

RÉPUBLIQUE DU CAMEROUN
REPUBLIC OF CAMEROON
Peace – Work – Fatherland

UNIVERSITÉ DE DSCHANG
UNIVERSITY OF DSCHANG
Scholae Thesaurus Dschangensis Ibi Cordum

BP 96, Dschang (Cameroun) – Tel (237) 233 45 13 81
Website: <http://www.univ-dschang.org>.
E-mail : udsrectorat@univ-dschang.org



INSTITUT UNIVERSITAIRE DE TECHNOLOGIE
FOTSO VICTOR DE BANDJOUN
FOTSO VICTOR UNIVERSITY
INSTITUTE OF TECHNOLOGY

Département de Génie Informatique
Department of Computer Engineering

BP 134, Bandjoun – Tel (237) 299 31 61 30 / 70 64 23 92
Website: <http://www.univ-dschang.org/iutfv/>
E-mail: iut.fotsovictor@univ-dschang.org

PROJET DE FIN D'ETUDES

SYSTEME DE RECONNAISSANCE ET D'EXTRACTION DES FORMES IDENTIFIABLES A PARTIR DES DONNEES GEOGRAPHIQUES

Réalisé par :

FOKO Alexis Gyslain

QSIR **CM-UDS-20IUT0296**

NGUETSA MEINKONDEM Darius Chatelain

CDRI **CM-UDS-20IUT1682**

TSAGUE KENFACK Joyce Laura

CDRI **CM-UDS-22IUT1158**

En vue de l'obtention de la

Licence de Technologie

Mention : **Génie Informatique**

Parcours : **Informatique et Réseaux**

Option : **Concepteur Développeur Réseaux Internet (CDRI)**

Qualité et Sécurité Internet Réseaux (QSIR)

Sous l'encadrement de :

Dr. FOTSING Eric

M. KAKEU Severin

Enseignant chargé de cours à l'IUT-FV de
Bandjoun

Enseignant vacataire à l'IUT-FV de Bandjoun

Membres du jury :

Président : **Pr. TANGUY Evani**, enseignant à l'IUT-FV de Bandjoun

Membres : **Dr. TABUEU FOTSO Laurent**, enseignant à l'IUT-FV de Bandjoun

M. AYEMTSA KUETE, enseignant à l'IUT-FV de Bandjoun

Année Académique 2022-2023

DEDICACE

A NOS FAMILLES

AVANT-PROPOS

L’Institut Universitaire de Technologie FOTSO VICTOR de BANDJOUN (en abrégé IUT FV) est un établissement de formation professionnel de l’université de Dschang. Il est né suite à la réforme universitaire de 1993, suivant l’arrêté présidentiel N°008/CAB/PR du 19 Janvier 1993.

L’IUT FV a pour vocation principale d’assurer la formation professionnelle des citoyens camerounais et étrangers, dans les domaines industriels et commerciaux. À ce titre, il fournit aux entreprises ou administrations des prestations de recherches appliquées, des services ou des formations professionnelles correspondant à leurs besoins. L’IUT FV de BANDJOUN a pour devoir de former les jeunes titulaires d’un baccalauréat ou équivalent, de les préparer aux examens nationaux du **BTS, DUT, LICENCE PROFESSIONNELLE et LICENCE TECHNOLOGIQUE**.

L’IUT FV DE BANDJOUN comporte plusieurs cycles de formation :

➤ **Diplôme Universitaire et Technologique (DUT), obtenu en deux ans dans les filières suivantes :**

- ✓ Génie Électrique (GE) ;
- ✓ Génie Informatique (GI) ;
- ✓ Génie de Télécommunication et Réseau (GTR) ;
- ✓ Génie Civil (GC) ;
- ✓ Maintenance Industrielle et Productique (MIP) ;
- ✓ Génie Thermique Energétique et Environnement (GTEE) ;
- ✓ Mécatronique Automobile (MKA) ;
- ✓ Gestions des Entreprises et des Administrations (GEA).

➤ **Brevet des techniciens supérieur (BTS) obtenu en deux ans dans les filières suivantes :**

- ✓ Maintenance des Systèmes Electroniques (MSE) ;
- ✓ Electrotechnique (ET) ;
- ✓ Froid et Climatisation (FC) ;

- ✓ Génie civil (**GC**) ;
- ✓ Gestion des Entreprises et administration (**GEA**) ;
- ✓ Banque Finance (**BF**) ;
- ✓ Assistant Manager (**ASM**) ;
- ✓ Marketing Commerce Vente (**MCV**) ;
- ✓ Collectivité Territoriale (**CT**).

➤ **Licences de technologiques (LT), Mention :**

- ✓ Gestion et Maintenance des Installations Électriques (**GMIE**) ;
- ✓ Informatique et Réseau (**IR**) parcours Concepteur et Développement Réseau Internet (**CDRI**) ;
- ✓ Génie Électrique (**LGE**) ;
- ✓ Génie Civil (**GC**) ;
- ✓ Maintenance Industrielle et Productique (**MIP**) ;
- ✓ Ingénierie des Réseaux et Télécommunications (**IRT**) ;
- ✓ Génie Géomatique (**GG**) ;
- ✓ Génie Thermique et Energétique (**GTE**).

➤ **Licence professionnelles (LP), Mention :**

- ✓ Marketing Manager Opérationnel (**MMO**) ;
- ✓ Banque Gestion des Relations Clientèle (**BGRC**) ;
- ✓ Gestion Administrative, Management des Organisations (**GAMO**) ;
- ✓ Gestion Comptable et Financière (**GCF**).

L'IUT-FV offre plusieurs types de formation :

- ❖ La formation Initiale ;
- ❖ La formation continue ;
- ❖ La formation à distance.

REMERCIEMENTS

Ce travail de recherche est avant tout une œuvre collective. Avant de mettre à la disposition des lecteurs la substance de ce rapport, l'occasion nous est offerte ici d'adresser nos sincères remerciements à tous ceux qui ont aidés de près ou de loin à la réalisation de ce projet. On ne saurait citer les noms de façon exhaustive. C'est ainsi que nous pensons :

- ✓ Au Directeur de l'IUT-FV **Pr. TCHINDA René** pour son engagement à faire de ses étudiants les meilleurs de la nation de demain ;
- ✓ Au **Pr. TAYOU DJAMEGNI Clémentin**, chef de Département du Génie Informatique de l'IUT-FV de Bandjoun pour l'attention qu'il porte à l'égard de notre formation et sa disponibilité ;
- ✓ A nos encadreurs **Dr. FOTSING Eric** et **M. KAKEU Severin** pour leurs disponibilités, explications, conseils, remarques et critiques ; à qui nous disons merci pour tout ce qu'ils nous ont inculqués non seulement en termes de connaissances mais aussi en termes de valeurs ;
- ✓ Aux membres du jury **Pr. TANGUY Evani**, **Dr. TABUEU FOTSO Laurent**, **M. AYEMTSA KUETE** pour leurs critiques et suggestions d'amélioration de notre travail ;
- ✓ Aux Enseignants du Département du Génie Informatique ainsi que le personnel de l'IUTFV de Bandjoun pour le dévouement qu'ils accordent à notre formation ;
- ✓ A nos très chers parents pour leurs perpétuelles efforts et sacrifices dans le sens de notre réussite.

LISTE DES ABREVIATIONS

Abréviation / Sigle	Définition du sigle
API	Application Programming Interface
CNN	Convolutional Neural Networks
CV2	Computer Vision version 2
GAN	Generative Adversarial Networks
GPU	Graphics Processing Unit
IoU	Intersection Over Union
LiDAR	Light Detection and Ranging
ML	Machine Learning
NLP	Natural Langage Processing
OpenCV	Open-Source Computer Vision Library
RAM	Random Access Memory
RNN	Recurrent Neural Networks
RSO	Radar à Synthèse d'Ouverture
SIG	Systèmes d'Informations Géographiques
TPU	Tensor Processing Unit
WandB	Weights and Biases

RESUME

Un système de reconnaissance et d'extraction de formes est un système capable de reconnaître une forme, de l'extraire et de l'identifier. Cette forme peut être contenue sur une image satellite. Ce système peut être utilisé dans des domaines tels que la gestion de l'environnement, la planification urbaine, la cartographie des ressources naturelles, l'agriculture, la sécurité nationale en automatisant la tâche d'analyse des données géographiques. Un tel système peut aider à économiser du temps et des ressources humaines, tout en permettant une analyse plus précise et approfondie des données. Pour implémenter ce système nous avons utilisé des algorithmes de Machine Learning et de Deep Learning pour l'entraînement de notre modèle basé sur les technologies telles que Google Colab, WanDB, TensorFlow et Keras ainsi que le langage de programmation Python. Une application web a été mise sur pied pour faciliter l'utilisation de notre modèle. Les résultats obtenus du système sont l'identification des formes telles que les routes, les bâtiments, les cours d'eau, les végétations qui pourront être utiles dans la cartographie, les planifications urbaines, la gestion des ressources naturelles, les surveillances environnementales et bien d'autres. Les principales difficultés que nous avons rencontrées sont la prise en main de cette nouvelle science qui est l'intelligence artificielle, l'acquisition d'un *dataset* propre au Cameroun, le manque de ressources matérielles pour l'apprentissage de notre modèle ainsi que son déploiement sur une application web.

Mots clés :

- Machine Learning
- Deep Learning
- Segmentation d'images
- Dataset

ABSTRACT

A pattern recognition and extraction system are a system able to recognize a pattern, extract it and identify it. This shape can be contained on a satellite image. This system can be used in fields such as environmental management, urban planning, natural resource mapping, agriculture, national security by automating the task of geographic data analysis. Such a system can help save time and human resources, while enabling more accurate and in-depth data analysis. To implement this system, we used Machine Learning and Deep Learning algorithms to train our model based on technologies such as Google Colab, WanDB, TensorFlow and Keras as well as the Python programming language. A web application has been set up to facilitate the use of our model. The results obtained from the system are the identification and extraction of shapes such as roads, buildings, waterways, vegetation that could be useful in cartography, urban planning, natural resource management, environmental monitoring etc. The main difficulties we encountered are the handling of this new science which is artificial intelligence, the lack of material resources for learning our model as well as its deployment on a web application.

Key words:

- Machine Learning
- Deep Learning
- Image segmentation
- Dataset

SOMMAIRE

DEDICACE	i
AVANT-PROPOS	ii
REMERCIEMENTS	iv
LISTE DES ABREVIATIONS	v
RESUME.....	vi
ABSTRACT	vii
SOMMAIRE	viii
LISTE DES FIGURES	x
LISTE DES TABLEAUX	xi
INTRODUCTION.....	1
CHAPITRE I: PRESENTATION DU PROJET	2
INTRODUCTION.....	2
I.1- PRESENTATION DE L'ARRETE DU PROJET	2
I.2- PROBLEMATIQUE	2
I.3- OBJECTIFS DU PROJET	3
I.4- ACTEURS CIBLES DU PROJET	3
I.5- RESULTATS ATTENDUS	4
I.6- ETAT DE L'ART	4
I.7- CHRONOGRAMME DES ACTIVITES	6
CONCLUSION	7
CHAPITRE II : ANALYSE FONCTIONNELLE ET TECHNIQUE DE CONCEPTION.....	8
INTRODUCTION.....	8
II.1- ANALYSE FONCTIONNELLE DU SYSTEME	8
II.2- CONCEPTION TECHNIQUE ET FONCTIONNELLE DU SYSTEME	9
II.3- INTERACTION ENTRE LES ACTEURS ET LE SYSTEME	23
II.3- COMPORTEMENT INTERNES ET DIFFERENTS ETATS LORS DU TRAITEMENT D'UNE IMAGE	25
CONCLUSION	27
CHAPITRE III: IMPLEMENTATION ET RESULTATS	28
INTRODUCTION.....	28
III.1- TECHNOLOGIES UTILISEES	28
III.2- ENVIRONNEMENT DE DEVELOPPEMENT	29
III.3- ENVIRONNEMENT DE PRODUCTION	31

III.4- EVOLUTION DE L'APPRENTISSAGE DU MODELE	32
III.5 RESULTATS OBTENUS	35
III.6- APPLICATION DES RESULTATS	42
CONCLUSION	42
CHAPITRE IV: DIFFILCUTES ET SUGGESTIONS	43
INTRODUCTION	43
IV.1- EVALUATION DES COUTS	43
IV.2- DIFFICULTES RENCONTREES	44
IV.3- SUGGESTIONS	45
CONCLUSION	46
REFERENCES BIBLIOGRAPHIQUES	xii
ANNEXE	xiii
TABLE DES MATIERES	xiv

LISTE DES FIGURES

Figure 1: Exemple de dataset Scalaire	9
Figure 2: Image du dataset	10
Figure 3: Mask d'une image du dataset	11
Figure 4: Image en échelle de gris	12
Figure 5: Image en couleur.....	13
Figure 6: Images et mask au patch size.....	14
Figure 7: Classes identifiables.....	15
Figure 8: Encodage OneHot.....	16
Figure 9: Image après encodage OneHot	17
Figure 10: Image avec translation	18
Figure 11: Aspect général d'un réseau de neurones à convolution	19
Figure 12: Convolution	20
Figure 13: Max Pooling.....	21
Figure 14: Résultats attendus à l'issu de traitement d'image	22
Figure 15: Architecture Unet.....	22
Figure 16: Diagramme de cas d'utilisation	24
Figure 17: Diagramme d'activité : traiter une image	26
Figure 18: Diagramme d'état transition	27
Figure 19: Evolution de l'apprentissage du modèle	32
Figure 20: Evolution des pertes sur les données d'entraînement et de validation	34
Figure 21: Evolution du coefficient de Jaccard sur les données d'entraînement et de validation	35
Figure 22: Image de test : ville de Dubaï	36
Figure 23: Différents patchs obtenus (40)	36
Figure 24: Résultat de la prédiction des différents patchs.....	37
Figure 25: Image globale prédict	37
Figure 26: Interface de l'application web	38
Figure 27: Patchs issus de l'image rogner	39
Figure 28: Patchs predictions and statistics 1.....	40
Figure 29: Summarize of the process	41
Figure 30: Patchs predictions and statistics 2.....	xiii

LISTE DES TABLEAUX

Tableau 1: Chronogramme des activités effectuées	6
Tableau 2: Evaluation des coûts	43

INTRODUCTION

L'analyse de données géographiques est devenue essentielle dans de nombreux domaines tels que la gestion de l'environnement, le transport, l'urbanisme, la cartographie et l'agriculture. Cependant, l'analyse manuelle de ces données peut être fastidieuse et coûteuse en temps et en ressources humaines. Par conséquent, les systèmes de reconnaissance et d'extraction des formes identifiables à partir de données géographiques ont été développés pour aider à automatiser cette tâche. Dans ce contexte, notre projet vise à développer un système de reconnaissance et d'extraction des formes identifiables à partir de données géographiques. Le système est capable d'analyser des données géographiques brutes telles que les images satellites et d'extraire des informations importantes telles que les routes, les bâtiments, les cours d'eau, les forêts et les sols. Le système sera basé sur des techniques de traitement d'image et de vision par ordinateur, telles que la segmentation d'image.

Plus qu'un Art, L'intelligence artificielle est une discipline jeune d'une soixantaine d'années, qui réunit des sciences, théories et techniques (notamment logique, mathématique, statistiques, probabilités, neurobiologie computationnelle et informatique) et dont le but est de parvenir à faire imiter par une machine les capacités cognitives d'un être humain. Notre système est une forme d'intelligence artificielle capable d'effectuer la segmentation sémantique d'images.

Dans la suite de notre analyse nous expliciterons les différentes techniques, technologie et étapes qui entrent dans sa mise en place. Tout d'abord nous allons présenter une description de notre projet au chapitre 01, puis une analyse fonctionnelle et technique de conception au chapitre 02, ensuite l'implémentation de notre solution, les résultats obtenus au chapitre 03, enfin les difficultés rencontrées et suggestions au chapitre 04.

CHAPITRE I: PRESENTATION DU PROJET

INTRODUCTION

Ce chapitre est consacré à la présentation générale du projet. Tout d'abord nous allons présenter la problématique qui s'est posée, puis l'état de l'art, ensuite les objectifs du projet et enfin les acteurs cibles de ce projet.

I.1- PRESENTATION DE L'ARRETE DU PROJET

L'analyse de données géographiques est devenue essentielle dans de nombreux domaines tels que la gestion de l'environnement, le transport, l'urbanisme, la cartographie, et l'agriculture. Cependant, l'analyse manuelle de ces données peut être fastidieuse et coûteuse en temps et en ressources humaines. Par conséquent, les systèmes de reconnaissance et d'extraction des formes identifiables à partir de données géographiques ont été développé pour aider à automatiser cette tâche.

Le projet consiste à développer un système de reconnaissance et d'extraction des formes identifiables à partir de données géographiques. Ce système sera capable d'analyser des données géographiques brutes telles que des images satellites et d'extraire des informations importantes telles que les routes, les bâtiments, les cours d'eau, les forêts, et les sols.

Le système est basé sur des techniques de traitement d'images et de vision par ordinateur, telles que la segmentation d'image, la reconnaissance de forme, l'apprentissage automatique et la classification. Des algorithmes sont développés pour identifier les formes géographiques spécifiques et extraire les informations pertinentes. Le projet comprend également une évaluation de la performance du système en utilisant des données de test.

I.2- PROBLEMATIQUE

L'analyse manuelle des données géographiques est une activité très fastidieuse et coûteuse car les données géographiques sont souvent très complexes et peuvent inclure une grande quantité d'informations, la collecte de ces données peut être coûteuse et prendre beaucoup de temps. La nécessité d'analyses précises et détaillées de ces données, la nécessité

de prise de décisions rapides et la gestion de zones géographiques vastes ou difficile d'accès sont les principaux problèmes liés au traitement de données géographiques.

I.3- OBJECTIFS DU PROJET

Ce projet a pour but d'aboutir à son terme à un système automatiques de traitement des données géographiques. Ces objectifs spécifiques sont :

- ✓ **Automatiser le traitement de données géographiques** pour réduire le temps et les coûts associés.
- ✓ **Améliorer la précision et la fiabilité des analyses géographiques**
- ✓ **Faciliter la prise de décisions** dans de nombreux domaines, tels que la gestion de l'environnement et la planification urbaine.
- ✓ **Réduire les coûts de collecte de données géographiques** en utilisant des données existantes, telles que des images satellites ou des cartes topographiques.
- ✓ **Surmonter les défis liés à la gestion de zones géographiques vastes ou difficiles d'accès** en utilisant des données existantes pour identifier et extraire des informations pertinentes.
- ✓ **Améliorer la cartographie** : Le système peut aider à identifier et donc aider dans la cartographie de manière plus précise les caractéristiques géographiques telles que les routes, les bâtiments, les cours d'eau, les zones boisées, les zones humides et bien d'autres.
- ✓ **Aider dans la planification urbaine** : Les administrations locales peuvent utiliser ce système pour planifier l'utilisation des terres en fonction des caractéristiques géographiques de la région.
- ✓ **Aider dans la surveillance environnementale** : Les autorités environnementales peuvent utiliser ce système pour surveiller l'évolution des forêts, des cours d'eau et des autres écosystèmes.

I.4- ACTEURS CIBLES DU PROJET

Les principaux acteurs cibles de notre projet sont :

- ✓ **Les décideurs politiques** : qui peuvent utiliser ce système pour prendre des décisions en matière de planification urbaine, de gestion de l'environnement, de surveillance de la sécurité nationale.
- ✓ **Les professionnels de la cartographie et de la géomatique** : qui peuvent utiliser un tel système pour automatiser le traitement de données géographiques et améliorer la précision et la fiabilité de leurs analyses.
- ✓ **Les chercheurs et les scientifiques** : qui peuvent utiliser les résultats de ce système pour mener des études et des recherches dans des domaines tels que la biologie, l'écologie et la géologie.
- ✓ **Les organisations gouvernementales et non gouvernementales** : qui peuvent utiliser les résultats d'un tel système pour mener des activités de surveillance et de gestion de l'environnement.
- ✓ **Les entreprises** : qui peuvent utiliser les résultats de ce système pour prendre des décisions en matière de développement immobilier et de transport, des décisions relatives à la localisation de leurs activités.

I.5- RESULTATS ATTENDUS

Le principal résultat attendu à l'issu de ce projet est un système de traitement automatiques de données géographique. Ce système sera donc capable de :

- ✓ Reconnaître une forme sur image satellite,
- ✓ Extraire cette forme,
- ✓ L'identifier,
- ✓ Et traiter les résultats obtenus.

I.6- ETAT DE L'ART

Le but essentiel de notre projet est de réaliser un système capable de reconnaître, d'extraire et d'identifier des formes identifiables sur des données géographique. Notre application devra permettre d'extraire un ensemble d'informations qui pourront être exploiter dans de nombreux domaines tels que le transport, la cartographie et la gestion de l'environnement pour par exemple faire des études statistiques sur de vaste zone non accessible (quantité de végétation, cours d'eaux).

Une étude similaire fut déjà menée au Cameroun en 2008 par une équipe de chercheur composée de **FOTSING Janvier, TALLA Narcisse, TONYE Emmanuel, et KANNA Thomas**. Le but de cette étude était de fournir une nouvelle approche pour l'analyse d'image **Radar à Synthèse d'Ouverture** sur une image provenant de la mangrove dans la région du Littoral. Pour ce faire la technologie qu'ils ont utilisées étaient basées sur un modèle **d'algorithme de classification non supervisée** et les résultats de leur travail est illustrée dans un article ayant pour titre **Classification non supervisée d'image RSO en vue de la cartographie des couverts végétaux de la mangrove littorale Camerounaise** publié sur le site [www.resarchgate.net](http://www.researchgate.net).

De tel système sont en constante évolution en raison des avancées technologiques récentes dans le domaine de l'intelligence artificielle et de la vision par ordinateur.

I.6.1- Techniques et approches utilisées

Voici quelques-unes des techniques et des approches actuelles utilisées dans ce domaine

- ❖ **Réseaux de neurones convolutifs** : Les CNN sont actuellement l'une des techniques les plus populaires pour la reconnaissance et l'extraction de formes géographiques à partir d'images. Les CNN sont des réseaux de neurones profonds qui apprennent à détecter automatiquement les caractéristiques pertinentes des images, telles que les formes, les couleurs et les textures. Les CNN ont été utilisés avec succès pour la détection de bâtiments, la classification de l'occupation des sols, la détection de cours d'eau ainsi que pour la segmentation d'images.
- ❖ **Traitements d'images satellitaires** : Les images satellitaires sont souvent utilisées pour la reconnaissance et l'extraction de formes géographiques en raison de leur couverture géographique globale et de leur haute résolution spatiale. Les techniques de traitement d'images peuvent être utilisées pour extraire des informations à partir de ces images, telles que l'occupation des sols, la végétation, les cours d'eau, les routes et les bâtiments. Les techniques de traitement d'images courantes comprennent la classification d'images, la segmentation d'images, l'analyse de texture, l'analyse de forme.
- ❖ **Analyse de données LiDAR** : Les données LiDAR sont des données de télédétection qui utilisent des lasers pour mesurer la distance entre l'émetteur et la surface terrestre. Les analyses de données LiDAR peuvent être utilisées pour générer des modèles 3D de la surface terrestre, ainsi que pour extraire des informations sur les caractéristiques du terrain. Les techniques courantes d'analyse de données LiDAR comprennent la

segmentation de nuages de points, la classification d'objets et la détection de changements.

- ❖ **Systèmes d'informations géographiques** : Les SIG sont des outils logiciels utilisés pour stocker, gérer, analyser et visualiser des données géographiques. Les SIG peuvent être utilisés pour intégrer des données géographiques provenant de différentes sources, telles que des images satellites, des données LiDAR ou même des cartes topographiques afin de faciliter la reconnaissance et l'extraction de caractéristiques géographiques particulières.
- ❖ **Techniques de segmentation d'images** : Les techniques de segmentation d'images sont utilisées pour diviser une image en régions homogènes en termes de texture, de couleur et de forme. Les techniques courantes de segmentation d'images comprennent la segmentation basée sur les seuils, la segmentation par croissance de régions et la segmentation par régions actives.
- ❖ **Deep Learning** : Le Deep Learning est une sous-catégorie de l'apprentissage automatique qui utilise des réseaux de neurones profonds pour apprendre automatiquement à partir de données. Le Deep Learning a été utilisé avec succès pour la reconnaissance et l'extraction de formes géographiques à partir de données d'images et de nuages de points LiDAR. Les techniques courantes de Deep Learning comprennent les CNN, les RNN et les GAN.

Ces techniques sont en constante évolution et continuent de s'améliorer grâce aux avancées technologiques récentes.

I.7- CHRONOGRAMME DES ACTIVITES

Afin de mieux structurer notre travail tout en restant cohérent dans nos objectifs, nous avons consigné l'ensemble de nos tâches dans le tableau suivant :

Tableau 1: Chronogramme des activités effectuées

TACHES	DUREE	DATE DE DEBUT	DATE DE FIN
Réception du thème de projet	//	07 Février 2023	//
Analyse du projet	02 semaines	07 Février 2023	21 Février 2023
Elaboration du cahier de charge	01 semaine	22 Février 2023	01 Mars 2023
Analyse des besoins	02 semaines	02 Mars 2023	15 Mars 2023

Acquisitions du <i>dataset</i>	03 semaines	15 Mars 2023	05 Avril 2023
Prétraitements des données	04 semaines	06 Avril 2023	06 Mai 2023
Entrainement du model	05 semaines	07 Mai 2023	14 Juin 2023
Intégration du système à l'application web	02 semaines	15 Juin 2023	28 Juin 2023
Dernière mise au point et test	02 semaines	29 Juin 2023	12 Juillet 2023
Rédaction du rapport	//	21 Février 2023	13 Juillet

CONCLUSION

Parvenu au terme de ce chapitre où nous avons présenté la problématique résolue par ce projet, les objectifs et les acteurs cibles, ainsi qu'une description de ce projet, dans la suite de notre travail nous allons présenter une analyse fonctionnelle et technique de conception.

CHAPITRE II : ANALYSE FONCTIONNELLE ET TECHNIQUE DE CONCEPTION

INTRODUCTION

Dans ce chapitre nous allons étudier en détail les concepts clés nécessaires pour la conception et la réalisation de notre système. Nous commencerons par présenter les fonctionnalités de notre système, puis de donner les détails techniques liés à leur implémentation et à la conception de notre système.

II.1- ANALYSE FONCTIONNELLE DU SYSTEME

II.1.1- Exigences fonctionnelles du système

Elles définissent ce que le système doit être capable de faire ou de fournir pour satisfaire les besoins des utilisateurs. C'est une spécification décrivant les fonctionnalités du système. Ici les principales fonctionnalités qui doivent être implémenter sont :

- **Reconnaitre et extraire les zones d'intérêts (classes) sur une image :** Le système doit être capable de détecter et reconnaître des zones sur une image satellitaire, telles que les zones constituées de bâtiments, de végétation, de cours d'eau et de sols
- **Traiter les données obtenues à des fin statistique :** Le système doit être capable de produire un diagramme pour visualiser les données en indiquant le pourcentage de chaque classe détecter sur l'image passé en entrée.

II.1.1- Exigences non fonctionnelles du système

Celle-ci décrivent comment le système doit fonctionner et quelles sont les qualités attendues de celui-ci. Les principales exigences non fonctionnelles de ce système sont :

- **Fiabilité :** il doit être capable d'assurer de manière continu le service attendu.
- **Précision :** Le système doit fournir des résultats de reconnaissance et d'extraction des zones géographiques avec une précision élevée pour garantir la fiabilité des données extraites.

- **Portabilité** : Il doit être facilement déployable sur d'autre environnements matériels ou logiciel nouveaux.
- **Maintenabilité** : le système doit être facilement utilisable et facilement maintenable.
- **Compatibilité** : il doit être facilement combinables à d'autres systèmes.

II.2- CONCEPTION TECHNIQUE ET FONCTIONNELLE DU SYSTEME

II.2.1- Acquisition d'un *dataset*

Afin de réaliser un modèle de segmentation sémantique d'image il est nécessaire d'avoir un *dataset* assez particulier.

- ***Dataset* en ML Classique**

Dans d'autre problème classique de ***supervised Learning*** (apprentissage supervisé) le ***dataset*** est sous forme de matrice contenant des scalaires. Ces scalaires sont de deux catégories on a en premier lieu les ***features*** puis la ***target***.

Les ***features*** désignent ici un ensemble de valeurs qui seront pris en paramètres dont dépendent les valeurs de notre ***target***

Les ***targets*** désignent les valeurs en résultantes des ***features***.

<i>y</i>	<i>x₁</i>	<i>x₂</i>	<i>x₃</i>
Prix	Surface m²	N chambres	Qualité
€ 313,000.00	124	3	1.5
€ 2,384,000.00	339	5	2.5
€ 342,000.00	179	3	2
€ 420,000.00	186	3	2.25
€ 550,000.00	180	4	2.5
€ 490,000.00	82	2	1
€ 335,000.00	125	2	2

Figure 1: Exemple de dataset Scalaire

La figure ci-dessus nous montre l'exemple d'un dataset. Les valeurs de **X** sont nos **features** et les valeurs de **Y** sont nos **targets**. Le but de ce type de modèle est d'étudier la relation entre les **features** et les **targets** afin de fournir une fonction sous la forme $f(x_1, x_2, \dots, x_n) = y$ qui permettra de prédire les valeurs de **Y** en sorties

- **Dataset en Deep Learning**

Le Deep Learning vient comme une sorte d'extension au Machine Learning. Il permet de faire les mêmes opérations que celle faites en Machine Learning mais sur une quantité significativement plus importante de données. Dans le cadre de notre projet notre dataset sera constituer de ce qu'on appelle **les tile images** et **les tile masks** comme illustre la figure ci-dessous



Figure 2: Image du dataset



Figure 3: Mask d'une image du dataset

- Une **tile image** : celle-ci désigne un morceau d'une image satellitaire de type **matrice** qui sera traiter afin de servir à entraîner notre modèle. Une **tile image** est en réalité un morceau d'une image satellitaire beaucoup plus grande cette technique consistant à découper les images satellitaires est nécessaire afin de faciliter le stockage, le traitement et la diffusion de ce type d'image.
- Une **mask image** : chaque tile image du *dataset* possède son **mask image**. Cette image est dite « masquer » ou « filtrer » le but de filtrer ici est d'éliminer les zones de l'images qui ne sont pas pertinentes afin de ne conserver que nos zones d'intérêts. Chaque zone d'intérêts est représentée par une couleur et en ce qui concerne ce projet nous en avons 06 au total

L'élaboration d'un telle *dataset* est assez complexe car il est nécessaire de photographier la zone en question avec l'aide d'un satellite ou d'un drone et ensuite il faut pouvoir extraire les zones d'intérêt ce qui est une tâche assez fastidieuse.

Notre *dataset* a été élaborer à partir de données recueillies dans la ville de **Dubaï au Emirates Arabes Unies** et qui est mis à la disposition de tous sur le site web **Kaggle.com**.

II.2.2- Traitement des images

II.2.2.a- Représentation d'une image en mémoire

Pour l'élaboration de modèle de segmentation sémantique d'image il est important de premièrement comprendre comment une image est stockée en mémoire. Il faut d'abord que l'on distingue les types d'images : Les images en échelle de gris et les images en couleur.

Dans le cas particulier des images en échelle de gris, chaque **pixel** est stocké sur un **octet** c'est-à-dire une valeur entre 0 et 255. La bibliothèque **opencv** de python nous permet de lire des images sous ce format avec la commande « cv2.imread('/path/image.jpg', 0) » comme l'illustre la figure suivante



- Dimension : (644, 797)
- Taille : 513268 pixels

Figure 4: Image en échelle de gris

Ce type d'image est représentée dans la mémoire de l'ordinateur sous forme d'une **matrice à deux dimensions**.

Nous avons aussi les images de couleur. La couleur ici est formée d'une association de valeur de trois couleurs fondamentales le **rouge, le vert et le bleu** (RGB). Chaque pixel ici au lieu d'avoir une simple valeur est constitué d'un vecteur de 03 valeurs donc chaque valeur est représentée sur 1 octet. C'est typiquement ce type d'images que sera utiliser pour l'entraînement de notre modèle.



- Dimension : (644, 797, 3)
- Taille : 1539804 pixels

Figure 5: Image en couleur

II.2.2.b- Pré-traitement d’images

Le **pré-traitement** des images est une étape primordiale pour la conception d’un modèle de segmentation sémantique d’images. Le but essentiel de cette étape est de pouvoir préparer les images à être utiliser pour entraîner notre modèle. Les images présentent dans le *dataset* n’ont pas toujours les mêmes dimensions or pour entraîner un modèle de ce type il est nécessaire que toutes les données en entrée soit de la même dimension appelé **patch size**.

➤ Fixation du patch size

Le **patch size** est la valeur de dimension que nous utiliserons pour notre modèle cette taille est généralement de 256×256 pixels mais elle dépend grandement de la résolution de l’image.

Pour des images de hautes résolution le patch size peut aller de 16×16 à 64×64 pixels.

Pour des images de moyennes résolutions le patch size peut aller de 128×128 à 512×512 pixels.

Il est important de noter que plus la résolution de l’image satellitaire est grande plus la taille des pixels sera réduite et par conséquent une petite zone de l’image contiendra une grande

quantité d'information c'est cela qui explique le fait que la taille du patch size est inversement proportionnelle à la résolution de l'image.

De façon général la segmentation sémantique permet d'attribuer une classe à chaque pixel de l'images c'est pourquoi il est nécessaire que le patch size soit aussi réduit afin de faciliter la classification de pixels. De plus il est nécessaire de trouver un certain équilibre en choisissant la valeur du patch size. Une patch size trop petit cela va grandement augmenter la charge de calcul et réduire la vitesse de traitement et choisir un patch size trop faible causera un manque de précision du modèle

Il est aussi important de noter à la fois les ***tile images*** et les ***mask images***

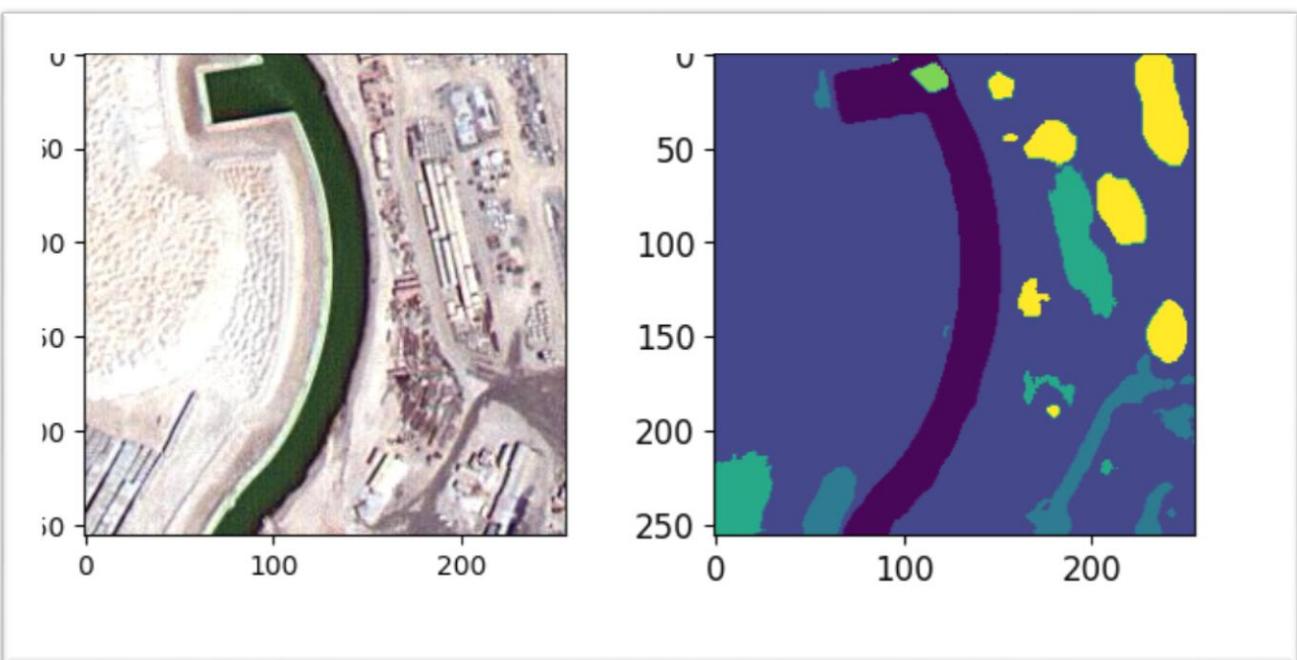


Figure 6: Images et mask au patch size

La figure ci-dessus nous montre le résultat que nous avons obtenu après avoir découper à la fois une ***tile image*** et sa ***mask image***. Ce tracé a été obtenu avec la bibliothèque **matplotlib** de python et nous montre parfaitement que chaque image est de la taille du patch size c'est-à-dire de 256×256 pixels.

➤ Classification des pixels

Cette opération est uniquement effectuer sur les ***tile mask*** de notre *dataset*. Lors du *preprocessing* des données il est crucial de pouvoir identifier le ***land marks*** (zones d'intérêt de notre modèle ou classe d'objet) et **associer chaque pixel de chaque tile image à une classe**

d'objet dans le cadre de ce projet nous identifierons 06 classes d'objet avec chacun ayant sa propre couleur en hexadécimale

```
'classes': [
    {"title": "Water", "shape": "polygon", "color": "#50E3C2", "geometry_config": {}},
    {"title": "Land (unpaved area)", "shape": "polygon", "color": "#F5A623", "geometry_config": {}},
    {"title": "Road", "shape": "polygon", "color": "#DE597F", "geometry_config": {}},
    {"title": "Building", "shape": "polygon", "color": "#D0021B", "geometry_config": {}},
    {"title": "Vegetation", "shape": "polygon", "color": "#417505", "geometry_config": {}},
    {"title": "Unlabeled", "shape": "polygon", "color": "#9B9B9B", "geometry_config": {}}], "tags": []
```

Figure 7: Classes identifiables

Un fois les classe identifier nous allons créer une nouvelle matrice qui contient **la valeur de l'indice de la classe de chaque pixel dans le fichier ci-dessus** pour bien comprendre cette étape on va prendre l'exemple d'une image M

Notre image M est une image en couleur ce qui signifie qu'elle peut être assimiler à un **tenseur d'ordre 3** (vecteur à 3 dimensions) notre image M est alors constituée de N pixel. Chaque pixel P est caractérisé par $P(l, c, v)$

$l = \text{indice sur ligne}$

$c = \text{indice colonne}$

$v = \text{vecteur } rgb(r, g, b)$

Si notre pixel fait par exemple partie du la classe « Water » une autre matrice M' sera générer et aura **l'indice de la classe de ce pixel P à cette position**

Ainsi à la fin de cette étape si on avait N images au départ, on aura également N matrices chacune caractérisant chaque image avec l'indice à la place de chaque pixel.

➤ Encodage **OneHot** Division du dataset

A cette étape d'avancement de réalisation notre *dataset* se divise en trois parties grandes parties

La partie A former de nos ***tile image***

La partie B former de nos ***tile mask***

Enfin la partie C constituer de nos matrices contenant la classe de chaque pixel pour chaque image de la partie B du *dataset*

- **Encodage *OneHot***

L'encodage *OneHot* est une méthode utilisée lors de la segmentation sémantique d'image afin de pouvoir plus facilement stocker la partie C du *dataset* en mémoire. En effet pour que la valeur de l'indice du pixel n'est pas d'incidence significatif lors de l'entraînement du modèle (par exemple la classe « water » à pour valeur « 0 » et la classe « vegetation » à pour valeur « 4 ») il est nécessaire de trouver une méthode qui permettra non seulement d'éviter le problème énoncé précédemment mais en plus qui rendra la tâche de traitement de l'ordinateur plus simple à réaliser c'est donc là l'importance de cette méthode d'encodage son principe de fonctionnement est le suivant :

Premièrement nous allons identifier le nombre total de classe dans notre dataset (6 dans notre cas particulier) puis nous allons créer un vecteur contenant 6 valeurs soit à « 0 » ou à « 1 » **la seul valeur à « 1 » de ce vecteur sera celle qui correspond à l'indice de la classe qui lui est associer dans notre dataset.**

Par exemple si nous avons un pixel P qui appartient à la classe « vegetation » par exemple (indice 4) ce pixel sera représenté par un vecteur V sous la forme V [0, 0, 0, 0, 1, 0]. Cet encodage est rendu possible grâce à la fonction « **to_categorical** » de la bibliothèque « **tensorflow.keras.utils** » illustrer comme suit

```
In [41]: labels_categorical_dataset = to_categorical(labels, num_classes=total_classes)
```

Figure 8: Encodage OneHot

Cette fonction prend 02 paramètres : le premier est le nombre total de classe présent de notre *dataset* (*num_class*) et la second est le *dataset* précédent contenant les valeurs de l'indice de chaque classe (*labels*). A la fin cette étape chaque image sera sous la forme suivante :

```
In [42]: labels_categorical_dataset[0]
Out[42]: array([[[0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   ...,
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.]],

   [[0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   ...,
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.]],

   [[0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   ...,
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0., 0.]],

   ...]
```

Figure 9: Image après encodage OneHot

- **Division du dataset**

Dans le Machine Learning en général il est toujours nécessaire de séparer le *dataset* de départ en 04 parties comme suit :

X_train : c'est la partie des ***tile image* du dataset** qui va être utiliser pour entraîner notre modèle

Y_train : c'est la partie des ***tile mask*** qui sera utiliser pour entraîner notre modèle

X_test : il s'agit de la partie des ***tile image*** sur lesquelles nous allons tester notre modèle

Y_test : qui représente ici nos données de validation c'est à partir de ces ***tile mask*** que nous allons voir à quel point notre modèle épouse la réalité et est précis

Il existe aussi une fonction qui nous permet de réaliser cette séparation du *dataset* du non de « **train_test_split** » de la bibliothèque « **sklearn.model_selection** ». Cette opération est réalisée comme suit :

```
x_train, x_test, y_train, y_test
= train_test_split(master_training_dataset, labels_categorical_dataset, test_size
= 0.15, random_state = 100)
```

La fonction « **train_test_split** » prend ici 04 paramètres

master_training_dataset : représente notre *dataset* contenant les ***tile image***

labels_categorical_dataset : représente notre *dataset* contenant les ***tile mask***

test size : représente la proportion entre les images contenues dans la X_test et dans la X_train

random_state : indique que les images doivent être prise de façon aléatoire dans le *dataset*

Une fois notre *dataset* diviser nous sommes désormais prêts à entamer l'entraînement du modèle.

II.2.1.d- Notion de réseaux de neurones convolutif et architecture U-net

❖ *Notion de réseaux de neurones à convolution*

Le monde visuel se caractérise par un ensemble de propriétés à savoir

- Une invariance par translation : par exemple une image avec peut avoir un élément à l'intérieur qui occuper différentes positions mais qui représente un même objet comme le montre la figure suivante :

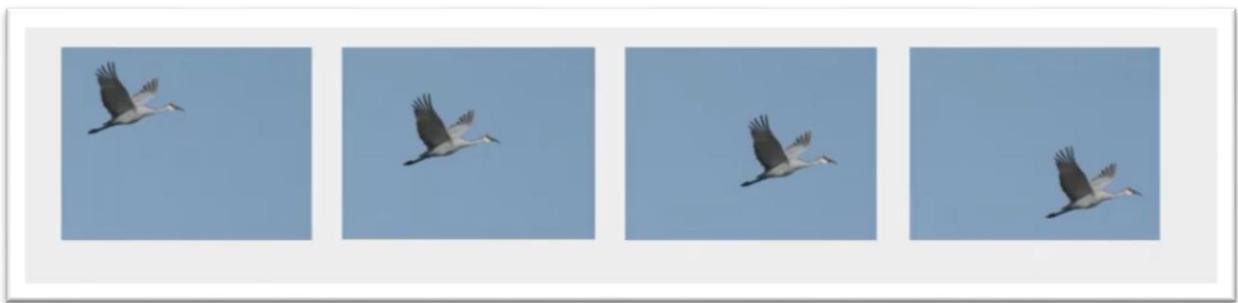


Figure 10: Image avec translation

- La localité : qui désigne le fait que les pixels qui sont très rapprochés entre eux peuvent appartenir à une même entité.
- Une hiérarchie spatiale : c'est-à-dire qu'un visage par exemple qui est une forme complexe est constituer d'entité beaucoup plus simple (oreilles, yeux, nez).

Ces spécificités rendent le monde visuels inadaptés pour les modèles d'apprentissage classique notamment pour la détection d'éléments locaux sur une images car en effet ce type de modèle traite en entrée les données dans leurs globalités et n'essaye pas de détecter ou d'extraire des formes de l'images.

C'est pour s'affranchir de ces limites imposées par le monde du visuel que sont apparues **les réseaux neurones à convolution**.

Les réseaux de neurones à convolution sont un type de réseau de neurones spécialisé dans le traitement d'image qui au lieu de traiter l'image dans son entiereté sont capables d'apprendre à reconnaître certaines particularités de l'image afin d'optimiser leurs traitements.

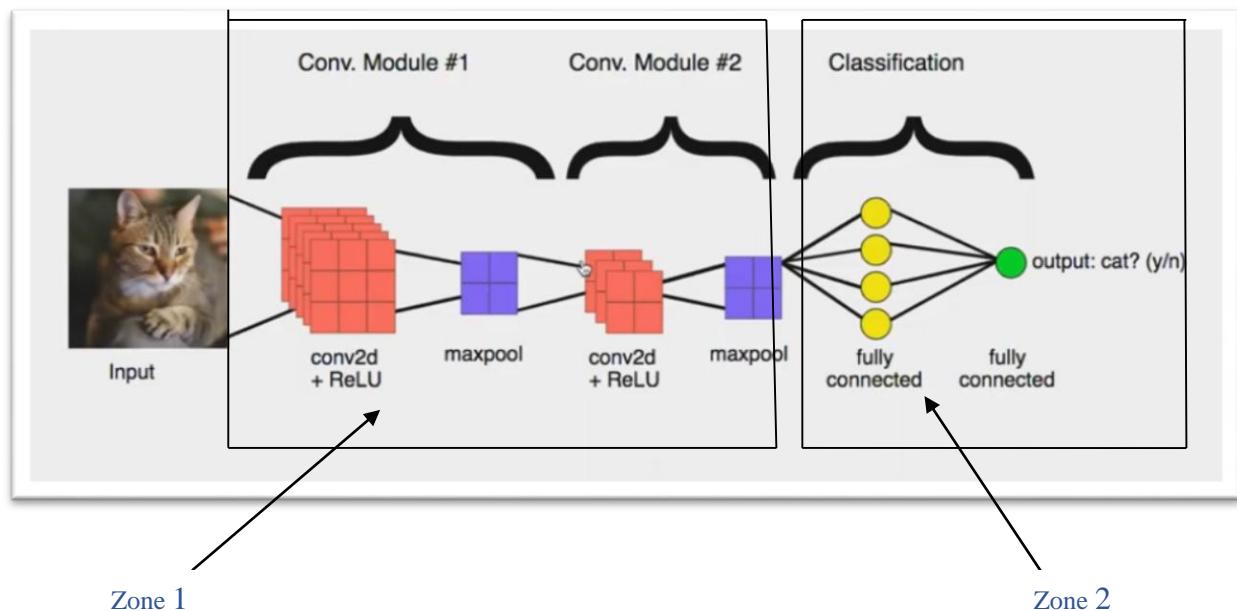


Figure 11: Aspect général d'un réseau de neurones à convolution

La figure ci-dessus nous présente l'aspect général d'un réseau de neurones à convolution.

- ✓ La « zone 1 » représente la **zone de convolution**. Celle-ci est constituée d'un certain nombre de couche de **convolution** et de couche dite de **pooling**.

La **convolution** un outil mathématique très largement utilisé dans le traitement d'images ce qui explique le fait que les réseaux de neurones à convolutions sont parfaitement adaptés pour cette discipline. La convolution agit comme un **filtrage**. On définit une taille de fenêtre qui va se balader à travers toutes l'image (Il faut garder à l'esprit qu'une image est perçue comme un tableau à plusieurs dimensions)

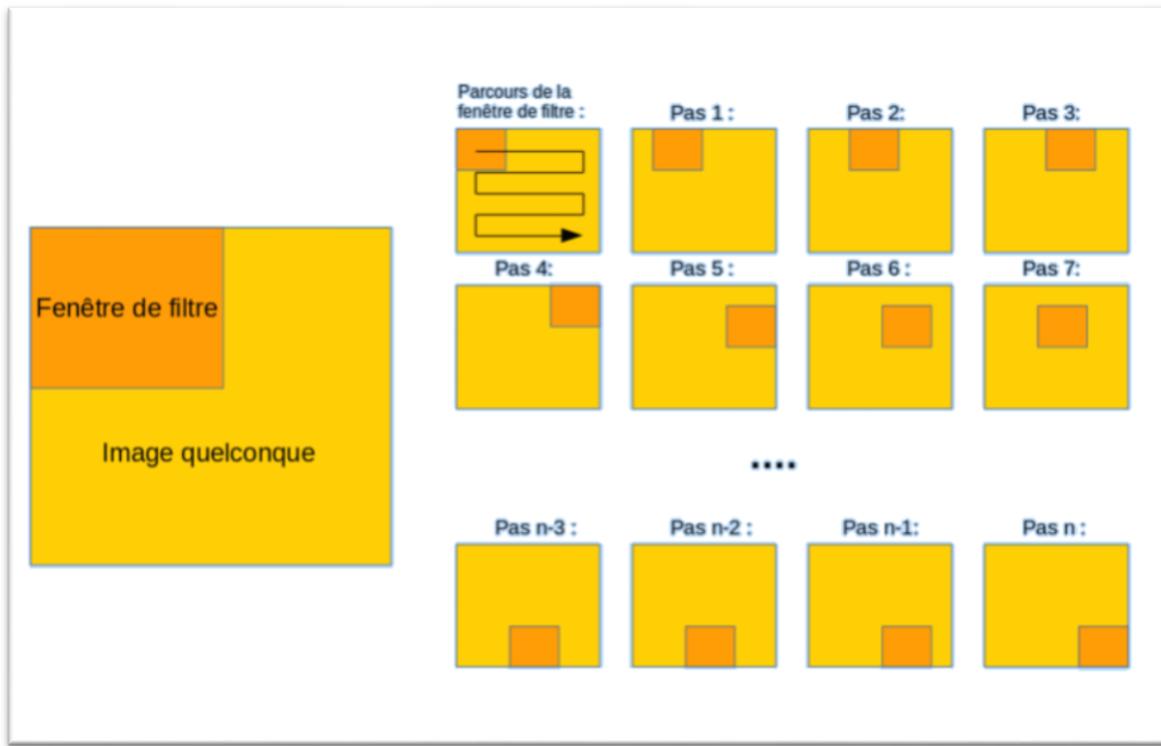


Figure 12: Convolution

Les paramètres possibles d'une couche de convolution sont :

- **La dimension de l'image** (Image 2D ou 3D)
- **La taille de la fenêtre** (*matrice 2×2 dans l'exemple de la zone 1*)
- **Le nombre de filtres** en effet une même couche peut avoir plusieurs filtres et chacun d'entre eux produira sa propre sortie.
- **Le pas** définit la vitesse d'évolution du filtre sur l'image. Il est généralement égal à 1.
- **Le padding** spécifie un ensemble de lignes et de colonnes à ajouter sur l'image d'entrée afin de contrôler les valeurs de la dimension de l'image en sortie.

Le Pooling est une opération qui consiste à prélever uniquement le caractère dominant sur l'image issue de la convolution comme l'illustre la figure ci-dessous

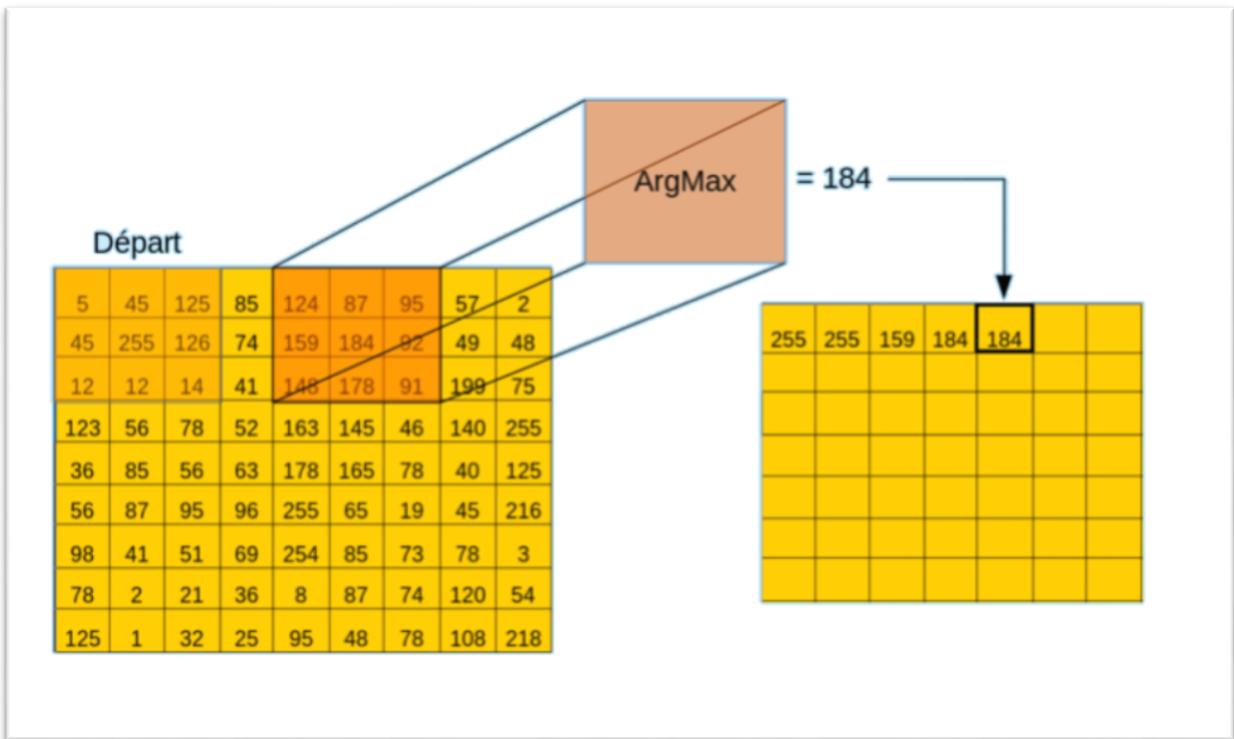


Figure 13: Max Pooling

Dans cet exemple le *pooling* prend en sortie le pixel avec la plus grande valeur cependant d'autre type de *pooling* existent.

- ✓ La « zone 2 » représente la couche de classification. Elle est constituée de couches dites « entièrement connecter (Couche dense) ». La fonctionnalité de cette zone est semblable à celle des réseaux de neurones classiques c'est-à-dire de pouvoir effectuer la classification des données qu'elle reçoit en entrée

❖ *Architecture U-net*

Les opérations effectuer dans la cadre du traitement d'images ont pour objectif de réaliser principalement les tâches que sont : **la détection d'objet** sur une image, **la segmentation sémantique** une image et **la segmentation par instance** d'une image. Les résultats attendus à l'issue de ces opérations sont illustrés par la figure ci-dessous

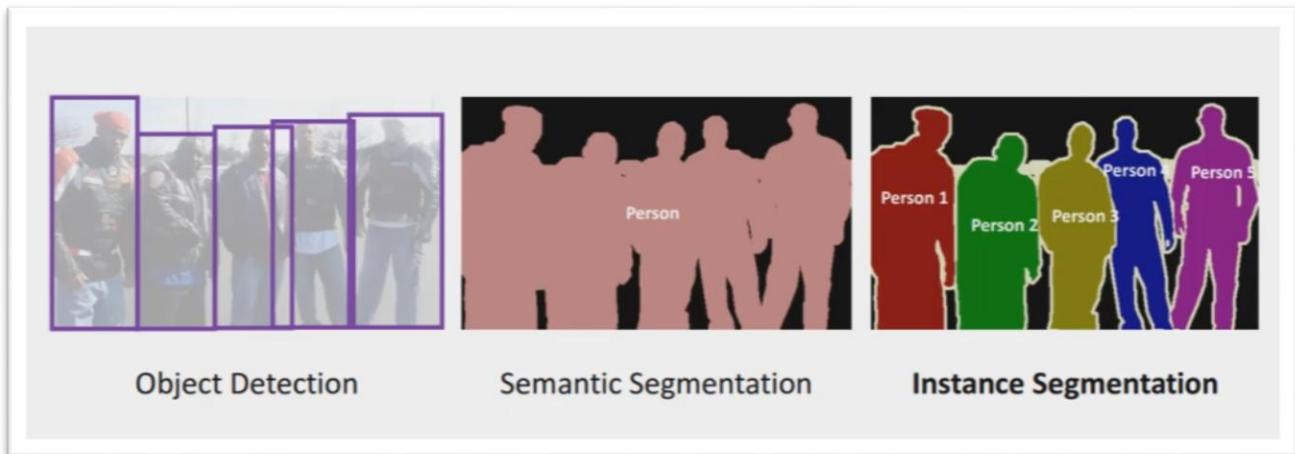


Figure 14: Résultats attendus à l'issu de traitement d'image

Dans le cadre de notre projet nous allons réaliser **une segmentation sémantique et multi-classes d'image**. Pour réaliser une telle tâche il est nécessaire que les différentes couches de **convolution** et de **Pooling** du modèle respectent une certaine architecture et, l'architecture la plus adaptée est **l'architecture Unet** qui a été spécialement conçue pour réaliser avec la plus grande précision possible la segmentation sémantique d'images.

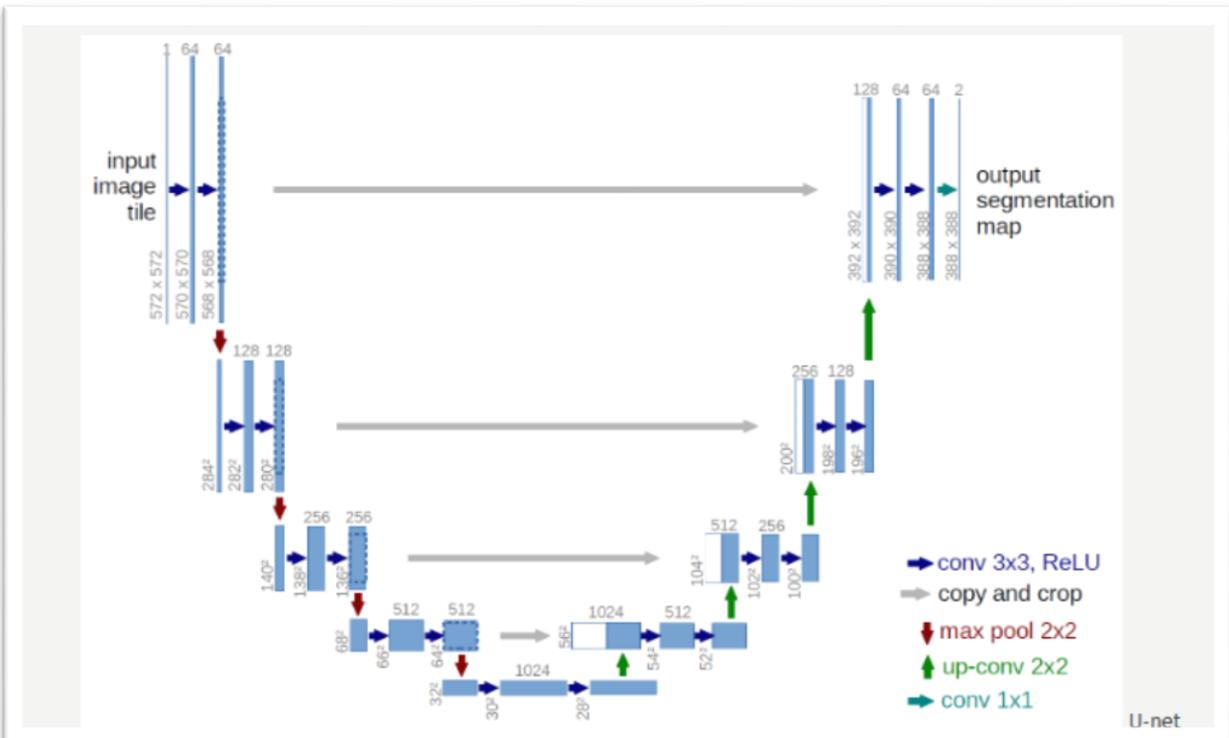


Figure 15: Architecture Unet

L'architecture Unet a été proposée en 2015 par **Olaf Ronneberger, Phillip Fischer et Thomas Brox** de l'université de **Freiburg** en Allemagne. Cette architecture a été proposer à la base pour l'analyse de données biomédicales en particulier la segmentation sémantique d'images de microscope électronique mais depuis lors **Unet** est devenu une architecture beaucoup plus populaire et trouve son application dans divers autres domaines à l'instar de la vision par ordinateur et l'imagerie satellitaire.

Cette architecture tire son nom de sa forme caractéristique en forme de « **U** » et elle décrit une suite de couche de **convolution**, de **MaxPooling** et de **déconvolution** que doivent subir les images en entrée afin de servir à entraîner le modèle.

II.3- INTERACTIONS ENTRE LES ACTEURS ET LE SYSTEME

II.3.1- Acteurs et cas d'utilisation du système

Le principal acteur de notre système est l'utilisateur. Les différents cas d'utilisation de notre système sont les suivants :

- Sélectionner une image
- Traiter une image
- Télécharger une image

Les fonctionnalités du système du point de vue utilisateur ainsi que les actions qu'ils peuvent effectuer sont représentées par la figure suivante :

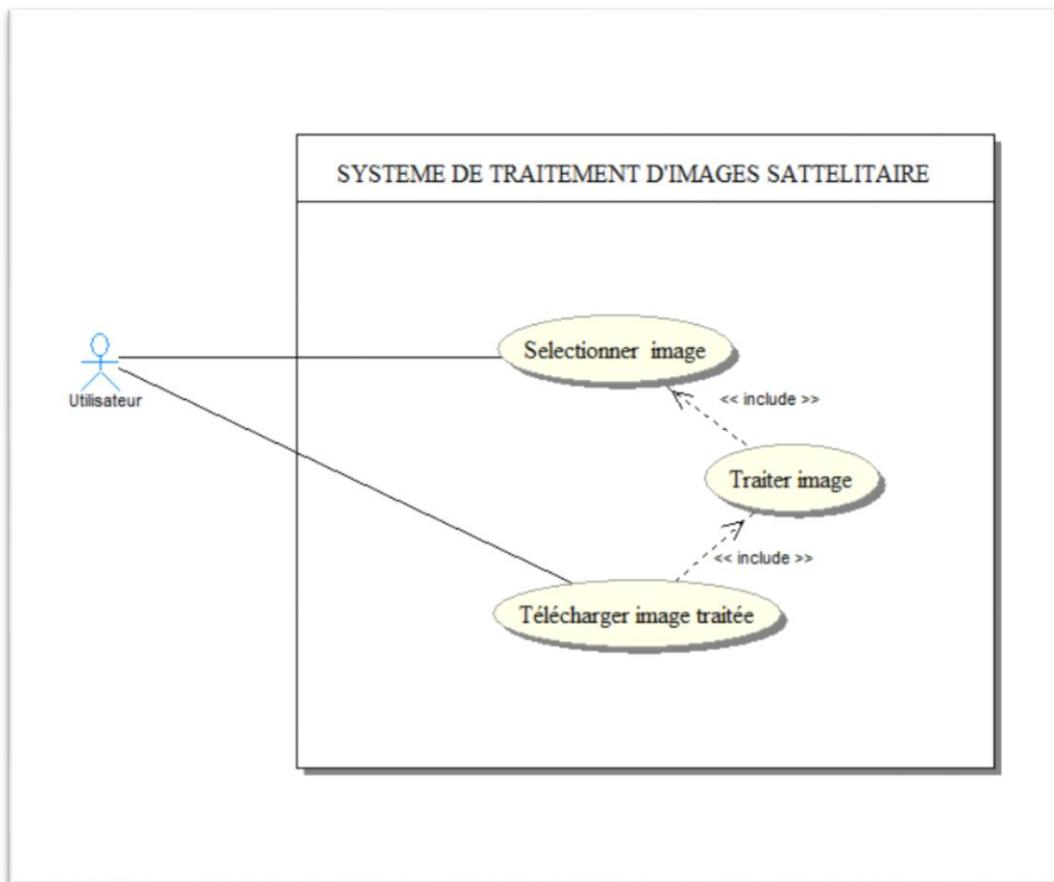


Figure 16: Diagramme de cas d'utilisation

II.2.2- Description textuelle des cas d'utilisation

Le tableau suivant décrit de façons plus simple et explicite le cas d'utilisation principal de notre système : **Traiter image**

Cas d'utilisation	Traiter image
Objectifs	Ce cas d'utilisation a pour objectif de reconnaître, d'extraire et d'identifier les formes identifiables sur une image satellitaire
Acteurs	Utilisateur
Précondition	Sélectionner l'image
Responsables	FOKO, NGUETSA, TSAGUE
Date	20/03/2023
Version	1.0

Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'image 2. Puis clique sur le bouton « Process image » 3. Le système rogne l'image 4. Puis affiche les différents patchs obtenus 5. Ensuite prédit chaque patch et affiche le résultat de la prédiction ainsi que les statistiques obtenues 6. Enfin reconstitue l'image prédictive à partir des patchs prédits et affiche cette image ainsi que les statistiques globales obtenues
Scenario alternatif	<ol style="list-style-type: none"> 1. Erreur l'image choisie est invalide 2. Le cas d'utilisation reprend à l'action 1 du scenario nominal
Postcondition	Aucune

II.3- COMPORTEMENTS INTERNES ET DIFFERENTS ETATS LORS DU TRAITEMENT D'UNE IMAGE

II.3.1- Comportement interne du système

Le comportement du système durant le processus de traitement interne d'une image peut être illustrer par le diagramme ci-dessous :

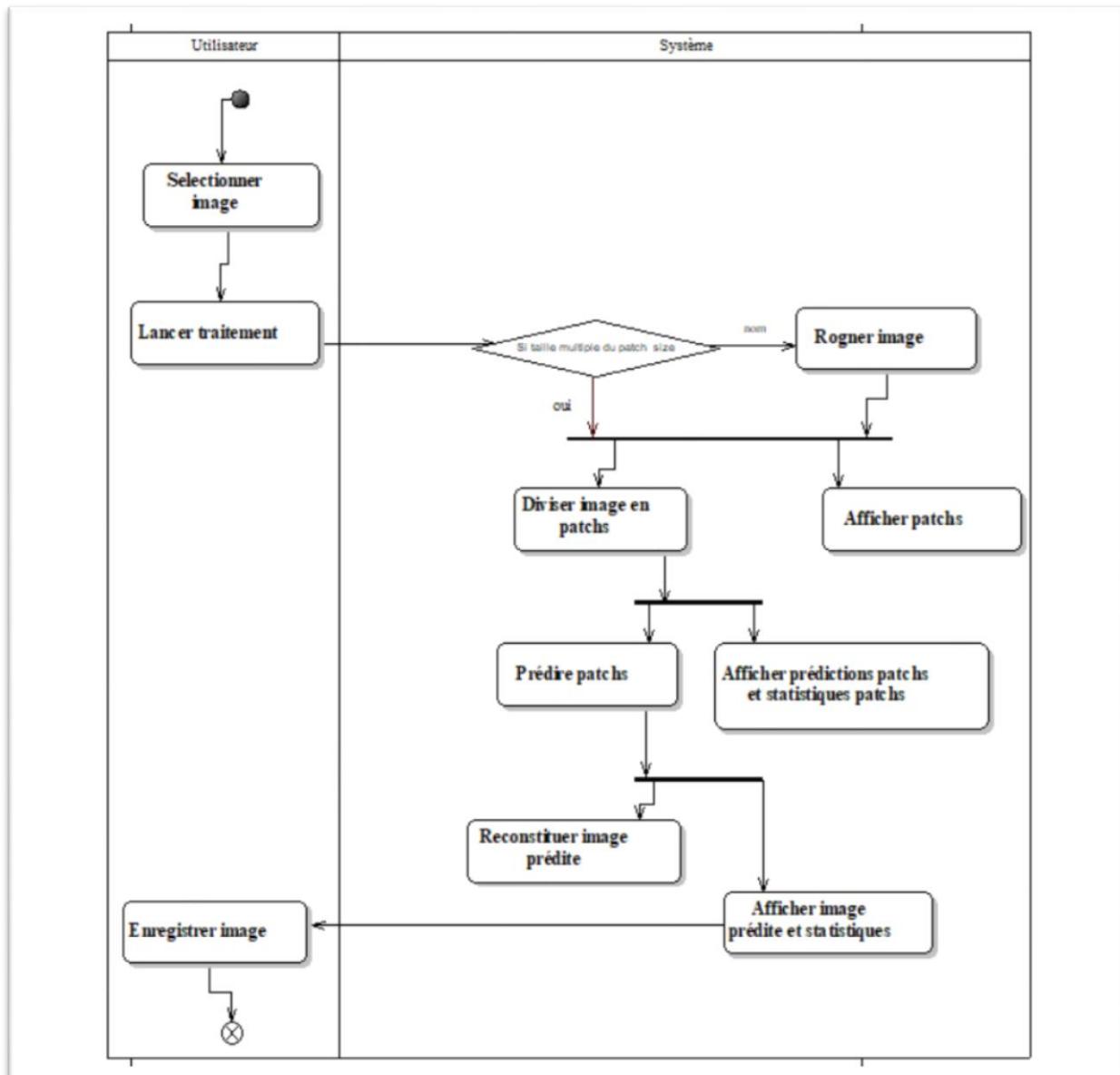


Figure 17: Diagramme d'activité : traiter une image

II.3.2- Différents états d'une image lors de son traitement

Pendant le traitement d'une image, il est nécessaire que les données d'entrées qui est l'image à traiter traverse un ensemble d'état tel qu'illustré par le diagramme ci-dessous :

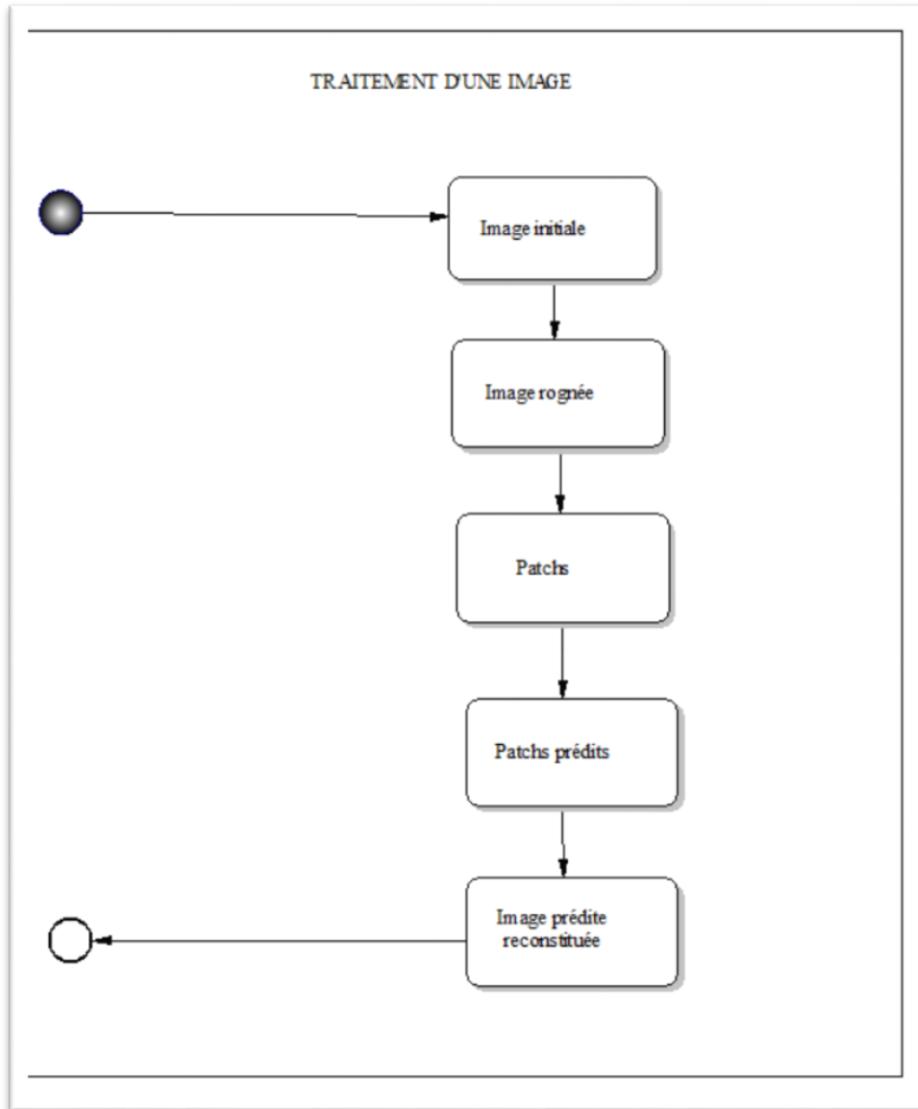


Figure 18: Diagramme d'état transition

Nous constatons donc que notre image lors de son traitement par le système traverse 5 états avant de pouvoir être exploitable par l'utilisateur.

CONCLUSION

Dans ce chapitre nous avons relevé les différents besoins fonctionnels et non fonctionnels de notre système et nous avons présenté les différentes étapes et techniques de conception liées à la réalisation de notre système. Nous allons présenter dans la suite de notre travail les différents outils et technologies utilisés pour implémenter notre système ainsi que les résultats obtenus.

CHAPITRE III: IMPLEMENTATION ET RESULTATS

INTRODUCTION

Ce chapitre a pour objectif majeur de présenter le produit final. C'est la phase d'entraînement de notre modèle et son déploiement sur une application web en utilisant des technologies spécifiques. Il est composé de trois parties, la première partie présente les différentes technologies utilisées, la seconde l'environnement de développement et l'environnement de production, et la dernière partie concerne les principaux résultats obtenus.

III.1- TECHNOLOGIES UTILISEES

Dans le cadre de la réalisation de ce projet nous avons utilisé les technologies spécifiques telles que **TensorFlow** et **Keras**, ainsi que des plateformes telles que **WandB** et **Hugging Face**.

III.1.1- TensorFlow



TensorFlow est une bibliothèque open source de Google pour l'apprentissage automatique et le traitement des données. Elle est utilisée pour la création et l'entraînement de modèles de ML, y compris les réseaux de neurones convolutifs, les réseaux de neurones récurrents et les réseaux de neurones profonds. Elle nous a permis de définir l'architecture, d'entrainer, d'évaluer, de sauvegarder et de charger notre modèle de réseaux de neurones convolutif de segmentation d'images.

III.1.2- Keras

Keras est une bibliothèque open source de haut niveau pour l'apprentissage automatique qui fournit une interface simple pour la construction de modèles de ML. Elle est compatible avec plusieurs moteurs de calcul de base, notamment TensorFlow et Theano, et est souvent utilisée pour les tâches de vision par ordinateur, de traitement du langage naturel et de prédition de séries temporelles. Grâce à son API couplé à TensorFlow nous l'avons utilisé pour le prétraitement de nos données d'apprentissage c'est-à-dire la

normalisation des données, la mise en forme des images et la transformation de données non structurées en données structurées ; la construction de notre modèle en utilisant ses fonctions telles que : ***MaxPooling2D*** pour ajouter des couches de ***pooling***, et ***Dense*** pour ajouter des couches entièrement connectées. Elle nous a également aidé à configurer notre modèle en définissant le nombres de couches, la taille des filtres de convolution, le taux d'apprentissage et les fonctions d'activations ; puis d'entrainer notre modèle grâce à ses fonctions telles que ***compile*** pour compiler le modèle avec une fonction de perte et une fonction d'optimisation, ***fit*** pour entraîner le modèle sur les données d'entraînement et de validation; enfin d'évaluer notre modèle en utilisant des métriques telles que l'exactitude, la précision et le rappel.

III.1.3- WanDB



Abréviation de "Weights and Biases", est une plateforme de gestion expérimentale pour la science des données et l'apprentissage automatique. La plateforme permet aux utilisateurs de suivre, visualiser et partager les résultats de leurs expériences d'apprentissage automatique de manière collaborative. Elle nous a permis de visualiser les résultats d'apprentissage de notre modèle notamment l'évolution du coefficient de ***jaccard***, de la précision et la perte sur les données d'apprentissage et de validation.

III.2- ENVIRONNEMENT DE DEVELOPPEMENT

Le développement de notre système a été subdivisé en deux grandes parties à savoir l'entraînement du modèle et son intégration sur une application web. Nous allons présenter ici l'environnement de développement utilisé pour l'entraînement de notre modèle et la création de notre application web.

Nous avons utilisé la version premium de la plateforme **Google Colab** pour l'entraînement de notre modèle ainsi que la création de notre application web.



est une plateforme de développement de ML basée sur le cloud qui permet aux utilisateurs de créer, de partager et de collaborer sur des projets de ML. Ces principales fonctionnalités sont :

- **Environnement de développement** : Google Colab fournit un environnement de développement basé sur Jupyter Notebook qui permet aux utilisateurs d'écrire et d'exécuter du code Python dans leur navigateur web. Il est équipé de toutes les bibliothèques couramment utilisées en ML telles que TensorFlow, Keras et PyTorch.
- **Accès au GPU et TPU** : Google Colab permet aux utilisateurs d'accéder gratuitement à des ressources de calcul haute performance telles que les GPU et les TPU. Cela permet aux utilisateurs de former leurs modèles de ML plus rapidement et plus efficacement.
- **Stockage de données** : Les utilisateurs peuvent stocker leurs données sur Google Drive, Google Cloud Storage ou localement sur leur ordinateur. Colab permet également de charger facilement des ensembles de données à partir de sources publiques telles que Kaggle et GitHub.
- **Collaboration** : Google Colab permet aux utilisateurs de collaborer sur des projets de ML en temps réel. Les utilisateurs peuvent partager des notebooks avec d'autres personnes et travailler ensemble sur le même notebook. Les utilisateurs peuvent également commenter et discuter du code à l'intérieur du notebook.
- **Sauvegarde des notebooks** : Les notebooks créés dans Google Colab sont automatiquement sauvegardés sur Google Drive. Les utilisateurs peuvent également télécharger leur notebook au format **.ipynb** pour le sauvegarder localement ou le partager avec d'autres personnes.

III.2.1- Environnement matériel

Pour des raisons de performances nous avons utilisé la version pro de Google Colab qui donne l'accès aux ressources matérielles suivantes :

- RAM : 25 Go
- CPU : Xeon 2,30 GHz à 2,80 GHz avec 8 cœurs virtuels
- Disque : 166.8 Go
- GPU : NVIDIA A100 Tensor Core et NVIDIA V100 Tensor Core.

III.2.2- Environnement logiciel

Google Colab Pro fournit un environnement logiciel complet et puissant pour l'apprentissage automatique et l'analyse de données, avec de nombreuses bibliothèques et outils populaires préinstallés telles que :

- **Python** : Google Colab Pro prend en charge Python 2 et Python 3
- **Bibliothèques de données et d'apprentissage automatique** : Google Colab Pro est préconfiguré avec de nombreuses bibliothèques de données et d'apprentissage automatique, notamment NumPy, Pandas, Matplotlib, TensorFlow, Keras, PyTorch, Scikit-learn, et bien d'autres.
- **Outils de développement** : Google Colab Pro propose des outils de développement pratiques tels que **Jupyter Notebook** qui a été notre principal éditeur, qui permettent de créer des notebooks interactifs, ainsi que des éditeurs de texte et des terminaux pour travailler avec du code.

En plus de ces outils et bibliothèques offerts par Google Colab, nous avons également utilisé la bibliothèque de développement **Gradio** qui nous a permis de créer notre application web.



est une bibliothèque open-source pour créer des interfaces utilisateur interactives pour les modèles d'apprentissage automatique. Il permet de créer rapidement des interfaces utilisateur pour leurs modèles d'apprentissage automatique sans avoir à écrire de code d'interface utilisateur personnalisé à partir de zéro.

Gradio est compatible avec de nombreux *Framework* d'apprentissage automatique populaires tels que TensorFlow, PyTorch, Scikit-learn, Keras et bien d'autres encore. Il propose également une interface utilisateur conviviale pour tester les modèles à partir d'exemples de données préchargés ou en entrant des données personnalisées.

III.3- ENVIRONNEMENT DE PRODUCTION

Après avoir entraîné notre modèle et l'intégrer à une application web, nous avons déployé cette application sur la plateforme **Hugging Face**.



Hugging Face

Est une entreprise de technologie basée à New York qui se concentre sur les technologies de traitement du langage naturel. Elle propose une plateforme open-source pour développer, entraîner et déployer des modèles d'apprentissage en NLP, ainsi que des API pour accéder à ces modèles. Elle est compatible avec de nombreux Framework populaires tels que PyTorch et TensorFlow. Elle offre un plan gratuit et des plans payants en fonction des ressources fournies. Dans notre cas nous avons utilisé le plan gratuit qui donne accès aux ressources matériels suivantes :

- Deux processeurs virtuels
- 16 Go de RAM

III.4- EVOLUTION DE L'APPRENTISSAGE DU MODELE

La figure suivante illustre les différentes courbes d'évolution de l'entraînement de notre modèle, notamment l'évolution du coefficient de **Jaccard**, de la **précision** et la **perte** sur les données d'apprentissage et de validation issu de la plateforme **WanDB** :

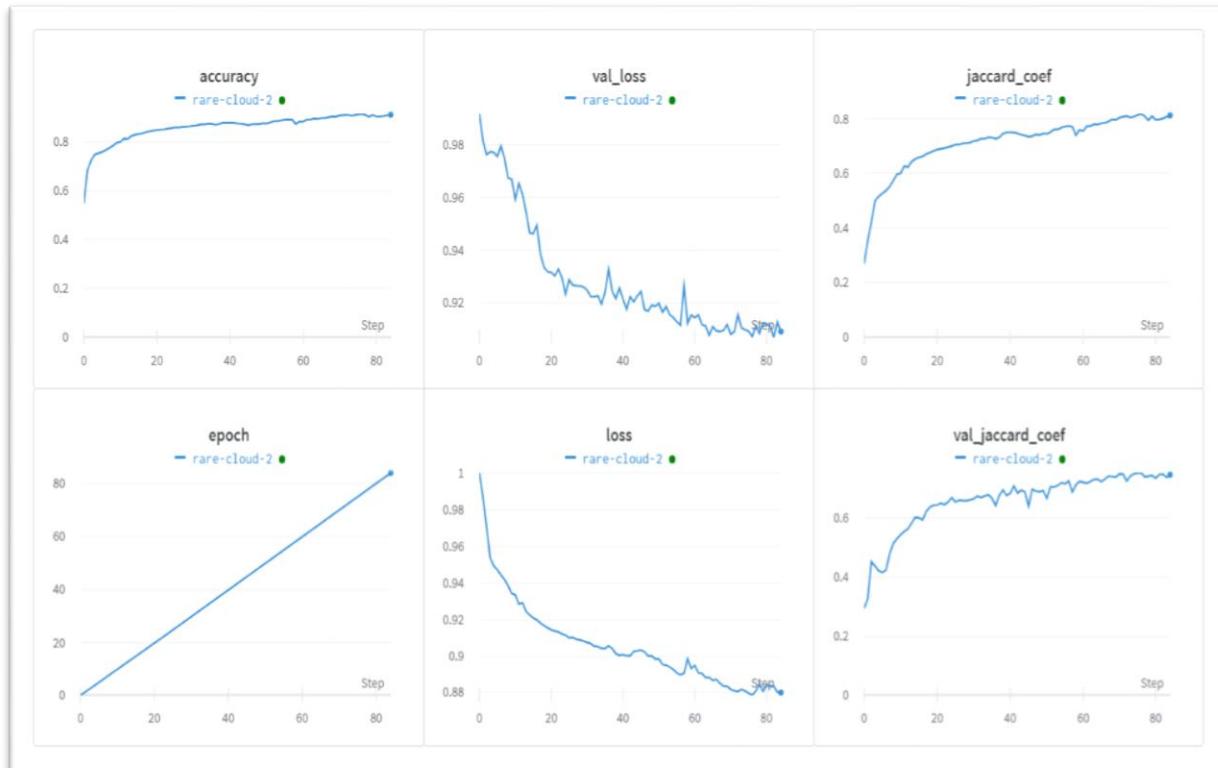


Figure 19: Evolution de l'apprentissage du modèle

- ✓ La première courbe « *jaccard_coef* » représente l'évolution du coefficient de Jaccard ou coefficient d'IoU sur les données d'entraînement. Le coefficient de Jaccard est utilisé pour évaluer la qualité de la segmentation en comparant la zone de chevauchement entre la segmentation prédictive et la segmentation réelle.
- ✓ La deuxième courbe « *val_jaccard_coef* » représente l'évolution du coefficient de Jaccard sur les données de validation.
- ✓ La troisième courbe « *val_loss* » représente la perte sur les données de validation. C'est une mesure utilisée pour évaluer la différence entre les prédictions du modèle et les valeurs réelles de la sortie souhaitée.
- ✓ La quatrième courbe « *epoch* » représente l'évolution du nombre d'époques. Nous avons entraîné notre modèle sur 85 époques pour une précision maximale dans la prédiction et pour éviter l'*overfitting* (cas où les performances du modèle sur l'ensemble de données d'entraînement sont améliorées au prix de moins bonnes performances sur des données non vues pendant l'entraînement, telles qu'un ensemble de données de test d'exclusion, de validation ou de nouvelles données).
- ✓ La sixième courbe « *val_accuracy* » représente l'évolution de la précision sur les données de validation. C'est une mesure qui permet d'évaluer la performance du modèle en termes de précision des prédictions, c'est-à-dire le pourcentage de prédictions correctes sur l'ensemble des prédictions effectuées sur les données de validation.
- ✓ La dernière courbe « *accuracy* » représente l'évolution de la précision sur les données d'entraînement. C'est une mesure qui permet d'évaluer la performance du modèle en termes de précision des prédictions, c'est-à-dire le pourcentage de prédictions correctes sur l'ensemble des prédictions effectuées sur les données d'entraînement.

La figure suivante illustre l'évolution des pertes sur les données d'entraînement et les données de validation :

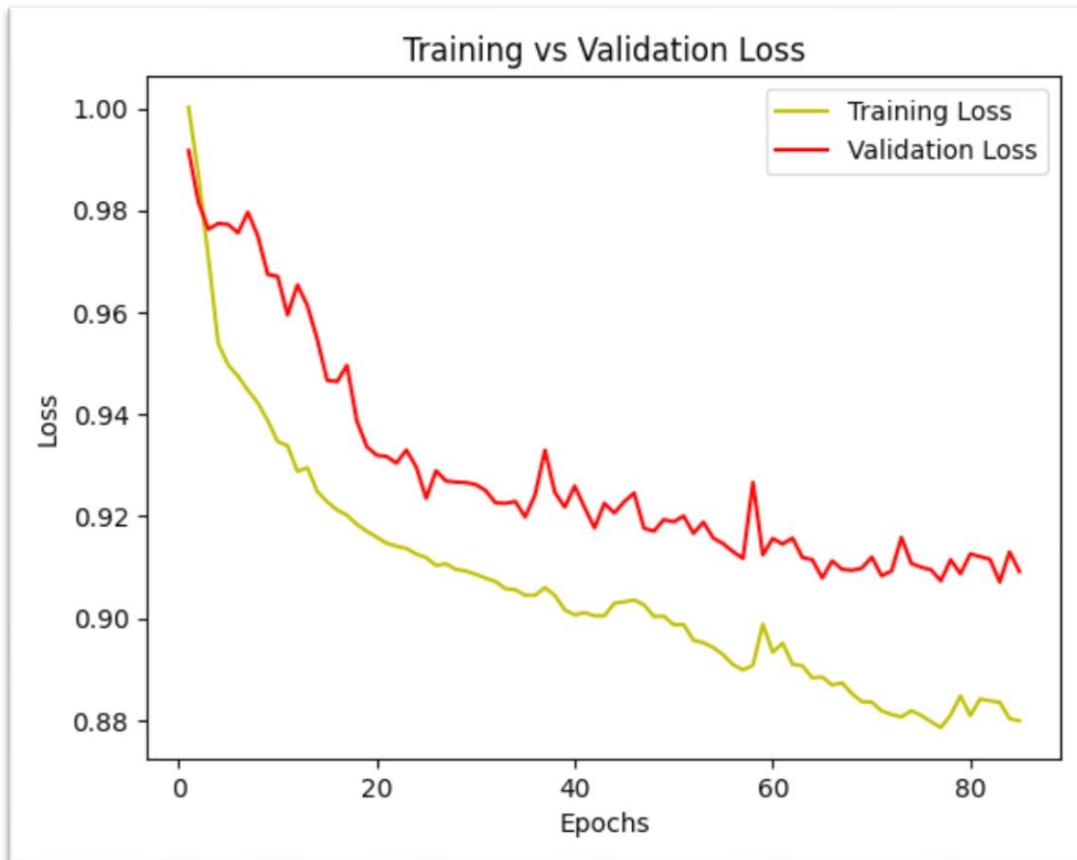


Figure 20: Evolution des pertes sur les données d'entraînement et de validation

On constate ici que les pertes sur les données d'entraînement évoluent de façons décroissantes et se rapproche des pertes sur les données de validations. Ce qui signifie qu'au fur et à mesure les prédictions faites par le modèle se rapproche des valeurs réelles de la sortie souhaitée.

La figure suivante illustre l'évolution du coefficient de Jaccard sur les données d'entraînement et les données de validation :

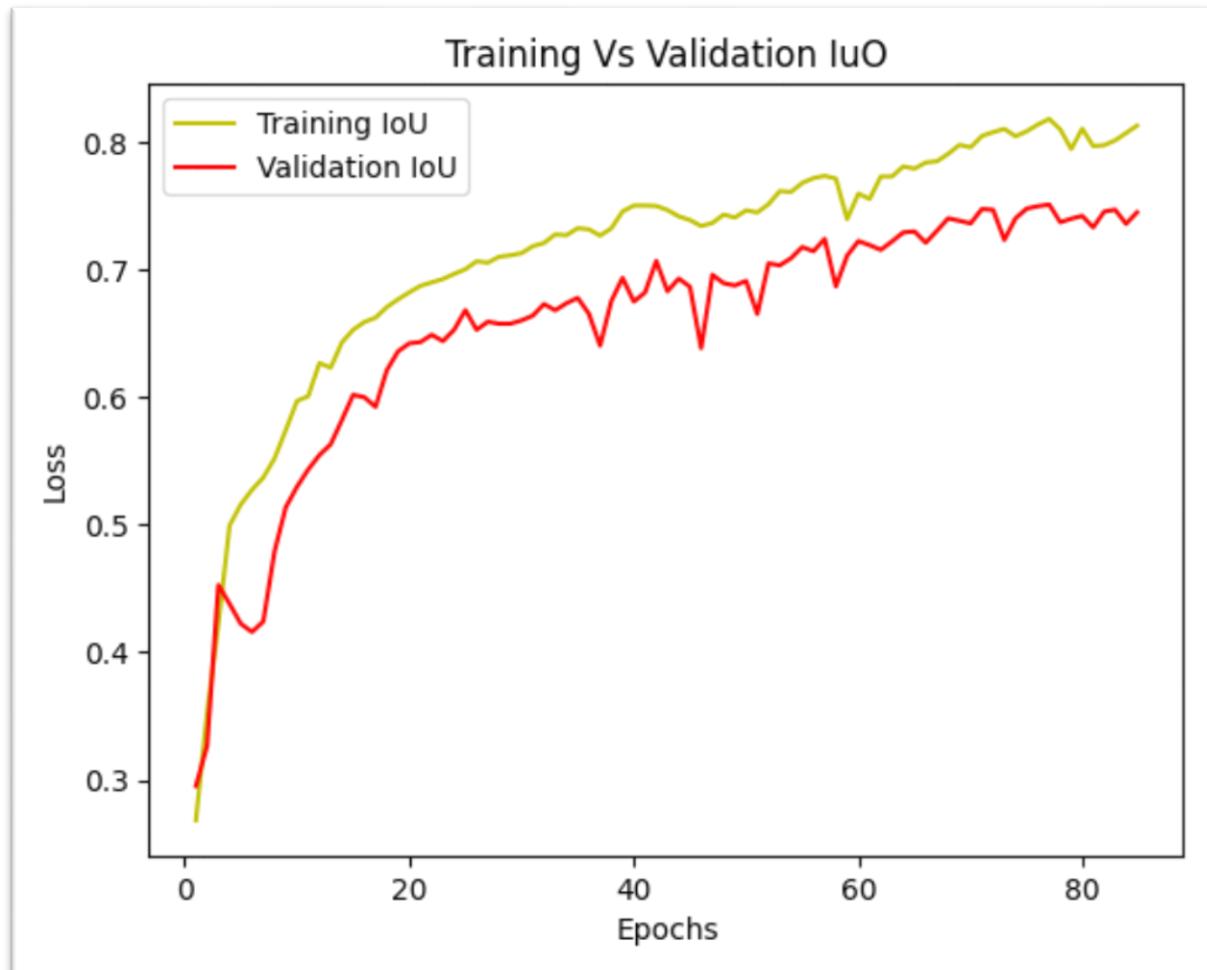


Figure 21: Evolution du coefficient de Jaccard sur les données d'entraînement et de validation

On constate ici l'évolution croissante du coefficient de *Jaccard* sur les données d'entraînement et les données de validation qui représente en effet l'évolution de la précision de prédiction du modèle.

Nous avons utilisé les proportions de données suivantes pour l'entraînement de notre modèle : 70 % pour l'apprentissage de notre modèle et 30% pour le test.

III.5 RESULTATS OBTENUS

III.5.1- Test du model

Après l'apprentissage de notre modèle nous avons effectués des tests avec l'image de taille 2149x1480 pixels suivante :

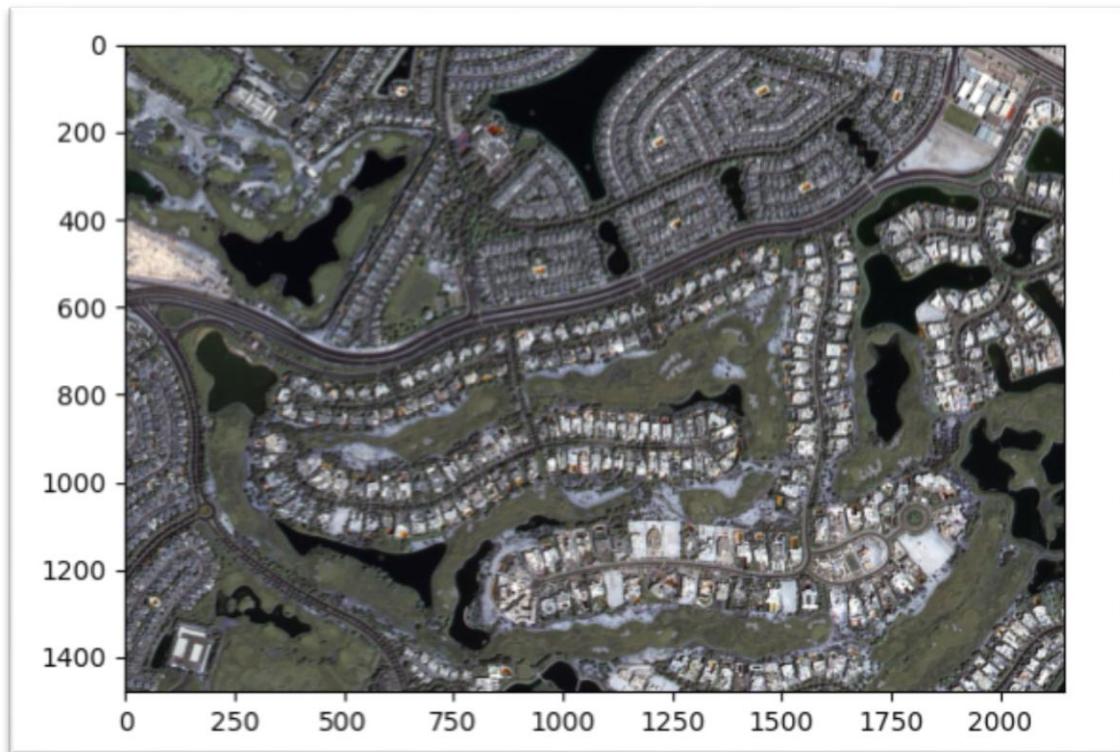


Figure 22: Image de test : ville de Dubaï

Tout d'abord l'image est rognée pour avoir une taille qui est multiple de 256×256 et divisée en patch de petites images de taille 256×256 . La figure suivante illustre les différents patchs obtenus à l'issu de cette étape :



Figure 23: Différents patchs obtenus (40)

Après avoir divisé l'image en patchs, ces différents patchs sont prédits par le modèle et les résultats obtenus sont les suivants :

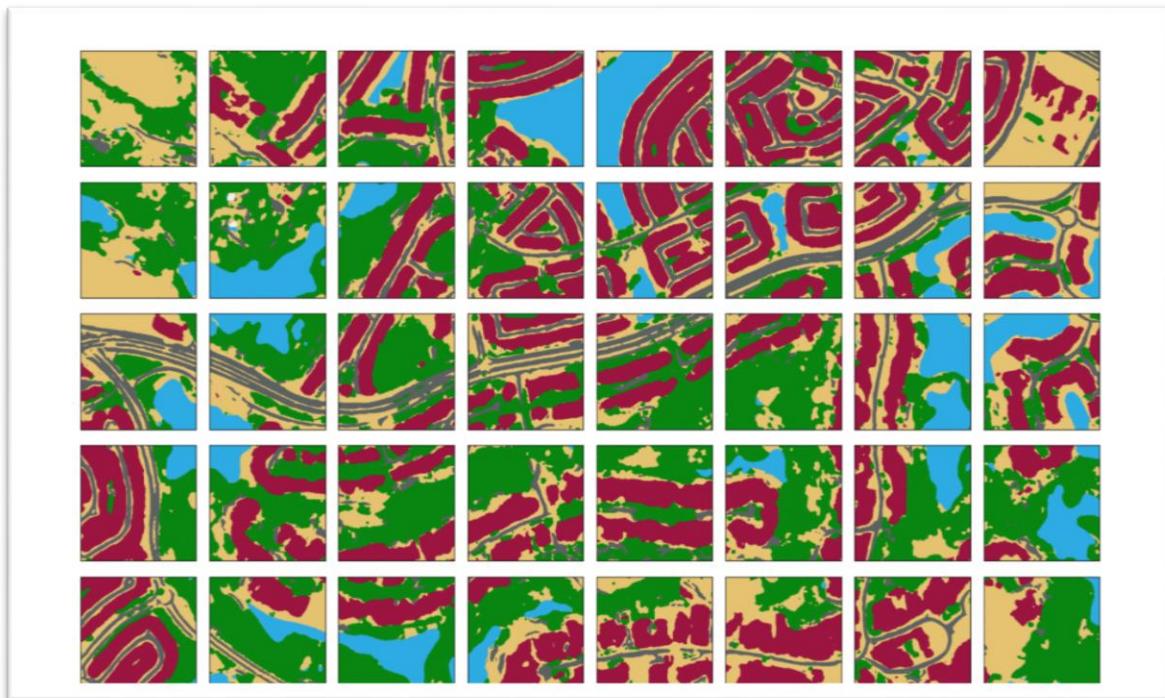


Figure 24: Résultat de la prédiction des différents patchs

Enfin l'image globale de prédiction est reconstituée à partir de ces différents patchs prédits et on obtient l'image suivante :

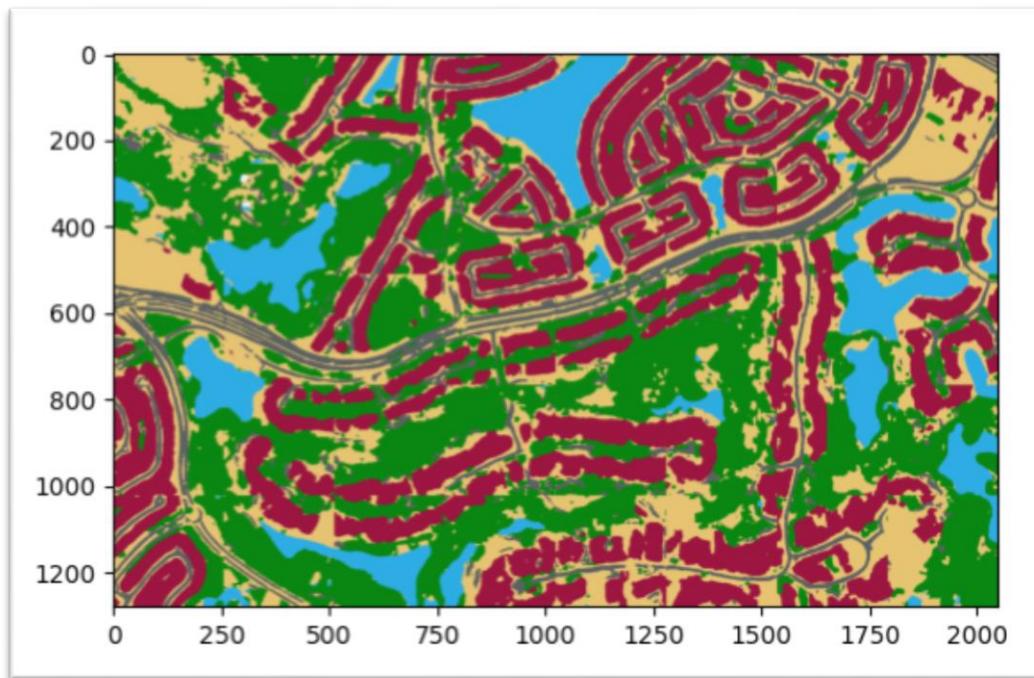


Figure 25: Image globale prédite

On peut donc constater que sur chaque patch prédit le modèle a pu reconnaître, extraire et identifier des formes présentes en donnant une couleur spécifique pour chaque forme identifiée. Ces différentes couleurs ont les significations suivantes :

Végétation (Vegetation)

Terre (Land)

Bâtiments ou construction (Building)

Non labelisé ou formes inconnues (unlabeled)

Eau (Water)

Route (Road)

III.5.2- Application web

Nous avons intégré notre modèle à une application web pour faciliter son utilisation.

La figure suivante présente l'interface d'accueil de notre application :

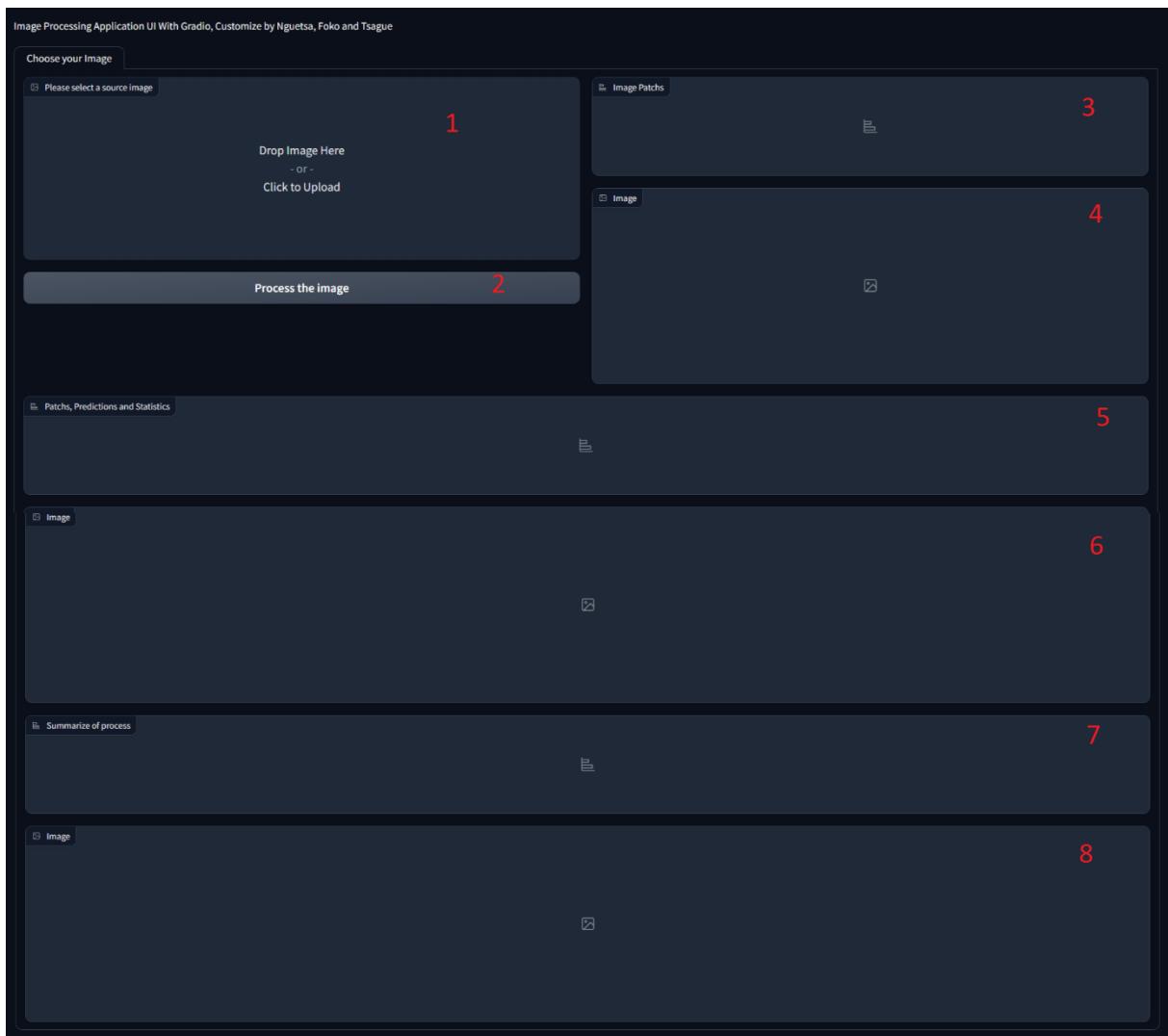


Figure 26: Interface de l'application web

- Le cadre permet à l'utilisateur de choisir ou de glisser et déposer l'image qu'il souhaite traiter [1]. Lorsque l'image est sélectionnée, elle s'affiche dans ce cadre.
- Le bouton [2] permet à l'utilisateur lorsqu'il a sélectionné l'image de lancer le processus de traitement de l'image. Lorsque l'utilisateur clique sur ce bouton, l'image est traitée suivant les différentes étapes décrites plus haut.
- L'ensemble des patchs issus de l'image rogner est affiché dans le cadre [4], ensuite les prédictions effectuées sur chaque patch ainsi que le diagramme de proportion sont affichées dans le cadre [6] et enfin l'image original, l'image rogner, l'image globale prédite ainsi que le diagramme de proportion globale sont affichées dans le cadre [8].
- Les cadres [3], [5] et [7] contiennent les titres des figures qui sont affichées.

Après traitement d'une image on obtient les résultats suivants :

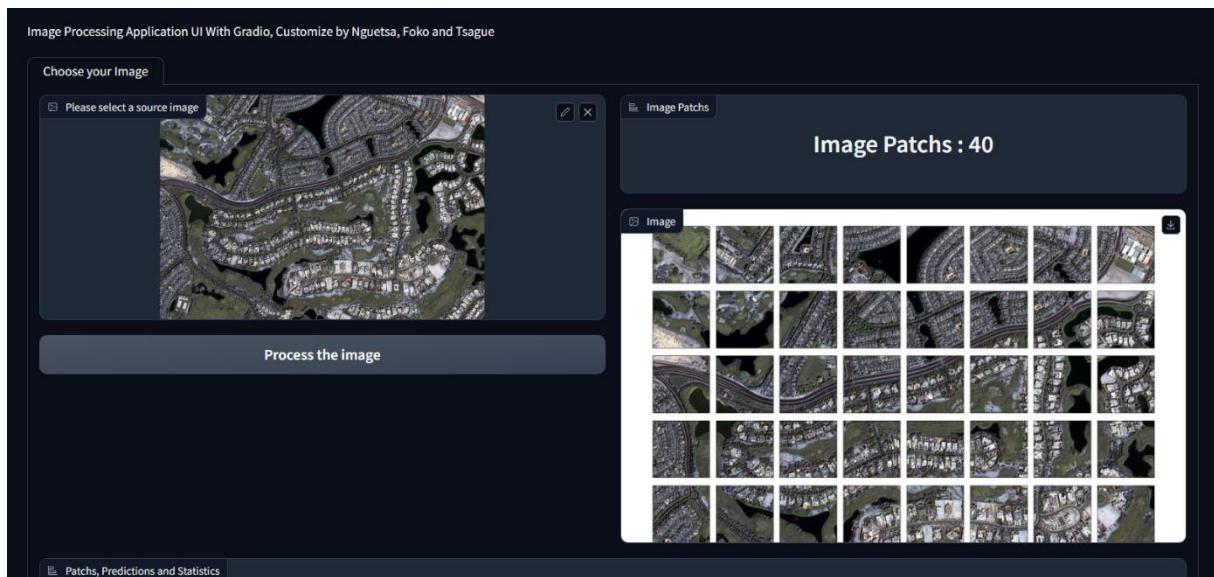
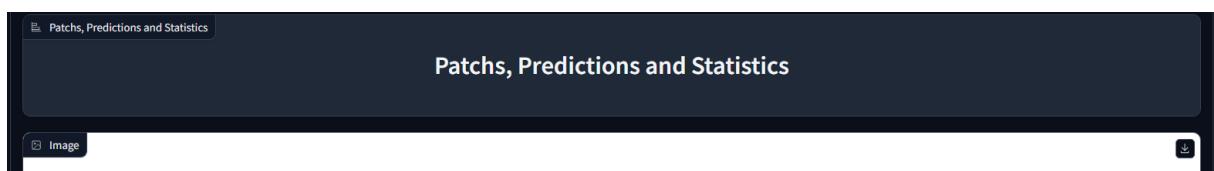


Figure 27: Patches issus de l'image rogner

Comme nous l'avons décrit plus haut on a sur cette figure l'image initial qui doit être traiter et l'ensemble des patchs dont 40 patchs pour cette image. Ces patchs sont donc prédis l'un a la suite de l'autre et on obtient donc le résultat suivant :



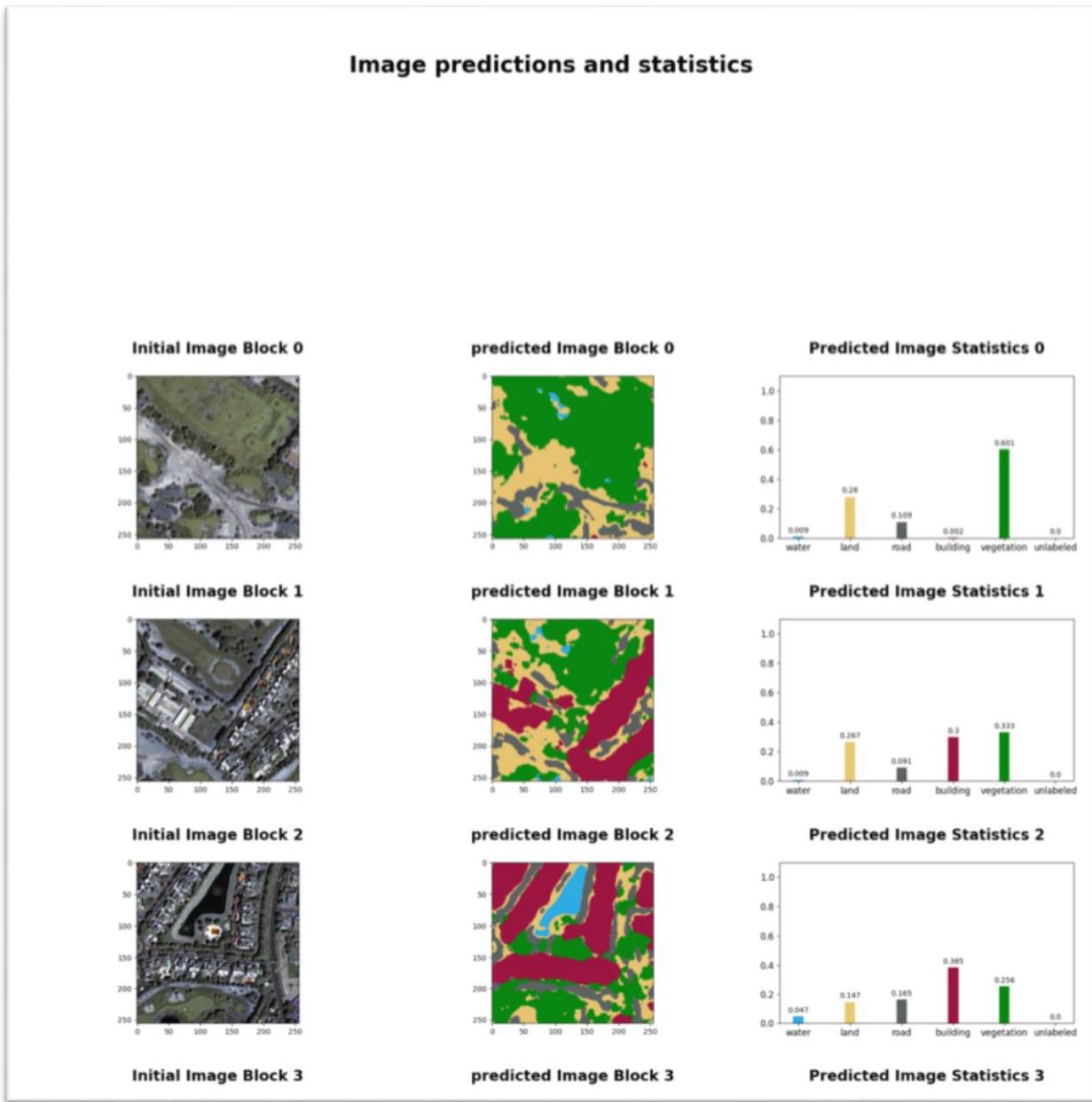


Figure 28: Patchs predictions and statistics 1

Vous trouverez la suite de ces résultats de prédictions des patchs en annexe.

Pour chaque patch on a la prédition correspondante ainsi qu'un diagramme en bâton qui illustre le pourcentage d'occupation de chaque classe sur le dit patch. Pour le premier patch par exemple on a 0.009 % d'eau, 0.28 % de sol, 0.109 % de route, 0.002 % de construction, 0.401 % de végétation et 0 % de formes inconnues, ce qui fait donc un total d'environ 1.

Les différents patchs prédits, l'image globale prédictée est reconstituée à partir de ceux-ci et on obtient le résultat suivant :

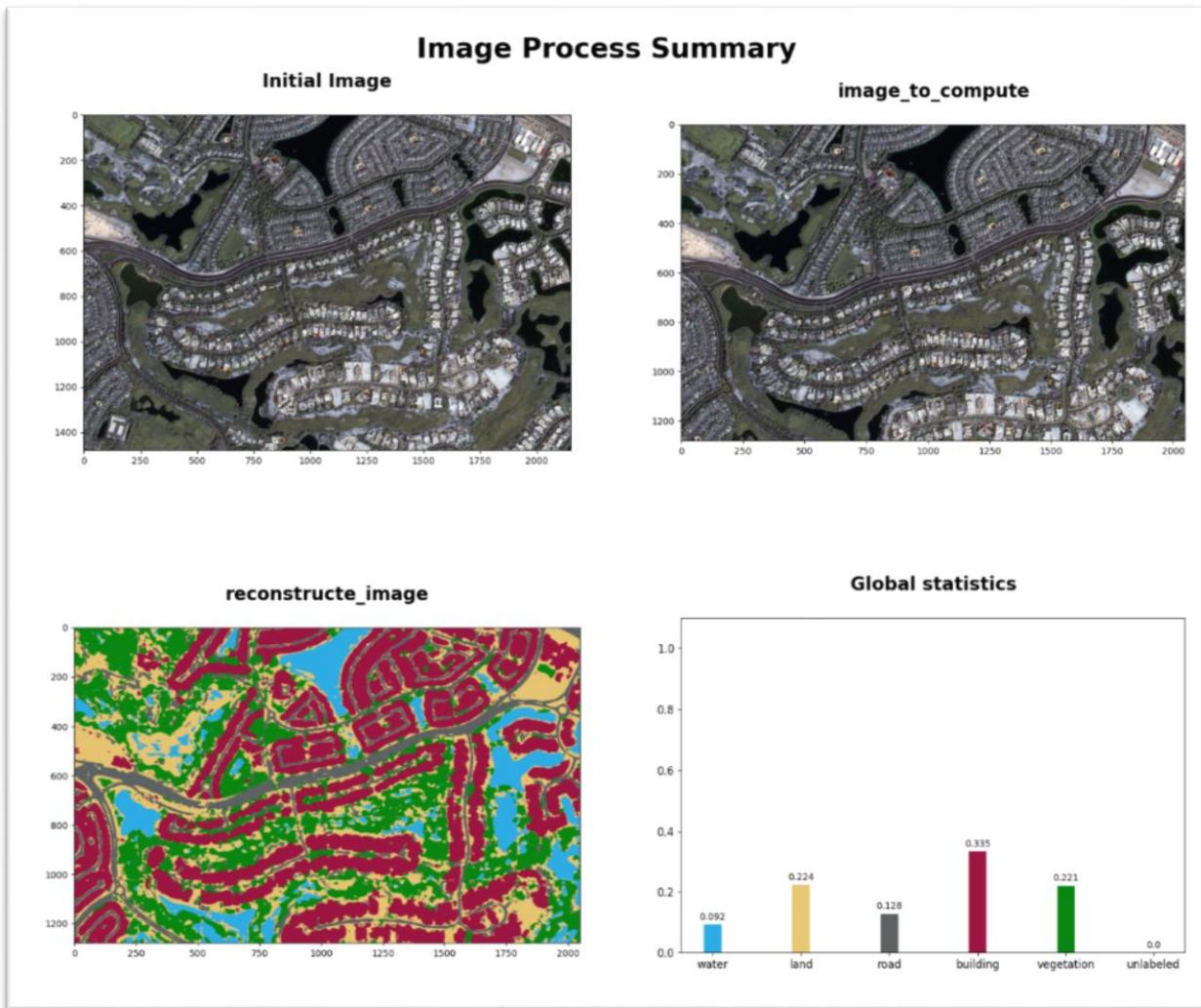


Figure 29: Summarize of the process

- La première image sur cette figure représente l'image sélectionnée par l'utilisateur, la deuxième image représente l'image qui est finalement traiter par l'application. Cette deuxième image est obtenue après avoir rogner l'image initial pour avoir une taille multiple de 256x256 pixels.
- La troisième image représente l'image rogné prédite qui est reconstituer à partir des différents patchs prédits à l'étape précédente.
- Enfin le digramme de battons illustre le pourcentage d'occupation globale de chaque classe sur l'image prédite. On a donc pour cette image 0.093 % d'eau, 0.224 % de sol, 0.1 % de route, 0.307 % de construction, 0.276 % de végétation et 0 % de formes inconnues, ce qui fait un total d'environ 1.

III.6- APPLICATION DES RESULTATS

Les résultats issus de cette application peuvent être utilisés dans plusieurs domaines pour aider à la prise de décisions notamment :

- ✓ **Agriculture** : Les données de pourcentage de végétation peuvent aider les agriculteurs à suivre l'état de croissance des cultures ;
- ✓ **Urbanisme** : Les données de pourcentage de bâtiments peuvent être utilisées pour cartographier les zones urbaines et pour planifier le développement urbain en identifiant les zones à forte densité de population et les zones où il y a une demande pour plus de logements ou d'infrastructures ;
- ✓ **Environnement** : Les données de pourcentage d'eau peuvent être utilisées pour suivre les changements climatiques et les impacts sur les ressources en eau. Elles peuvent également être utilisées pour identifier les zones à risque d'inondation et pour planifier les mesures de prévention et de réduction des risques ;
- ✓ **Gestion des ressources naturelles** : Les données de pourcentage de sol peuvent être utilisées pour cartographier les sols et pour planifier les activités agricoles. Les données de pourcentage de végétation peuvent également être utilisées pour suivre la déforestation et la restauration des écosystèmes.

CONCLUSION

Les outils et technologies présenté dans ce chapitre nous ont permis d'implémenter notre système et d'obtenir les résultats que nous avons présenté ci-dessus. Comme tout projet lors de la réalisation de ce projet nous avons été confronté à plusieurs difficultés que nous présenterons dans la suite de ce travail ainsi que des suggestions d'amélioration de notre système actuel.

CHAPITRE IV: DIFFILCUTES ET SUGGESTIONS

INTRODUCTION

Dans ce chapitre nous allons tout d'abord présenter une évaluation de coût lié à la réalisation de ce projet, ensuite les différentes difficultés que nous avons rencontrées au cours de la réalisation de ce projet et enfin quelques suggestions d'amélioration.

IV.1- EVALUATION DES COUTS

Les couts liés à la réalisation de ce projet sont :

- **Ressources humaines** : Pour une équipe constituée de trois ingénieurs en intelligence artificielle, en raison d'un salaire de 2 500 000 FCFA par mois sur une durée de 4 mois le coût total liés à la ressource humaine s'élevé donc à **2 500 000 (salaire) * 3 (ingénieurs) * 4 (mois) = 30 000 000 FCFA**.
- **Ressources matérielles** : Nous avons utilisé la version pro de Google Colab qui coûte 10.74 dollars soit environ 6 500 FCFA par mois ce qui fera donc au total **6 500 (par mois) * 4 (mois) * 3 (ingénieurs) = 78 000 FCFA**. En plus de Google Colab, nous avons utilisés des machines qui coûtes 250 000 FCFA par machine soit au total **250 000 (par machines) * 3 (ingénieurs) = 750 000 FCFA**. D'où le coût total lié aux ressources matérielles suivant : **78 000 + 750 000 = 828 000 FCFA**.
- **Connexion internet** : En raison d'une connexion internet de 25 000 FCFA par mois sur une durée de 4 mois nous avons donc le coût total suivant : **25 000 (par mois) * 4 (mois) * 3 (ingénieurs) = 300 000 FCFA**.

Le tableau suivant récapitule les couts liés à la réalisation de ce projet :

Tableau 2: Evaluation des coûts

Ressources	Coût (FCFA)
Ressources humaines	30 000 000
Ressources matérielles	828 000
Connexion internet	300 000
Total	31 128 000

IV.2- DIFFICULTES RENCONTREES

Les principales difficultés rencontrées lors de la réalisation de ce projet sont les suivantes :

- **La prise en main et la compréhension de cette nouvelle science qu'est l'intelligence artificielle** : l'intelligence artificielle est une science assez vaste et dont l'apprentissage est assez fastidieux. Il était donc nécessaire pour nous de se familiariser rapidement afin d'exploiter tout le potentiel de cette discipline dans le cadre de la réalisation de notre projet.
- **Acquisition du dataset** : Dans le cadre de la réalisation d'un modèle de segmentation sémantique d'image il est nécessaire d'avoir un *dataset* particulier constituer d'une image et du masque associer à cette dernière. Ce type de *dataset* n'est pas commun et est assez difficile à obtenir gratuitement surtout s'il s'agit d'image provenant de satellite. Concernant le Cameroun nous n'avons pas trouvé un tel *dataset* ce qui explique pourquoi nous avons utilisé un *dataset* propre à la ville de Dubai que nous avons pu obtenir sur le site <https://www.kaggle.com>. Ce système peut être implémenter dans le contexte du Cameroun en utilisant des données d'entraînement propre à celui-ci, cependant la constitution d'un tel *dataset* est une tâche très fastidieuse et prendrait énormément de temps ce qui n'était pas l'objectif premier de ce projet.
- **La compréhension des concepts nécessaire au traitement d'images** : En effet, pour la réalisation de notre projet il était nécessaire de comprendre des principes clés liés au traitement d'images à savoir : La manière dont est représenter une image en mémoire, comment traiter les différents types d'images (Images RASTER et Images Vecteur), la différence entre une image en **échelle de gris** et une image **rgb**.
- **L'instabilité de la connexion à internet** : Comme mentionné préalablement nous avons utilisé des outils comme **Google Colab** pour la réalisation de notre projet. Cet outil n'est disponible qu'en ligne et une connexion à internet rapide et stable est crucial pour son utilisation ce qui n'est pas souvent le cas au Cameroun et en Bandjoun en particulier.

IV.3- SUGGESTIONS

Malgré le fait que notre projet soit achevé il reste néanmoins ouvert à diverses améliorations à savoir :

- La possibilité d'entrainer un second modèle qui pourra effectuer des prédictions à partir des données fournies par notre système actuel. Une des limites de notre application est qu'elle fournit des données uniquement valides pendant un instant données ou sur une courte période de temps il serait donc intéressant de coupler à notre modèle un second modèle capable de faire des prédictions évolutives dans le temps à partir des données qu'il reçoit de notre modèle.
- Il est également possible d'associer notre système à un drone. Cette association augmentera considérablement l'efficacité de notre système car un drone facilitera l'acquisition des images nécessaire pour l'entraînement du modèle et ce quel que soit la zone où les prédictions doivent être effectuées. Notre système sera ainsi beaucoup plus flexible et gagnera en mobilité.
- L'augmentation des classes d'objet et de la quantité de données utilisées pour l'entraînement du modèle. Dans la version actuelle de notre système nous ne pouvons identifier que 06 classes d'objets par conséquent il manque de précision. Il serait donc important d'augmenter le nombre de classes détectable afin de la rehausser. En ce qui concerne les données d'entraînement plus cette quantité est grande et diverse, plus le système sera fiable et polyvalent.

CONCLUSION

Après avoir présenté les différents coûts liés à la réalisation de ce projet nous avons présenté les difficultés auxquelles nous avons été confrontés ainsi que quelques suggestions d'amélioration de notre système.

CONCLUSION

Un système de reconnaissance et d'extraction des formes identifiables sur une image est un système capable de reconnaître, identifier et extraire les formes sur une image. Pour réaliser notre système nous avons commencé par acquérir les données nécessaires pour l'entraînement du modèle, puis nous avons prétraité ces données, ensuite entraîner notre modèle pour qu'il soit capable de reconnaître et d'extraire les formes sur une image et enfin nous avons intégré notre système à une application web pour faciliter son utilisation. A la fin de notre réalisation nous avons obtenus un système capable de reconnaître les bâtiments, les routes, les cours d'eau, la végétation, le sol sur une image satellitaire. Nous avons été confronter à plusieurs difficultés tel que la prise en main et la compréhension de cette nouvelle science qui est l'intelligence artificielle, une connexion internet instable et bien d'autres. Comme perspectives d'amélioration de notre système, nous pouvons entraîner un second modèle qui pourra effectuer des prédictions à partir des données fournies par notre système actuel, associer notre système à un drone et augmenter des classes d'objets et la quantité de données utilisées pour l'entraînement du modèle pour pouvoir améliorer son champ d'action et sa précision.

REFERENCES BIBLIOGRAPHIQUES

- [1] (2023, Janvier 30). Récupéré sur tensorflow.org: <https://www.tensorflow.org/guide>
- [2] (2023, Avril 05). Récupéré sur huggingface.co: <https://huggingface.co>
- [3] (2023, Avril 10). Récupéré sur wandb.ia: <https://wandb.ia>
- [4] (2023, Mars 12). Récupéré sur kaggle.com: <https://kaggle.com>
- [5] CHauhan, A. (2023, janvier 15). *Prodramp*. Récupéré sur Youtube:
<https://www.youtube.com/@Prodramp/playlists/Deep Learning>
- [6] Datensctutz. (2015). *U-NET: Convolutionnal Networks for Biomedical Image Segmentation*. Freiburg.
- [7] LES RESEAU DE NEURONES CONVOLUTIFS. (2023, juillet 14). Récupéré sur natural-solution.eu/: <https://www.natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-rseaux-de-neurones-convolutifs>
- [8] Nada, B. (2023, Janvier 20). *detection-images-yolo-tensorflow*. Récupéré sur blent.ia:
<https://blent.ai/detection-images-yolo-tensorflow/>
- [9] Saint-Cirges, G. (2019). *Apprendre le Machine Learning* . Paris.
- [10] Saint-Cirges, G. (2022). *Apprendre le Machine Leaning en une samaine*. Récupéré sur Machine Learnia: <https://machinelearnea.com>
- [11] Saint-Cirges, G. (2023, janvier 20). *Machine Learnea*. Récupéré sur Youtube:
<https://www.youtube.com/@MachineLearnia/playlists/Machine Learning Français Formation complete>
- [12] Saint-Cirges, G. (2023, janvier 20). *Machine Learnea*. Récupéré sur Youtube:
<https://www.youtube.com/@MachineLearnia/playlists/Formation Deep Learning>

ANNEXE

Suite des résultats des prédictions des différents patchs :

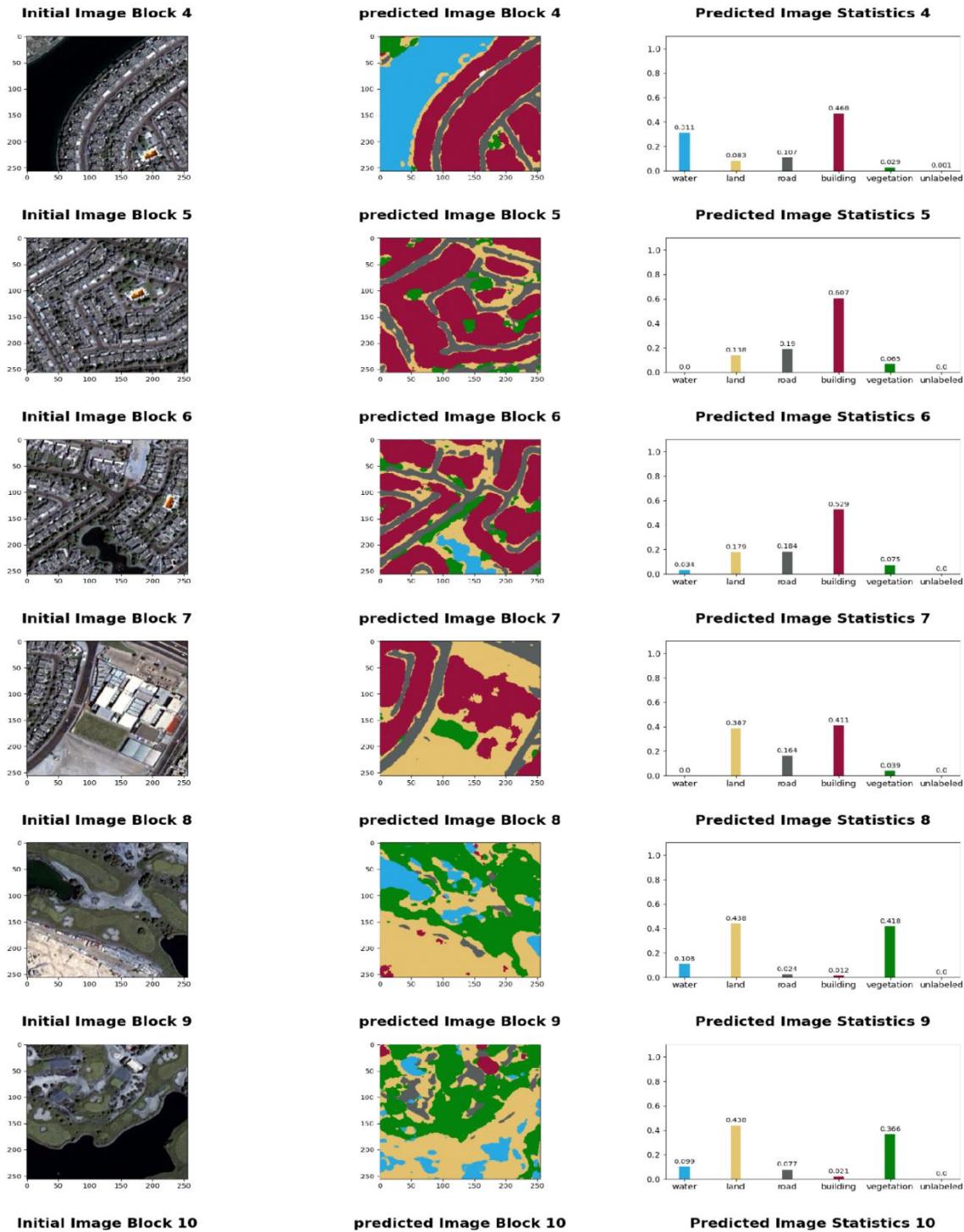


Figure 30: Patchs predictions and statistics 2

TABLE DES MATIERES

DEDICACE.....	i
AVANT-PROPOS	ii
REMERCIEMENTS	iv
LISTE DES ABREVIATIONS	v
RESUME.....	vi
ABSTRACT	vii
SOMMAIRE	viii
LISTE DES FIGURES	x
LISTE DES TABLEAUX	xi
INTRODUCTION.....	1
CHAPITRE I: PRESENTATION DU PROJET	2
INTRODUCTION.....	2
I.1- PRESENTATION DE L'ARRETE DU PROJET	2
I.2- PROBLEMATIQUE	2
I.3- OBJECTIFS DU PROJET	3
I.4- ACTEURS CIBLES DU PROJET	3
I.5- RESULTATS ATTENDUS	4
I.6- ETAT DE L'ART	4
I.7- CHRONOGRAMME DES ACTIVITES	6
CONCLUSION	7
CHAPITRE II : ANALYSE FONCTIONNELLE ET TECHNIQUE DE CONCEPTION.....	8
INTRODUCTION.....	8
II.1- ANALYSE FONCTIONNELLE DU SYSTEME	8
II.2- CONCEPTION TECHNIQUE ET FONCTIONNELLE DU SYSTEME	9
II.3- INTERACTION ENTRE LES ACTEURS ET LE SYSTEME	23
II.3- COMPORTEMENT INTERNES ET DIFFERENTS ETATS LORS DU TRAITEMENT D'UNE IMAGE	25
CONCLUSION	27
CHAPITRE III: IMPLEMENTATION ET RESULTATS	28
INTRODUCTION.....	28
III.1- TECHNOLOGIES UTILISEES	28
III.2- ENVIRONNEMENT DE DEVELOPPEMENT	29

III.3- ENVIRONNEMENT DE PRODUCTION	31
III.4- EVOLUTION DE L'APPRENTISSAGE DU MODELE	32
III.5 RESULTATS OBTENUS	35
III.6- APPLICATION DES RESULTATS	42
CONCLUSION	42
CHAPITRE IV: DIFFILCUTES ET SUGGESTIONS	43
INTRODUCTION	43
IV.1- EVALUATION DES COUTS	43
IV.2- DIFFICULTES RENCONTREES	44
IV.3- SUGGESTIONS	45
CONCLUSION	45
CONCLUSION	46
REFERENCES BIBLIOGRAPHIQUES	xii
ANNEXE	xiii
TABLE DES MATIERES	xiv