

République Tunisienne
Ministère de l'Enseignement Supérieur et de
la recherche scientifique

Université de Gafsa
Institut Supérieur des Sciences
Appliquées et de la Technologie de Gafsa



Cycle de Formation en Mastère
Professionnelle dans la Discipline
Science de données

Mémoire de MASTERE
Science des données
N° d'ordre : 02-MSD

MEMOIRE

Présenté à

L'Institut Supérieur des Sciences Appliquées et de Technologie de Gafsa

(Département Informatique et télécommunication)
En vue de l'obtention Diplôme en

MASTERE

Dans la discipline Sciences des données

Par

Sameh Ghilene

**REALISATION D'UN MODELE D'ATTAQUE DES IMAGES
NUMERIQUE TATOUEES MULTI-BITS A
L'AIDE D'UNE APPROCHE DE RESEAU DE
NEURONE CONVOLUTIF BASE SUR L'U-NET**

Soutenu devant le jury composé de :

M. Said TAIEB

Président

Mme Wajdi SAADAoui

Rapporteur

M. Ahmed REKIK

Encadreur

M. Mounir TELLi

Co-Encadreur

A.U : 2022 – 2023

Dédicace

Je dédie ce travail à :

Mes chers parents que Dieu vous protège et vous accordent une longue vie pleine d'amour et de joie infinies car aucune dédicace ne saurait exprimer l'amour, l'estime et le respect que j'ai toujours eu pour vous.

Mon cher papa « Ghilene Ezzdine »

Qui était mon super homme et mon support pour toute la vie.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ma chère maman « Raddaoui Jemiaa »

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

A mes sœurs et mes frères

J'exprime envers vous une profonde admiration et un attachement inconditionnel. Je vous souhaite toute la joie et le succès dans vos parcours

Tous mes

Collègues et tous les ami(e)s que je connais à ISSAT.

Enfin, je remercie tous ceux qui m'ont supporté durant mes études et ma vie.

Esameh ghilene

Remerciements

Je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui m'avez donné la force et la patience d'accomplir ce modeste travail.

Je tiens à présenter ma reconnaissance et mes remerciements à mon encadrant

«Telli Mounir» pour le temps consacré à la lecture et aux réunions qui ont rythmé les différentes étapes de notre mémoire.

Les discussions que nous avons partagées ont permis d'orienter mon travail d'une manière pertinente. Je la remercie aussi pour sa disponibilité à encadrer ce travail à travers ses critiques et ses propositions d'amélioration.

Je tiens également à remercier les membres du jury qui ont accepté

D'examiner mon mémoire.

J'adresse aussi mes remerciements à tous mes professeurs à ISSAT pour la formation et l'expérience qu'ils m'ont transmise.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui nous ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

Esameh ghilene

Table des matières

Liste des figures	
Liste des Abréviations.....	
Introduction générale	1
Chapitre1	4
Etude préliminaire	4
Introduction.....	4
1. Tatouage numérique multi-bits :.....	4
1.1. C'est quoi le tatouage numérique :.....	4
1.2. C'est quoi le tatouage numérique multi-bits :	5
1.3. Techniques de tatouage numérique multi-bits :.....	6
1.4. Algorithmes de tatouage numérique multi-bits :	7
2. Méthodes d'attaque des images tatouées :.....	8
2.1. Principe d'attaque de l'image tatouée :	8
2.2. Attaque de l'image tatouée :.....	9
2.3. Outils de détection des attaques :	9
3. Présentation de projet :.....	10
3.1. Problématiques :	10
3.2. Objectifs :	10
3.3. Solutions existantes :	11
3.4. Solution proposée :	12
Conclusion	12
Chapitre2.....	13
SEGMENTATION D'IMAGES EN DEEP LEARNING : L'APPROCHE	13
U-NET.....	13
Chapitre2	14
Segmentation d'images en Deep Learning : L'approche U-net	14
Introduction.....	14
1. Le Deep Learning :	14
1.1. Présentation :	14

1.2.	Principe de fonctionnement :	15
1.3.	Architecture pour la segmentation d'image :	16
1.4.	Méthodes d'apprentissage :	17
2.	Les réseaux de neurones convolutionnels (CNN) :	19
2.1.	Présentation :	19
2.2.	Fonctionnement et architecture :	20
2.3.	CNN pour le traitement des images :	21
3.	La segmentation d'images :	21
3.1.	Présentation :	21
3.2.	Méthodes de segmentation d'images :	22
3.3.	Les modèles de segmentation d'images utilisant les CNN :	24
4.	Le U-Net :	25
4.1.	Présentation :	25
4.2.	Architecture et fonctionnement :	26
4.3.	U-Net pour la segmentation des tatouages numériques :	26
	Conclusion :	27
	Chapitre3.....	28
	Expérimentations	28
	Chapitre3	29
	Expérimentations	29
	Introduction.....	29
1.	Outils de développement :	29
1.1.	Google Colab :	29
1.2.	Langage Python :	29
2.	Bibliothèques pour la préparation des données :	30
2.1.	NumPy :	30
2.2.	OpenCV :	31
2.3.	PIL (Python Imaging Library) :	31
3.	Outils pour l'implémentation d'U-Net :	32
3.1.	TensorFlow :	32

3.2. Keras :	32
3.3. PyTorch :	33
4. Source des données :	34
5. Tatouage des images :	34
6. Implémentation de modèle U-Net :	36
7. Résultats obtenus :	40
Conclusion	41
Conclusion générale	42
Webographie	43
Résumé	46
Abstract	46

Liste des figures

Figure 1- 1 Eléments d'un système de tatouage numérique	5
Figure 1- 2 Classification des algorithmes de tatouage numérique [4]	7
Figure 2- 1 principe de fonctionnement de réseaux de neurones.....	15
Figure 2- 2 Architecture d'un réseau de segmentation d'images [26]	16
Figure 2- 3 Architecture classique d'un réseau de neurones convolutif [27]	20
Figure 2- 4 Segmentation basée sur seuillage [28]	22
Figure 2- 5 Segmentation basée sur la région [29]	23
Figure 2- 6 Segmentation basée sur la détection de contours [30]	23
Figure 2- 7 U-net pour la segmentation d'image [31]	25
Figure 3- 1 Logo Google Colab [32]	29
Figure 3- 2 Logo Google Python[33].....	30
Figure 3- 3 Logo NumPy [34]	31
Figure 3- 4 Logo OpenCV [35]	31
Figure 3- 5 Logo PIL [36].....	31
Figure 3- 6 Logo TensorFlow [37]	32
Figure 3- 7 Logo Keras[38]	33

Figure 3- 8 Logo PyTorch [39].....	34
Figure 3- 9 TinyImageNetDataset	34
Figure 3- 10 Tatouage de l'image	35
Figure 3- 11 Affichage d'une Image non tatouée	36
Figure 3- 12 Affichage d'exemple d'une image tatouée	36
Figure 3- 13 Organigramme de code de tatouage.....	37
Figure 3- 14 Organigramme de code de création de modèle U-Net.....	37
Figure 3- 15 Modèle U-net	38
Figure 3- 16 Entraînement Vs Validation.....	39
Figure 3- 17 Entraînement de modèle.....	39
Figure 3- 18 Accuracy de modèle U-Net.....	40
Figure 3- 19 Loss de modèle U-Net.....	40

Liste des Abréviations

API: Application Programming Interface

CNN: Convolutional Neural Network

DCT : Discrete Cosine Transform

RNN: Recurrent Neural Network

GAN: Generative Adversarial Network

AE: Autoencoder

LSTM: Long Short-Term Memory

LSB : Least Significant Bit

ML: Machine Learning

Introduction générale

Le tatouage numérique est une technique de dissimulation d'informations ou de marquage invisible qui est appliquée aux images pour diverses raisons, telles que la protection des droits d'auteur, la vérification de l'authenticité ou le suivi des images.

La détection de tatouages numériques est devenue un enjeu important dans le domaine de la sécurité des données et de la protection de la propriété intellectuelle. De nombreuses méthodes de détection ont été développées, mais elles présentent souvent des limitations en termes de précision et de robustesse. C'est là que l'approche basée sur U-Net entre en jeu.

U-Net est une architecture de réseau neuronal convolutionnel (CNN) largement utilisée dans la segmentation d'images. Sa structure en forme de U et ses mécanismes de connexion permettent une segmentation précise des régions d'intérêt dans les images. Dans notre projet, nous exploitons les capacités d'U-Net pour la détection des tatouages numériques.

L'objectif principal de ce projet est de concevoir et d'implémenter un modèle U-Net qui sera entraîné sur un ensemble de données d'images contenant des tatouages numériques. Le modèle sera ensuite capable de détecter si une image est tatouée ou non.

Ce rapport vise à fournir une étude approfondie du tatouage numérique multi-bits, des méthodes d'attaque des images tatouées et de l'approche de segmentation d'images basée sur le modèle U-net. Nous explorerons les concepts, les techniques et les outils nécessaires à la compréhension et à la mise en œuvre de ces méthodes. Les résultats obtenus dans le cadre de ce rapport contribueront à une meilleure compréhension et à une application pratique du tatouage numérique et de la segmentation d'images.

Le premier chapitre sera consacré à l'étude préliminaire du tatouage numérique multi-bits et des méthodes d'attaque des images tatouées. Dans ce chapitre, nous aborderons les concepts fondamentaux du tatouage numérique. Nous examinerons également les différentes méthodes d'attaque utilisées pour altérer ou détecter les images tatouées, ainsi que les outils de détection des attaques.

Ensuite, dans le deuxième chapitre, nous nous concentrerons sur la segmentation d'images en utilisant l'approche U-net basée sur le Deep Learning. Nous commencerons par introduire le Deep Learning, en expliquant son fonctionnement et son application à la segmentation d'images. Nous explorerons également les réseaux de neurones convolutionnels (CNN) utilisés dans le traitement des images, ainsi que les méthodes de segmentation d'images existantes. Enfin, nous présenterons en détail l'architecture U-net et son utilisation spécifique

dans la segmentation des tatouages numériques.

Le dernier chapitre sera consacré aux outils indispensables pour la mise en œuvre du modèle U-net. Nous présenterons les outils de développement, les principales bibliothèques de Deep Learning telles que TensorFlow, Keras et PyTorch, qui permettent l'implémentation du modèle U-net et la source des données utilisées pour notre étude. Enfin, nous présenterons les résultats obtenus.

Chapitre 1

ETUDE PRELIMINAIRE

Chapitre1

Etude préliminaire

Introduction

Dans ce chapitre, nous allons étudier la notion de tatouage numérique, les méthodes d'attaques des images tatouées et les méthodes l'intelligence artificielle pour la détection des images tatoués.

1. Tatouage numérique multi-bits :

1.1.C'est quoi le tatouage numérique :

Le tatouage numérique, également appelé filigrane numérique ou watermarking, est une technique qui permet d'ajouter une marque distinctive ou un identifiant unique à un fichier numérique, tel qu'une image, une vidéo ou un fichier audio. Cette marque peut être visible ou invisible et est souvent utilisée pour protéger les droits d'auteur et la propriété intellectuelle de l'auteur ou du propriétaire du fichier.

Le tatouage numérique est une technique souvent utilisée dans l'industrie de la musique, du cinéma et de la photographie, ainsi que dans d'autres domaines où la protection de la propriété intellectuelle est cruciale.

Il existe deux types de tatouage numérique :

- **Tatouage numérique visible [44]** : est une technique qui implique l'ajout d'une marque distinctive visible sur le contenu numérique, comme une image ou une vidéo. Cette marque est souvent un logo, une signature ou un texte et peut être placée dans un coin de l'image ou de la vidéo. L'objectif du watermarking visible est d'identifier le propriétaire ou l'auteur du contenu numérique et de décourager la copie ou le vol de contenu.
- **Tatouage numérique invisible [45]** : est une technique qui ajoute une marque distinctive à l'intérieur du contenu numérique, de manière à ne pas être perceptible à l'œil humain. Cette technique est souvent utilisée pour protéger les droits d'auteur et la propriété intellectuelle du contenu numérique. Le watermarking invisible peut être utilisé pour identifier l'auteur ou le propriétaire du contenu numérique, pour assurer la traçabilité et la sécurité des produits numériques, ou pour lutter contre la copie illégale ou le piratage.

Le tatouage numérique visible est utilisé pour identifier le propriétaire ou l'auteur du contenu numérique, tandis que le tatouage numérique invisible est utilisé pour protéger les droits d'auteur et la propriété intellectuelle du contenu numérique et pour assurer la traçabilité et la sécurité des produits numériques.

Un système de tatouage numérique ou de watermarking peut être considéré comme un système de communication composé de trois éléments principaux : un émetteur, un canal de communication et un récepteur. L'insertion de l'information à dissimuler dans le signal hôte correspond à la transmission des données, tandis que tout traitement appliqué au signal hôte après l'insertion de l'information, ainsi que l'interaction entre les données de l'information dissimulée et celles du signal hôte lui-même, représentent la transmission à travers un canal de communication. La récupération de l'information dissimulée à partir des données du signal hôte correspond au rôle du récepteur.

En analogie avec les systèmes de communication, tout système de tatouage numérique prend la forme représentée sur la figure suivante ;

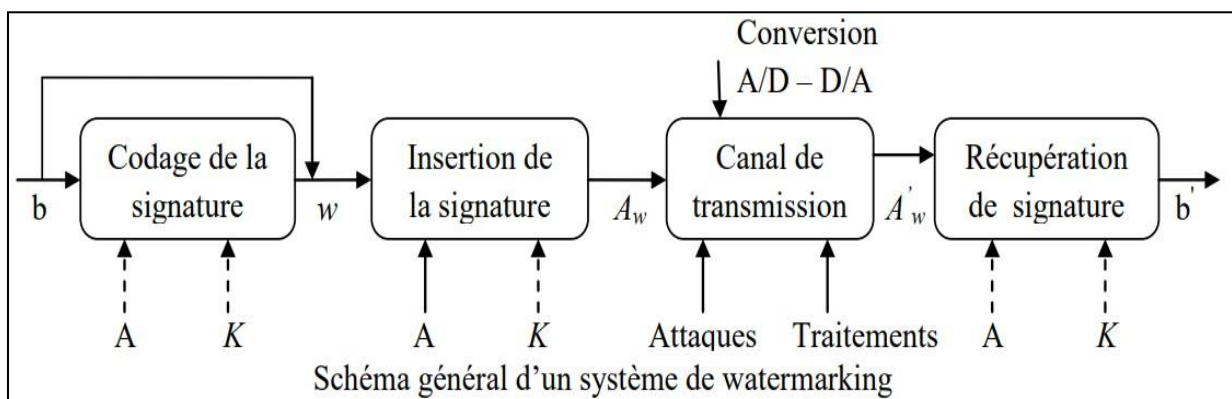


Figure 1- 1 Eléments d'un système de tatouage numérique

1.2.C'est quoi le tatouage numérique multi-bits :

Le tatouage numérique multi-bit est une technique de tatouage numérique qui permet d'ajouter plusieurs bits d'information à un fichier numérique, tels que des images, des vidéos, des fichiers audio ou des documents. Contrairement au tatouage numérique classique, qui ne permet d'ajouter qu'un seul bit d'information, le tatouage numérique multi-bit permet d'ajouter plusieurs bits, ce qui permet d'augmenter la complexité de la marque distinctive et d'améliorer sa résistance aux attaques de contrefaçon ou de falsification.

Le tatouage numérique multi-bit est souvent utilisé pour protéger les droits d'auteur et la propriété intellectuelle, en permettant de marquer un fichier numérique avec des informations spécifiques telles que le nom de l'auteur, le titre, la date, le numéro de série ou d'autres informations pertinentes. Cette technique peut également être utilisée pour identifier l'auteur

ou le propriétaire d'un fichier, pour assurer la traçabilité et la sécurité des produits, ou pour lutter contre la copie illégale ou le piratage.

La complexité du tatouage numérique multi-bit dépend du nombre de bits d'information ajoutés et de l'algorithme de tatouage utilisé pour l'ajout. Les algorithmes de tatouage multi-bit doivent être conçus pour minimiser la perte de qualité du fichier numérique original tout en garantissant la fiabilité et la sécurité de la marque distinctive.

1.3.Techniques de tatouage numérique multi-bits :

Il existe plusieurs techniques de tatouage numérique multi-bits qui peuvent être utilisées pour ajouter des informations à un fichier numérique. Voici quelques-unes des techniques les plus courantes :

- **La technique de substitution de bits :** Cette technique consiste à remplacer certains bits dans le fichier numérique par les bits de l'information à ajouter. Cette technique est relativement simple à mettre en œuvre mais elle peut avoir un impact sur la qualité de l'image originale.
- **La technique de modulation de phase :** Cette technique consiste à moduler la phase de certains pixels de l'image afin d'y incorporer des informations. Cette technique est plus complexe que la technique de substitution de bits mais elle offre une meilleure résistance à la compression et à la conversion de format.
- **La technique de codage par blocs :** Cette technique consiste à diviser l'image en blocs et à ajouter de l'information à chaque bloc. Cette technique permet d'ajouter plus d'informations que les techniques précédentes et elle est également plus résistante aux attaques de contrefaçon.
- **La technique de codage par ondelettes :** Cette technique consiste à utiliser une transformation en ondelettes pour ajouter de l'information à l'image. Cette technique est plus complexe que les autres techniques mais elle offre une résistance supérieure aux attaques de contrefaçon et une meilleure qualité de l'image originale.

Il est important de noter que chaque technique de tatouage numérique multi-bits a ses avantages et ses inconvénients. Le choix de la technique dépendra des besoins spécifiques de chaque application.

1.4.Algorithmes de tatouage numérique multi-bits :

Il existe une multitude les algorithmes de tatouage numériques développés pour diverses applications. Ces algorithmes sont utilisés pour ajouter des informations à un fichier numérique tout en minimisant l'impact sur la qualité de l'image originale. Chacun d'entre eux a ses avantages et ses inconvénients. Le choix de l'algorithme dépendra des besoins spécifiques de chaque application.

La classification de ces algorithmes peut se faire selon différents critères tel que : le domaine d'insertion, la robustesse, la technique d'insertion utilisée, le mode d'extraction, la perception de la marque et la préservation de l'image originale. La figure ci-dessous présente un organigramme de cette classification.

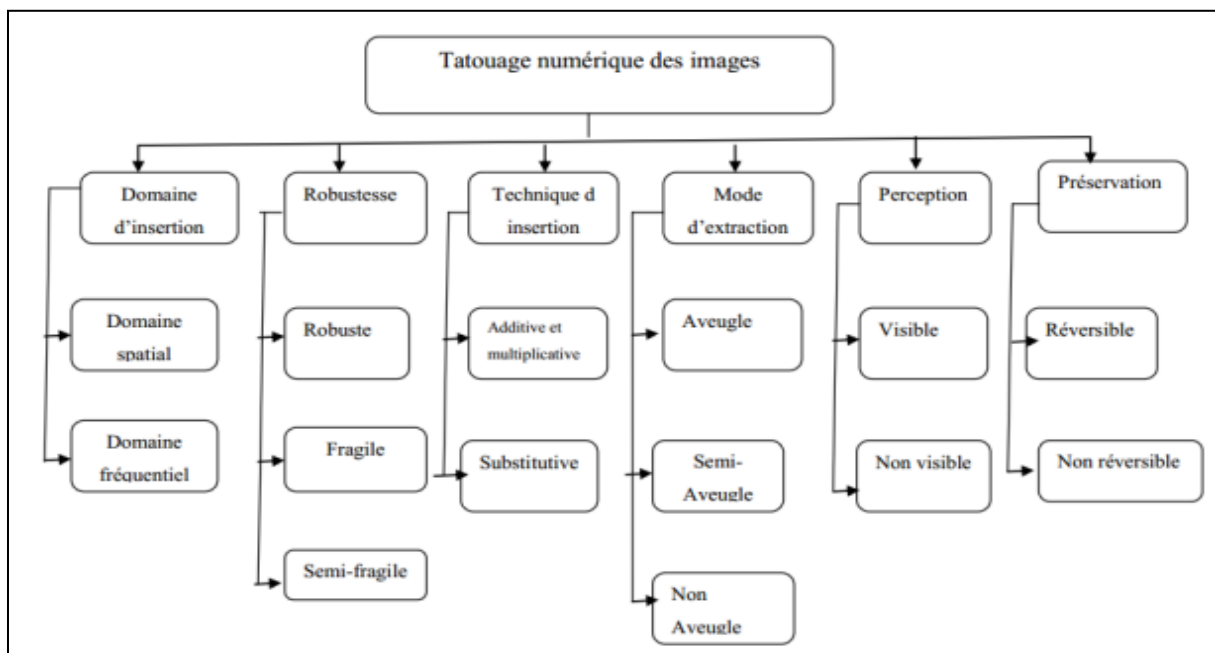


Figure 1- 2 Classification des algorithmes de tatouage numérique [4]

Parmi les algorithmes les plus courants, on trouve [12] :

- **L'algorithme Spread Spectrum** : Cet algorithme utilise une technique de modulation de phase pour ajouter des informations à l'image. Cette technique consiste à répartir les informations à ajouter sur une bande de fréquences et à les ajouter aux pixels en modifiant leur phase. Cette technique est très résistante aux attaques de contrefaçon et peut être utilisée pour ajouter plusieurs bits d'information à une image.
- **L'algorithme Least Significant Bit (LSB)** : Cet algorithme utilise la technique de substitution de bits pour ajouter des informations à l'image. Cette technique consiste à remplacer les bits de poids faible de chaque pixel de l'image par les bits de

l'information à ajouter. Cette technique est relativement simple à mettre en œuvre, mais elle peut affecter la qualité de l'image originale.

- **L'algorithme DiscreteCosineTransform (DCT) :** Cet algorithme utilise une transformation en cosinus discrète pour ajouter des informations à l'image. Cette technique consiste à diviser l'image en blocs et à appliquer la transformation en cosinus discrète à chaque bloc pour ajouter des informations. Cette technique est plus résistante aux attaques de contrefaçon que la technique LSB, mais elle peut avoir un impact sur la qualité de l'image originale.
- **L'algorithme de codage par ondelettes (DWT) :** Cet algorithme utilise une transformation en ondelettes pour ajouter des informations à l'image. Cette technique consiste à appliquer la transformation en ondelettes à l'image pour extraire les coefficients d'ondelettes et à modifier certains de ces coefficients pour y ajouter des informations. Cette technique est très résistante aux attaques de contrefaçon et peut être utilisée pour ajouter plusieurs bits d'information à une image.

2. Méthodes d'attaque des images tatouées :

2.1.Principe d'attaque de l'image tatouée :

Le principe d'attaque des images tatouées consiste à essayer de retirer ou de modifier le tatouage numérique d'une image sans altérer la qualité visuelle de l'image originale. Les attaques peuvent être effectuées par des personnes mal intentionnées pour essayer de falsifier l'image, ou par des processus de compression d'image qui peuvent affecter la qualité du tatouage numérique.

Les méthodes d'attaque des images tatouées sont généralement classées en deux catégories ;

- **Les attaques actives** sont des attaques qui essaient de retirer le tatouage numérique en altérant intentionnellement l'image. Cela peut être fait en modifiant certains pixels de l'image, en supprimant des parties de l'image ou en ajoutant du bruit à l'image pour masquer le tatouage numérique.
- **Les attaques passives** sont des attaques qui essaient de retirer le tatouage numérique sans altérer intentionnellement l'image. Cela peut être fait en comprimant l'image ou en la recadrant pour enlever des parties de l'image qui contiennent le tatouage numérique. Ces attaques sont plus difficiles à détecter que les attaques actives car elles ne modifient pas directement l'image.

2.2. Attaque de l'image tatouée :

Les images tatouées peuvent être soumises à diverses attaques visant à éliminer ou à altérer le tatouage numérique. Ces attaques peuvent être utilisées individuellement ou en combinaison les unes avec les autres pour altérer le tatouage numérique et rendre plus difficile sa récupération. Les techniques de tatouage numérique sont donc continuellement améliorées pour être plus robustes aux différentes attaques possibles.

Parmi les méthodes d'attaque les plus courantes, on peut citer ;

- **Les attaques par filtrage** consistent à appliquer différents types de filtres sur l'image tatouée pour supprimer ou atténuer le tatouage numérique. Les filtres les plus courants sont les filtres de passe-bas, qui éliminent les hautes fréquences de l'image.
- **Les attaques par compression** consistent à comprimer l'image tatouée en utilisant des algorithmes de compression tels que JPEG, ce qui peut entraîner une perte d'informations et une altération du tatouage numérique.
- **Les attaques par recadrage** consistent à recadrer l'image tatouée pour éliminer le tatouage numérique. Cette méthode peut être efficace si le tatouage est situé dans une région spécifique de l'image.
- **Les attaques par transformation géométrique** consistent à appliquer des transformations géométriques telles que la rotation, l'homothétie, la translation ou la projection sur l'image tatouée pour altérer le tatouage numérique.
- **Les attaques par ajout de bruit** consistent à ajouter du bruit (le bruit gaussien, le bruit de Poisson, ...) à l'image tatouée pour altérer le tatouage numérique.

2.3. Outils de détection des attaques :

Il existe différentes techniques et outils pour détecter les attaques sur les images tatouées qui peuvent être même utilisés de manière combinée pour améliorer la détection des attaques sur les images tatouées multi-bits., en voici quelques-uns :

- **La méthode de corrélation** : consiste à comparer l'image originale et l'image tatouée pour déterminer si le tatouage numérique a été altéré. Si la corrélation entre les deux images est faible, cela peut indiquer une attaque.
- **Les méthodes de signature** : utilisent des algorithmes de signature pour détecter les altérations dans l'image tatouée. Les signatures peuvent être basées sur les caractéristiques du tatouage numérique ou sur des propriétés statistiques de l'image.

- **Les méthodes de compression** : consistent à compresser l'image tatouée et à comparer la taille de la version compressée avec la taille attendue. Si la taille est différente, cela peut indiquer une attaque.
- **Les méthodes de filtrage** : consistent à appliquer des filtres sur l'image tatouée pour détecter les altérations. Les filtres peuvent être basés sur des propriétés statistiques de l'image ou sur les caractéristiques du tatouage numérique.
- **Les méthodes de watermarking actif** : consistent à insérer un tatouage numérique supplémentaire dans l'image tatouée pour détecter les altérations. Le tatouage supplémentaire peut être conçu pour être invisible à l'œil nu, mais détectable par des algorithmes spécifiques.
- **Les méthodes de deeplearning** : utilisent des réseaux de neurones pour détecter les altérations dans l'image tatouée. Les réseaux peuvent être entraînés sur des images tatouées originales et altérées pour apprendre à détecter les différences.

3. Présentation de projet :

3.1.Problématiques :

La détection de tatouage numérique est devenue une problématique majeure dans le domaine de la sécurité numérique. En effet, avec l'avancée de la technologie numérique, les techniques de tatouage numérique sont de plus en plus courantes dans les médias numériques (images et vidéos). Ces techniques ont pour but d'ajouter des informations cachées ou visibles dans les images ou les vidéos pour différentes raisons, surtout pour la protection des droits d'auteur, la traçabilité des images, ou la détection de contrefaçons.

Malheureusement, ces mêmes techniques peuvent également être utilisées à des fins malveillantes, telles que l'usurpation d'identité ou la falsification d'images. Dans ces cas, il est crucial de pouvoir détecter la présence d'un tatouage numérique, afin de préserver l'intégrité et la sécurité des données numériques. En effet, la capacité à détecter un tatouage numérique permet de garantir l'authenticité et la fiabilité des images et vidéos utilisées dans diverses applications, telles que la sécurité informatique, la médecine, la surveillance et d'autres domaines où l'exactitude et la précision sont essentielles.

3.2.Objectifs :

Avec l'augmentation de l'utilisation de techniques de tatouage numérique, il est devenu crucial de pouvoir détecter la présence d'un tatouage numérique afin de prévenir la falsification

d'images ou l'usurpation d'identité.

Les images numériques peuvent être altérées ou modifiées par des personnes malveillantes pour diffuser des informations fausses ou trompeuses. Dans ce contexte, le but de notre projet de détection de tatouage numérique est de fournir une solution efficace et précise pour détecter les tatouages numériques dans les images. Cette solution permettra de garantir l'intégrité et la sécurité des données numériques en détectant et en identifiant les modifications qui ont été apportées aux images. En fin de compte, le projet vise à aider les utilisateurs à protéger leurs données numériques et à prévenir les abus et les utilisations malveillantes de ces données.

3.3.Solutions existantes :

Il existe plusieurs solutions existantes pour la détection de tatouage numérique, on peut citer par exemples ;

- **La solution DeTOX** : C'est une solution développée par la société Eurocom pour la détection de tatouages numériques dans des images. Cette solution utilise des techniques avancées de traitement d'image et de vision par ordinateur pour identifier les zones modifiées d'une image. Elle prend en compte les différents types de tatouages numériques, tels que les tatouages à base de bits et les tatouages à base de points, ainsi que les différentes altérations possibles, telles que la compression ou la rotation.

La solution DeTOX utilise des algorithmes de détection de contours pour extraire les contours de l'image et les comparer avec ceux d'une image originale non modifiée. Elle utilise ensuite des techniques de corrélation pour détecter les zones modifiées en comparant la similarité entre les différentes parties de l'image. Enfin, elle utilise des méthodes pour apprendre à détecter les tatouages numériques.

DeTOX est considéré comme l'une des solutions les plus performantes pour la détection de tatouages numériques, grâce à sa grande précision et à sa capacité à détecter une large gamme de tatouages numériques et d'altérations d'image. Elle est utilisée par de nombreuses entreprises et organisations pour protéger leur propriété intellectuelle et leurs données numériques.

Cette solution est spécifique au traitement des vidéos.

- **Restb.ai WatermarkDetection** : Cette solution de vision artificielle pour la détection de tatouage numérique est dotée d'une API RESTful, qui permet de détecter

facilement la présence de filigranes, de logos ou de textes artificiellement ajoutés dans des images. Cette solution est conçue pour détecter ces éléments, qu'ils soient opaques ou presque transparents. Grâce à cette API, les clients peuvent automatiser la recherche et la classification des images qui contiennent des filigranes, protégeant ainsi leurs propres droits d'auteur et évitant les violations de propriété intellectuelle.

Cette solution est payante et offre des options de base dans sa version gratuite.

- **Détection basée sur l'estimation bayésienne :** La détection de tatouage numérique basée sur l'estimation bayésienne est une approche statistique pour identifier la présence d'un tatouage numérique dans une image. Cette méthode suppose que la probabilité d'avoir un tatouage numérique dans une image suit une distribution de probabilité aléatoire. En utilisant des techniques d'estimation bayésienne, cette méthode est capable de modéliser cette distribution et de déterminer la présence ou l'absence d'un tatouage numérique dans une image donnée. Elle est particulièrement efficace pour détecter les tatouages numériques ajoutés subtilement à l'image, qui peuvent être difficiles à repérer avec d'autres méthodes.

3.4.Solution proposée :

Nous avons choisi la méthode de détection de tatouage en utilisant U-Net. Cette méthode utilise le réseau de neurones convolutionnels U-Net pour détecter la présence d'un tatouage numérique dans une image. Le réseau U-Net est un modèle de segmentation d'image qui a été conçu pour traiter des images médicales en segmentant des structures complexes telles que les organes ou les tissus. Nous allons adapter ce modèle pour la détection de tatouage numérique en l'entraînant sur un ensemble de données d'images tatouées et non tatouées. Le modèle doit être capable de segmenter les pixels de l'image en deux classes : tatoué ou non tatoué.

Conclusion

Dans ce chapitre, nous avons étudié les techniques de tatouages numériques, les méthodes d'attaque. Dans le reste de ce rapport, nous allons présenter notre modèle d'attaque des images numériques tatouées multi-bits à l'aide d'une approche de réseau de neurones convolutifs basée sur le U-Net.

Par la suite nous avons présenté quelques solutions existantes de détection de tatouage numérique et nous avons choisi d'utiliser l'approche U-net pour la détection de tatouage.

Le chapitre suivant sera consacré à la présentation de l'approche U-net pour la segmentation d'images.

Chapitre2

SEGMENTATION D'IMAGES EN DEEP LEARNING : L'APPROCHE U-NET

Chapitre2

Segmentation d'images en Deep Learning : L'approche U-net

Introduction

Le tatouage numérique consiste à ajouter des informations ou des marques invisibles à l'œil nu sur une image numérique. Ces marques sont utilisées pour diverses raisons telles que l'authentification, la traçabilité ou encore la protection de la propriété intellectuelle. Pour extraire ces marques, il est nécessaire de segmenter l'image, c'est-à-dire de séparer les différentes parties de l'image pour isoler les zones d'intérêt. C'est là que la segmentation d'image en Deep Learning, en particulier l'approche U-net, intervient. En effet, U-net est un modèle de segmentation d'images qui permet de séparer les zones d'intérêt d'une image, ce qui peut être utilisé pour extraire les marques de tatouage numérique. Ainsi, le lien entre le tatouage numérique et la segmentation d'image est étroit, car la segmentation est une étape cruciale pour extraire les informations de tatouage numérique et l'approche U-net est l'une des approches les plus efficaces pour cette tâche.

Dans ce chapitre, nous allons étudier l'approche U-net pour la segmentation d'image dans le but d'extraire des informations de tatouage.

1. Le Deep Learning :

1.1.Présentation :

Le Deep Learning est une branche de l'apprentissage automatique ou Machine Learning qui s'appuie sur l'utilisation de réseaux de neurones artificiels pour apprendre à partir de données. Il est inspiré du fonctionnement du cerveau humain, où des réseaux de neurones interconnectés travaillent ensemble pour effectuer des tâches complexes.

Le Deep Learning permet de résoudre des problèmes de classification, de reconnaissance de formes, de traitement du langage naturel, de vision par ordinateur ou autres en apprenant à partir d'un grand nombre de données. Contrairement à l'apprentissage machine traditionnel, où les caractéristiques des données doivent être extraites manuellement, le Deep Learning est capable de détecter automatiquement les caractéristiques importantes des données d'entrée à partir de plusieurs couches de traitement.

Le Deep Learning a connu un grand succès ces dernières années, en particulier dans des domaines tels que la reconnaissance d'images, la reconnaissance vocale et la traduction automatique.

1.2.Principe de fonctionnement :

Le principe de fonctionnement du Deep Learning est basé sur l'utilisation de réseaux de neurones artificiels qui simulent le fonctionnement du cerveau humain. Ces réseaux de neurones sont constitués de plusieurs couches de neurones interconnectés, chacune effectuant des opérations mathématiques sur les données d'entrée pour produire une sortie.

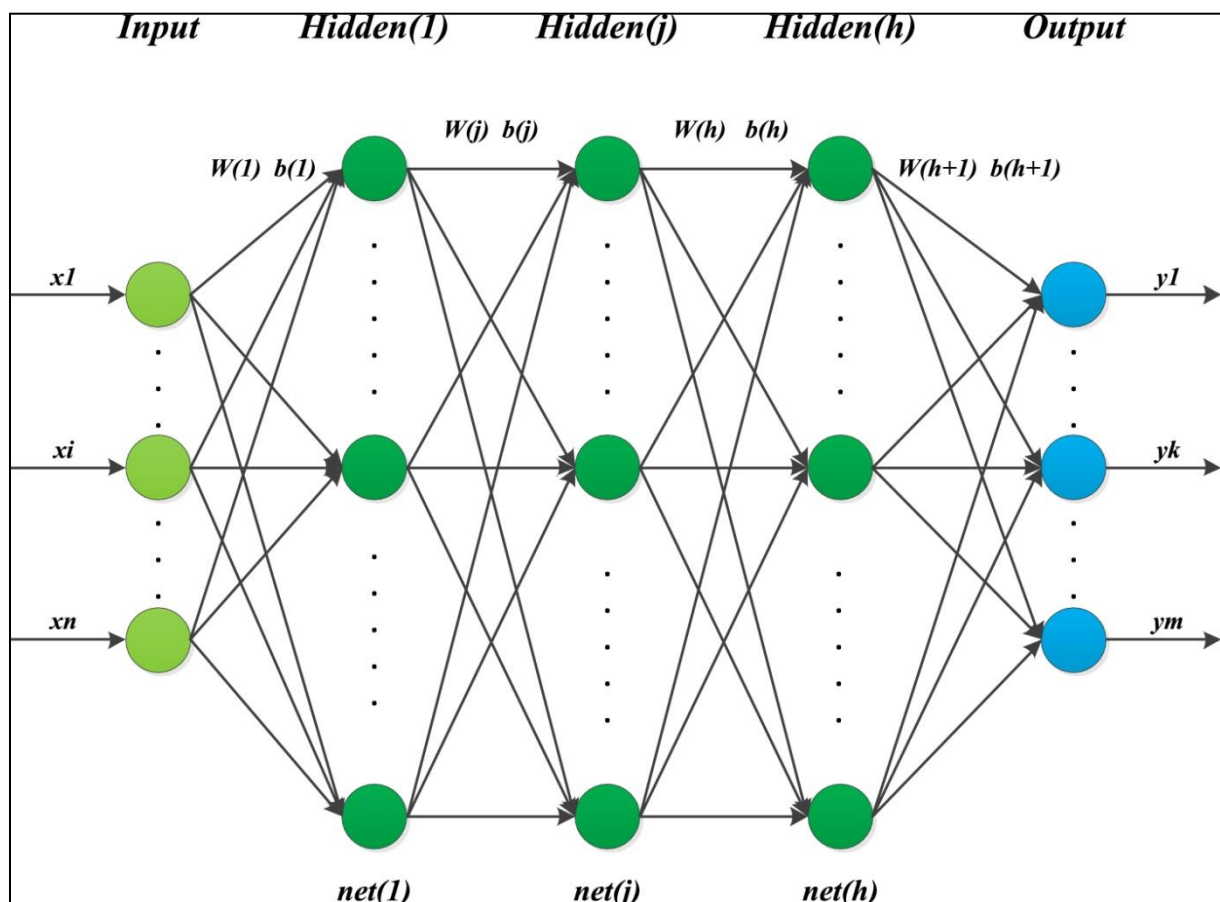


Figure 2- 1 principe de fonctionnement de réseaux de neurones

Les données d'entrée sont traitées par la première couche de neurones, qui extrait les caractéristiques basiques des données. Les sorties de cette première couche sont ensuite utilisées comme entrée pour la couche suivante, qui extrait des caractéristiques plus complexes. Ce processus se répète pour chaque couche, en augmentant graduellement la complexité des caractéristiques extraites jusqu'à la sortie finale.

Le réseau de neurones apprend à partir d'un ensemble de données d'entraînement en ajustant les poids des connexions entre les neurones pour minimiser une fonction de coût. Une fois

que le réseau a été entraîné, il peut être utilisé pour prédire la sortie pour de nouvelles données.

Le deeplearning est particulièrement adapté pour des problèmes de grande complexité et de grande dimensionnalité, tels que la reconnaissance d'images, la traduction automatique, la reconnaissance vocale, et bien d'autres encore.

1.3.Architecture pour la segmentation d'image :

L'architecture d'un réseau de neurones pour la segmentation d'image peut varier selon les besoins et les spécificités de chaque problème, mais elle suit généralement ce principe d'encodeur-décodeur relié par une passerelle.

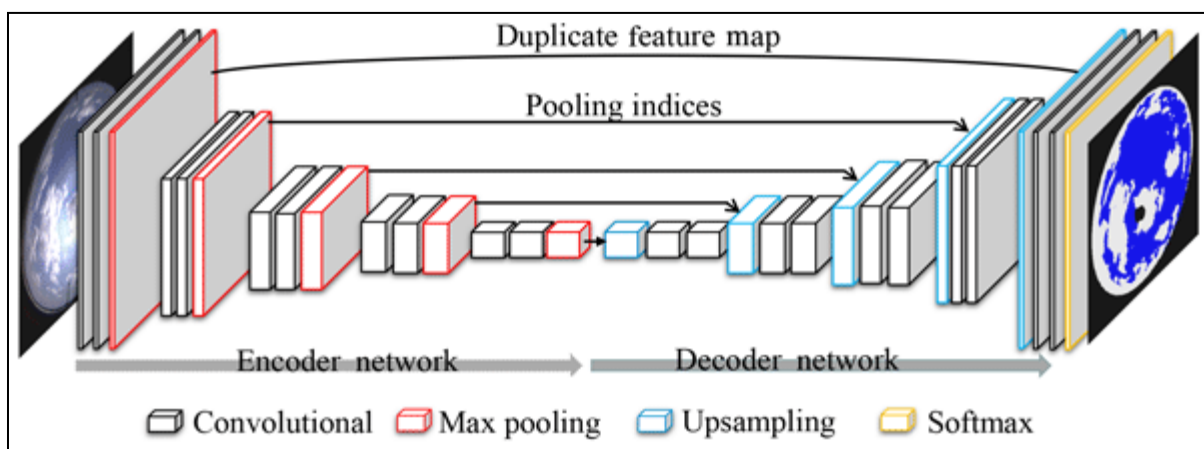


Figure 2- 2 Architecture d'un réseau de segmentation d'images [26]

L'encodeur est responsable de la réduction de la résolution spatiale de l'image d'entrée, tout en augmentant la profondeur des caractéristiques extraites. Cette réduction de la résolution spatiale se fait en général à l'aide de couches de convolution et de pooling, qui permettent de réduire les dimensions spatiales de l'image tout en conservant les informations les plus importantes.

Le décodeur, quant à lui, utilise des opérations de convolution transposée pour agrandir la résolution spatiale de la carte des caractéristiques, tout en réduisant sa profondeur. Cette étape permet de reconstruire une image segmentée de la même taille que l'image d'entrée.

La passerelle est une couche de convolution qui permet de connecter l'encodeur et le décodeur tout en préservant les informations spatiales. Cette passerelle peut être composée de plusieurs couches pour améliorer les performances du modèle.

1.4.Méthodes d'apprentissage :

1.4.1. Apprentissage automatique :

L'apprentissage automatique ou Machine Learning est une branche de l'intelligence artificielle qui vise à permettre à un système informatique d'apprendre automatiquement à partir de données, sans être explicitement programmé pour cela. Le système apprend à partir d'exemples et de modèles, et est capable de généraliser ses connaissances pour faire des prédictions sur de nouvelles données. L'apprentissage automatique est utilisé dans de nombreux domaines, tels que la reconnaissance vocale, la reconnaissance d'images, la recommandation de produits, la détection de fraudes, etc. Il existe différents types d'apprentissage automatique [16], tels que ;

- **L'apprentissage supervisé :** Il s'agit de l'utilisation d'un ensemble de données étiquetées pour entraîner un algorithme à reconnaître des modèles dans de nouvelles données non étiquetées. Le processus d'apprentissage consiste à présenter des exemples étiquetés à l'algorithme, qui ajuste ses paramètres pour minimiser l'erreur de prédiction entre les sorties attendues et les sorties produites. Les applications courantes de l'apprentissage supervisé comprennent la classification, la régression et la prédiction.

Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée, on trouve l'approche bayésienne, la régression linéaire, la machine à vecteur de support (SVM), l'arbre de décision, les forêts d'arbres décisionnels, ...

- **L'apprentissage non supervisé :** cette méthode consiste à trouver des structures ou des modèles dans les données sans utiliser d'étiquettes ou de réponses préétablies. L'algorithme est donc laissé à lui-même pour découvrir des associations et des regroupements dans les données. Les applications courantes de l'apprentissage non supervisé incluent la segmentation, la détection d'anomalies et la compression de données.

On distingue plusieurs algorithmes d'apprentissage non supervisé tels que par exemple le K-moyennes (KMeans), le FuzzyKMeans, l'Espérance-Maximisation (EM),...

- **L'apprentissage par renforcement :** cette méthode implique l'apprentissage d'un agent à interagir avec un environnement pour maximiser une récompense donnée. L'agent prend des actions pour atteindre un objectif et reçoit une récompense ou une punition en fonction de son comportement. L'objectif est de maximiser la récompense

cumulée au fil du temps. Les applications courantes de l'apprentissage par renforcement incluent les jeux, la robotique et la gestion de ressources.

1.4.2. Apprentissage profond :

L'apprentissage profond, également connu sous le nom de DeepLearning, est une branche de l'apprentissage automatique qui utilise des réseaux de neurones artificiels pour apprendre des représentations de données hiérarchiques. Contrairement à l'apprentissage machine traditionnel, qui utilise des caractéristiques prédéfinies pour représenter les données, l'apprentissage profond permet aux algorithmes d'apprendre des caractéristiques directement à partir des données.

Les réseaux de neurones profonds sont des réseaux de neurones artificiels qui comportent plusieurs couches cachées, ce qui leur permet de modéliser des relations complexes entre les données d'entrée et les sorties souhaitées. Chaque couche transforme les entrées en sorties en utilisant des poids ajustables, qui sont optimisés pendant l'entraînement pour minimiser l'erreur de prédiction.

L'apprentissage profond est utilisé dans de nombreux domaines, tels que la vision par ordinateur, la reconnaissance vocale, la traduction automatique, la génération de texte et la reconnaissance d'objets. Il a permis des avancées significatives dans ces domaines, en particulier dans la reconnaissance d'images, où les réseaux de neurones profonds ont surpassé les humains dans certains défis de classification d'images.

Parmi les algorithmes d'apprentissage profond, on peut citer les suivants sachant que ces algorithmes sont largement utilisés dans de nombreuses applications d'apprentissage profond, et il en existe de nombreuses autres variantes et extensions ; [21]

- **Réseaux de neurones convolutifs (CNN)** : les CNN sont souvent utilisés pour l'analyse d'images, la reconnaissance de formes, la classification d'images, la segmentation d'images, etc. Ils ont une architecture profonde et convolutive, qui leur permet de traiter de grandes quantités de données d'entrée et d'apprendre des caractéristiques pertinentes de manière autonome.
- **Réseaux de neurones récurrents (RNN)** : les RNN sont utilisés pour l'analyse de séquences, la reconnaissance de la parole, la traduction automatique, etc. Ils ont la capacité de prendre en compte des informations contextuelles, en utilisant une boucle de rétroaction qui leur permet de prendre en compte les sorties précédentes.

- **Réseaux de neurones générateurs adverses (GAN) :** les GAN sont utilisés pour la génération de données synthétiques réalistes, tels que des images, des vidéos, des sons, etc. Ils sont composés de deux réseaux de neurones, un générateur et un discriminateur, qui s'entraînent de manière compétitive pour produire des données synthétiques de plus en plus réalistes.
- **Réseaux de neurones autoencodeurs (AE) :** les AE sont utilisés pour la réduction de dimensionnalité, la reconstruction d'images, la détection d'anomalies, etc. Ils apprennent une représentation compressée des données d'entrée en utilisant un encodeur et un décodeur.
- **Réseaux de neurones à mémoire à court terme (LSTM) :** les LSTM sont une variante des RNN, conçue pour surmonter les problèmes de disparition du gradient et pour prendre en compte des informations à long terme. Ils sont utilisés pour la génération de texte, la modélisation de séries temporelles, etc.
- **Réseaux de neurones résiduels (ResNet) :** les ResNet sont utilisés pour la reconnaissance d'images, en particulier pour la reconnaissance d'objets dans des images de grande taille et de haute résolution. Ils utilisent des blocs de résidus pour faciliter la propagation de l'information et éviter la dégradation de la performance avec l'augmentation de la profondeur du réseau.

2. Les réseaux de neurones convolutionnels (CNN) :

2.1.Présentation :

Les réseaux de neurones convolutionnels (Convolutional Neural Networks ou CNN en anglais) sont un type de réseau de neurones artificiels qui ont révolutionné la reconnaissance d'images et la vision par ordinateur en général. Les CNN sont basés sur la structure biologique du cerveau et plus particulièrement sur le cortex visuel.

Leur architecture est caractérisée par l'utilisation de couches de convolution, qui permettent de détecter des motifs locaux dans une image en effectuant des opérations de convolution entre des filtres et l'image en entrée. Les couches de convolution sont souvent suivies de couches de pooling, qui permettent de réduire la dimension de l'image en entrée en sélectionnant les caractéristiques les plus pertinentes.

Les réseaux de neurones convolutionnels sont généralement composés de plusieurs couches convolutionnelles et de pooling, suivies de couches entièrement connectées, qui permettent de classifier les caractéristiques extraites de l'image en entrée. L'apprentissage de ces réseaux se fait de manière supervisée, en minimisant une fonction de coût grâce à la rétro propagation du

gradient.

Les CNN ont permis des avancées significatives dans de nombreux domaines, tels que la reconnaissance de visages, la détection d'objets, la segmentation d'images, ou encore la traduction automatique.

2.2.Fonctionnement et architecture :

Le fonctionnement des CNN repose sur la convolution, une opération mathématique qui permet de transformer une image en une carte de caractéristiques (featuremap) en appliquant des filtres (ou noyaux) sur l'image d'entrée. Ces filtres permettent de détecter des motifs spécifiques dans l'image, tels que des bords, des coins, des textures, etc. Les résultats de la convolution sont ensuite passés à travers une fonction d'activation, qui introduit une non-linéarité dans le modèle.

Les CNN se composent généralement de plusieurs couches de convolution, entrecoupées de couches de pooling (ou de sous-échantillonnage) qui réduisent la taille de la carte de caractéristiques, et de couches entièrement connectées qui effectuent la classification ou la segmentation finale.

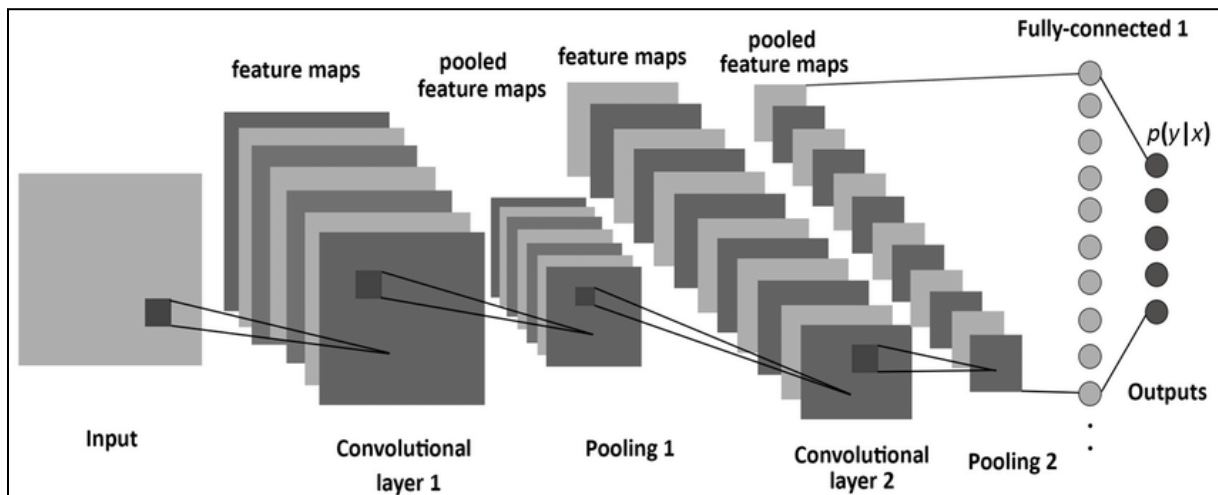


Figure 2- 3 Architecture classique d'un réseau de neurones convolutif [27]

L'architecture des CNN peut varier en fonction de la tâche à accomplir et de la complexité de l'image d'entrée. Les architectures les plus courantes sont le LeNet-5, le AlexNet, le VGGNet, le InceptionNet, et le ResNet.

Les CNN ont connu un grand succès dans de nombreuses applications de la vision par ordinateur, notamment la reconnaissance faciale, la reconnaissance d'objets, la segmentation d'images médicales, la détection d'objets, la reconnaissance de texte dans les images, et bien d'autres.

2.3.CNN pour le traitement des images :

Il existe de nombreuses applications pour les réseaux de neurones convolutionnels (CNN), on trouve principalement :

- Classification d'images : Les CNN sont couramment utilisés pour classer les images en différentes catégories, comme la classification de voitures, d'animaux, de personnes, de paysages, ...
- Détection d'objets : Les CNN sont également utilisés pour détecter des objets dans des images, en encadrant les objets d'une boîte de délimitation et en les étiquetant avec une étiquette appropriée.
- Segmentation d'images : Les CNN peuvent être utilisés pour segmenter des images, c'est-à-dire diviser une image en différentes parties pour identifier différents objets ou régions.
- Reconnaissance de visages : Les CNN peuvent également être utilisés pour la reconnaissance faciale, en apprenant à identifier différentes caractéristiques faciales telles que les yeux, le nez, la bouche, ...
- Traitement du langage naturel : Les CNN peuvent également être utilisés pour le traitement du langage naturel, en apprenant à extraire des caractéristiques importantes des mots et des phrases pour effectuer des tâches telles que la classification de texte, la traduction de texte, ...
- Diagnostic médical : Les CNN sont également utilisés dans le domaine médical pour aider à diagnostiquer des maladies en analysant des images médicales telles que les rayons X, les IRM, les scanners, ...
- Reconnaissance de caractères : Les CNN peuvent également être utilisés pour la reconnaissance de caractères manuscrits, en apprenant à identifier différentes caractéristiques des caractères tels que les boucles, les courbes, les points, ...

3. La segmentation d'images :

3.1.Présentation :

La segmentation d'images est une technique de traitement d'images qui consiste à diviser une image en plusieurs parties, appelées segments ou régions, afin d'en faciliter l'analyse et la compréhension. Cette tâche est souvent réalisée manuellement par des experts en imagerie, ce qui peut être fastidieux et coûteux en temps et en ressources.

Avec l'avènement des techniques de Deep Learning, la segmentation d'images peut désormais être effectuée automatiquement par des réseaux de neurones convolutionnels (CNN). Ces derniers sont capables d'apprendre à extraire les caractéristiques d'une image et de les utiliser pour identifier les limites et les contours des objets qui s'y trouvent.

La segmentation d'images peut avoir de nombreuses applications dans différents domaines, tels que la médecine, l'industrie, la surveillance, la robotique, etc. Elle peut être utilisée pour la détection d'objets, la reconnaissance de formes, la mesure de distances et de surfaces, la création de cartes en 3D, la surveillance de l'environnement, ...

3.2.Méthodes de segmentation d'images :

Il existe plusieurs méthodes de segmentation d'images, qui sont utilisées en fonction des caractéristiques de l'image à segmenter et des besoins spécifiques de l'application. Voici quelques-unes des méthodes les plus couramment utilisées :

- **Segmentation basée sur seuillage :** Cette méthode consiste à diviser une image en plusieurs segments en fonction des valeurs des pixels. Les pixels dont la valeur est supérieure ou inférieure à un seuil prédéfini sont attribués à des segments différents. Cette méthode est simple et rapide, mais elle peut ne pas être adaptée pour les images complexes.

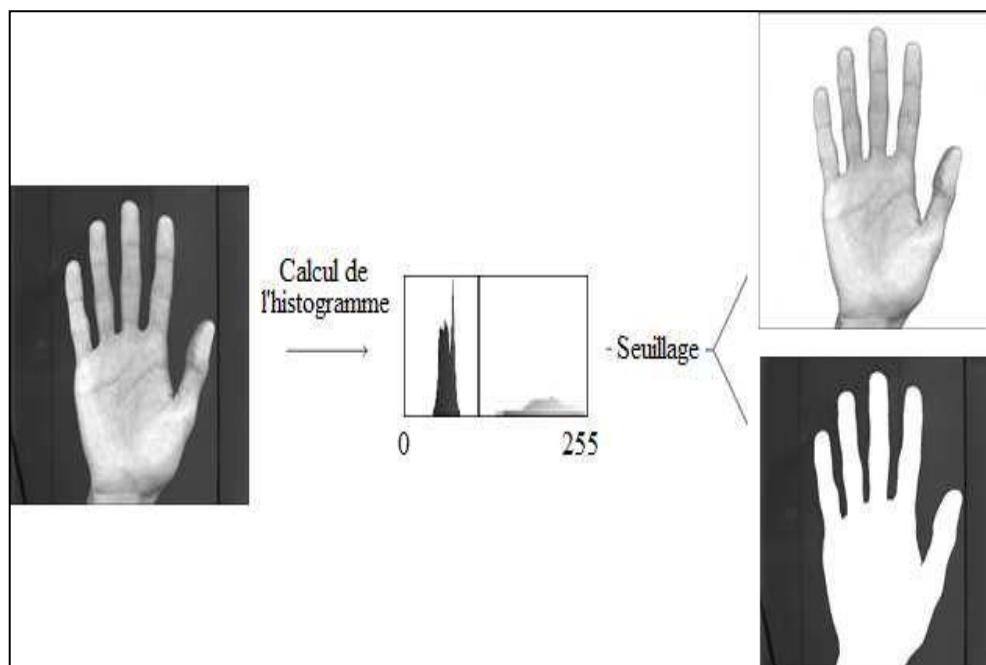


Figure 2- 4 Segmentation basée sur seuillage [28]

- **Segmentation basée sur la région** : Cette méthode utilise des informations sur la texture, la couleur et la forme des objets dans l'image pour créer des régions cohérentes. Elle commence par la segmentation des régions de l'image et leur fusion pour former des objets complets.



Figure 2- 5 Segmentation basée sur la région [29]

- **Segmentation basée sur la détection de contours** : Cette méthode détecte les contours des objets dans l'image en utilisant des algorithmes de détection de contours tels que le filtre de Sobel, le filtre de Canny, etc. Une fois que les contours ont été détectés, la segmentation peut être réalisée en utilisant des méthodes de remplissage de contour.

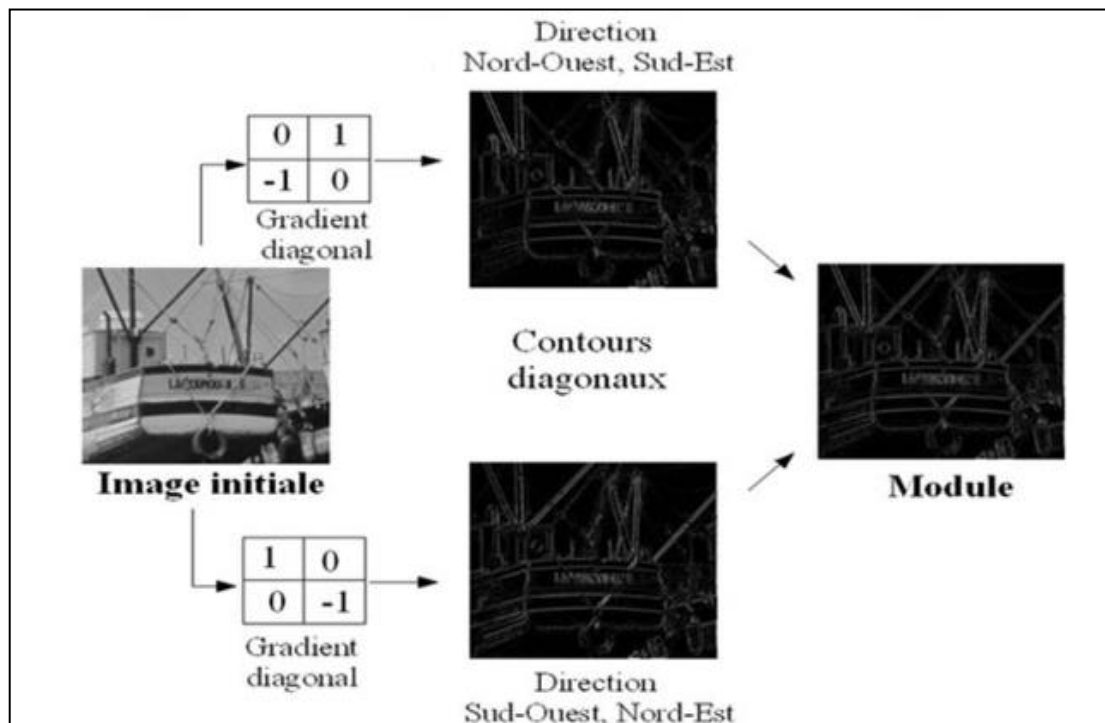


Figure 2- 6 Segmentation basée sur la détection de contours [30]

- **Segmentation basée sur l'apprentissage automatique :** Cette méthode utilise des algorithmes d'apprentissage automatique tels que les réseaux de neurones convolutifs pour apprendre à segmenter les images. Les algorithmes sont entraînés sur des données d'entraînement avec des images annotées pour prédire les classes d'objets dans de nouvelles images.

3.3.Les modèles de segmentation d'images utilisant les CNN :

Les modèles de segmentation d'images utilisant les CNN (réseaux de neurones convolutionnels) sont des algorithmes de segmentation d'images basés sur l'apprentissage en profondeur. Ils sont conçus pour détecter des régions d'intérêt dans des images en apprenant des caractéristiques à partir de données d'entraînement. Les modèles de segmentation d'images CNN sont souvent utilisés pour la segmentation sémantique, qui consiste à étiqueter chaque pixel d'une image avec une étiquette de classe correspondant à la catégorie d'objet à laquelle il appartient. Ces modèles sont très performants dans des tâches telles que la segmentation d'organes dans des images médicales, la détection de voitures dans des images satellites, et la reconnaissance d'objets dans des images de surveillance.

Parmi les modèles de segmentation d'images utilisant des CNN :

- U-Net
- Mask R-CNN
- FCN (FullyConvolutional Networks)
- DeepLab v3+
- SegNet
- PSPNet (PyramidSceneParsing Network)
- ENet (Efficient Neural Network for Real-time Semantic Segmentation)
- ICNet (Image Cascade Network)
- RefineNet
- DeepVOG (DeepVariational Optical Flow Guided Network)

Nous allons utiliser le U-net pour le reste de notre travail. Le choix de ce modèle est imposé par le cahier des charges du projet.

4. Le U-Net

4.1.Présentation :

U-Net est un modèle de segmentation d'images basé sur les réseaux de neurones convolutionnels (CNN), introduit par Olaf Ronneberger, Philipp Fischer et Thomas Brox en 2015. Le modèle a été initialement développé pour la segmentation sémantique des images biomédicales, mais il a depuis été utilisé avec succès dans diverses applications de segmentation d'images.

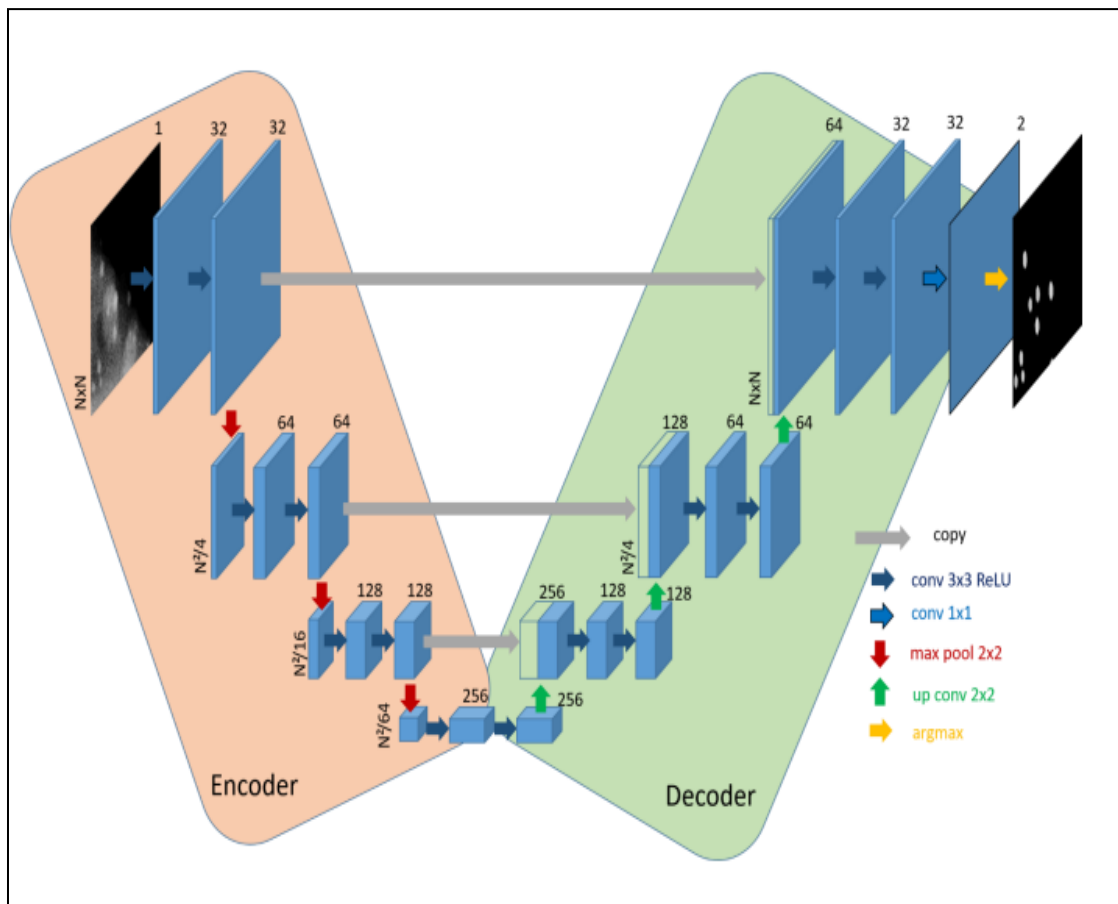


Figure 2- 7 U-net pour la segmentation d'image [31]

Ce modèle de segmentation d'images basé sur les réseaux de neurones convolutionnels qui utilise des couches de convolution pour extraire les caractéristiques de l'image, une phase de réduction pour réduire la taille de l'image et une phase de convolution inverse pour produire la carte de segmentation finale. Le modèle utilise également des connexions en saut pour améliorer la précision de la segmentation.

Il est constitué d'une série de couches de convolution qui réduisent progressivement la taille de l'image d'entrée tout en augmentant le nombre de canaux de caractéristiques. Cette phase de réduction est suivie par une phase de convolution inverse qui rétablit la taille de l'image en produisant une carte de segmentation de la même taille que l'image d'entrée.

4.2.Architecture et fonctionnement :

L'architecture U-Net se compose d'un encodeur, qui capture les caractéristiques de l'image d'entrée et les réduit en dimensionnalité, et d'un décodeur, qui utilise ces caractéristiques pour reconstruire l'image segmentée. Contrairement aux architectures classiques de CNN, U-Net utilise des connexions résiduelles pour permettre une propagation directe des informations des couches d'encodeur aux couches de décodeur correspondantes, permettant ainsi une meilleure localisation des objets dans l'image segmentée.

Le principe de base de U-Net est d'utiliser un encodeur pour extraire les caractéristiques de l'image d'entrée, puis un décodeur pour produire une carte de segmentation pixel par pixel. L'encodeur est constitué de couches de convolutions et de max-pooling qui réduisent la résolution spatiale de l'image et augmentent le nombre de canaux de la carte de caractéristiques. Le décodeur utilise des couches de déconvolution pour augmenter la résolution spatiale et des couches de concaténation pour fusionner les cartes de caractéristiques de l'encodeur avec les cartes de caractéristiques du décodeur.

L'architecture en forme de U du réseau permet de combiner à la fois des informations contextuelles et spatiales pour produire des résultats de segmentation précis. Cette architecture est souvent utilisée pour la segmentation d'images biomédicales, telles que la segmentation de cellules, d'organes et de lésions, mais elle peut également être appliquée à d'autres domaines, tels que la segmentation d'images satellites et la segmentation de documents.

4.3.U-Net pour la segmentation des tatouages numériques :

L'approche U-Net peut être utilisée pour détecter et segmenter des tatouages numériques dans des images. La segmentation des tatouages numériques est importante pour la reconnaissance de contenu d'image, l'authentification d'images, la protection de la vie privée et la détection de manipulations d'images.

Le modèle U-Net permet d'obtenir des résultats précis en segmentant les tatouages numériques dans les images avec une précision élevée et une faible erreur de segmentation. Les avantages de cette approche comprennent une haute précision de segmentation, une faible erreur de segmentation, une réduction du temps de traitement et une meilleure robustesse à la variation des images.

En utilisant le modèle U-Net pour la segmentation des tatouages numériques, il est possible de détecter les tatouages numériques dans des images et de les segmenter avec précision. Cela

peut aider à protéger la vie privée des individus, à détecter la falsification d'images et à améliorer la reconnaissance de contenu d'image.

Conclusion :

En conclusion, la segmentation d'images est un domaine clé de la vision par ordinateur qui permet de résoudre de nombreux problèmes, tels que la détection d'objets, la reconnaissance de formes, la détection de contours. Parmi les méthodes de segmentation, l'approche basée sur les réseaux de neurones convolutionnels (CNN) a connu un grand succès ces dernières années, en particulier avec l'utilisation de l'architecture U-Net. Cette architecture se distingue par sa capacité à traiter des images de taille variable et à obtenir des résultats précis pour la segmentation d'images complexes, telles que les images de tatouages numériques. L'utilisation de l'approche U-Net pour la segmentation d'images présente de nombreux avantages, notamment une grande précision, une vitesse de traitement élevée et une flexibilité en termes de taille d'image. Dans ce qui suit de ce rapport nous allons focaliser nos études sur cette approche.

Chapitre3

Expérimentations

Chapitre3

Expérimentations

Introduction

Dans ce chapitre, nous allons étudier la notion de tatouage numérique, les méthodes d'attaques des images tatouées et les méthodes l'intelligence artificielle pour la détection des images tatoués.

1. Outils de développement :

1.1.Google Colab :

Google Colaboratory, également connu sous le nom de "Google Colab", est un environnement de développement de machine Learning basé sur le Cloud et gratuit. Il permet aux utilisateurs d'écrire et d'exécuter du code Python dans un environnement Jupyter Notebook, sans avoir à installer de logiciel sur leur ordinateur local. Colab utilise des machines virtuelles hébergées par Google qui offrent un accès gratuit à des ressources de calcul, notamment des processeurs, des GPU et des TPU, pour des tâches de machine Learning et d'apprentissage profond. Les utilisateurs peuvent également partager leurs notebooks Colab avec d'autres utilisateurs pour la collaboration en temps réel.



Figure 3- 1 Logo Google Colab [32]

1.2.Langage Python :

Python est un langage de programmation interprété, conçu pour être facile à lire et à écrire. Il est utilisé pour développer des applications dans une grande variété de domaines, tels que la science des données, l'apprentissage automatique, la visualisation de données, la création de sites Web, l'automatisation de tâches, etc.

Python est maintenant l'un des langages de programmation les plus populaires au monde, grâce à sa simplicité et sa polyvalence. Il est apprécié pour sa syntaxe concise et claire, qui facilite la lecture et la maintenance du code, ainsi que pour sa grande variété de bibliothèques et de Framework disponibles pour faciliter le développement de projets.

Python est un langage interprété, ce qui signifie que le code source est exécuté directement par l'interpréteur Python, sans nécessiter de compilation préalable. Cela rend le processus de développement plus rapide et plus souple, car les développeurs peuvent tester leur code en temps réel et effectuer des modifications rapidement.

Python est également connu pour sa communauté active et engagée, qui contribue à la création et à la maintenance de nombreuses bibliothèques et Framework open source, tels que NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch et bien d'autres. Ces bibliothèques ont grandement contribué à faire de Python un langage de choix pour le développement d'applications de science des données et d'apprentissage automatique.



Figure 3- 2 Logo Google Python[33]

2. Bibliothèques pour la préparation des données :

2.1.NumPy :

NumPy est une bibliothèque open-source de calcul numérique pour Python. Elle fournit des objets de tableaux multidimensionnels hautement performants et des fonctions pour travailler avec ces tableaux. Les tableaux NumPy permettent de représenter et de manipuler facilement des données numériques telles que des images, des sons et des données scientifiques. Les tableaux NumPy sont souvent utilisés en conjonction avec des bibliothèques de visualisation de données telles que Matplotlib et des bibliothèques d'apprentissage automatique telles que TensorFlow et PyTorch.

NumPy est largement utilisé dans le domaine du calcul scientifique et de l'apprentissage automatique en raison de sa rapidité, de sa facilité d'utilisation et de sa capacité à effectuer des calculs sur des tableaux de grandes dimensions.



Figure 3- 3 Logo NumPy [34]

2.2.OpenCV :

OpenCV (Open Source Computer Vision Library) est une bibliothèque open-source de traitement d'images et de vision par ordinateur. Elle offre un grand nombre d'algorithmes de traitement d'images et de vidéos, ainsi que des fonctions pour l'analyse et la reconnaissance de formes. Elle est écrite en C++ mais dispose de bindings pour d'autres langages de programmation tels que Python. OpenCV est largement utilisée dans la recherche et l'industrie pour des applications telles que la reconnaissance de caractères, la surveillance vidéo, la vision industrielle, la réalité augmentée, la robotique, etc.



Figure 3- 4 Logo OpenCV [35]

2.3.PIL (Python Imaging Library) :

PIL (Python Imaging Library) est une bibliothèque Python utilisée pour le traitement d'images. Elle permet de créer, ouvrir, modifier et enregistrer différents formats d'images tels que JPEG, BMP, PNG, GIF, TIFF, etc. La bibliothèque PIL fournit des outils pour effectuer diverses opérations sur les images, telles que la conversion de couleurs, le redimensionnement, la rotation, le recadrage, le masquage, ...

PIL est une bibliothèque très utilisée pour le traitement d'images en Python en raison de sa facilité d'utilisation et de sa compatibilité avec d'autres bibliothèques telles que NumPy et OpenCV. En outre, PIL est une bibliothèque open-source, ce qui permet aux utilisateurs de l'utiliser gratuitement et de la personnaliser selon leurs besoins.



Figure 3- 5 Logo PIL [36]

3. Outils pour l'implémentation d'U-Net :

3.1.TensorFlow :

TensorFlow est une plateforme open-source développée par Google pour construire et déployer des modèles d'apprentissage automatique. C'est l'un des Framework de deep Learning les plus populaires utilisés par les chercheurs et les développeurs pour créer des réseaux neuronaux pour différentes tâches, y compris la reconnaissance d'images et de la parole, le traitement du langage naturel et bien plus encore. TensorFlow est conçu pour fournir un environnement flexible et évolutif pour la construction et la formation de modèles d'apprentissage automatique, avec prise en charge de plusieurs langages de programmation, y compris Python, C++, Java et plus encore.

L'une des principales caractéristiques de TensorFlow est sa capacité à créer et à entraîner des modèles de réseaux neuronaux complexes à l'aide de graphes de flux de données. La plateforme fournit une large gamme de couches de réseaux neuronaux prédéfinies, de fonctions d'activation et d'algorithmes d'optimisation, qui peuvent être facilement combinés et personnalisés pour créer des modèles très précis pour diverses applications.

TensorFlow offre également plusieurs APIs et bibliothèques de haut niveau, notamment Keras, TensorFlowEstimators et TensorFlowDatasets, qui facilitent la création et la formation de modèles d'apprentissage automatique pour les développeurs. De plus, TensorFlow prend en charge le calcul distribué, permettant aux utilisateurs de former des modèles sur plusieurs dispositifs et machines, ce qui est utile pour augmenter l'échelle des modèles volumineux et les former plus rapidement.

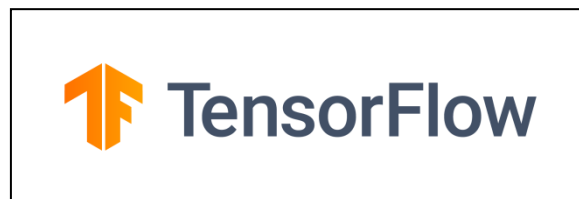


Figure 3- 6 Logo TensorFlow [37]

3.2.Keras :

Keras est une bibliothèque open source d'apprentissage en profondeur qui permet de créer et de former des modèles de réseaux de neurones artificiels.

Keras fournit une interface de haut niveau pour des bibliothèques de calculs mathématiques sous-jacentes telles que TensorFlow, Theano et Microsoft Cognitive Toolkit. Il a été conçu pour faciliter et accélérer le processus de développement de modèles de réseaux de neurones en fournissant une API simple et intuitive pour la création et la formation de modèles.

Keras prend en charge une variété de types de modèles, tels que les réseaux de neurones convolutionnels (CNN), les réseaux de neurones récurrents (RNN), les réseaux de neurones récurrents avec mémoire à court terme (LSTM) et les réseaux de neurones à convolutions 1D et 2D. Il offre également un large éventail de fonctions d'activation, de fonctions de perte et d'optimiseurs pour personnaliser les modèles selon les besoins spécifiques de chaque application.

Keras est également conçu pour être facile à utiliser et à comprendre, ce qui en fait une bibliothèque populaire. Il permet également une certaine flexibilité dans le choix du backend de calcul, de sorte que les utilisateurs peuvent choisir le backend de leur choix, tel que TensorFlow ou Theano.



Figure 3- 7 Logo Keras[38]

3.3.PyTorch :

PyTorch est une plate-forme open-source de deep Learning développée principalement par Meta. Il s'agit d'un Framework Python qui permet de construire des modèles de machine learning et de deeplearning. PyTorch est très apprécié par les chercheurs en intelligence artificielle pour sa simplicité et sa flexibilité.

L'un des avantages de PyTorch est sa capacité à fournir des graphes de calcul dynamiques, qui permettent aux utilisateurs de modifier les graphes de calcul au moment de l'exécution, contrairement à TensorFlow où les graphes sont statiques. Cette fonctionnalité facilite le débogage des modèles de deep Learning, car les erreurs peuvent être facilement localisées et corrigées.

PyTorch propose également une large gamme de modules de réseaux de neurones pré-construits, d'optimiseurs, de fonctions d'activation et de couches qui peuvent être facilement combinés pour créer des architectures de modèles complexes. De plus, PyTorch est très flexible et permet aux utilisateurs de concevoir des architectures de modèles personnalisées et de créer des modules de réseaux de neurones personnalisés.

PyTorch a également une forte communauté de développeurs, ce qui signifie qu'il existe de nombreuses ressources disponibles pour l'apprentissage et la résolution de problèmes. En outre, PyTorch supporte l'apprentissage profond distribué sur des clusters de serveurs pour accélérer l'entraînement des modèles de grande envergure.



Figure 3- 8 Logo PyTorch [39]

4. Source des données :

Afin d'implémenter notre solution et évaluer un modèle d'apprentissage automatique nous avons utilisé le DataSet TinyImageNet. La collection de données est disponible via le lien suivant (<https://paperswithcode.com/dataset/tiny-imagenet>). Il s'agit d'un dossier contenant un ensemble des images de taille 64x64 pixels comme le montre la figure suivante.

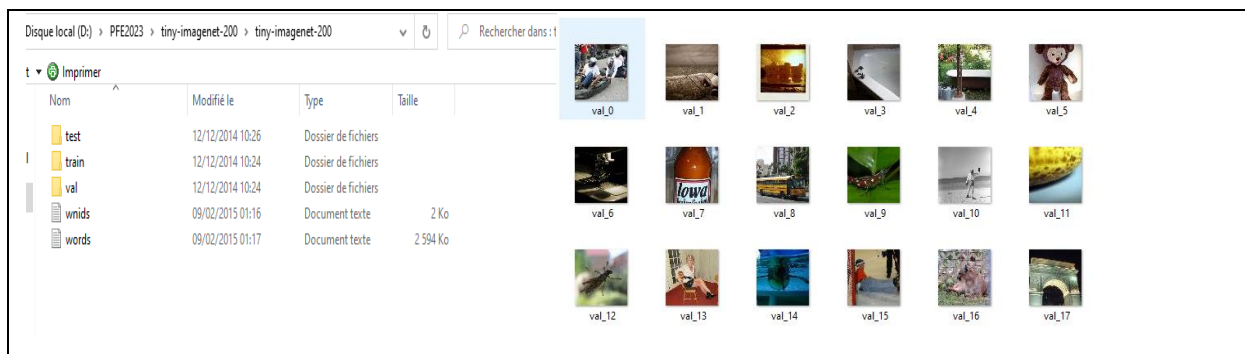


Figure 3- 9 TinyImageNetDataset

Cet ensemble de données contient 200 classes d'images avec 500 images par classe soit un total de 100 000 images. Chaque classe possède 500 images d'entraînement, 50 images de validation et 50 images de test. [45]

Nous allons utiliser ce DataSet du fait qu'il est utilisé comme un ensemble de données de référence pour les tâches de classification d'images et est souvent utilisé pour évaluer la performance de nouveaux modèles de réseaux de neurones convolutifs.

5. Tatouage des images :

Nous avons tatoué l'ensemble de 10000 images dans le DataSet. Le processus de tatouage est le suivant :

- 1- On lit l'image à partir de dataSET enregistré sur Google Drive.
- 2- On fait le redimensionnement de l'image en 256x256 pixels.

- 3- On définit le texte de tatouage, la forme et l'emplacement dans l'image.
L'emplacement est généré d'une façon aléatoire.
- 4- On écrit le texte sur l'image.
- 5- On convertie l'image pour faciliter le traitement ultérieur.
- 6- On enregistre l'image dans un dossier séparé avec les images tatouées.

```

# Définir la police et la taille de texte
font = ImageFont.truetype(font_path, 8)
# Définir le texte à tatouer
text = "PFE SAMEH 2023 - TATOUAGE NUMÉRIQUE"
# Taille de police
font_size = 8

# Parcours de tous les fichiers du dossier
for filename in os.listdir(folder_path):
    # Chemin complet de l'image
    image_path = os.path.join(folder_path, filename)

    # Charger l'image
    image = Image.open(image_path).convert("RGBA")

    # Redimensionner l'image
    new_image = image.resize((250, int(image.height * (250 / image.width))))
    h, w = new_image.size

    # Définir la police et la taille de texte
    font = ImageFont.truetype(font_path, font_size)

    # Dessiner le texte sur l'image
    draw = ImageDraw.Draw(new_image)
    text_width, text_height = draw.textsize(text, font)
    draw.text((random.randint(0,w//6),random.randint(0,h//6)),
              str(text), fill=(255,255,255), font=font) #Afficher le tatouage en haut de l'image

    # Enregistrer l'image résultante
    output_path = os.path.join(image_tatouee_path, f"tatouage_{filename}")
    result = cv2.cvtColor(np.array(new_image), cv2.COLOR_RGBA2BGRA)
    cv2.imwrite(output_path, result)

```

Figure 3- 10 Tatouage de l'image

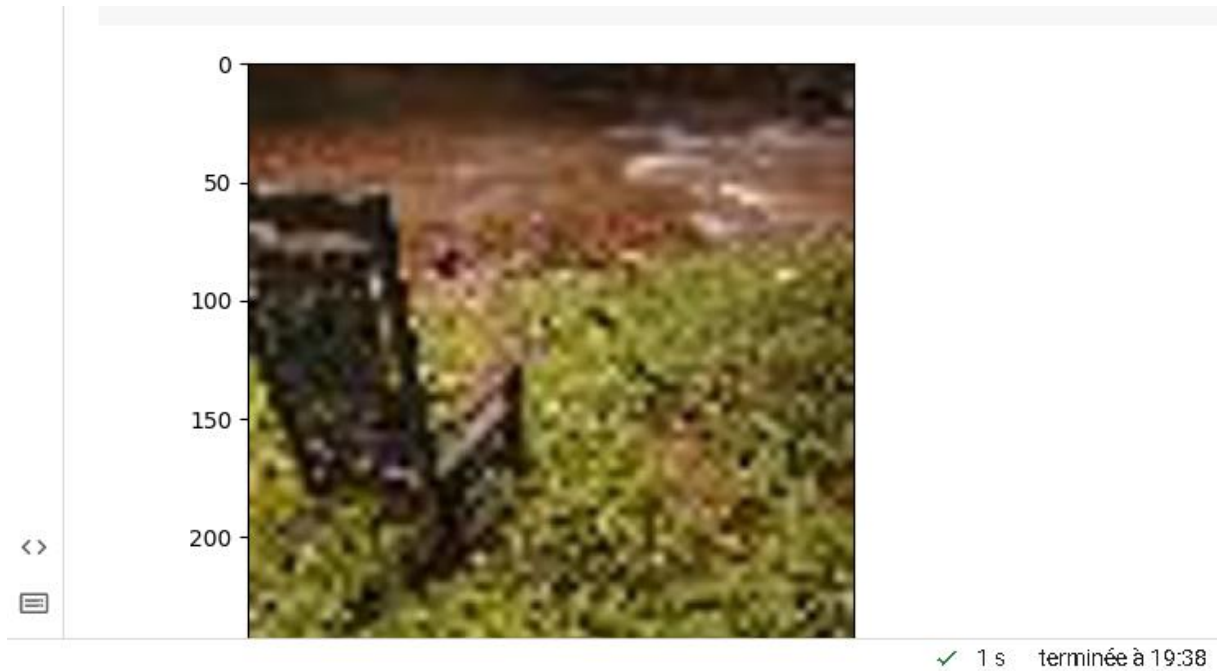


Figure 3- 11 Affichage d'une Image non tatouée

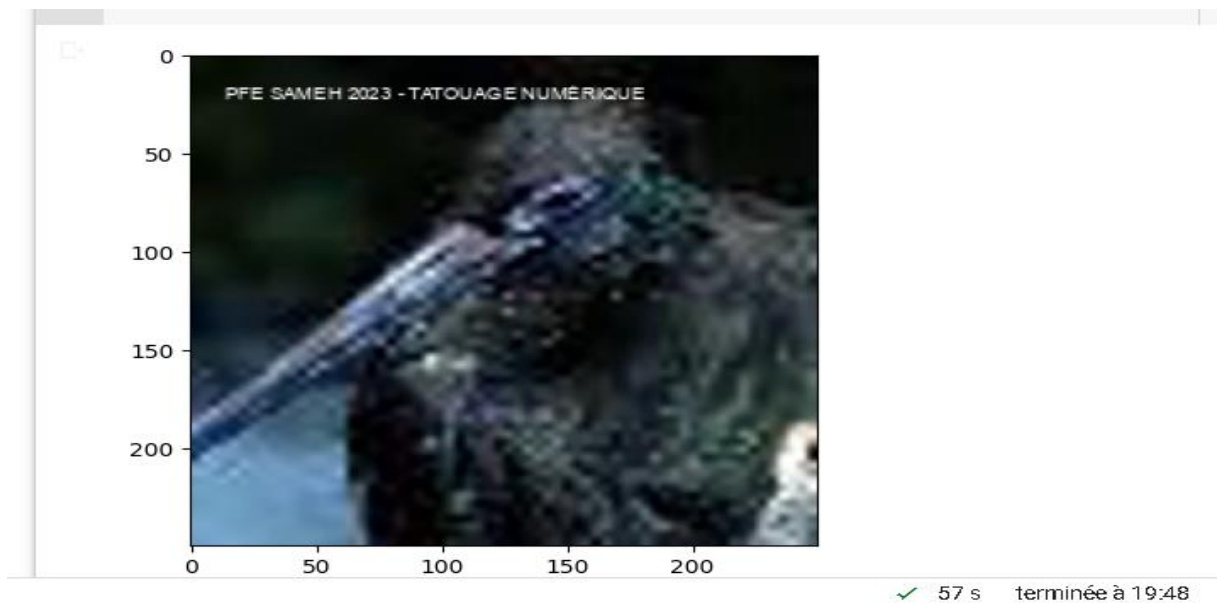


Figure 3- 12 Affichage d'exemple d'une image tatouée

6. Implémentation de modèle U-Net :

Pour l'implémentation de modèle U-net, nous avons commencé par la mise en place de modèle U-net. Le modèle prend en entrée image d'entrée au modèle de taille 256x256 en 3 couleurs.

Le modèle commence par un initialiseur aléatoire, par la suite il construit une série des couches.

L'organigramme de code de tatouage des images est donné par la figure suivante :

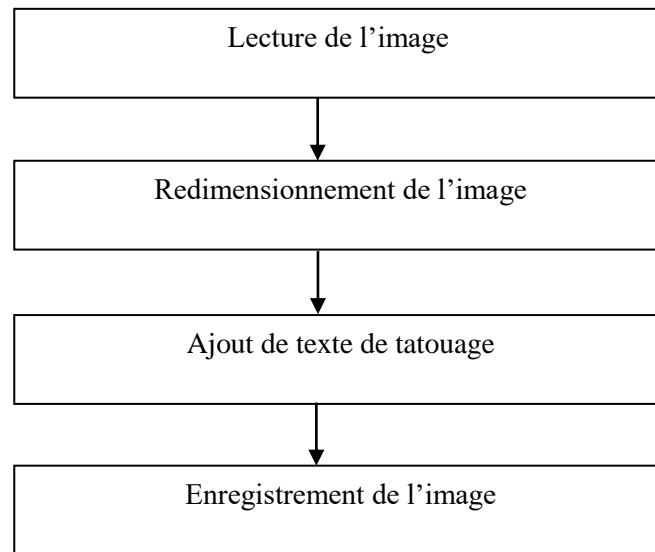


Figure 3- 13 Organigramme de code de tatouage

L'organigramme de code de création de modèle U-Net est donné par la figure suivante ;

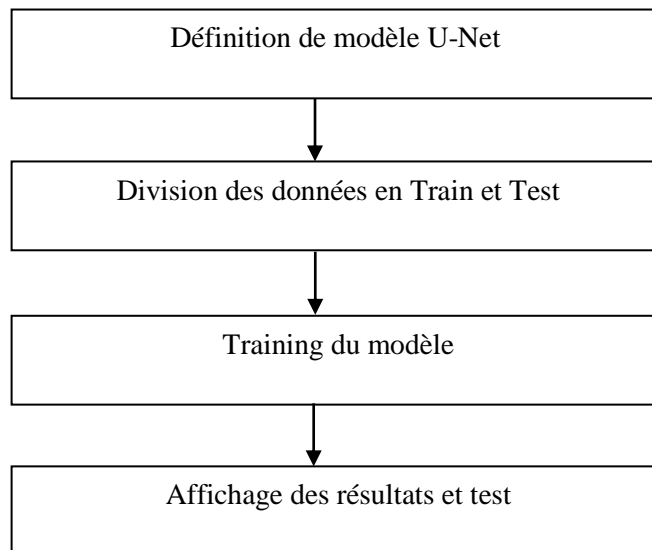


Figure 3- 14 Organigramme de code de création de modèle U-Net

Ce modèle est défini dans la figure suivante ;

```
def UnetModel():
    inputs = Input(shape=[256, 256, 3])
    initializer = tf.keras.initializers.RandomNormal(mean=0.0, stddev=0.02, seed=None)
    conv1 = Conv2D(64, 3, activation='relu', padding='same', kernel_initializer=initializer)(inputs)
    conv1 = Conv2D(64, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(128, 3, activation='relu', padding='same', kernel_initializer=initializer)(pool1)
    conv2 = Conv2D(128, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

    conv3 = Conv2D(256, 3, activation='relu', padding='same', kernel_initializer=initializer)(pool2)
    conv3 = Conv2D(256, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

    conv4 = Conv2D(512, 3, activation='relu', padding='same', kernel_initializer=initializer)(pool3)
    conv4 = Conv2D(512, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv4)
    drop4 = Dropout(0.5)(conv4)
    pool4 = MaxPooling2D(pool_size=(2, 2))(drop4)

    conv5 = Conv2D(1024, 3, activation='relu', padding='same', kernel_initializer=initializer)(pool4)
    conv5 = Conv2D(1024, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv5)
    drop5 = Dropout(0.5)(conv5)

    up6 = Conv2D(512, 2, activation='relu', padding='same', kernel_initializer=initializer)(
        UpSampling2D(size=(2, 2))(drop5))
    merge6 = concatenate([drop4, up6], axis=3)
    conv6 = Conv2D(512, 3, activation='relu', padding='same', kernel_initializer=initializer)(merge6)
    conv6 = Conv2D(512, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv6)

    up7 = Conv2D(256, 2, activation='relu', padding='same', kernel_initializer=initializer)(
        UpSampling2D(size=(2, 2))(conv6))
    merge7 = concatenate([conv3, up7], axis=3)
    conv7 = Conv2D(256, 3, activation='relu', padding='same', kernel_initializer=initializer)(merge7)
    conv7 = Conv2D(256, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv7)

    up8 = Conv2D(128, 2, activation='relu', padding='same', kernel_initializer=initializer)(
        UpSampling2D(size=(2, 2))(conv7))
    merge8 = concatenate([conv2, up8], axis=3)
    conv8 = Conv2D(128, 3, activation='relu', padding='same', kernel_initializer=initializer)(merge8)
    conv8 = Conv2D(128, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv8)

    up9 = Conv2D(64, 2, activation='relu', padding='same', kernel_initializer=initializer)(
        UpSampling2D(size=(2, 2))(conv8))
    merge9 = concatenate([conv1, up9], axis=3)
    conv9 = Conv2D(64, 3, activation='relu', padding='same', kernel_initializer=initializer)(merge9)
    conv9 = Conv2D(64, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv9)
    conv9 = Conv2D(2 * 6, 3, activation='relu', padding='same', kernel_initializer=initializer)(conv9)
    conv10 = Conv2D(3, 1, activation='softmax')(conv9)

    return Model(inputs=inputs, outputs=conv10)
```

Figure 3- 15 Modèle U-net

A ce stade, nous avons divisé notre collection des images en 2 parties (80% pour l'entraînement et 20% pour la validation) comme le montre la figure suivante ;

```
datagen = ImageDataGenerator(rescale=1. / 255, validation_split=0.2)
train_generator = datagen.flow_from_directory(
    directory='/content/drive/MyDrive/TinyImagetNET/',
    target_size=(image_height, image_width),
    batch_size=batch_size,
    class_mode='input',
    subset='training'
)
validation_generator = datagen.flow_from_directory(
    directory='/content/drive/MyDrive/TinyImagetNET/',
    target_size=(image_height, image_width),
    batch_size=batch_size,
    class_mode='input',
    subset='validation'
)
```

Figure 3- 16 Entraînement Vs Validation

Nous avons par la suite entraîné notre modèle comme le montre la figure suivante ;

```
epochs=epochs)

... Epoch 1/10
1865/1998 [=====>..] - ETA: 14:03 - loss: 0.5244 - accuracy: 0.7738 - precision: 0.2798

Exécution (3 h 17 min 41 s)... <cell line: 2> > fit() > error_handler() > __call__() > _call() > _call__() > _call_flat() > call() > quick_execute()
```

Figure 3- 17 Entraînement de modèle

7. Résultats obtenus :

Après l'entraînement de notre modèle nous avons obtenu les résultats à partir des mesures de performance calculées à chaque epoch. Le taux de précision (accuracy) est donné par la figure suivante :

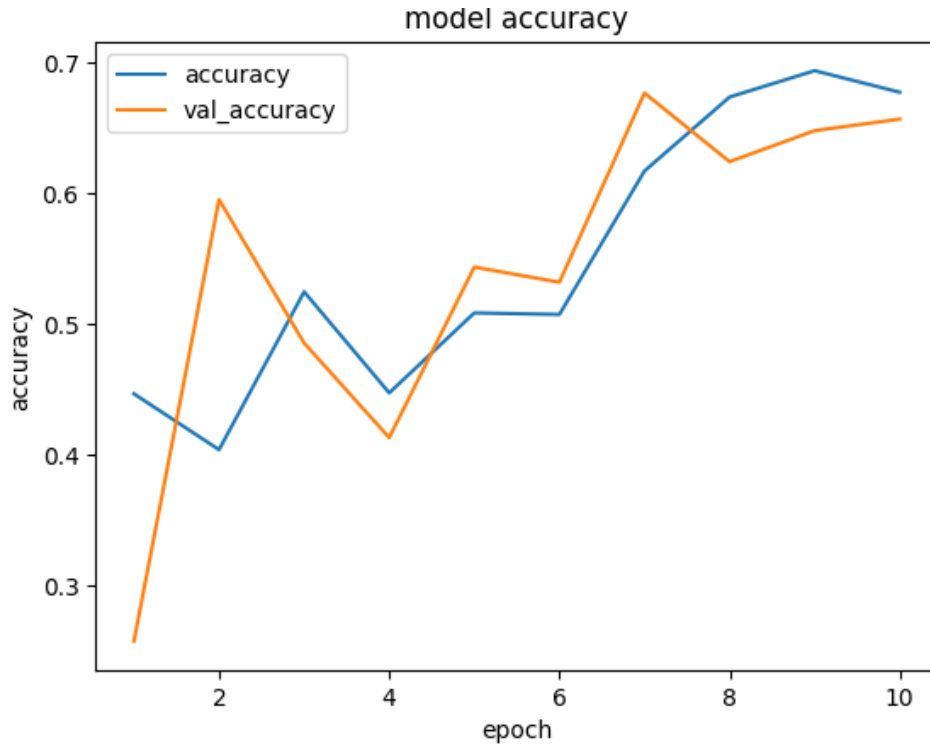


Figure 3- 18 Accuracy de modèle U-Net

La valeur de la fonction de perte est donnée par la figure suivante :

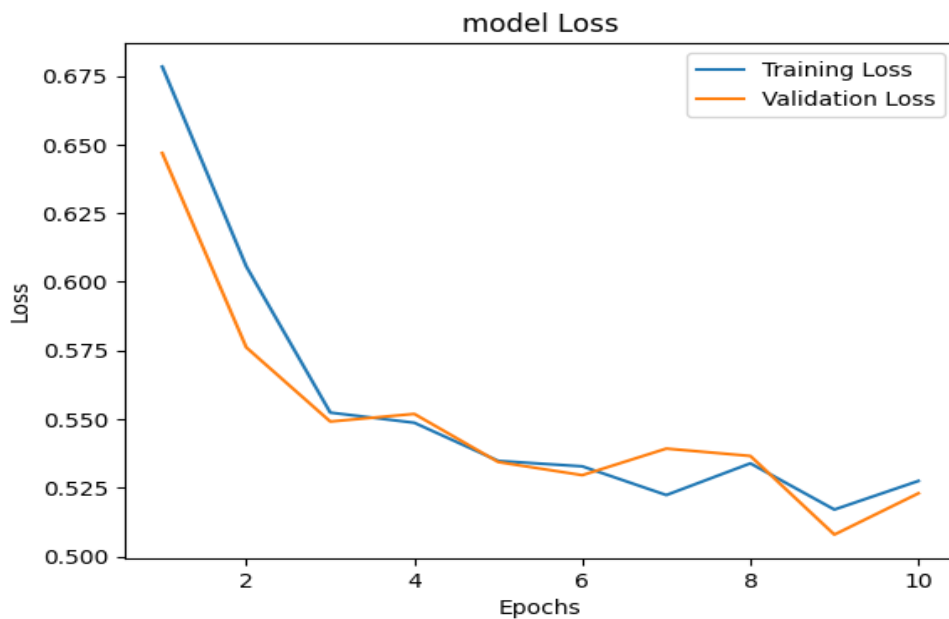


Figure 3- 19 Loss de modèle U-Net

Conclusion

En conclusion, la segmentation d'images est une technique importante pour extraire des informations utiles à partir d'images. Les réseaux de neurones convolutionnels, en particulier l'architecture UNet, ont montré de très bons résultats dans la segmentation d'images, notamment pour la détection de tatouages numériques. La mise en œuvre de ces réseaux peut être facilitée grâce à des outils tels que TensorFlow, Keras, PyTorch, NumPy, OpenCV et PIL. L'utilisation de ces bibliothèques peut être effectuée sur différentes plateformes, y compris Google Colab.

Conclusion générale

En conclusion, ce rapport a présenté une étude approfondie sur la détection de tatouage numérique en utilisant l'approche U-Net. Nous avons examiné les principes et techniques du tatouage numérique multi-bits, ainsi que les méthodes d'attaque des images tatouées. Ensuite, nous avons présenté notre projet de détection de tatouage numérique en utilisant l'architecture U-Net.

L'implémentation du modèle U-Net a permis d'obtenir des résultats encourageants. L'entraînement du modèle a montré une progression de la performance au fil des epochs, avec une diminution de la perte et une amélioration de l'exactitude et de la précision. Cependant, il est important de noter que les performances obtenues dépendent de la qualité et de la diversité des données d'entraînement.

Malgré les résultats prometteurs, il reste des pistes d'amélioration. Il serait intéressant d'explorer des techniques de prétraitement plus avancées, d'augmenter la taille de l'ensemble de données et d'optimiser les paramètres du modèle.

Finalement, ce rapport a contribué à l'étude de la détection de tatouage numérique en utilisant l'approche U-Net, offrant des perspectives intéressantes pour la sécurisation des images et la protection de la propriété intellectuelle. Il ouvre la voie à de futures recherches et développements dans ce domaine passionnant.

Webographie

- [1] <http://eprints.univ-batna2.dz/653/1/sce%20GOLEA%20NOUR%20EL%20HOUDA.pdf> [Consulté le 14/02/2023]
- [2] <https://apcpedagogie.com/quest-ce-que-le-watermarking/> [Consulté le 12/02/2023]
- [3] https://www.lirmm.fr/~chaumont/download/cours/tatouage/01_tatouage_4_transparents_par_page.pdf [Consulté le 13/02/2023]
- [4] http://biblio.univ-antananarivo.mg/pdfs/ratsimbazafyTsioryH_ESPA_MAST_16.pdf [Consulté le 14/02/2023]
- [5] <https://fr.theastrologypage.com/digital-watermarking> [Consulté le 14/02/2023]
- [6] <https://core.ac.uk/download/pdf/15495171.pdf> [Consulté le 14/02/2023]
- [7] <https://mmagister.univ-setif.dz/images/facultes/TEC/2014/Farouk%20ZEHDA.pdf> [Consulté le 14/02/2023]
- [8] <https://core.ac.uk/download/pdf/15486467.pdf> [Consulté le 14/02/2023]
- [9] http://www.univ-usto.dz/theses_en_ligne/doc_num.php?explnum_id=2801 [Consulté le 14/02/2023]
- [10] <https://core.ac.uk/download/pdf/19937961.pdf> [Consulté le 14/02/2023]
- [11] <https://www.lirmm.fr/~chaumont/download/cours/tatouage/JPO-Mars2009-tatouage.pdf> [Consulté le 14/02/2023]
- [12] https://www.researchgate.net/publication/309674087_Technique_Hybride_de_Compression_pour_le_Tatouage_des_Images [Consulté le 15/02/2023]
- [13] https://theses.hal.science/file/index/docid/668566/filename/Mathon_-_DA_veloppement_de_mA_thodes_de_tatouage_sA_res_pour_le_traA_age_de_contenus_multim_A_dia.pdf [Consulté le 16/02/2023]
- [14] <https://hal.science/hal-03696016/document> [Consulté le 16/02/2023]
- [15] <https://www.cairn.info/revue-les-cahiers-du-numerique-2003-3-page-135.htm> [Consulté le 16/02/2023]
- [16] <https://course.elementsofai.com/fr/4/1> [Consulté le 16/02/2023]

-
- [17] <https://datascientest.com/apprentissage-automatique> [Consulté le 16/02/2023]
- [18] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501333-deep-learning-definition-et-principes-de-l-apprentissage-profond/> [Consulté le 16/02/2023]
- [19] <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/> [Consulté le 16/02/2023]
- [20] <https://datascientest.com/deep-learning-definition#:~:text=Le%20Deep%20Learning%20ou%20apprentissage,%C3%A0%20l'%C3%A9tablissement%20des%20r%C3%A8gles.> [Consulté le 16/02/2023]
- [21] <https://www.jedha.co/formation-ia/algorithmes-deep-learning> [Consulté le 16/02/2023]
- [22] <https://www.stemmer-imaging.com/fr-ch/conseil-technique/apprentissage-automatique-et-apprentissage-profond/> [Consulté le 16/02/2023]
- [23] <https://datascientest.com/u-net> [Consulté le 17/02/2023]
- [24] https://nchlis.github.io/2019_10_30/page.html [Consulté le 17/02/2023]
- [25] <https://www.sciencedirect.com/science/article/pii/S2352152X21011270> [Consulté le 28/04/2023]
- [26] <https://amt.copernicus.org/articles/13/1953/2020/> [Consulté le 28/04/2023]
- [27] https://www.researchgate.net/figure/Architecture-classique-dun-reseau-de-neurones-convolutif-Une-image-est-fournie-en_fig5_330995099 [Consulté le 28/04/2023]
- [28] https://www.researchgate.net/figure/Segmentation-par-seuillage-a-partir-de-lhistogramme-des-niveaux-de-gris-dune-image_fig3_261401416 [Consulté le 28/04/2023]
- [29] <https://slideplayer.com/slide/4955917/> [Consulté le 28/04/2023]
- [30] <https://slideplayer.fr/slide/1296291/> [Consulté le 28/04/2023]
- [31] https://www.creatis.insa-lyon.fr/~grenier/wp-content/uploads/teaching/UNet/UNet_InBrief.pdf [Consulté le 28/04/2023]
- [32] <https://ruslanmv.com/assets/images/posts/2022-01-24-How-to-connect-Google-Colab-to-your-computer/colab.png> [Consulté le 04/05/2023]
- [33] <https://python.developpez.com/tutoriels/debuter-avec-python-au-lycee/?page=comment-suivre-ce-tutoriel> [Consulté le 04/05/2023]

-
- [34] <https://en.wikipedia.org/wiki/NumPy> [Consulté le 04/05/2023]
- [35] https://content.axopen.com/uploads/opencv_logo_62fb531c30.png [Consulté le 04/05/2023]
- [36] <https://raw.githubusercontent.com/python-pillow/pillow-logo/main/pillow-logo-dark-text-1280x640.png> [Consulté le 04/05/2023]
- [37] <https://github.com/tensorflow/tensorflow> [Consulté le 04/05/2023]
- [38] <https://www.actuia.com/keras/> [Consulté le 04/05/2023]
- [39] <https://www.kernix.com/pytorch-le-framework-a-la-croissance-la-plus-rapide-en-deep-learning/> [Consulté le 04/05/2023]
- [40] <https://theses.hal.science/tel-03659821/document> [Consulté le 12/05/2023]
- [41] <https://core.ac.uk/download/pdf/15486467.pdf> [Consulté le 12/05/2023]
- [42] <https://www.ircam.fr/projects/pages/detox> [Consulté le 12/05/2023]
- [43] <https://rapidapi.com/restb/api/restb-ai-watermark-detection/details> [Consulté le 12/05/2023]
- [44] <https://www.eurasip.org/Proceedings/Eusipco/Eusipco2009/contents/papers/1569186222.pdf>
[Consulté le 13/05/2023]
- [45] <https://paperswithcode.com/dataset/tiny-imagenet> [Consulté le 12/05/2023]

Résumé

Le tatouage numérique multi-bits a été développé pour améliorer la robustesse contre opérations de traitement du signal et distorsions géométriques.

A cet effet, des outils qui testent la robustesse en appliquant des attaques simulées sur des images tatouées sont disponibles. Cependant, il existe des limites à ces attaques générales car elles ne peuvent pas exploiter les caractéristiques spécifiques du tatouage numérique multi-bits ciblé. De plus, ces attaques sont généralement conçues sans considération de la qualité visuelle. Pour répondre à ces limitations, nous proposons un réseau d'attaque basé sur les réseaux des neurones convolutifs (CNN) et U-NET, un outil entièrement entraînable sur la base d'apprentissage *TinyImagenet* et qui utilise les points faibles du tatouage numérique multi-bits cible.

Mots clés : Google Colab, python, CNN, Tatouage numérique, Intelligence artificielle.

Abstract

Multi-bit watermarking was developed to improve robustness against signal processing operations and geometric distortions.

For this purpose, tools that test the robustness by applying simulated attacks on watermarked images are available. However, there are limitations to these general attacks as they cannot exploit the specific characteristics of the targeted multi-bit watermark. Moreover, these attacks are usually designed without consideration of visual quality. To address these limitations, we propose an attack network based on convolutional neural networks (CNN) and U-NET, a fully trainable tool on the TinyImagenet learning base and which uses the weak points of multi-channel digital watermarking. target bits.

Keywords: Google Colab, python, CNN, Watermarking, Artificial Intelligence.