenons aux concepts de base. Les réseaux de neurones sont des algorithmes d'intelligence artificielle inspirés du fonctionnement des réseaux de neurones du cerveau humain. Parmi ces algorithmes, les réseaux de neurones à convolution(CNN) permettent l'analyse des images. Ils constituent à ce jour l'une des tentatives les plus abouties de doter une IA de capacités s'approchant de la vision humaine.

Ils sont constitués de plusieurs couches, elles-mêmes constituées de plusieurs neurones. Chaque couche reçoit les informations de la couche précédente, traite ces informations et les renvoie à la couche suivante. Chaque neurone d'une couche est donc relié aux neurones de la couche suivante. Ces liens sont pondérés par des poids et, à l'image du signal électrique émis par les neurones biologiques, vont influencer la propagation de l'information à travers le réseau de neurones à convolution. [3]

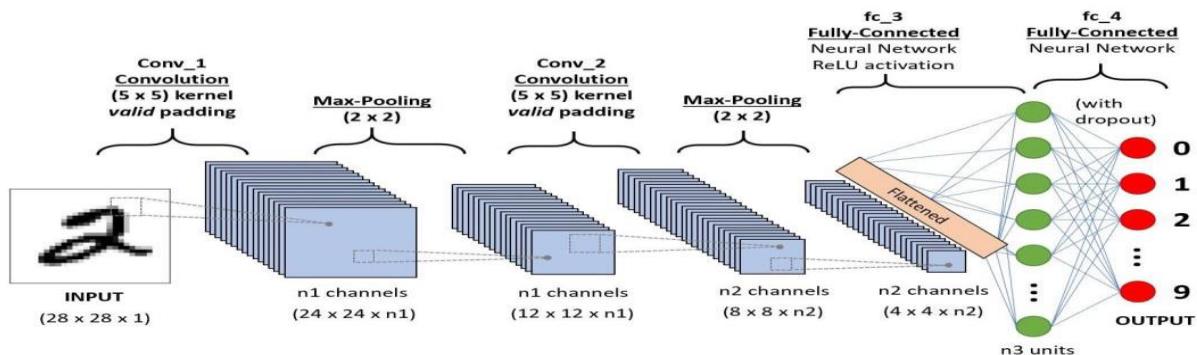


Figure 15 : Architecture du CNN.

Ces modèles neuronaux sont composés de différents types de couches neuronales, correspondant à plusieurs étapes de l'analyse :

2.3.1. Couche de convolution

Ces modèles traitent les images “fragments par fragments”, ils comparent ces “fragments” à des caractéristiques spécifiques. Ces images sont ainsi “filtrées” par une méthode mathématique appelée convolution : des filtres *Feature Detector*, permettant la détection de caractéristiques spécifiques, sont appliqués sur l'image et aboutissent à la création de plusieurs *Feature Map* ou "cartes des caractéristiques" correspondant à ces différents filtres. L'image de départ est donc transformée en plusieurs matrices selon les différentes caractéristiques avec lesquelles on l'a filtrée.^[3]

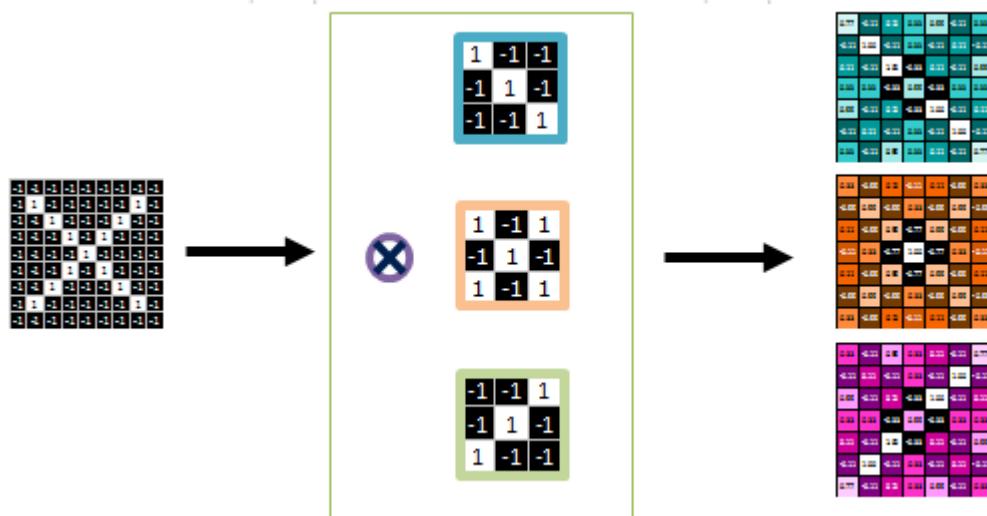


Figure 16 : Application du filtre sur une image.

2.3.2. Couche de mise en commun (Pooling)

Le *pooling* est une étape qui permet de réduire le nombre de paramètres descriptifs de l'image et d'éviter le sur-apprentissage. C'est l'étape qui consiste à diminuer la taille d'une image tout en conservant le maximum d'information en passant une petite “fenêtre”, généralement de 2X2 pixels sur les *Feature Map* obtenues par convolution. Ici le pixel avec la valeur maximale est conservé et donc les caractéristiques les plus importantes. ^[3]

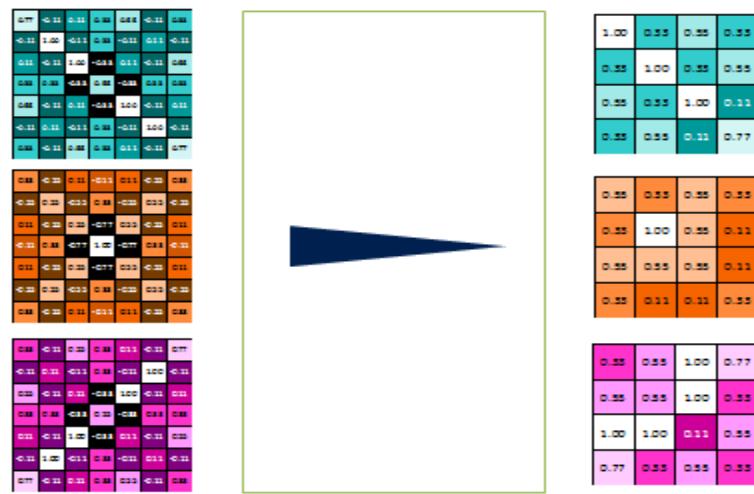


Figure 17 : Explication du Pooling.

2.3.3. Couche dense (full connexion)

Après la convolution et le pooling, la carte de caractéristiques finale utilisée pour la tâche de « classification » est exécutée par la couche entièrement connectée. Ces couches entièrement connectées ne peuvent accepter que des données unidimensionnelles. Par conséquent, les données 3D doivent être converties en 1D. Pour cela, les fonctions d'aplatissement Python standard peuvent être utilisées. Les neurones de la couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente, et ils fonctionnent exactement comme le font les réseaux de neurones normaux. [4]

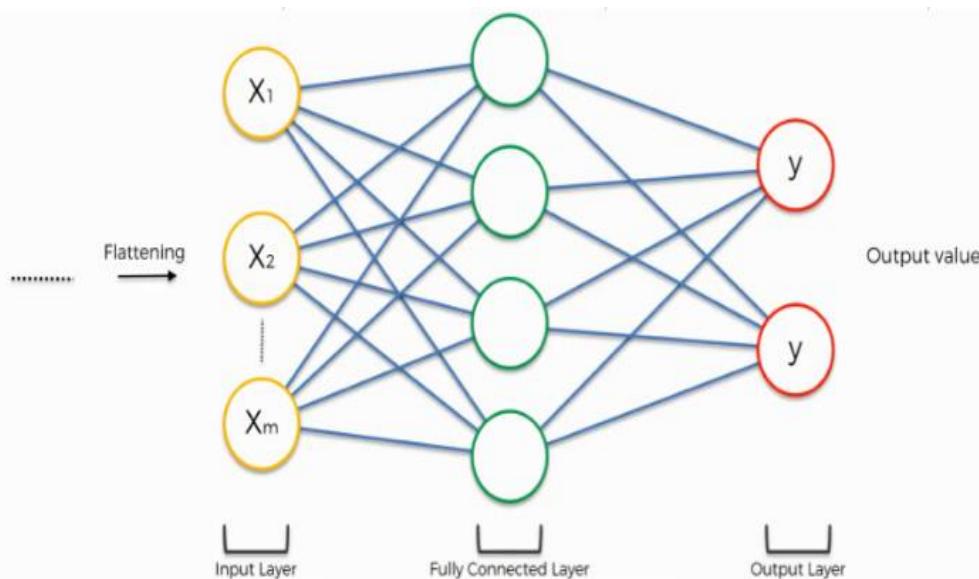


Figure 18 : Représentation de la couche full connexion.

2.4. Mask R-CNN

Le modèle Mask R-CNN, permet de détecter des objets et de les classifier. Sa particularité est d'ajouter à cette tâche de détection la segmentation d'instance, c'est à dire que chaque pixel de l'image sera classé. Ainsi, ce double “compétence” représente un avantage par rapport à des modèles de détection, elle vient affiner le résultat proposé. De plus, contrairement à la segmentation sémantique, qui permet d'associer à chaque pixel un label, la segmentation d'instance associe un masque et un label à chaque objet, même si ces objets appartiennent à la même classe.

Mask R-CNN est une extension du modèle Faster R-CNN. Aux deux types de sorties générées par ce dernier, qui sont la classe de l'objet présent sur l'image et la boîte englobante associée, s'ajoute une troisième branche dont la sortie est le masque de l'objet.

Cet algorithme créé par Facebook est une combinaison de modèles déjà existants : *le réseau convolutif ResNet101* ; un réseau de proposition de régions (*regions proposal network, RPN*) ; un classificateur binaire de masque. [5]

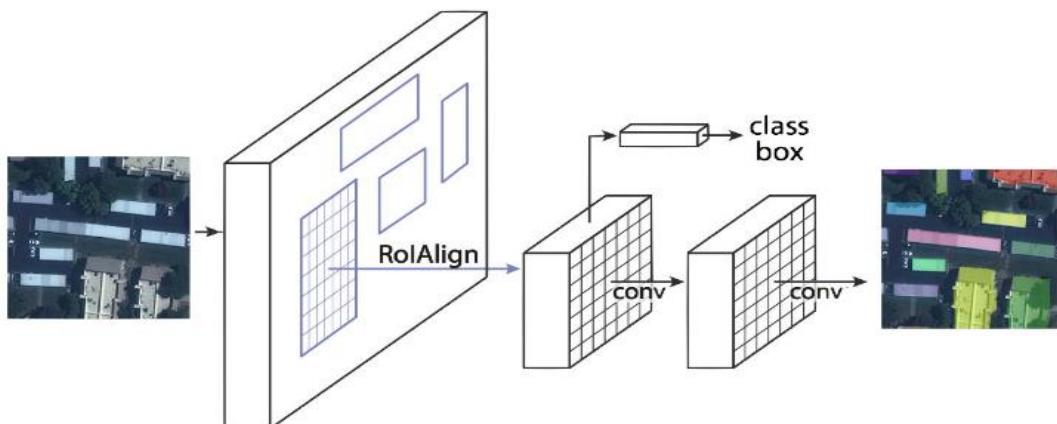


Figure 19 : Application du Mask R-CNN.

2.5. VGG 16^[6]

VGG est un réseau neuronal convolutif proposé par K. Simonyan et A. Zisserman de l'Université d'Oxford et a acquis une notoriété pour avoir remporté le concours ILSVRC (ImageNet Large Scale Visual Recognition Challenge) en 2014. Le modèle a atteint une précision de 92,7 % sur Imagenet, ce qui est l'un des meilleurs résultats obtenus. Par rapport aux modèles précédents, il marque une amélioration en proposant un noyau de plus petite taille (3×3) dans les couches convolutives qu'auparavant. Le modèle a été formé pendant des semaines à l'aide de cartes graphiques à la pointe de la technologie.

2.5.1. Architecture

Dans les faits, il existe deux algorithmes disponibles : **VGG16** et **VGG19**. Dans ce travail, nous allons concentrer sur l'architecture du premier.

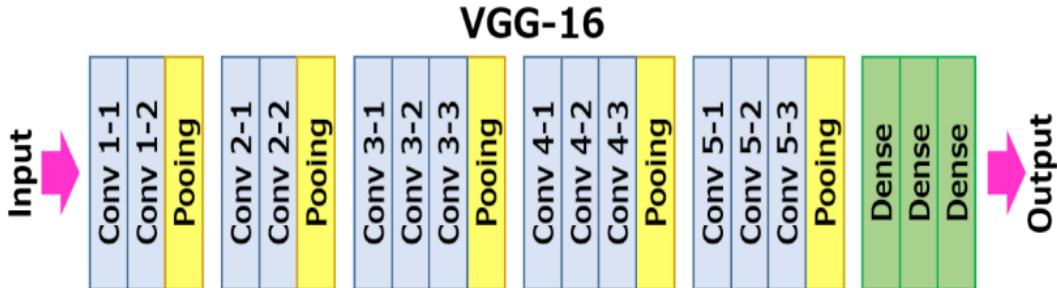


Figure 20 : Architecture Algorithme VGG16.

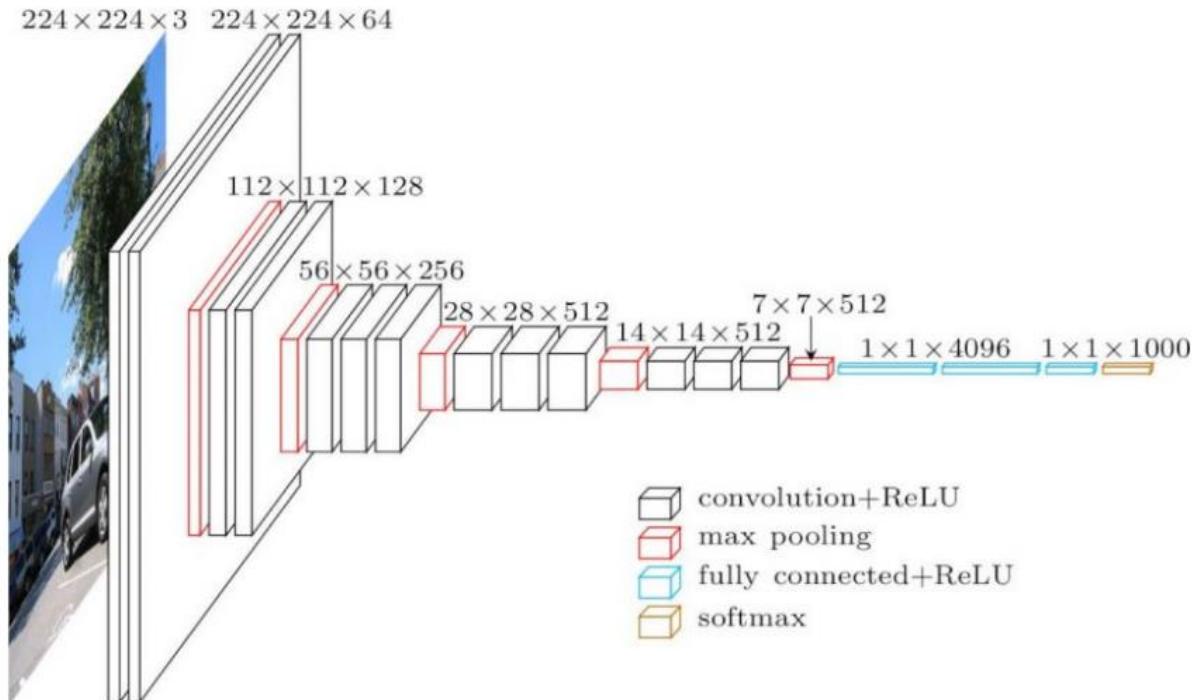


Figure 21 : Structure Algorithme VGG16.

L'entrée de la couche de convolution est une image RVB de taille fixe 224×224 .

L'image est passée à travers un empilement de couches convolutives, où les filtres ont été utilisés avec un très petit champ récepteur : 3×3 (qui est la plus petite taille pour capturer la notion de gauche/droite, haut/bas, centre). Dans l'une des configurations, il utilise également des filtres de convolution 1×1 , qui peuvent être considérés comme une transformation linéaire des canaux d'entrée (suivie d'une non-linéarité). La foulée de convolution est fixée à 1 pixel ; le rembourrage spatial de l'entrée de la couche convective d'entrée est tel que la résolution

spatiale est conservée après la convolution, c'est-à-dire que le remplissage est de 1 pixel pour 3×3 couches convolutives. La mise en commun spatiale est effectuée par cinq couches de mise en commun maximale, qui suivent certaines des couches convolutives (toutes les couches convolutives ne sont pas suivies par un regroupement maximal). La mise en commun maximale est effectuée sur une fenêtre de 2×2 pixels, avec une foulée de 2.

Trois couches entièrement connectées (FC) suivent une pile de couches convolutives (qui ont une profondeur différente dans différentes architectures) : les deux premières ont 4096 canaux chacune, la troisième effectue une classification ILSVRC à 1000 voies et contient donc 1000 canaux (un pour chaque classe). La couche finale est la couche soft max. La configuration des couches entièrement connectées est la même dans tous les réseaux.

Toutes les couches cachées sont équipées du non linéarité de rectification (ReLU). Il est également à noter qu'aucun des réseaux (sauf un) ne contient la normalisation de la réponse locale (LRN), une telle normalisation n'améliore pas les performances sur l'ensemble de données ILSVRC, mais entraîne une augmentation de la consommation de mémoire et du temps de calcul.

2.6.YOLO [7]

YOLO (You Only Look Once) est un algorithme qui utilise des réseaux de neurones pour fournir une détection d'objets en temps réel. Cet algorithme est populaire en raison de sa vitesse et de sa précision. Il a été utilisé dans diverses applications pour détecter les feux de circulation, les personnes, les animaux, etc...

L'algorithme YOLO utilise des réseaux de neurones convolutifs (CNN) pour détecter des objets en temps réel. Comme son nom l'indique, l'algorithme ne nécessite qu'une seule propagation vers l'avant à travers un réseau de neurones pour détecter les objets. Cela signifie que la prédiction dans l'ensemble de l'image est effectuée en une seule exécution d'algorithmes. Le CNN est utilisé pour prédire simultanément diverses probabilités de classe et cadres de délimitation.

L'algorithme YOLO fonctionne à l'aide des trois techniques suivantes :

Blocs résiduels.

Régression de la boîte englobante.

Intersection sur Union (IOU).

2.6.1. Blocs résiduels

Tout d'abord, l'image est divisée en différentes grilles. Chaque grille a une dimension de $S \times S$. L'image suivante montre comment une image d'entrée est divisée en grilles.



Figure 22 : Illustration des blocs résiduels.

Dans l'image ci-dessus, il existe de nombreuses cellules de grille de dimension égale. Chaque cellule de la grille détectera les objets qui y apparaissent. Par exemple, si un centre d'objet apparaît dans une certaine cellule de grille, alors cette cellule sera chargée de le détecter.

2.6.2. Régression de la boîte englobante

Un cadre de délimitation est un contour qui met en évidence un objet dans une image. Chaque cadre de délimitation de l'image se compose des attributs suivants :

- Largeur (pc)
- Hauteur (bh)
- Classe (par exemple, personne, voiture, feu de circulation, etc.) - Ceci est représenté par la lettre c.
- Centre de la boîte englobante (bx,by)

L'image suivante montre un exemple de cadre de délimitation. La boîte englobante a été représentée par un contour jaune.

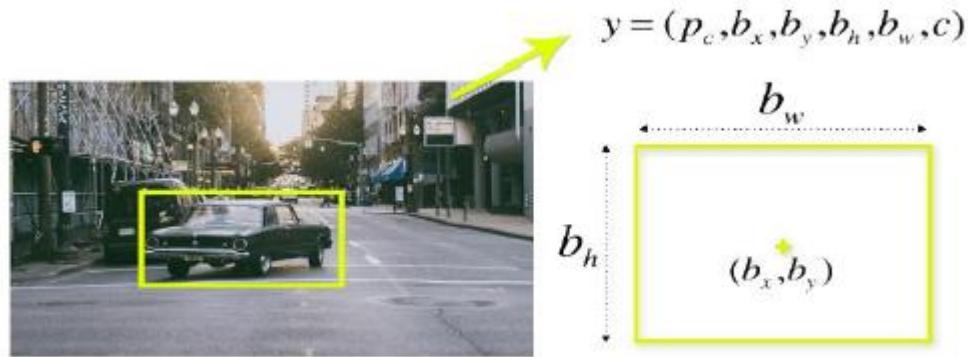


Figure 23 : Illustration d'une boîte englobante.

YOLO utilise une régression de cadre de délimitation unique pour prédire la hauteur, la largeur, le centre et la classe des objets. Dans l'image ci-dessus, représente la probabilité qu'un objet apparaisse dans la boîte englobante.

2.6.3. Intersection sur union (IOU)

L'intersection sur l'union (IOU) est un phénomène de détection d'objets qui décrit comment les boîtes se chevauchent. YOLO utilise IOU pour fournir une boîte de sortie qui entoure parfaitement les objets.

2.6.4. Combinaison des trois techniques

L'image suivante montre comment les trois techniques sont appliquées pour produire les résultats de détection finaux.

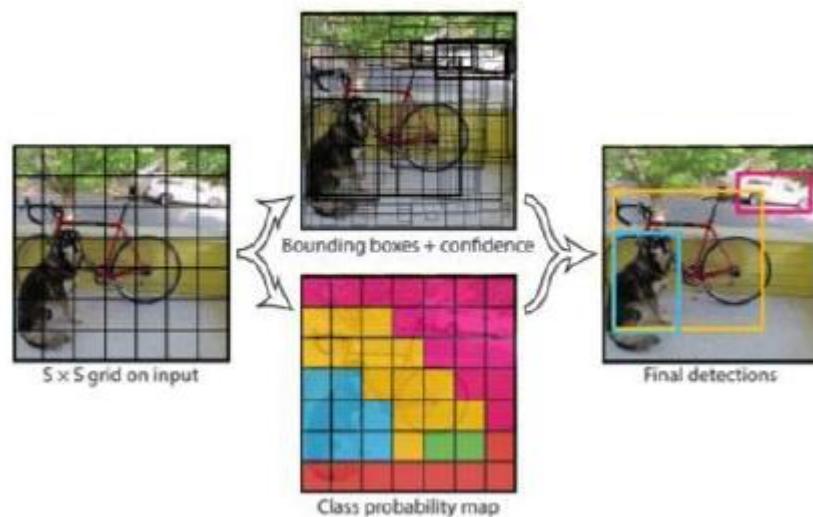


Figure 24 : Illustration d'une détection YOLO.

Tout d'abord, l'image est divisée en cellules de grille. Chaque cellule de la grille prévoit B cadres de délimitation et fournit leurs scores de confiance. Les cellules prédisent les probabilités de classe pour établir la classe de chaque objet.

Par exemple, on peut remarquer au moins trois classes d'objets : une voiture, un chien et un vélo. Toutes les prédictions sont faites simultanément à l'aide d'un seul réseau de neurones convolutif.

L'intersection sur l'union garantit que les boîtes englobantes prédites sont égales aux boîtes réelles des objets. Ce phénomène élimine les cadres de délimitation inutiles qui ne correspondent pas aux caractéristiques des objets (comme la hauteur et la largeur). La détection finale consistera en des cadres de délimitation uniques qui s'adapteront parfaitement aux objets.

Par exemple, la voiture est entourée par la boîte englobante rose tandis que le vélo est entouré par la boîte englobante jaune. Le chien a été mis en évidence à l'aide de la boîte englobante bleue.

2.7.Description de l'architecture:

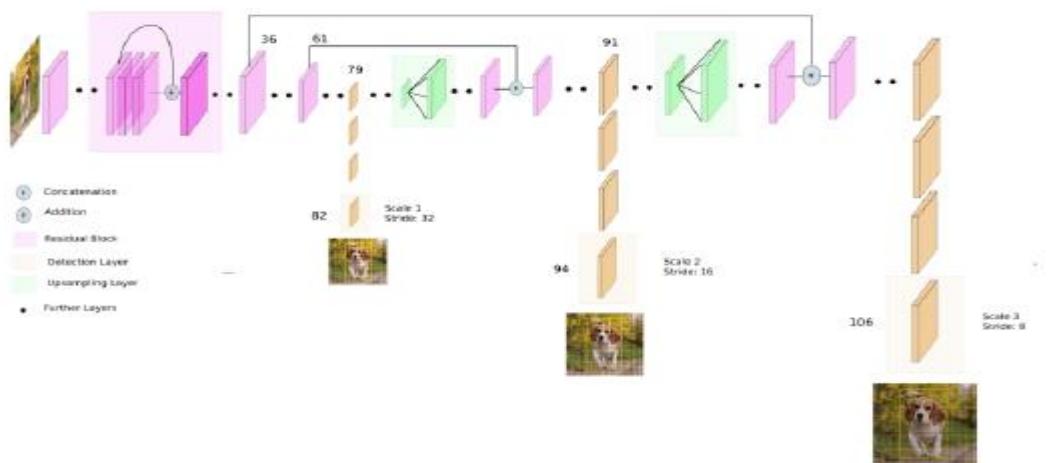


Figure 25 : Architecture de YOLO.

Les entrées sont un lot d'images de forme $(m, 416, 416, 3)$. YOLO transmet cette image à un réseau de neurones convolutifs (CNN). Les deux dernières dimensions de la sortie ci-dessus sont aplatis pour obtenir un volume de sortie de $(19, 19, 425)$:

Ici, chaque cellule d'une grille 19×19 renvoie 425 numéros.

$425 = 5 * 85$, où 5 est le nombre de boîtes d'ancre par grille.

$85 = 5 + 80$, où 5 est (pc, bx, by, bh, bw) et 80 est le nombre de classes que nous voulons détecter.

La sortie est une liste de cadres de délimitation avec les classes reconnues. Chaque cadre de délimitation est représenté par 6 nombres (pc, bx, by, bh, bw, c). Si nous développons c en un vecteur à 80 dimensions, chaque cadre englobant est représenté par 85 nombres.

Enfin, nous effectuons l'IOU (Intersection over Union) et la suppression non maximale pour éviter de sélectionner des cases qui se chevauchent.

2.8. Etude de l'existant

Le trafic ferroviaire dédié au transport des voyageurs et des marchandises se doit au jour d'aujourd'hui d'être très performant en termes de sécurité et de stabilité. Pour cela, les différentes entreprises de transport ferroviaire dans le monde ont mis en place plusieurs systèmes en essayant de résoudre la même problématique, dont nous retrouvons les suivants:

2.8.1. Vossloh [8]

Vossloh compte parmi les leaders mondiaux dans le domaine de la technique ferroviaire. L'infrastructure ferroviaire constitue le cœur de métier du Groupe. Ils offrent à leurs clients du monde entier des solutions intégrées livrées par un fournisseur unique. La maintenance efficace et économique d'un réseau passe par un planning reposant sur une base solide : l'analyse précise de l'état actuel. Cette analyse, constituée de différents contrôles et de diverses mesures, consiste à documenter les défauts de rail en prenant en compte leur gravité et leur positionnement exact. En fonction des tolérances de la norme à appliquer, Vossloh définit ensuite un entretien des rails idéal. Vossloh propose à l'achat des appareils et des véhicules de contrôle et de mesure, incluant des logiciels et des formations. Mais ils proposent aussi des services de mesure, de contrôle, de mise en forme des résultats et de conseil. Parmi ces services nous trouvons :

2.8.1.1. Service d'inspection

Pour la saisie et la documentation de l'état des rails, Vossloh fait appel aux technologies de contrôle et de mesure les plus modernes et à des collaborateurs hautement spécialisés et certifiés qui possèdent une connaissance complète de toutes les phases du cycle de vie du rail.

Divers procédés de contrôle sont possibles pour détecter les irrégularités présentes sur les rails. Les résultats de contrôle sont analysés et préparés pour évaluation au moyen d'un logiciel spécial.

À partir des données collectées, les experts de Vossloh sélectionnent la procédure d'entretien la plus rationnelle en termes de coûts et déterminent ses paramètres.

Les méthodes de contrôle utilisées par Vossloh sont les suivantes :

1. Contrôle visuel.
2. Contrôle par courants de Foucault.
3. Contrôle par ultra-sons.
4. Contrôle par magnétoscopie.
5. Contrôle magnétique par flux de fuite.

2.8.1.2. Service de mesure

On ne peut obtenir une représentation fiable de l'état du rail que si les résultats de l'analyse des fissures sont complétés par des mesures portant sur la géométrie du rail et de la voie. C'est ce type de mesure qui permet de déterminer l'évolution de l'usure des rails, des aiguillages, des croisements ou encore des appareils de dilatation et de la comparer à des normes, à des règlementations ou à des valeurs nominales définies par les entreprises. Il s'agit des mesures suivantes :

Mesure du profil transversal du champignon.

Mesure de l'ondulation longitudinale du rail.

Mesure de la voie et de l'aiguillage.

Mesure de l'enlèvement de matière.

2.8.2. Speno^[9]

Au fil des passages de trains, les rails s'usent. Pour les entretenir, Speno conçoit des trains équipés de meules. Ils embarquent des systèmes de mesure du profil des rails. Le meulage est effectué en fonction des résultats obtenus. Exploitant jusqu'alors des capteurs de mesure par contact, la société suisse a modernisé ses systèmes de mesure en les dotant de technologies optiques. Elle en a profité pour faire évoluer la plate-forme PXI de traitement et de visualisation. L'application programmée sous Labview tourne désormais sur un seul contrôleur PXI doté d'un processeur multi-cœur...

Essentiel

Speno a remplacé les capteurs traditionnels de ses systèmes de mesures du profil des rails par des solutions de mesures sans contact.

Le nouveau système de mesure placé dans le bogie des trains de meulage a ainsi gagné en compacité.

Le dispositif de traitement et la visualisation des mesures embarqué dans la cabine du conducteur a également évolué.

L'application logicielle développée sous Labview a été simplifiée et le nombre de contrôleur PXI a été réduit.

2.8.2.1. Solution proposée par Speno

Speno embarque sur ses trains des systèmes de mesure qui relèvent en temps réel les profils longitudinaux et transversaux du rail. En replaçant les capteurs de déplacement traditionnels par des technologies optiques sans contact, le système de mesures a gagné en compacité et la conception mécanique de l'ensemble a été considérablement simplifiée. Le système de mesure est désormais directement supporté par le bogie.

2.8.3. BLET instruments et contrôles [10]

Parmi les instruments de mesure pour le ferroviaire, BLET Measurment Group propose la jauge de profil de rail, un appareil ou profilomètre qui permet de contrôler l'usure d'un rail de chemin de fer. Cette jauge de profil de rail est conçue pour la mesure de la coupe transversale de la face d'usure du rail.

Les principales fonctions de la jauge de profil de rail sont les suivantes :

1. obtenir les informations sur la coupe transversale de la face d'usure du rail.
2. Réaliser un profil complet et analyser l'usure du rail.
3. Visualiser sur l'écran de la jauge de profil de rail, la comparaison entre le rail actuel et un nouveau rail.
4. La procédure de mesure à suivre pour l'utilisation de la jauge de profil de rail est la suivante:

Fixer la jauge de mesure sur le rail.

Allumer la jauge laser en pressant le bouton ON/OFF sur la batterie.

Allumer l'écran en pressant sur le bouton. Une communication sans fil s'établit entre la jauge de profil de rail et l'écran.

Appuyer sur le bouton Mesure situé sur l'écran. La jauge de profil de rail va scanner la surface du rail. Une lumière LED rouge est allumée pendant la durée du scan, d'environ trois secondes. Une fois le scan terminé, l'écran affiche les valeurs des paramètres sélectionnés.

5. Pour visualiser le profil du rail, appuyer sur le bouton Profil. L'écran affiche le profil du rail ainsi que les paramètres mesurés.
6. Le logiciel fourni avec la jauge de profil de rail permet de tenir une base de données sur l'usure du rail dans un ordinateur.

2.9. Conclusion

A travers ce chapitre, il a été présenté deux grandes parties, tout d'abord, la description de l'aspect théorique de l'IA et le Deep Learning avec ses domaines d'applications, ensuite une présentation des modèles de segmentation. Et pour finir, une étude de l'existant a été établie à travers quelques articles expliquant la solution au problème de détection des défaillances des rails ferroviaires.

Le chapitre suivant sera consacré à la conception du système proposé, en présentant les algorithmes utilisés pour ce dernier.

Chapitre 3

3. Conception

3.1. Introduction

Ce chapitre est consacré à la présentation détaillé de la conception du modèle de la prédiction proposé. Nous le commençons par expliquer les types d'usures des rails ferroviaires étudiés dans ce travail, ensuite nous présentons l'architecture générale de notre système avec ses différentes phases. Puis nous passons à une explication détaillée des différentes parties du système.

3.2. Définition du rail [11]

Rail: Un rail est une barre en acier à laquelle on a donné une forme particulière. Le rail est utilisé pour la circulation des véhicules de chemin de fer. Deux files de rails forment une voie ferrée. Les rails gardent alors leur écartement constant grâce à des traverses.

Le rail assure le guidage, car une roue de chemin de fer possède un *boudin* collé à l'intérieur du rail. Le rail assure aussi la transmission de la signalisation, puisqu'il est conducteur de signaux électriques.

Le rail est le premier élément en contact entre le véhicule et la voie. Il se charge de transmettre et de répartir les forces du véhicule sur le système en dessous. Il y a différents types de rails avec différentes caractéristiques.

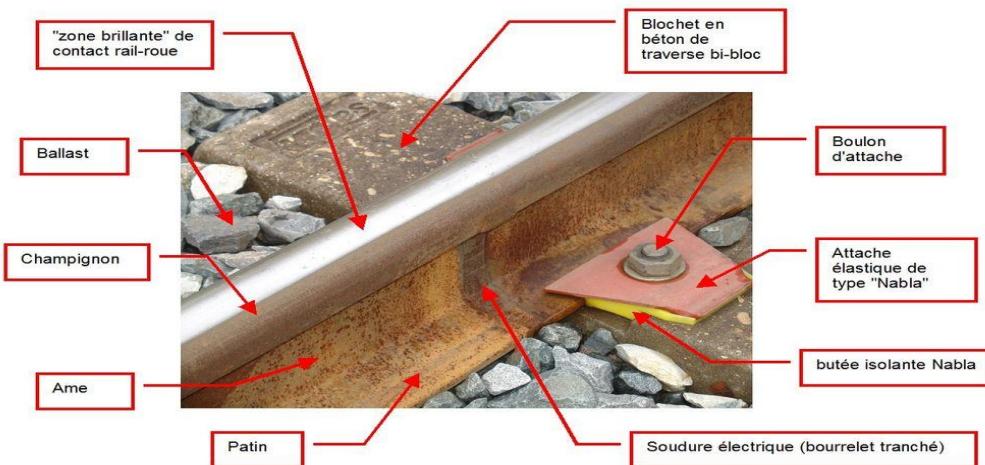


Figure 26 : Détails du couple rail-traverse.

3.2.1. Type de rails ^[12]

Il existe plusieurs types de rails:

Le rail Vignole est le plus couramment utilisé. Il est composé de deux parties, le patin qui repose sur la traverse et le champignon où prennent appui les roues des trains.

Le rail de type Broca à gorge est le rail utilisé pour les tramways. Il est constitué d'une gorge qui reçoit le boudin de la roue, car la voie est encastrée dans la chaussée routière.

Le rail à double champignon était utilisé avant la Seconde Guerre mondiale. Une fois que l'un des champignons était usé, les ouvriers le retournaient. Cela permettait des économies pour les compagnies. Mais le deuxième champignon était écrasé par le passage des trains et il fallait le changer. On trouve encore ce type de rail en France, notamment sur les voies peu fréquentées et peu entretenuées, et aussi sur les voies de service.

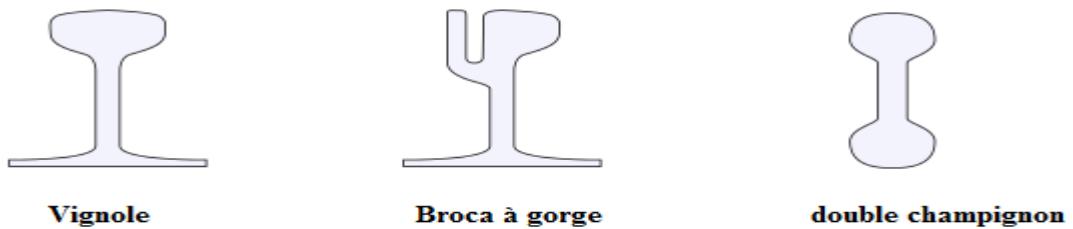


Figure 27 : Les types de rails.

3.3.Définition de l'usure

Usure : Détérioration par un usage prolongé, par le frottement, etc. », le petit ROBERT.

3.3.1.Les types d'usures

Les phénomènes naturels (la chaleur extrême par exemple) ainsi que l'impact qu'ont les trains sur les rails font que des anomalies apparaissent sur celles-ci, mettant en risque la sécurité des voyageurs. La déformation des rails peut survenir entre autres du fait de l'existence de traverses endommagées sur la voie ferrée. Ci-dessous nous discutons de trois types d'usures que nous allons traiter dans notre système:

- **Usure ondulatoire [13]**

L'usure ondulatoire (UO) des rails est un phénomène qui touche aussi bien les trains que les trams, les métros ou tous autres types de transports ferroviaires (Collette, 2007). Elle demeure l'un des phénomènes les plus mal connus pouvant détériorer les profils des roues et des rails dans le domaine ferroviaire (Mai, 2011). Cette usure se manifeste sous forme d'ondulations étendues sur les surfaces des profils dans la direction du roulement. Elle constitue ainsi des sources d'excitation de vibrations inconfortables (bruit) et des sources de fatigue. Bien que les travaux existants ne modélisent qu'imparfaitement le processus de formation de ce phénomène, la modélisation proposée par (Carson, Johnson et Gray, 1971) reste la plus communément admis. Elle suggère la combinaison de deux mécanismes : un mécanisme de fixation de longueur d'onde et un mécanisme d'endommagement.



Figure 28 : Usure ondulatoire.

- **Usure latérale du rail (latérales du flanc de champignon et latérale de la bavette)**

L'usure latérale du rail est l'un des principaux paramètres qui permet de déterminer l'état des rails, en toute fiabilité. Parmi les principales causes d'apparition de ces usures nous trouvons :

-Défaut imputable à l'utilisation.

- L'usure latérale (ou chanfreinage) des rails situés sur les files hautes des courbes résultante de sollicitations du matériel roulant.

- Dans une même courbe, l'usure latérale présente généralement un aspect sinusoïdal.

- L'évolution de l'usure latérale dépend fortement de la nuance du rail et de la qualité du graissage des rails.

- L'usure latérale constitue un défaut à partir du moment où son importance est:

- ✓ nuisible à l'entretien de la voie (sur écartement excessif, valeur limite d'intervention), susceptible de provoquer une rupture par affaiblissement du profil (atteinte du congé inférieur du champignon)



Figure 29 : Usure latérale de la bavette.



Figure 30 : Usure de flanc de champignon.

3.4. Architecture générale du système

L'objectif de notre travail est de concevoir un système intelligent capable de prédire le type de l'usure et sa localisation exacte dans chaque image en utilisant une base de données réelle i.e. Ces données ne sont pas issues de travaux de recherche reliés.

Après la collecte de données, nous avons les préparer en utilisant plusieurs méthodes pour :

- Déetecter les contours nécessaires.
- Faire l'étiquetage des données.
- Faire enrichir la BDD.

Ensuite, ces données sont passées à nos modèles pour l'apprentissage, Après l'apprentissage, on utilise chacun des deux modèles pour faire une prédiction et sortir par la suite par deux résultats, la comparaison des résultats nous permet de choisir le meilleur modèle adéquat avec nos données.

La Figure 32 montre l'architecture du modèle proposé.

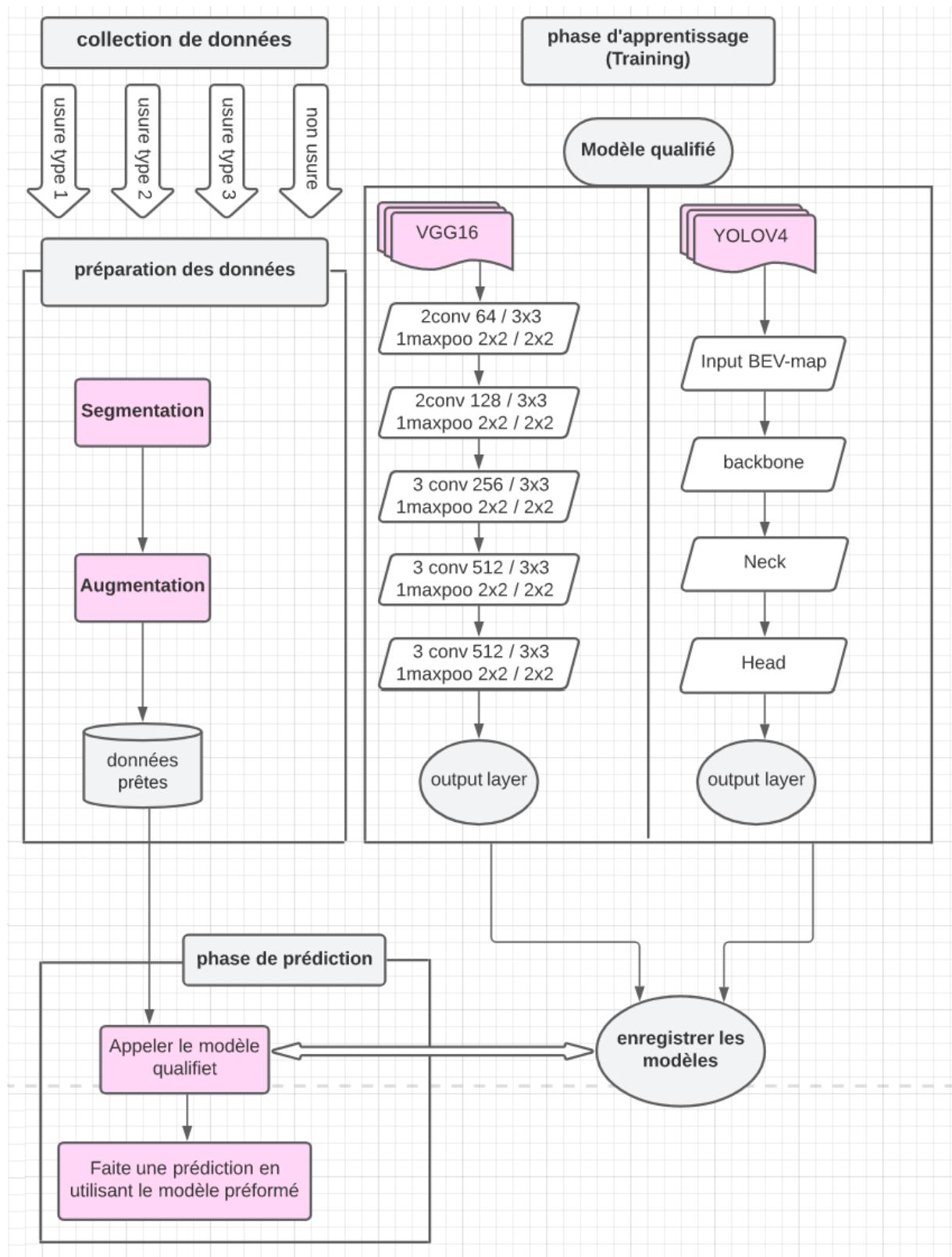


Figure 31 : Architecture du modèle proposé.

3.4.1. Collection de données

Etant donné l'absence d'une base de données spécialisée aux images d'usure de chemins de fer (rails du tramway) au niveau de l'organisme d'accueil, la collecte de données a été effectuée d'une manière manuelle en prenant en photo les rails avec un smartphone.

D'un côté nous avons fait une large étude sur les trois types d'usures que nous allons traiter, nous avons cherché ses causes et ses emplacements. Et de l'autre côté une autre étude était faite sur les rails ferroviaires installés sur Alger de DERGANA vers ROUISSEAU, et après avoir comparé nos résultats nous avons atteint les stations où chacun de ces types d'usures abonde. Ces stations sont : MIMOUNI HAMOUD, La cité Universitaire CUB1 BAB EZZOUAR et TAMARIS. Le processus de collecte de données s'est déroulé en une semaine, où les photos prises ont été collectées soigneusement.

Pour commencer, l'acquisition de données a débuté avec 350 images, 280 de l'entraînement et 70 de validation, mais après l'implémentation, la performance du modèle était très moyenne, ce qui nous a poussés à enrichir notre base de données avec 355 images supplémentaires.

L'attribution de la base de données s'est effectuée d'une manière standard c'est-à-dire, 80% pour l'entraînement et 20% pour la validation ce qui nous donne le tableau suivant :

Entrainement	Validation
564	141
705	

Tableau 1 : Tableau de la base de données.

3.4.2. Segmentation

La segmentation de ce système consiste à détecter les contours du rail ferroviaire à partir d'une photo prise par la caméra. Pour cela le modèle d'apprentissage profond utilisé est celui du Mask R-CNN pour sa performance dans la segmentation d'instance.

3.4.3. Annotation

L'annotation en machine Learning est un processus d'étiquetage des données sur les images. Les annotations peuvent être utilisées pour informer les machines de la présence de divers objets en l'occurrence dans ce modèle c'est de l'informer de la localisation de l'usure dans le rail.

3.4.3.1. Annotation pour Mask RCNN

Le logiciel d'annotation le plus adapté pour la segmentation est celui de LABELME, la figure 33 illustre un exemple d'annotation:

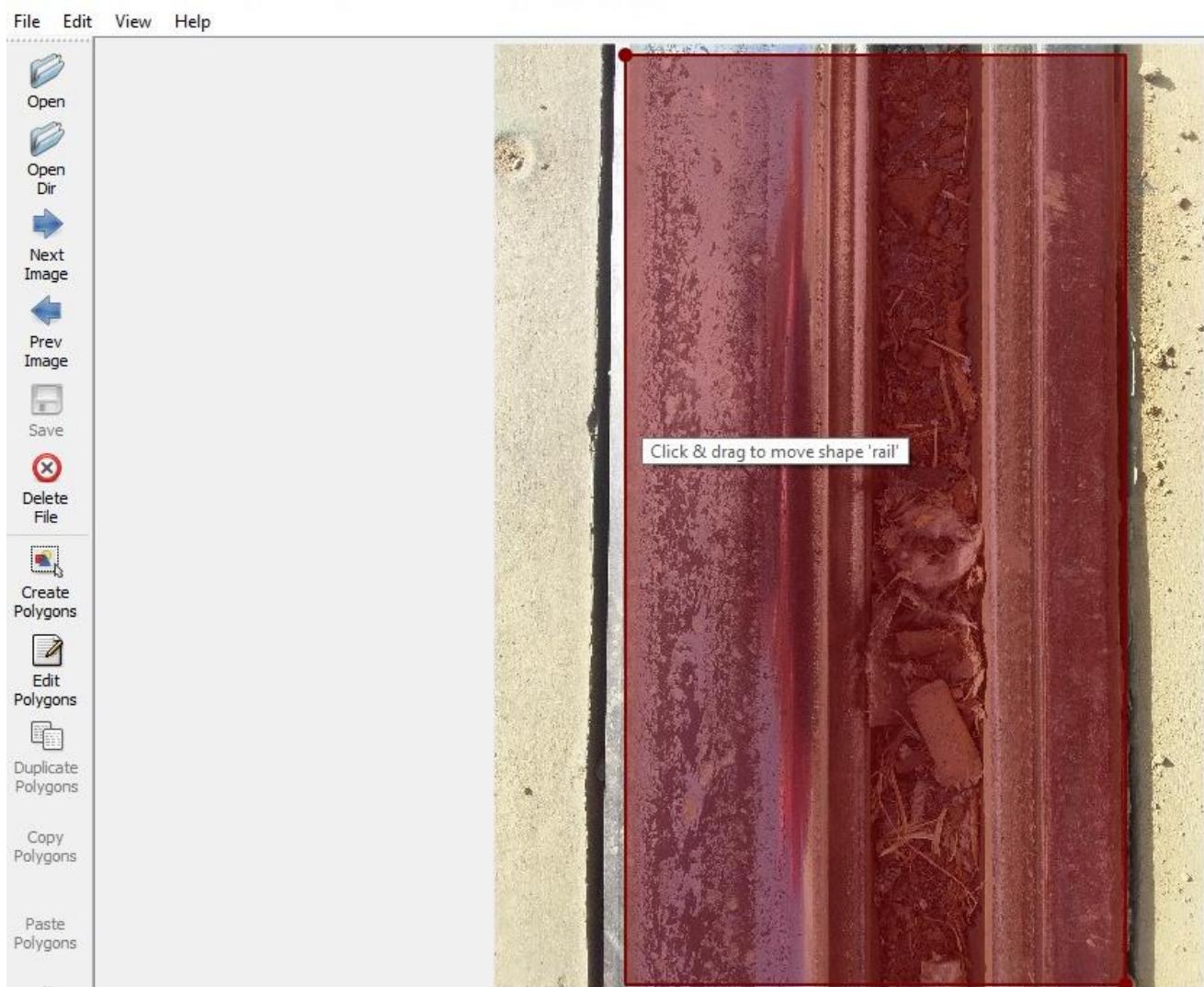


Figure 32 : Exemple d'annotation dans LABELME.

Ces informations sur les images des rails seront stockées dans un fichier JSON comme illustré dans l'image suivante :

```
{
  "version": "5.0.1",
  "flags": {},
  "shapes": [
    {
      "label": "rail",
      "points": [
        [
          5.0,
          784.1851851851851
        ],
        [
          2201.296296296296,
          1680.4814814814813
        ]
      ],
      "group_id": null,
      "shape_type": "rectangle",
      "flags": {}
    }
  ],
  "imagePath": "IMG_1.jpg",
  "imageData": "/9j/4AAQSkZJRgABAQAAAQABAAD/",
  "imageHeight": 2448,
  "imageWidth": 3264
}
```

Figure 33 : Fichier JSON généré dans l'annotation.

3.4.3.2. Configurations pour le MaskR-CNN: Avant de lancer l'entraînement une configuration des paramètres est de rigueur, ces derniers sont représentés dans le tableau ci-dessous :

Configurations	
Architecture	Resnet-101
Taille du batch	2
Momentum	0.9
Taux	0.001

Tableau 2 : tableau de configuration du MaskR-CNN.

Les figures suivantes illustrent des exemples de segmentation :



Figure 34 : Images avant et après la segmentation.

3.4.3.3. Annotation de la détection de l'usure

Afin d'annoter les quatre classes à savoir l'usure ondulatoire, l'usure latérale de la bavette, l'usure latérale du flanc du champignon et la classe sans usure, nous avons choisi d'utiliser la plateforme Web d'annotation Make sens, la figure – 27 illustre un exemple d'annotation illustré dans l'image suivante :

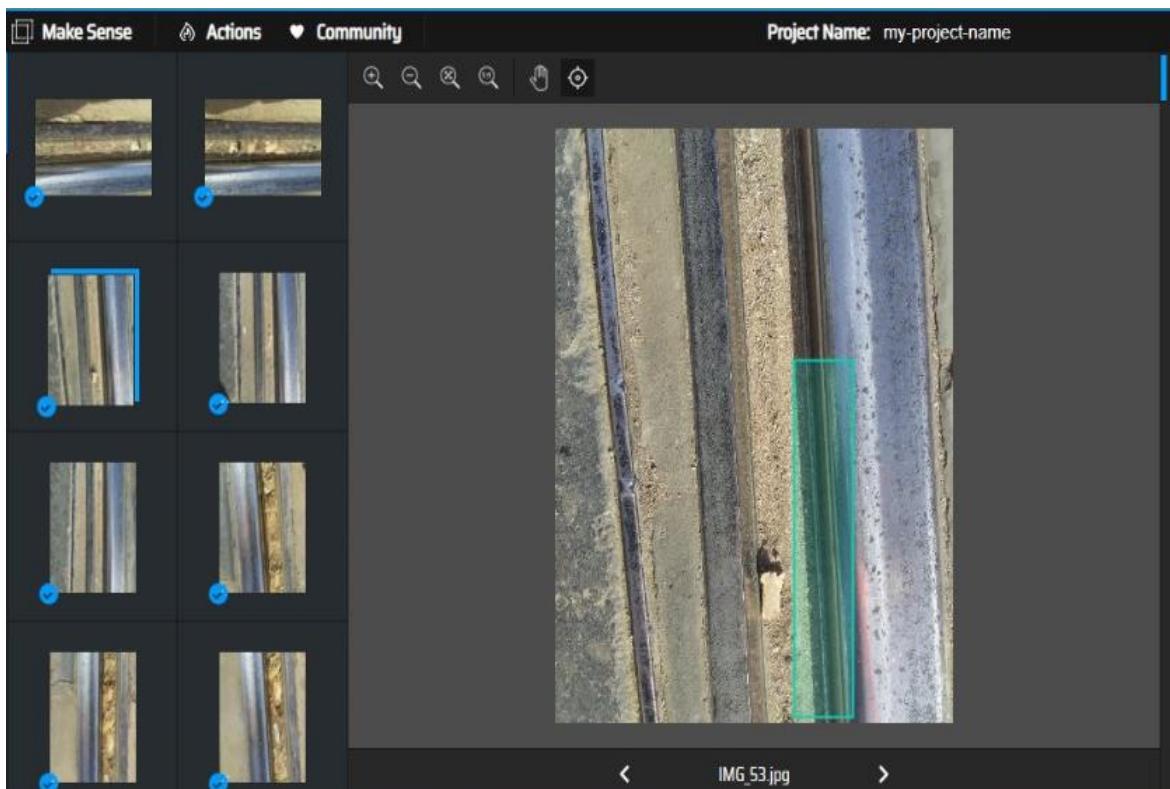


Figure 35 : Annotation de la détection de l'usure.

Ces informations sur les images des rails seront stockées dans un fichier texte comme illustré dans les images suivantes :

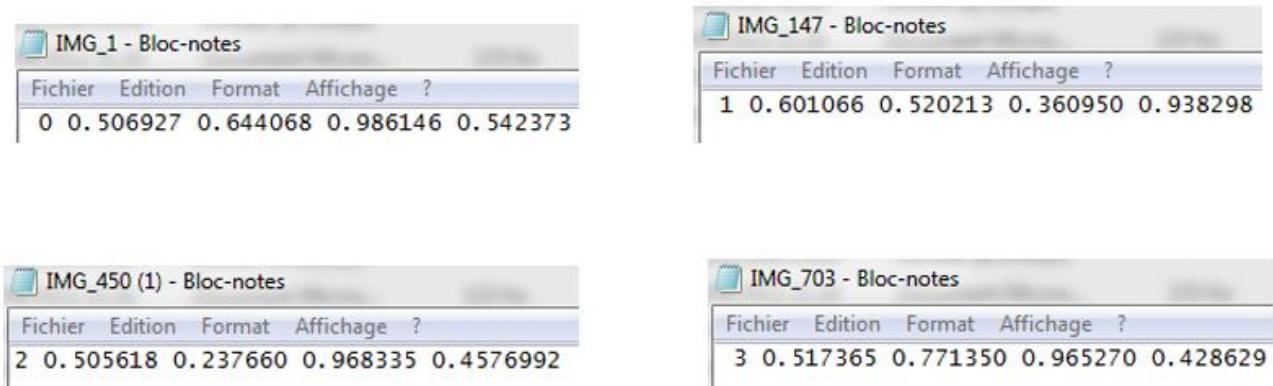


Figure 36 : Fichiers.txt généré dans l'annotation.

Remarque :

Chaque fichier contient une série des nombres, le premier chiffre représente la classe d'usure comme suit :

0 : usure latérale du flanc de champignon.

1 : usure latérale de la bavette.

2 : non usure.

3 : usure ondulatoire.

Et les autres chiffres représentent la localisation de l'usure dans l'image concernée.

3.4.4. Augmentation des données

Après avoir annoté les images de la BDD et générer les fichiers d'annotations, il reste une étape avant l'apprentissage de la détection, c'est celle de l'augmentation de données. En effet, c'est une étape importante afin d'avoir plus d'images pour l'apprentissage qui veut dire plus de données, elle consiste à faire des transformations sur les images existantes dans la BDD, quelques exemples de ces transformations comparées à l'originale sont illustrés dans les images suivantes :



Figure 37 : Floutage.



Figure 38 : Rotation à droite.



Figure 39 : Rotation à gauche.



Figure 40 : Rotation 180°.

Au total, quatre transformations ont été effectuées sur le DataSet, trois de rotations, à droite, à gauche et 180 degrés, et une celle de flottage de l'image.

Donc la nouvelle base de données obtenue contient 3525 images annotées. (le 5 c'est pour les images originales).

Sans augmentation	Avec augmentation
705	$705 * 5 = 3525$

Tableau 3 : Base de données avant et après l'augmentation de données.

3.5.Les modèles qualifiés

3.5.1. Configuration de VGG16

Les configurations ConvNet sont décrites dans le tableau 4. La largeur des couches convolutives (le nombre de canaux) est plutôt petite, commençant à 64 dans la première couche, puis augmentant d'un facteur 2 après chaque couche de regroupement maximum, jusqu'à atteindre 512.

Layer	Output	Layer	Output
Input	3 x 224 x 224	Conv4	512 x 28 x 28
Conv1	64 x 224 x 224		512 x 28 x 28
	64 x 224 x 224		512 x 28 x 28
Max Pool 1	64 x 112 x 112	Max Pool 4	512 x 14 x 14
Conv2	128 x 112 x 112	Conv5	512 x 14x 14
	128 x 112 x 112		512 x 14x 14
Max Pool 2	128 x 56 x 56		512 x 14x 14
Conv3	256 x 56 x 56	Max Pool 5	512 x 7 x 7
	256 x 56 x 56	FC1	4096
	256 x 56 x 56	FC2	1024
Max Pool 3	256 x 28 x 28	FC3	500
		SoftMax	500

Tableau 4 : Tableau de configuration de VGG16.

Remarque :

Pour pouvoir concevoir ce modèle, nous avons divisé nos données en quatre types, chaque type appartient à un dossier.

Le prétraitement et l'augmentation de données sont générés automatiquement.

L’attribution de la base de données s’est effectuée comme suit : 60% pour l’apprentissage (Training), 20% pour la validation et 20% pour le test.

Le nombre d’itération est fixé à 10.

3.5.2. Configuration de YOLOV4

YOLOV4 est développé par trois développeurs Alexey Bochkovskiy, Chien-Yao Wang et Hong-Yuan Mark Liao. YOLOv4 est le 4ème ajout à la famille des modèles de détecteurs d’objets YOLO. Il s’agit d’un modèle marquant qui a solidifié le nom et la position de YOLO dans le domaine de la vision par ordinateur. Il est sorti avec le concept des techniques BOF (Bag Of Freebies) et BOS (Bag Of Specials) pour améliorer les performances du modèle.

- **Fichier classes.names :**

Tout d’abord il faut créer un fichier contenant comme son nom l’indique, les noms des classes à détecter, l’image suivante illustre ce dernier :

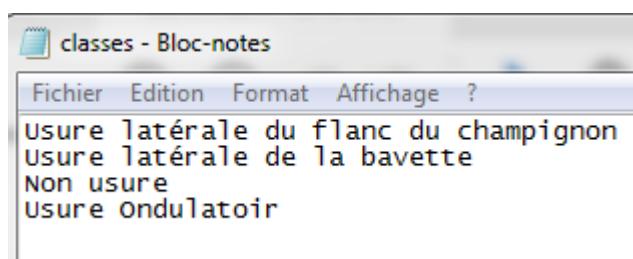


Figure 41 : illustration du fichier classes.names.

- **Fichier obj.data :**

Ensuite la création du fichier obj.data qui contient le nombre de classes et l’emplacement des chemins de la base de données, celui de l’entraînement et de validation, ainsi que le dossier où stocker les poids de l’entraînement. L’image suivante illustre ceci :

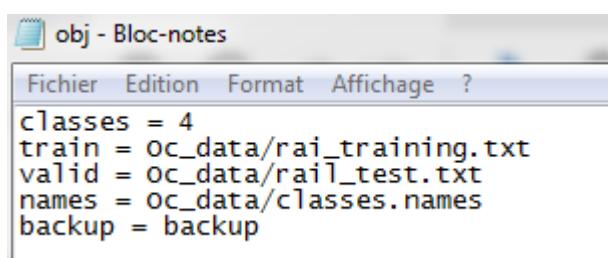
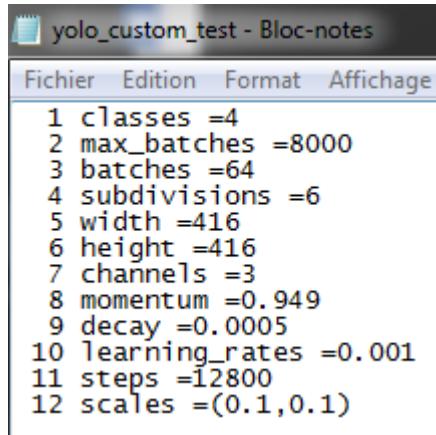


Figure 42 : illustration du fichier obj.data.

- **Fichier yolov4_config.py et yolo4.cfg :**

Et pour finir un paramétrage du modèle en modifiant le nombre de classes, nombres d'itérations, taux d'apprentissage et le nombre de filtres dans le fichier Yolo_custom_test.cfg



```
1 classes =4
2 max_batches =8000
3 batches =64
4 subdivisions =6
5 width =416
6 height =416
7 channels =3
8 momentum =0.949
9 decay =0.0005
10 learning_rates =0.001
11 steps =12800
12 scales =(0.1,0.1)
```

Figure 43 : Illustration du fichier yolo_custom_test.cfg.

3.6. Conclusion

Dans ce chapitre, toutes les étapes de l'architecture du système ont été expliquées de la collecte de données à l'extraction des caractères en passant par la segmentation en utilisant l'algorithme MASK-RCNN. Dans le chapitre suivant on exposera les résultats de ces derniers en discutant et analysant les performances et précisions avec la présentation d'une API qui concrétise le modèle.

Chapitre 4

Implémentation

4.1. Introduction

Dans ce dernier chapitre nous allons d'abord définir l'environnement de développement et le langage de programmation utilisés pour la réalisation de notre système. Nous avons présenté aussi la base d'apprentissage et de test utilisés, les détails de l'implémentation de l'architecture proposée et l'interface graphique de l'application.

4.2. Langages de programmation et environnement de travail

4.2.1. Python [\[14\]](#)

Le langage Python est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données ou la gestion d'infrastructures. Python est un Langage de programmation interprété permet l'exécution du code sur n'importe quel ordinateur. Utilisable aussi bien par des programmeurs débutants qu'experts, Python permet de créer des programmes de manière simple et rapide.



Figure 44 : Logo de Python

4.2.2. TensorFlow [\[15\]](#)

TensorFlow est un framework open source développé par les chercheurs de Google pour exécuter l'apprentissage automatique, l'apprentissage en profondeur et d'autres charges de travail d'analyse statistique et prédictive. Comme des plates-formes similaires, il est conçu

pour rationaliser le processus de développement et d'exécution d'applications d'analyse avancées pour les utilisateurs tels que les scientifiques des données, les statisticiens et les modélisateurs prédictifs. Le framework comprend des ensembles d'API de haut niveau et de bas niveau. Google recommande d'utiliser ceux de haut niveau lorsque cela est possible pour simplifier le développement du pipeline de données et la programmation d'applications.



Figure 45 : Logo TensorFlow.

4.2.3. Keras ^[16]

Keras est une API de réseau de neurones écrite en langage Python. Il s'agit d'une bibliothèque Open Source, exécutée par-dessus des frameworks tels que Theano et TensorFlow. Il est conçu pour être modulaire, rapide et simple à utiliser. Aujourd'hui, Keras est l'une des APIs de réseaux de neurones les plus utilisées pour le développement et le testing de réseaux de neurones. Elle permet de créer très facilement des « layers » pour les Neural Networks ou de mettre en place des architectures complexes.

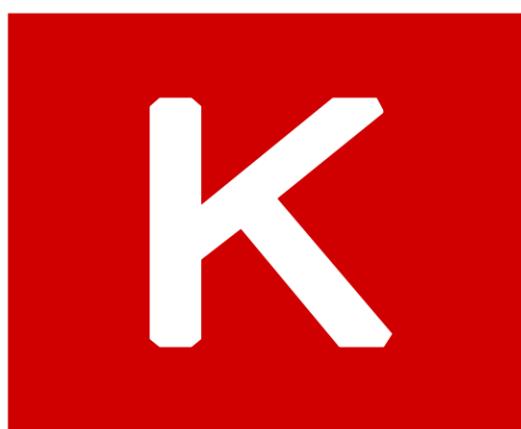


Figure 46 : Logo de Keras.

4.2.4. Google Colab [17]

Google Colab ou aussi appelé Colaboratory est un service dans le cloud, proposé par Google gratuitement. Il est basé sur l'environnement Jupyter Notebook et est destiné à la formation et à la recherche en apprentissage automatique. Cette plateforme permet de former des modèles de machine Learning directement dans le cloud. Il n'est pas nécessaire de l'installer sur l'ordinateur, les ressources informatiques peuvent donc être utilisées pour d'autres tâches. Google colab permet d'enregistrer les différents notebooks sur Google Drive, ce qui permet également de partager et de faire travailler plusieurs personnes sur le même document à la fois.



Figure 47 : Logo de Google Colab.

4.2.5. Flask [18]

Flask est un micro framework open-source de développement web en Python. Flask a pour objectif de créer des applications Web ainsi que des API assez rapidement et facilement.



Figure 48 : Logo de Flask.

4.2.6. Visual studio code [19]

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto-complétion, la coloration syntaxique, le débogage, et les commandes git.

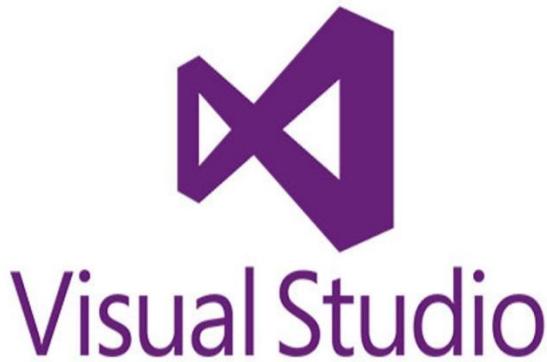


Figure 49 : Logo de Visual Studio.

4.2.7. Wampserver [20]

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.



Figure 50 : Logo de WampServer.

4.2.8. MySql [21]

MySQL Est une abréviation de "My Structured Query Language" ou mon langage de requêtes structuré.

MySQL est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde. Il est distribué sous double licence, une licence publique générale GNU

et une propriétaire selon l'utilisation qui en est faites. La première version de MySQL est apparue en 1995 et l'outil est régulièrement entretenu.



Figure 51 : Logo de MySql.

4.3. Configuration du matériel utilisée

La configuration matérielle utilisée pour l'entraînement et l'évaluation des deux modèles est la suivante:

Un PC Lenovo avec les caractéristiques suivantes:

- processeur intel(R) Core(TM) i3-6006U CPU @2.00GHz 1.99GHz.
- Carte graphique intel.
- RAM d'une taille de 4 GO.
- Système d'exploitation Windows10 professionnel64 bit, processeur x64.

Un PC FUJITSU avec les caractéristiques suivantes:

- processeur intel(R) Core i3 CPU P6200 @ 2.13GHz 2.13GHz.
- Carte graphique intel.
- RAM d'une taille de 4 GO.
- Système d'exploitation Windows 13 64 bits, processeur x64.

4.4.Résultats et discussions

Dans cette section nous allons présenter et discuter les résultats obtenus suite à l'application des deux modèles VGG16 et YOLOV4 sur la base d'images collecter manuellement. Nous allons également donner une comparaison des performances de ces deux modèles.

4.4.1. VGG16

L'entraînement du modèle de détection VGG16 a duré une demi-journée avec un nombre d'itération égal à 6.

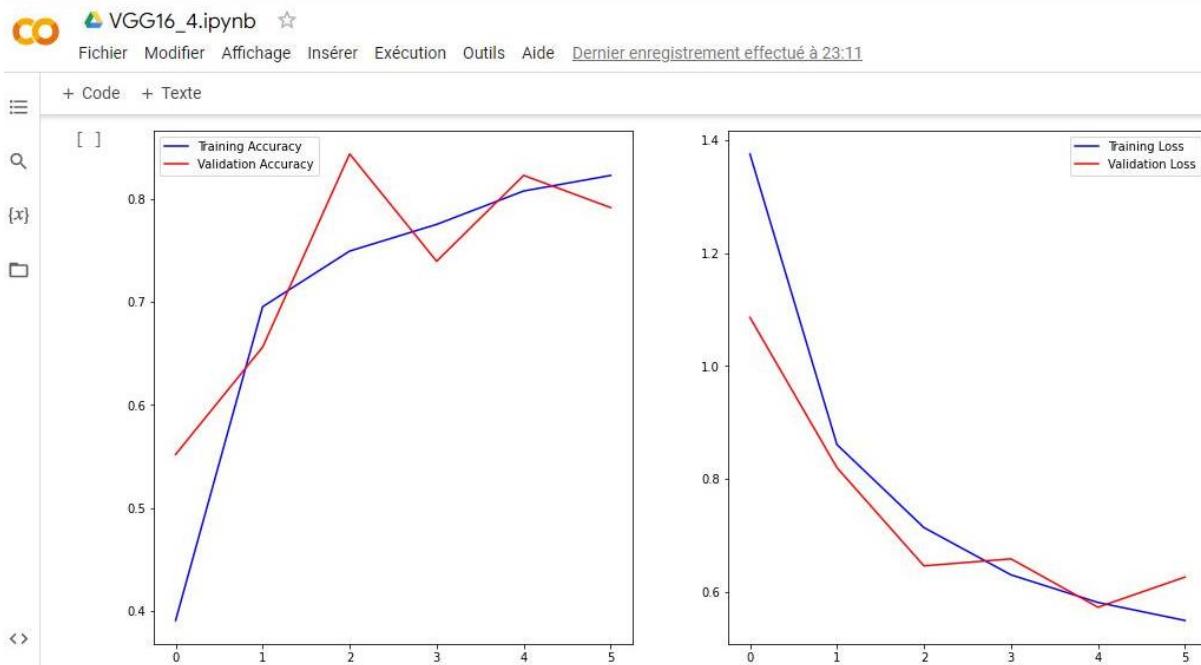


Figure 52 : courbe graphique de la précision et la perte pour 4 classes.

D'après la figure 52, la courbe graphique à gauche montre le résultat de précision de l'entraînement (Training) et de la validation, nous remarquons que cette précision a pu atteindre une valeur maximale dans la troisième itération.

La courbe graphique à gauche montre le résultat de taux d'erreur de l'entraînement (Training) et de la validation, nous remarquons que cette précision a pu atteindre une valeur minimale égale à 54%.

```

Epoch 1/10
29/29 [=====] - ETA: 0s - loss: 1.3757 - accuracy: 0.3909
Epoch 1: val_accuracy improved from -inf to 0.55208, saving model to 2022-08-16 20:10:38.426436_vgg16.h5
29/29 [=====] - 143s 5s/step - loss: 1.3757 - accuracy: 0.3909 - val_loss: 1.0860 - val_accuracy: 0.5521
Epoch 2/10
29/29 [=====] - ETA: 0s - loss: 0.8605 - accuracy: 0.6955
Epoch 2: val_accuracy improved from 0.55208 to 0.65625, saving model to 2022-08-16 20:10:38.426436_vgg16.h5
29/29 [=====] - 144s 5s/step - loss: 0.8605 - accuracy: 0.6955 - val_loss: 0.8196 - val_accuracy: 0.6562
Epoch 3/10
29/29 [=====] - ETA: 0s - loss: 0.7134 - accuracy: 0.7495
Epoch 3: val_accuracy improved from 0.65625 to 0.84375, saving model to 2022-08-16 20:10:38.426436_vgg16.h5
29/29 [=====] - 146s 5s/step - loss: 0.7134 - accuracy: 0.7495 - val_loss: 0.6455 - val_accuracy: 0.8438
Epoch 4/10
29/29 [=====] - ETA: 0s - loss: 0.6297 - accuracy: 0.7754
Epoch 4: val_accuracy did not improve from 0.84375
29/29 [=====] - 141s 5s/step - loss: 0.6297 - accuracy: 0.7754 - val_loss: 0.6580 - val_accuracy: 0.7396
Epoch 5/10
29/29 [=====] - ETA: 0s - loss: 0.5807 - accuracy: 0.8078
Epoch 5: val_accuracy did not improve from 0.84375
29/29 [=====] - 141s 5s/step - loss: 0.5807 - accuracy: 0.8078 - val_loss: 0.5722 - val_accuracy: 0.8229
Epoch 6/10
29/29 [=====] - ETA: 0s - loss: 0.5488 - accuracy: 0.8229
Epoch 6: val_accuracy did not improve from 0.84375
29/29 [=====] - 141s 5s/step - loss: 0.5488 - accuracy: 0.8229 - val_loss: 0.6257 - val_accuracy: 0.7917
Epoch 6: early stopping

```

Figure 53 : résultats détaillé de la précision et la perte pour 4 classes.

La figure 53 montre que le processus de la précision s'est arrêté à la sixième itération (Early stopping), car la précision n'a pas été améliorée de 84%.

Résultats du test

```
Model Test accuracy: 0.8348624110221863
Model Test Loss: 0.5411776900291443
```

Figure 54 : résultat du test pour le modèle VGG16.

La figure précédente représente le résultat du test des 4 classes, 84% pour la précision et 54% pour le taux d'erreur.

La figure suivante représente le résultat de test du modèle VGG16, qui montre cette image est appartient à la classe numéro 02 « non usure ».

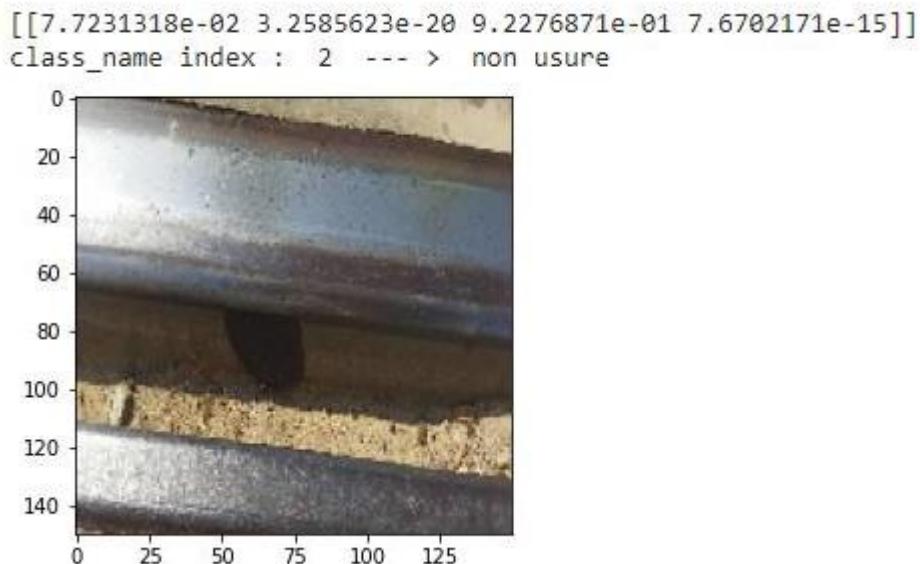


Figure 55 : Exemple de test du modèle VGG16

D'après les résultats trouvés nous pouvons conclure que la valeur de la précision est acceptable.

4.4.2. YOLOv4

L'entraînement du modèle de détection YOLOv4 a duré trois jours, avec un nombre d'itérations égales à 1200.

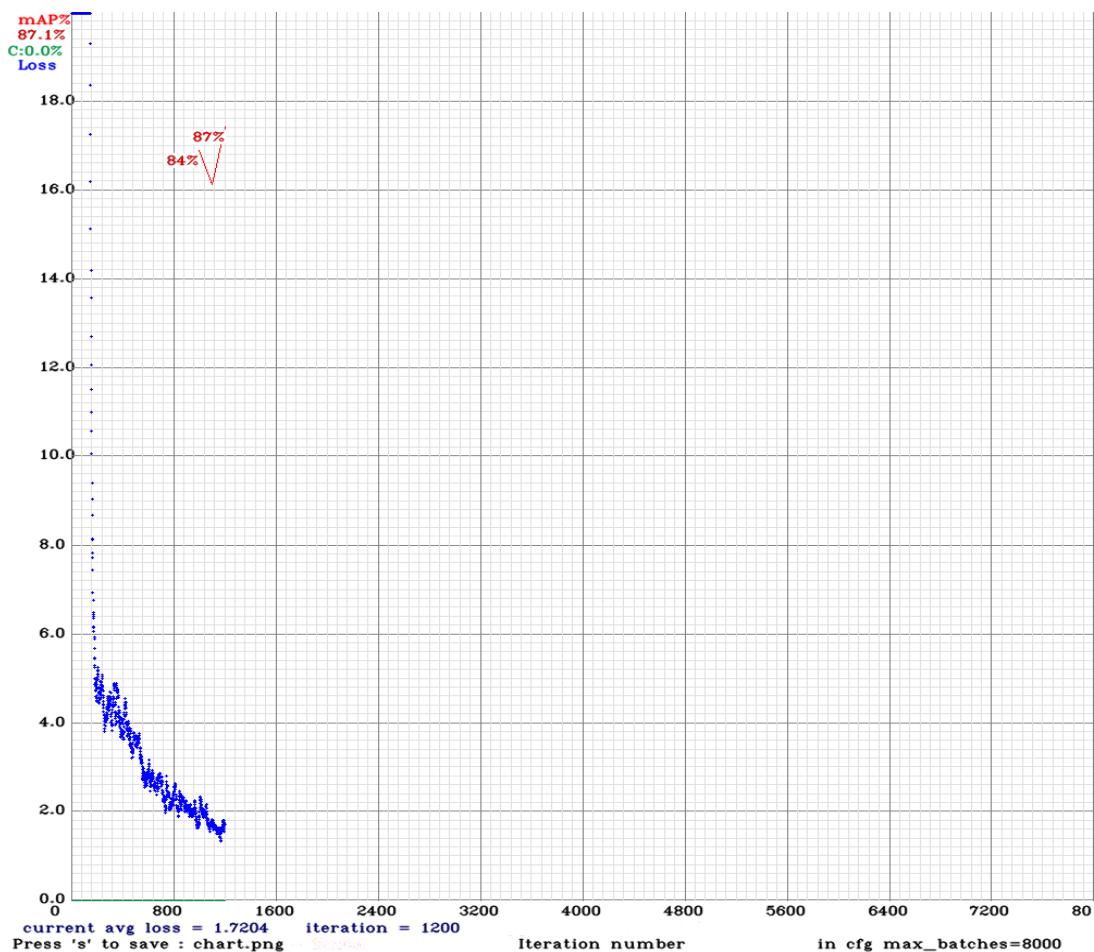


Figure 56 : Schéma d'entraînement de YOLOv4.

D'après cette figure nous pouvons voir que la précision d'entraînement est en train d'augmenter avec le nombre d'itérations, tandis que la perte d'entraînement diminue avec le nombre d'itérations.

Précision

La figure suivante montre le résultat du traitement de YOLOv4 après 1200 itérations :

```
for conf_thresh = 0.25, precision = 0.85, recall = 0.87, F1-score = 0.86
for conf_thresh = 0.25, TP = 176, FP = 32, FN = 27, average IoU = 61.27 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.872962, or 87.30 %
```

Figure 57 : précision d'entraînement.

La figure suivante montre le résultat de la précision pour chaque classe :

```

detections_count = 693, unique_truth_count = 203
class_id = 0, name = Usure latérale du flanc du champignon, ap = 73.33% (TP = 26, FP = 2)
class_id = 1, name = Usure latérale de la bavette, ap = 100.00% (TP = 3, FP = 3)
class_id = 2, name = Non usure, ap = 87.75% (TP = 107, FP = 21)
class_id = 3, name = Usure Ondulatoir, ap = 88.10% (TP = 40, FP = 6)

```

Figure 58 : précision de chaque classe.

D'après les figures illustrées au-dessus nous remarquons que le résultat de précision de ce modèle de détection est acceptable et égale à 87%, avec un taux d'erreur égale à 17%. Ainsi que la précision de chacune des quatre classes est supérieure à 70%. Cela dit que YOLOv4 a pu détecter tous les types.



Figure 59 : prédiction de la classe non usure.

Remarque

D'après la figure au-dessus nous remarquons que le traitement a été arrêté à 1200 itérations tandis que le nombre d'itérations a été fixé au début à 8000 itérations, cela est à cause d'un des obstacles que nous avons rencontrés qu'est la limitation d'utilisation du type d'exécution GPU de Google Colab.

4.5.Comparaison des deux modèles

	YOLOv4	VGG16
Précision	87	84
Taux de d'erreur	17	54
Classification	Oui	Oui
Localisation	Oui	Non

Tableau 05 : Tableau comparatif entre YOLOv4 et VGG16.

En comparant les résultats illustrés dans le tableau 05, nous pouvons constater que les valeurs de précisions pour les deux modèles sont proches, par contre le taux d'erreur du modèle VGG16 est très élevé par rapport à celui de YOLOv4. Donc VGG16 est moins performant que YOLOv4 cela veut dire que ce dernier est le plus adapté avec notre système (nos données).

Un autre avantage de YOLOv4 est qu'il nous a permis de localiser l'usure.

4.6.Application Web

Le modèle proposé a été complété par une interface graphique simple permettant son exécution facile. Tout d'abord, lorsque l'utilisateur ouvre l'application Web, la fenêtre principale s'affiche avec un seul bouton actif.



Figure 60 : L'interface graphique principale.

Le bouton « predict» nous permettons d'accéder à la deuxième interface.

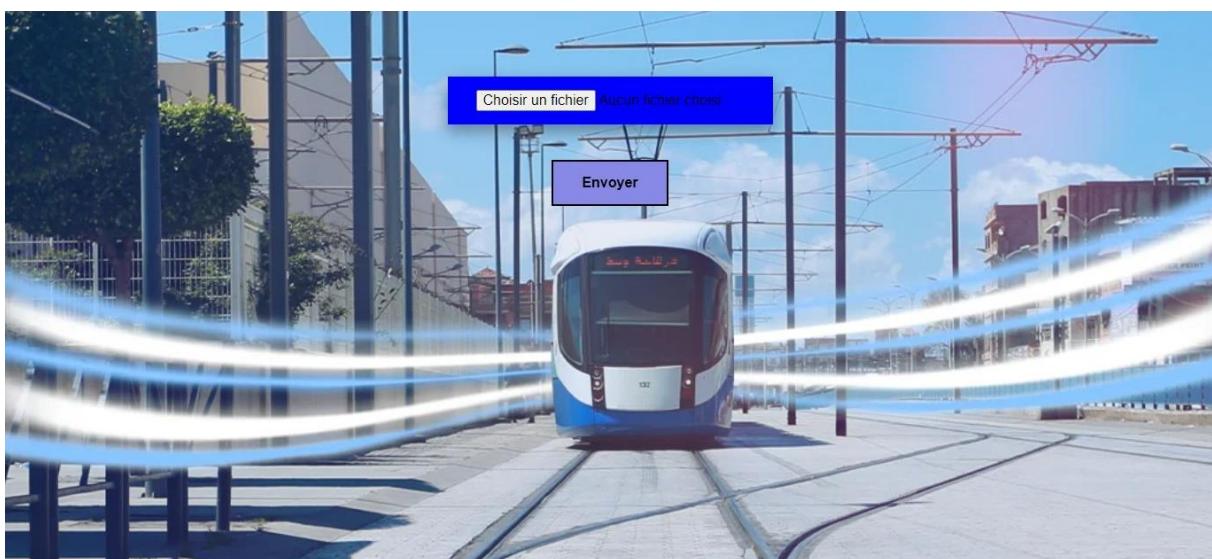


Figure 61 : Interface graphique 2.

Cette interface contient un champ pour insérer les images à prédire et les afficher et un bouton pour faire appel à notre modèle proposé et ensuite afficher le résultat.

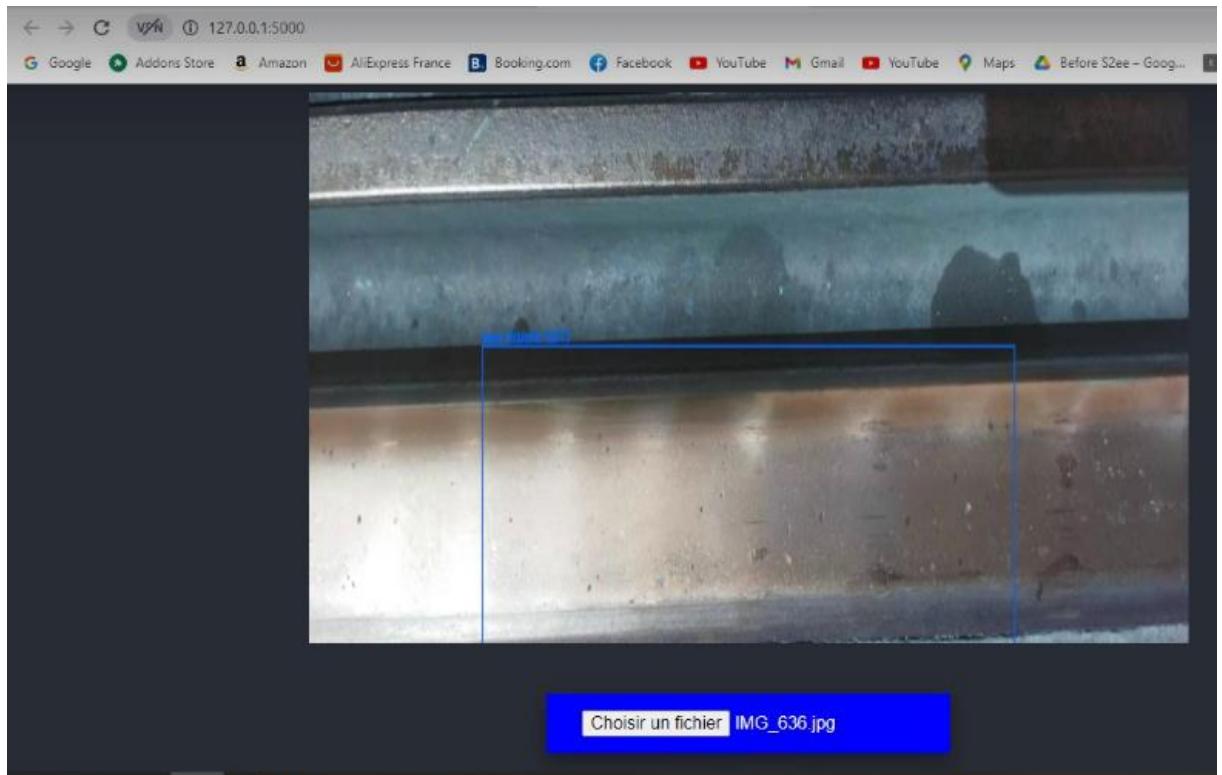


Figure 62 : prédiction de la classe usure Ondulatoire.

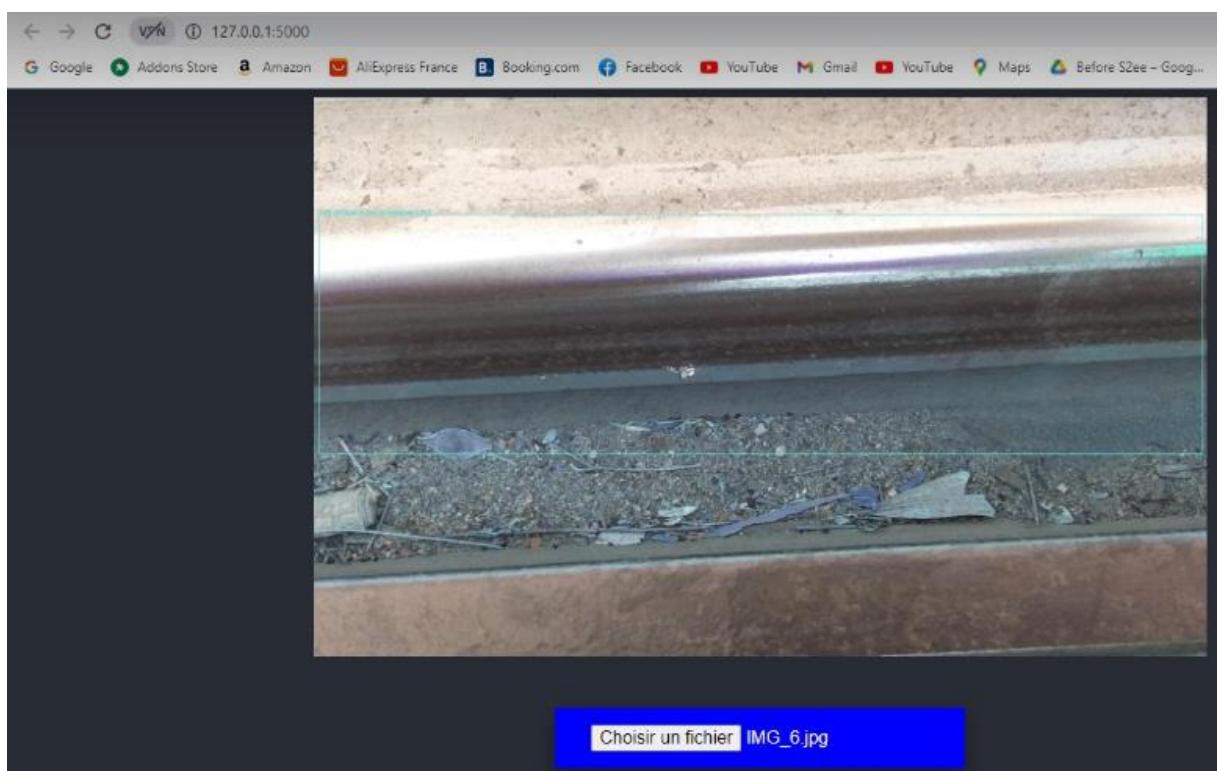
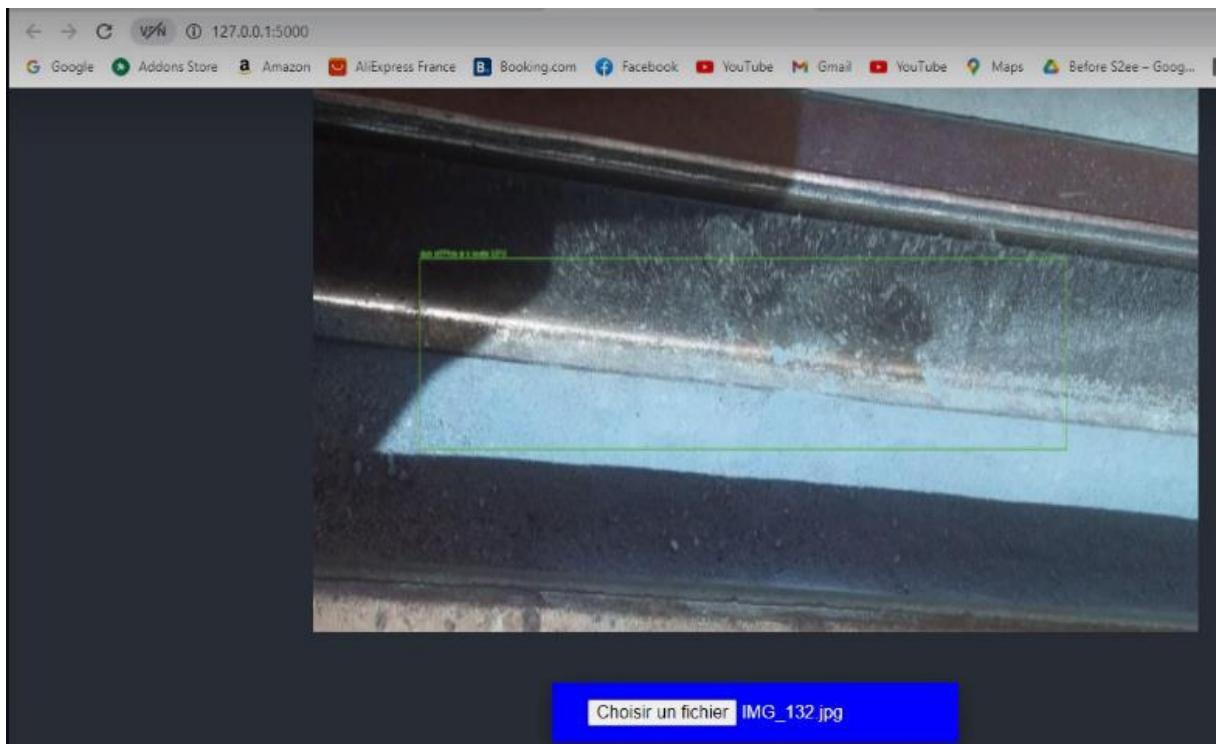
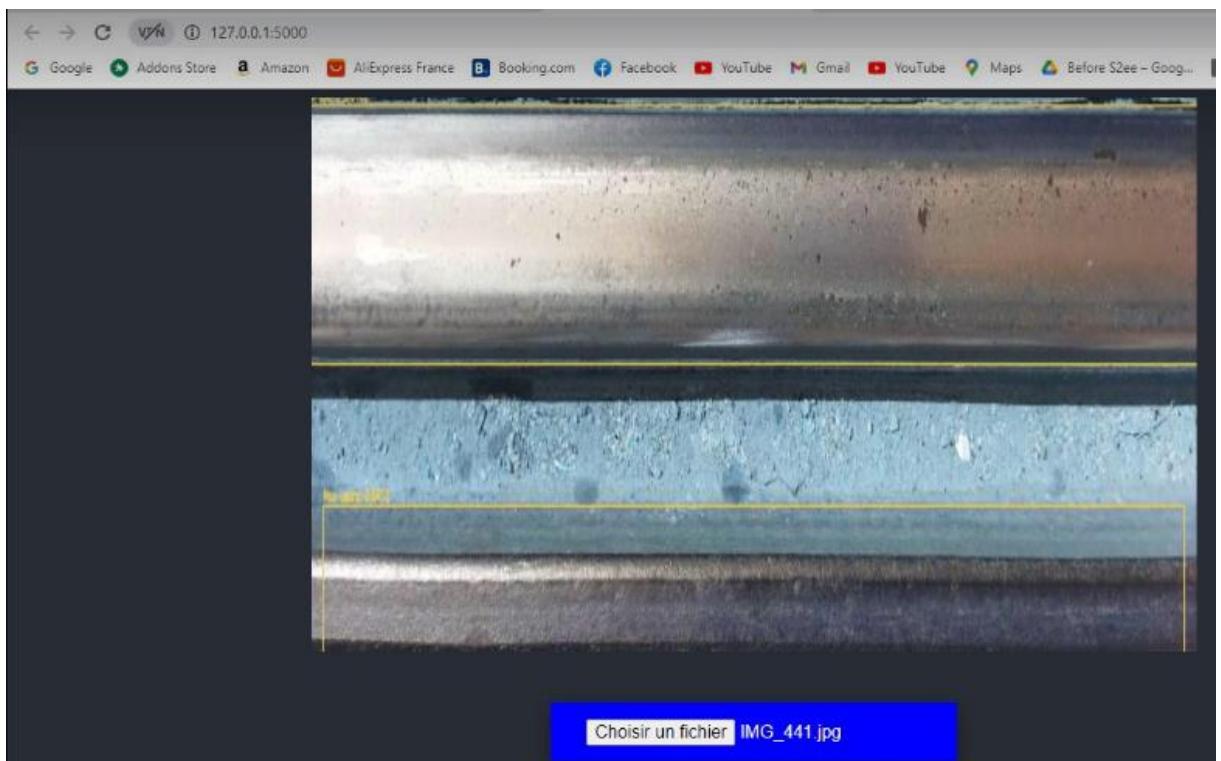


Figure 63 : prédiction de la classe usure de Champignon.**Figure 64 : Prédiction de la classe usure de la Bavette.****Figure 65 : Prédiction de la classe non usure.**

Remarque

Une autre application était créée pour tester la performance du deuxième modèle (VGG16), elle se trouve dans l'annexe.

4.7. Conclusion

Ce chapitre a été divisé en deux parties : implémentation, résultats et comparaisons.

Dans la première partie, nous avons présenté l'environnement de travail, le langage de programmation, les bases d'apprentissage et de test, les détails de l'implémentation de la phase d'apprentissage, ainsi que l'interface graphique. Dans la deuxième partie, nous avons présenté la configuration expérimentale, les résultats et les comparaisons avec d'autres travaux.

Conclusion générale

Le travail réalisé a été développé afin de satisfaire la demande de l'entreprise SETRAM qui souhaitait réaliser un système capable de prendre en entrée des images des rails ferroviaires et retourne le type d'usure associées à chaque image.

L'objectif de ce projet était de réaliser un système basé sur l'intelligence artificielle, permet de contrôler l'état des rails des Tramways afin de réduire un maximum d'erreurs que l'être humain peut causer telles que l'inexactitude du type de l'usure, la perte du temps et de l'argent.

Pour remédier à cette problématique, nous avons proposé une application web qui prend en entré une image afin de déterminer à quel type d'usure elle appartient, pour prendre les mesures nécessaires pour réparer le rail selon l'usure détectée dans l'image.

Pour concevoir ce système nous sommes passées par trois grandes étapes à savoir, segmentation de l'image; afin de délimiter les contours du rail dans chaque image. Ensuite la détection des usures qui se trouvent dans l'image segmentée et enfin l'extraction du type de l'usure trouvé dans l'image.

Afin d'implémenter ce système, nous avons utilisé le Deep Learning en utilisant un DataSet que nous avons collecté manuellement contenant 3525 images de rail qui seront annotées à l'aide du logiciels d'annotation adapté à chacune de ces étapes essentielles à savoir la segmentation de l'image et la détection des usures qui s'y trouvent.

Ce travail nous a permis d'approfondir nos connaissances dans le domaine de la vision par ordinateur en implémentant trois modèles d'apprentissage profond à savoir l'algorithme Mask-RCNN utilisé dans l'étape de segmentation. Pour celle de la détection, nous avons opté d'implémenter YOLOV4 et VGG16 afin de comparer leur performance et choisir lequel est le plus adapté pour notre DataSet.

Cependant, notre système reste ouvert à des améliorations en perspective, avec l'amélioration de la précision en augmentant le DataSet de manière considérable en mettant en place un réseau de capteurs qui collecte en temps réel les informations relatives aux

différents points des rails et les envoient à un serveur distant, et créer une autre application Web ou Mobile qui montre la localisation GPS exacte où se trouve l'usure pour la réparer.

Et la troisième application Web destiner à afficher le type de l'usure, son degré, la localisation et une proposition comment cette usure doit être réparé.

Références bibliographiques

- [1] : CHOLLET Francois. Deep learning with python. 2017
- [2] :ZACCONE Giancarlo, MD REZAUL Karim, MENSHAWY Ahmed. Deep learning with tensorflow. 2017
- [3] :<https://makina-corpus.com/sig-webmapping/extraction-dobjets-pour-la-cartographie-par-deep-learning-choix-du-modele>
- [4] :**Gollapudi, Sunila.** 2019. *Learn Computer Vision Using OpenCV With Deep Learning CNNs and RNNs*. s.l. : appress, 2019.
- [5] :<https://viso.ai/deep-learning/mask-r-cnn/>
- [6] :<https://datascientest.com/quest-ce-que-le-modele-vgg>
- [7] :<https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>.
- [8] :<https://vossloh.com>
- [9] : www.qmt-group.com
- [10] :<https://www.bletmesure.fr/actualites/E08/2019/12/25/002263-ART312002095-jauge-de-profil-de-rail-une-jauge-pour-mesurer-l-usure-d-un-rail.html>
- [11] :<https://fr.vikidia.org/wiki/Rail?oldid=1125595>
- [12] :Thèse de doctorat de l'Université Paris-Est dirigée par Fabien LEURENT, CONTRIBUTIONS A UNE APPROCHE PATRIMONIALE POUR LA VOIE FERREE DE TRAMWAY 9 juin 2020
- [13] : <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python>
- [14] :<https://www.futura-sciences.com/tech/definitions/informatique-python-19349>
- [15] :<https://www.searchdatamanagement.techttarget.com/definition/TensorFlow>.
- [16] :<https://datascientest.com/keras>
- [17] :<https://www.informatique-mania.com/linformatique/google-colaboratory>
- [18] :<https://flask.palletsprojects.com/en/2.0.x/>
- [19] :<https://framalibre.org/content/visual-studio-code>
- [20] :<https://www.wampserver.com/>
- [21] :<https://sql.sh/sqlite/mysq>

Annexe A

VGG16 pour deux classes

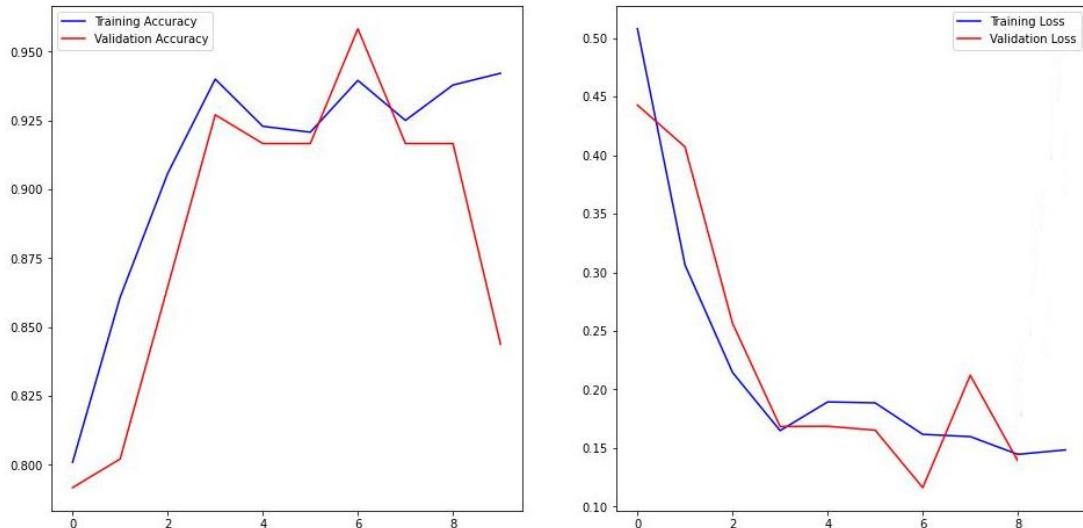


Figure 66 : courbe graphique de la précision et la perte VGG16 pour 2 classes.

```

Epoch 5: val_accuracy did not improve from 0.92708
30/30 [=====] - 140s 5s/step - loss: 0.1893 - accuracy: 0.9229 - val_loss: 0.1686 - val_accuracy: 0.9167
Epoch 6/10
30/30 [=====] - ETA: 0s - loss: 0.1885 - accuracy: 0.9208
Epoch 6: val_accuracy did not improve from 0.92708
30/30 [=====] - 139s 5s/step - loss: 0.1885 - accuracy: 0.9208 - val_loss: 0.1652 - val_accuracy: 0.9167
Epoch 7/10
30/30 [=====] - ETA: 0s - loss: 0.1616 - accuracy: 0.9396
Epoch 7: val_accuracy improved from 0.92708 to 0.95833, saving model to 2022-08-16 20:52:36.623050_vgg16.h5
30/30 [=====] - 141s 5s/step - loss: 0.1616 - accuracy: 0.9396 - val_loss: 0.1161 - val_accuracy: 0.9583
Epoch 8/10
30/30 [=====] - ETA: 0s - loss: 0.1597 - accuracy: 0.9251
Epoch 8: val_accuracy did not improve from 0.95833
30/30 [=====] - 138s 5s/step - loss: 0.1597 - accuracy: 0.9251 - val_loss: 0.2120 - val_accuracy: 0.9167
Epoch 9/10
30/30 [=====] - ETA: 0s - loss: 0.1445 - accuracy: 0.9379
Epoch 9: val_accuracy did not improve from 0.95833
30/30 [=====] - 142s 5s/step - loss: 0.1445 - accuracy: 0.9379 - val_loss: 0.1388 - val_accuracy: 0.9167
Epoch 10/10
30/30 [=====] - ETA: 0s - loss: 0.1483 - accuracy: 0.9422

```

Figure 67 : résultats détaillé de la précision et la perte pour 2 classes.

Après avoir analysé les résultats de VGG16 pour quatre classes nous avons remarqué que la valeur de la fonction de perte était très élevé, donc nous avons décider de tester le modèle avec deux classes (classe usure, classe non usure). Ce modèle est performant car

d'après les résultats illustrés dans les deux figures précédentes, nous voyons que la valeur de la fonction de précision est égale à 95% et la valeur de perte égale à 15%.

Annexe **B**

Application Web pour le VGG16



Figure 68 : Interface graphique pour VGG16.

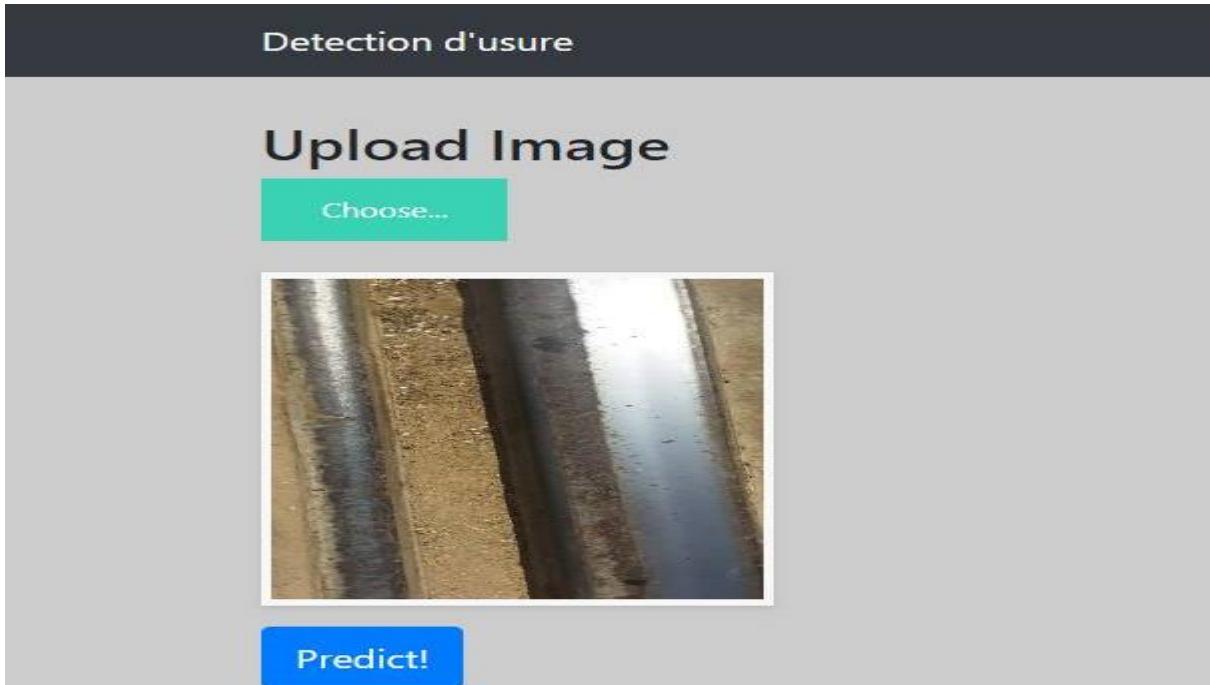


Figure 69: affichage de l'image à prédire.

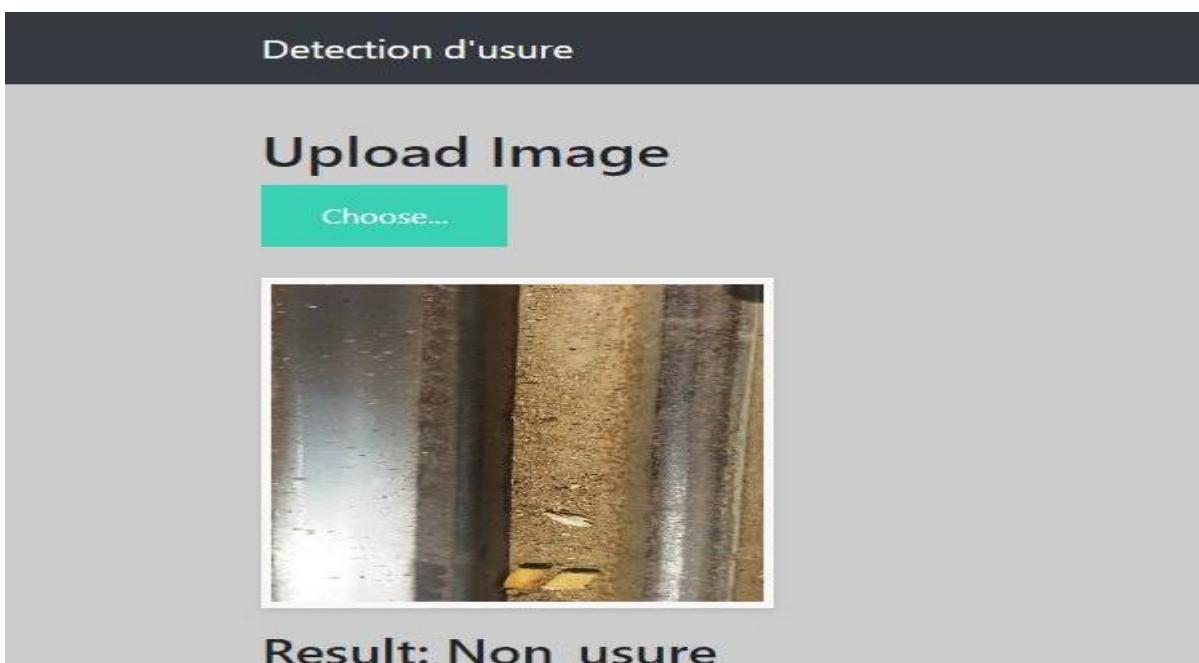


Figure 70 : Affichage du résultat de la prédiction.

Abstract

In recent years, more and more companies want to invest in Deep Learning and computer vision applications. Indeed, object detection is a very active area of research that aims to locate regions or areas of an image or video stream. In this project, we want to detect from an image, a railway rail and extract its failures (corrosion) so that the company can repair the corosions found in due time, so as not to waste time or money. And therefore provide better service to citizens.

In order to achieve these objectives, the SETRAM Company wishes to automate this process of detecting corrosion in railway rails. Such a transition involves the development of a platform to meet the needs of the company.

Our system includes a web application for employees so that they can insert an image of a rail, detect and extract the types of corrosion found there and finally send the experts to the field to repair the damage found.

Keywords: railway track, corrosion, Deep Learning, objects detection, VGG16, YOLOV4, Web application.

ملخص

في السنوات الأخيرة، ترحب المزيد من الشركات في الاستثمار في تطبيقات التعلم العميق ورؤية الكمبيوتر في الواقع.

يعد اكتشاف الأشياء مجالاً نشطاً جدًا في ميدان البحوث الذي يهدف إلى تحديد مناطق من صورة أو من فيديو، في هذا المشروع نريد أن نكتشف صورة سكة حديدية واستخراج أعطالها (تآكلها) حتى تتمكن الشركة من إصلاح التآكل الموجود في الوقت المناسب، من أجل توفير الوقت والمال، وبالتالي تقديم خدمة أفضل للمواطنين.

من أجل تحقيق هذه الأهداف، ترغب شركة SETRAM في أتمة عملية الكشف عن التآكل في قضبان السكك الحديدية. يتضمن هذا الانتقال تطوير منصة لتلبية احتياجات الشركة.

يشتمل نظامنا على تطبيق ويب للموظفين حتى يتمكنوا من إدخال صورة لسكة حديدية واكتشاف واستخراج أنواع التأكل الموجودة فيها وإرسال الخبراء في النهاية إلى الميدان لإصلاح الضرر الذي تم العثور عليه.

الكلمات المفتاحية: سكة حديدية، تأكل، تعلم عميق، كشف الأشياء، تطبيق الويب.