

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département du Génie Industriel

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel

Option : Data Science et Intelligence Artificielle

Développement d'un outil intelligent d'aide à la décision pour la
détection des manipulations dans les images et les vidéos

Réalisé par

OUCHENE Souhil

TOUATI Islem

Présenté et soutenu publiquement le : **11 Juillet 2024**

Devant le jury composé de :

Président	ZOUAGHI Iskander	MCA	ENP
Examinatrice	BELDJOUDI Samia	MCA	ENP
Promoteurs	ADNANE Mourad	Professeur	ENP
	LAADJEL Moussadek	MCA	CRD-GN

ENP 2024

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département du Génie Industriel

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel

Option : Data Science et Intelligence Artificielle

Développement d'un outil intelligent d'aide à la décision pour la
détection des manipulations dans les images et les vidéos

Réalisé par

OUCHENE Souhil

TOUATI Islem

Présenté et soutenu publiquement le : **11 Juillet 2024**

Devant le jury composé de :

Président	ZOUAGHI Iskander	MCA	ENP
Examinatrice	BELDJOUDI Samia	MCA	ENP
Promoteurs	ADNANE Mourad	Professeur	ENP
	LAADJEL Moussadek	MCA	CRD-GN

ENP 2024

∞ Remerciements ∞

Nous tenons à exprimer notre profonde gratitude, tout d'abord, envers Dieu le Tout-Puissant pour sa bénédiction, la force, la volonté et la patience qu'Il nous a accordées pour mener à bien ce travail.

Nous souhaitons adresser nos sincères remerciements à **M. Mourad ADNANE**, notre encadrant à l'école. Son aide précieuse, ses conseils éclairés et son soutien indéfectible ont été essentiels à la réalisation de ce projet.

Nous tenons également à témoigner notre profonde reconnaissance à **M. Moussadek LAADJEL**, notre encadrant au CRD-GN. Sa guidance, son expertise et son engagement constant ont été une source d'inspiration inestimable, contribuant à la réussite de ce travail.

Notre gratitude toute particulière va à **M. Iskander ZOUAGHI**, dont le rôle a été déterminant dans ce projet. Sa vision, son soutien et la confiance qu'il nous a accordée en nous offrant cette opportunité exceptionnelle ont été les catalyseurs de cette aventure enrichissante. Son engagement envers l'excellence et l'innovation a été une source d'inspiration constante, nous poussant à donner le meilleur de nous-mêmes tout au long de ce parcours.

Nous tenons à exprimer par avance notre gratitude **aux membres du jury** qui évalueront ce travail. Leur temps, leur expertise et leur contribution à l'amélioration de la qualité de notre projet sont grandement appréciés.

Notre reconnaissance s'étend également au corps professoral du département de génie industriel de l'École Nationale Polytechnique pour la formation de qualité et l'encadrement professionnel qu'ils nous ont prodigués.

Enfin, nous souhaitons remercier chaleureusement **les ingénieurs Menad KHELIFA, Mehdi AMOR OUAHMED et Mohamed El Amine MALEK** pour leur aide précieuse tout au long de ce projet.

∞ Dédicace ∞

Du fond de mon cœur, je dédie ce travail...

À vous, ma tendre **mère**, mon roc et mon étoile. Vos bras ont été mon refuge, votre amour ma force. Chaque sourire, chaque encouragement de votre part a illuminé mon chemin. Vous êtes mon héroïne, ma confidente, mon inspiration. Je vous aime d'un amour infini.

À votre mémoire, mon cher **père**. Bien que le temps nous ait séparés trop tôt, votre présence reste vivace dans chacun de mes actes. Vos rêves pour moi sont devenus les miens. J'espère que de là-haut, vous me regardez avec fierté. Votre amour continue de me porter chaque jour.

À vous, mes frères, mes complices de toujours : **Imad**, mon grand frère, mon modèle. Ta force tranquille et ta sagesse m'ont toujours guidé. **Nassim**, mon brillant aîné. Ton intelligence et ta persévérance m'ont poussé à me dépasser. **Aymen**, mon rayon de soleil. Ton rire et ta joie de vivre ont égayé mes journées les plus sombres. **Amir**, mon petit frère adoré. Ta pureté et ton innocence m'ont rappelé la beauté de la vie.

Vous êtes les piliers de mon existence, mes meilleurs amis, mes confidents. Notre lien fraternel est mon plus précieux trésor.

À vous, **Meriem**, **Ahlem**, et **Yousra**, mes belles-sœurs. Je vous suis reconnaissant pour votre amour et votre générosité.

À la douce mémoire de ma grand-mère **Yema Wardiya**. Votre tendresse continue de nous envelopper tous.

À mes chères **Ilène**, **Razane**, **Ayla** et au petit **Yazid**. Votre présence dans nos vies est une source de joie et d'inspiration constante.

Au "**meilleur futur docteur**", mon soutien. Ta présence a rendu ce chemin moins ardu.

À mes professeurs bienveillants, qui ont cru en moi et m'ont poussé à donner le meilleur.

À tous ceux qui ont croisé mon chemin et l'ont embelli, merci du fond du cœur. Ce travail est le fruit de votre amour et de votre soutien. Il vous est dédié, avec toute ma tendresse et ma gratitude.

Souhil

∞ Dédicace ∞

Du fond de mon cœur, je dédie ce travail...

À mon père **Saad** et ma mère **Habiba**, les personnes les plus importantes dans ma vie, et les plus chères à mon cœur, je souhaite exprimer ma profonde reconnaissance pour avoir été ma source de force. Je vous remercie sincèrement d'avoir toujours cru en moi et de m'avoir constamment encouragé à poursuivre mes rêves. Vous êtes toujours la source de ma motivation qui m'a permis d'être là aujourd'hui, sans vous je ne serais rien. Maman, Papa, je veux vous dire merci pour tout.

À mon frère **Karim**, ou bien mon deuxième père, à mon frère **Riad**, ma source de motivation, les deux personnes qui ont toujours cru en moi et m'ont poussé vers l'avant, les deux personnes qui ont tout fait pour moi depuis ma naissance. Je vous aime tous les deux et je vous promets incha'Allah de vous rendre fiers de moi. Votre petit frère

À mes sœurs **Hanane**, **Lamia** et **Boutheina**, mes mamans, source de ma motivation, vous êtes ma vie. Qu'Allah vous préserve et exauce tous vos souhaits.

À tous mes chers neveux et nièces : **Younes**, **Tedj Eddine**, **Khadija**, **Maria**, **Ishak**, **Ayoub**, **Sidra**, **Hadjer**, **Arwa**, **Zaid** et **Tasnim**, je souhaite beaucoup de succès dans votre vie, votre parcours et vos futures études.

À mes professeurs bienveillants, qui ont cru en moi et m'ont poussé à donner le meilleur.

À mes chers amis du groupe "EDHAY3IN" ainsi qu'**Ahmed**, **Oussama** et **Younes**, merci pour votre soutien et votre amitié inestimable.

Islem

المخلص:

تركز هذه الأطروحة على تطوير نظام ذكي للكشف عن التزوير في الصور ومقاطع الفيديو، باستخدام تقنيات متقدمة في معالجة الصور وتقنيات التعلم العميق. يتضمن المشروع تحليلاً متعمقاً لطرق الكشف القائمة على شذوذات الضوضاء وتناقضات الضوضاء وإزالة الفسيفساء. بالتوازي مع ذلك، قمنا بتطبيق نماذج التعلم العميق، مثل الشبكات العصبية التلافيفية (CNN)، باستخدام التعلم التناقلي ونهج التعلم التجميعي لتحسين دقة الكشف. وقد أدى استعمال نموذج U-Net لتجزئة الصور إلى تحديد أفضل. النظام الذي تم الوصول إليه يجمع بين واجهة مستخدم بديهية وبنية خلفية قوية أمثل للمعالجة في الوقت الحقيقي والتحليل التكيفي للصور والفيديو. تُظهر النتائج التي تم الحصول عليها تحسناً كبيراً في أداء الكشف، مما يمهد الطريق للتطبيقات المستقبلية في مختلف مجالات الأمن والعدالة.

الكلمات المفتاحية: العلوم الجنائية، كشف التزوير، الصور ومقاطع الفيديو، التعلم العميق، الرؤية الحاسوبية، معالجة الصور، الشذوذات الضوضائية، التزييف، الضغط، والشبكات العصبية التلافيفية، والتعلم التحويلي، تجزئة الصور، تصنيف الصور U-Net.

Abstract:

This thesis focuses on the development of an intelligent forgery detection system for images and videos, utilizing advanced image processing and deep learning techniques. The project includes an in-depth analysis of detection methods based on noise anomalies, noise inconsistencies, post-filtering noise residues, compression artifacts, and demosaicking.

In parallel, we have implemented deep learning models, such as convolutional neural networks (CNN), using transfer learning and ensemble learning approaches to improve detection accuracy. The integration of U-Net for image segmentation has been crucial in identifying manipulations. The deployed system combines an intuitive user interface with a robust backend architecture, optimized for real-time processing and adaptive analysis of images and videos.

The results obtained show a significant improvement in detection performance, paving the way for future applications in various fields of security and justice.

Keywords: forensics, forgery detection, images and videos, deep learning, computer vision, image processing, noise anomalies, artifacts, compression, convolutional neural networks, transfer learning, image segmentation, image classification, U-Net.

Résumé :

Ce mémoire se concentre sur le développement d'un système intelligent de détection des falsifications dans les images et vidéos, exploitant des techniques avancées de traitement d'images et d'apprentissage profond. Le projet inclut une analyse approfondie des méthodes de détection basées sur les anomalies de bruit, les incohérences de bruit, les résidus de bruit post-filtrage, les artefacts de compression, et le dé-mosaïquage. En parallèle, nous avons mis en œuvre des modèles d'apprentissage profond, tels que les réseaux de neurones convolutifs (CNN), en utilisant des approches d'apprentissage par transfert et d'ensemble learning pour améliorer la précision de la détection.

L'intégration d'UNET pour la segmentation d'images a permis une identification plus fine des manipulations. Le système déployé combine une interface utilisateur intuitive et une architecture backend robuste, optimisée pour le traitement en temps réel et l'analyse adaptative des images et vidéos.

Les résultats obtenus montrent une amélioration significative des performances de détection, ouvrant la voie à des applications futures dans divers domaines de la sécurité et de la justice.

Mots clés : Forensiques, détection de falsifications, images et vidéos, apprentissage profond, vision par ordinateur, traitement d'images, anomalies de bruit, artefacts, compression, réseaux de neurones convolutifs, Apprentissage par transfert, segmentation d'images, classification d'images, U-Net.

Table des Matières

Liste des tableaux	
Liste des Figures	
Introduction Générale.....	15
Chapitre I. Contexte du Projet.....	18
I.1 Introduction.....	19
I.2 Contexte du projet	19
I.2.1 Problématique détaillée	19
I.2.2 Objectifs du projet	21
I.3 Organisme d'accueil.....	22
I.3.1 L'organisation du CRD-GN.....	23
I.4 Étude de l'existant.....	24
Chapitre II. Généralités	25
II.1 Introduction.....	26
II.2 Concepts généraux sur les images.....	26
II.3 Concepts généraux sur la forensique numérique	31
II.3.1 Types de falsification dans les images	32
II.3.2 Catégorisation des techniques de détection de la falsification.....	35
II.3.3 Outils de Détection de Contenus Falsifiés	38
1.4 Conclusion	39
Chapitre III. Etat de l'Art	40
III.1 Introduction.....	41
III.2 Traitement d'images (Image processing).....	41
III.3 Apprentissage Profond (Deep Learning)	42
III.3.1 Processus CRISP-DM	42
III.4 Vision par ordinateur (Computer Vision)	43
III.4.1 Détection d'objets.....	45

III.4.2	Approches Avancées en Classification : Transfer Learning et Ensemble Learning.....	45
III.4.3	UNET et ses Applications en Segmentation d'images.....	48
III.5	Conclusion	52
Chapitre IV. Méthodes Statistiques de détection des falsifications d'images		53
IV.1	Introduction.....	54
IV.2	Détection Basée sur les Anomalies de Bruit	55
IV.2.1	Analyse des incohérences de bruit.....	55
IV.2.2	Analyse des résidus de bruit post-filtrage par filtre médian ...	59
IV.3	Détection basée sur les artefacts	61
IV.3.1	Analyse de la compression.....	61
IV.3.2	Dé-mosaïquage	72
IV.4	Conclusion	78
Chapitre V. Méthodes basées sur le Deep Learning.....		80
V.1	Introduction.....	81
V.2	Compréhension du problème (Business Understanding).....	81
V.2.1	Pourquoi la segmentation	81
V.3	Compréhension des données (Data Understanding)	82
V.4	Préparation des données (Data Preparation).....	86
V.4.1	Organisation des données	86
V.4.2	Analyse approfondie des caractéristiques	88
V.4.3	Préparation et traitement des données	91
V.5	La modélisation (modeling).....	96
V.5.1	Approches Spécifiques	97
V.5.2	Paramètres de modélisation	98
V.5.3	Outils, Technologies et Optimisations Clés	99
V.6	Évaluation Comparative des Approches et Résultats	100
V.6.1	Métriques d'Évaluation.....	101
V.6.2	Résultats.....	102

V.7	Retour à la compréhension du problème, approche de classification	106
V.7.1	Mise en concurrence des modèles et mécanisme de vote majoritaire	107
V.7.2	Evaluation et résultats.....	109
V.7.3	Pipeline de prédiction des falsifications : Analyse adaptative d'images et de vidéos.....	114
V.8	Déploiement.....	116
V.8.1	Frontend (Interface utilisateur)	117
V.8.2	Backend (logique de traitement)	120
V.8.3	Synergie Frontend-Backend : Orchestration du Flux de Données	121
V.8.4	Tests et Validation Future	123
V.9	Conclusion	123
	Conclusion Générale.....	126
	Perspectives d'amélioration.....	128
	Bibliographie.....	129
Annexes	133
Annexe A	134
Annexe B	142
Annexe C	183

Liste des tableaux

Tableau III-1: Segmentation d'images vs. Détection d'objets vs. Classification d'images	44
TABLEAU V-1: Récapitulatif des données utilisées	85
TABLEAU V-2: Comparaison détaillée des trois approches utilisées pour la segmentation.	97
TABLEAU V-3: Tableau récapitulatif des scores obtenus lors du training pour les différentes approches.....	102
TABLEAU V-4: Comparaison entre les modèles pré-entraînés.....	107
TABLEAU V-5: récapitulatif des scores obtenus lors du training des différents modèles	109
TABLEAU V-6: Flux de Données	122
Tableau V-7 : Comparaison entre les différentes approches	124

Liste des Figures

FIGURE I-1: Organisation interne de la gendarmerie nationale	23
Figure II-1 : Processus de numérisation d'une image	27
Figure II-2 : Différences entre les types d'images	27
Figure II-3 : Différentes résolutions courantes pour les images	28
Figure II-4 : Deux différentes profondeurs de couleur	28
Figure II-5 : Comparaison entre les différents formats d'images	29
Figure II-6 : Pipeline de traitement d'image numérique	30
Figure II-7 : Comparaison entre une image originale et une autre falsifiée.	32
Figure II-8 : Exemple de copy-move.	33
Figure II-9 : Démonstration de l'altération d'image par splicing.	33
Figure II-10 : Exemple d'inpainting généré avec la démo interactive de Nvidia	34
Figure II-11 : Exemple de suppression d'image : Effacement d'un élément	34
Figure II-12 : Classification des techniques de détection de la falsification d'images.....	35
Figure III-1 : Variations de traitement appliquées à une vue de Venise	41
Figure III-2 : Processus CRISP-DM	43
Figure III-3 : Comparaison entre le processus de traitement visuel entre la vision humaine et la vision par ordinateur	44
Figure III-4 : Exemple illustratif de détection d'objets.	45
Figure III-5 : principe de travail pour le bagging et le boosting	48
Figure III-6 : Exemple de segmentation d'image pour la détection de piétons et de véhicules dans une scène urbaine	49
Figure III-7 : Architecture U-Net	49
Figure III-8 : Principe de convolution et déconvolution	51
FIGURE IV-1: Organigramme des méthodes statistiques de détection de falsifications d'images et vidéos.....	54
FIGURE IV-2: Illustration de la transformée en ondelettes appliquée à une image x	57
FIGURE IV-3: Les coefficients diagonaux D (HH), verticaux V (LH), horizontaux H (HL) et d'approximation A (LL) de la transformée discrète en ondelettes.....	57
FIGURE IV-4: Application de la méthode d'analyse des inconsistances de bruit.....	59
FIGURE IV-5: Détection de Manipulation par Analyse des Résidus de Bruit	60
FIGURE IV-6: Exemple de crénelage (Aliasing)	61
FIGURE IV-7: Processus de compression d'une image au format JPEG.....	62
FIGURE IV-8: Différents modes de sous échantillonnage de la chrominance	63

FIGURE IV-9: Exemple de matrice de quantification pour une compression plus rapide	64
FIGURE IV-10: Parcours pour transformer la matrice en chaîne de nombre	64
FIGURE IV-11: Différences normalisées vs. Niveaux de re-quantification JPEG	68
FIGURE IV-12: Effet JPEG Ghost, une analyse multi-niveau de compression	69
FIGURE IV-13: Processus en entonnoir de l'Analyse du Niveau d'Erreur (ELA) pour la détection de falsifications d'images.....	70
FIGURE IV-14: Exemple d'artefacts de compression dans une image JPEG	71
FIGURE IV-15: Exemple de la détection de falsification à l'aide de la méthode ELA	72
FIGURE IV-16: Algorithme d'interpolation bilinéaire	74
FIGURE IV-17: Coefficients des 8 filtres pour une interpolation linéaire de haute qualité	75
FIGURE IV-18: Bayer CFA patterns	76
FIGURE IV-19: Exemple de la détection de falsification à l'aide de du dé-mosaïquage.	78
FIGURE V-1: Exemples d'images extraites des datasets	86
FIGURE V-2: Mask généré.....	87
FIGURE V-3: Mask binaire généré.....	88
FIGURE V-4: Diagramme en bâtons pour le nombre d'images par classe.....	89
FIGURE V-5: PDF pour la hauteur des images.....	90
FIGURE V-6: PDF pour la largeur des images.....	90
FIGURE V-7: Nombre de canaux pour les images et les masques	91
FIGURE V-8: Quelques exemples de l'augmentation des images.....	94
FIGURE V-9: Quelques exemples de la segmentation d'une image	95
FIGURE V-10: Exemple de Dropout	99
FIGURE V-11: Bar Chart récapitulatif des scores obtenus lors du training des différents modèles.....	103
FIGURE V-12: La comparaison des deux approches de segmentation.....	105
FIGURE V-13: Comparaison entre les performances des différents modèles.....	111
FIGURE V-14: illustration du principe de l'Early Stopping.	111
FIGURE V-15: Comparaison de résultats de l'approche automatisée par rapport à l'analyse humaine	113
FIGURE V-16: Comparaison entre les résultats obtenus par l'analyse humaine et l'analyse automatique.....	114
FIGURE V-17: Pipeline de prédiction.....	115
FIGURE V-18: Interface implémentée.....	117

Liste des symboles et abréviations

- **Adam** : Adaptive Moment Estimation (algorithme d'optimisation)
 - **AHE** : Adaptive Histogram Equalization
 - **API** : Application Programming Interface
 - **ASCII** : American Standard Code for Information Interchange
 - **Avi** : Audio Video Interleave
 - **CCD** : Charge-Coupled Device
 - **CDF** : Cumulative Distribution Function
 - **CLAHE** : Contrast Limited Adaptive Histogram Equalization
 - **CMOS** : Complementary Metal-Oxide-Semiconductor
 - **CRUD** : Create, Read, Update, Delete
 - **CRD-GN** : Centre de Recherche et Développement de la Gendarmerie Nationale
 - **CRF** : Camera Response Function
 - **CRISP-DM** : Cross-Industry Standard Process for Data Mining
 - **CSS** : Cascading Style Sheets
 - **CFA** : Color Filter Array
 - **DCT** : Discrete Cosine Transform
 - **DVD** : Digital Versatile Disc
 - **DWT** : Discrete Wavelet Transform
 - **EDA** : Exploratory Data Analysis
 - **ELA** : Error Level Analysis
 - **EXIF** : Exchangeable Image File Format
 - **FN** : False Negative
 - **FLOPs** : Floating Point Operations per Second
 - **FP** : False Positive
 - **GAN** : Generative Adversarial Network
 - **GIF** : Graphics Interchange Format
 - **GN** : Gendarmerie Nationale
 - **GPS** : Global Positioning System
 - **GPU** : Graphics Processing Unit
 - **HTML** : HyperText Markup Language
 - **HTTP** : HyperText Transfer Protocol
 - **HSL** : Hue, Saturation, Lightness
 - **HSV** : Hue, Saturation, Value
 - **IDE** : Integrated Development Environment
-

-
- **IPTC** : International Press Telecommunications Council
 - **JPEG** : Joint Photographic Experts Group
 - **JSON** : JavaScript Object Notation
 - **MAD** : Median Absolute Deviation
 - **MLE** : Maximum Likelihood Estimation
 - **MP** : Mégapixels
 - **MSE** : Mean Squared Error
 - **mp4** : MPEG-4 Part 14
 - **PDF** : Probability Density Function
 - **PNG** : Portable Network Graphics
 - **PPP** : Pixels Par Pouce
 - **PRNU** : Photo Response Non-Uniformity
 - **PSNR** : Peak Signal-to-Noise Ratio
 - **ReLU** : Rectified Linear Unit
 - **RLE** : Run-Length Encoding
 - **RGB** : Red, Green, Blue
 - **SSIM** : Structural Similarity Index
 - **TFA** : TensorFlow Addons
 - **TIFF** : Tagged Image File Format
 - **VSCoDe** : Visual Studio Code (éditeur de code)
 - **VP** : Vrai Positif
 - **VN** : Vrai Négatif
 - **XMP** : Extensible Metadata Platform
 - **YCrCb** : Luminance (Y) and Chrominance (Cr and Cb).
-

Introduction Générale

L'intelligence artificielle (IA) a révolutionné de nombreux domaines, offrant des opportunités significatives pour automatiser et optimiser des processus complexes. Dans le domaine de l'analyse d'images et de vidéos, en particulier, les avancées de l'IA ont commencé à remodeler la manière dont les falsifications visuelles sont détectées, en offrant des solutions innovantes pour l'identification des contenus altérés.

Avec l'évolution rapide de la technologie numérique, la manipulation des images et vidéos est devenue de plus en plus sophistiquée et accessible. Cette démocratisation des outils de retouche et de manipulation pose des défis importants pour garantir l'authenticité des contenus visuels, notamment dans les domaines juridiques, journalistiques et de la sécurité publique.

Les enjeux sont particulièrement critiques dans les zones de conflit, comme on peut l'observer actuellement dans le contexte palestinien. Des cas flagrants de désinformation visuelle ont été signalés, visant à minimiser ou déformer la réalité de la situation en Palestine. Ces manipulations médiatiques, orchestrées par ceux qui s'opposent à la cause palestinienne, cherchent à influencer l'opinion publique mondiale et à discréditer les témoignages authentiques des Palestiniens.

Par exemple, des vidéos montrant les conséquences des bombardements sur Gaza ont été falsifiées pour en diminuer la gravité. Des images de destructions massives ont été faussement attribuées à d'autres conflits pour nier leur réalité actuelle en Palestine. On a également observé la diffusion d'anciennes images de manifestations pacifiques, manipulées pour les faire passer pour des actes de violence.

Ces manipulations s'étendent au domaine audio, avec des enregistrements sonores falsifiés pour déformer les déclarations des leaders palestiniens ou pour créer de fausses confessions. L'objectif est clairement de saper la crédibilité du mouvement de résistance palestinien et de justifier des actions agressives à son encontre.

De plus, l'émergence de technologies d'IA générative, telles que les "deepfakes", a considérablement compliqué la tâche de distinction entre le vrai et le faux. Ces technologies permettent de créer des vidéos hyperréalistes de personnes disant ou faisant des choses qu'elles n'ont jamais dites ou faites, posant un défi majeur pour la vérification de l'authenticité des contenus médiatiques.

Face à cette problématique croissante, ce projet de fin d'études se concentre sur le développement d'un outil intelligent d'aide à la décision pour la détection de falsifications

dans les images et vidéos, visant à répondre aux besoins spécifiques des professionnels de la justice, des forces de l'ordre et des autres parties prenantes concernées.

L'objectif principal de ce projet est de créer un système robuste et fiable capable de détecter, d'analyser et de fournir des informations précises sur les images et vidéos falsifiées. Pour atteindre cet objectif, le système exploitera les techniques les plus avancées de l'apprentissage profond et de la vision par ordinateur, telles que les réseaux de neurones convolutifs (CNN) et l'apprentissage par transfert (transfer learning). Ces approches permettront d'identifier les manipulations avec une grande précision, même face à des techniques de falsification sophistiquées.

Ce mémoire, structuré en cinq chapitres, offre une exploration approfondie des différents aspects de la détection de falsifications dans les médias visuels :

1. Contexte du Projet : Cette section ancre le projet dans sa réalité opérationnelle. Elle détaille la problématique spécifique adressée, les objectifs précis du projet, et présente l'organisme d'accueil, le CRD-GN (Centre de Recherche et de Développement de la Gendarmerie Nationale). Une étude approfondie de l'existant permet de positionner le projet dans le contexte actuel des outils de détection de falsifications, en mettant l'accent sur les défis contemporains tels que ceux observés dans le conflit palestinien

2. Généralités : Cette partie pose les fondements théoriques nécessaires à la compréhension du sujet. Elle aborde les concepts généraux sur les images numériques, leur principe d'acquisition, et explore les différents types de manipulation couramment rencontrés. Une attention particulière est portée aux approches actuelles de détection des manipulations, accompagnée d'une présentation de logiciels de détection existants sur le marché.

3. État de l'Art : Cette partie offre une revue exhaustive des avancées technologiques et méthodologiques dans les domaines connexes. Elle couvre le traitement d'images, l'apprentissage profond, le processus CRISP-DM (Cross-Industry Standard Process for Data Mining), la vision par ordinateur, et la détection d'objets. Un accent particulier est mis sur les approches avancées en classification, notamment le transfer learning et l'ensemble learning, ainsi que sur l'architecture UNET et ses applications innovantes en segmentation d'images.

4. Méthodes Statistiques de Détection : Cette partie se penche sur les approches statistiques de détection de falsifications. Elle explore en détail les méthodes basées sur les anomalies de bruit, l'analyse des incohérences et des résidus de bruit, ainsi que la détection basée sur les artefacts. Cette partie aborde également l'analyse de la

compression et le dé-mosaïquage, offrant une compréhension approfondie des techniques statistiques employées dans la détection de manipulations.

5. Détection Basée sur le Deep Learning : La dernière partie constitue le cœur de l'innovation du projet. Elle suit le processus CRISP-DM pour structurer le développement de solutions basées sur l'apprentissage profond. Cette section couvre la compréhension du problème, l'analyse et la préparation des données, la modélisation avec des approches spécifiques au deep learning, et l'évaluation comparative des différentes approches. Elle culmine avec la présentation d'un pipeline complet de prédiction des falsifications, intégrant une analyse adaptative d'images et de vidéos. Les aspects de déploiement, incluant le développement d'une interface utilisateur et la logique de traitement backend, sont également abordés, ainsi que les perspectives de tests et de validation future.

Ce mémoire vise à contribuer significativement au domaine de la détection de falsifications visuelles en proposant une approche intégrée et innovante, combinant des méthodes statistiques éprouvées avec les dernières avancées en matière d'intelligence artificielle. L'objectif final est de fournir aux professionnels un outil puissant et fiable pour lutter contre la désinformation visuelle et renforcer l'intégrité des contenus médiatiques dans notre ère numérique, particulièrement dans des contextes sensibles comme les zones de conflit où la véracité de l'information est cruciale.

Chapitre I.
Contexte du projet

I.1 Introduction

Ce premier chapitre pose les fondements essentiels de notre projet. Nous y présentons une étude préliminaire approfondie, pierre angulaire de notre démarche. Notre analyse s'ouvre sur un examen détaillé de la problématique générale, soulignant son importance et ses implications dans le contexte actuel. Nous mettons ensuite en lumière les objectifs spécifiques de notre projet.

Dans un second temps, nous introduisons l'organisme d'accueil, en détaillant son rôle, ses missions et la problématique spécifique qu'il a identifiée. Nous exposons enfin les objectifs visés par ce travail, en relation directe avec les besoins, les attentes et les exigences de cet organisme.

I.2 Contexte du projet

Dans l'ère numérique actuelle, la prolifération des technologies de l'information et de la communication a engendré une augmentation exponentielle de la création et du partage de contenus visuels. Parallèlement à cette évolution, on observe une sophistication croissante des techniques de manipulation d'images et de vidéos, rendant la détection des falsifications de plus en plus complexe. Ce phénomène pose des défis majeurs pour les forces de l'ordre, en particulier dans le domaine de la forensique.

I.2.1 Problématique détaillée

Dans l'ère numérique actuelle, la prolifération des technologies de l'information et de la communication a engendré une augmentation exponentielle de la création et du partage de contenus visuels. Parallèlement à cette évolution, on observe une sophistication croissante des techniques de manipulation d'images et de vidéos, rendant la détection des falsifications de plus en plus complexe. Ce phénomène pose des défis majeurs pour les forces de l'ordre, en particulier dans le domaine de la forensique numérique.

L'importance stratégique de cette problématique ne peut être sous-estimée, en particulier pour des secteurs de sécurité comme la Gendarmerie Nationale. La manipulation de contenus visuels représente une menace directe pour la sécurité nationale, l'ordre public et l'intégrité des processus démocratiques. Les services de sécurité se trouvent confrontés à un défi de taille : maintenir leur capacité à distinguer le vrai du faux dans un environnement numérique de plus en plus complexe.

Pour la Gendarmerie Nationale, cette problématique revêt une importance cruciale à plusieurs niveaux : Elle peut être utilisée pour **propager de fausses informations**,

influencer l'opinion publique et déstabiliser les institutions gouvernementales, mettant en péril la stabilité du pays. Dans le domaine des enquêtes criminelles, ces manipulations peuvent **compromettre la fiabilité des preuves**, conduisant à des erreurs judiciaires ou à l'impunité de criminels. **Les groupes terroristes exploitent également ces technologies pour leur propagande**, leur recrutement et la planification d'attaques, rendant cruciale la capacité à identifier ces manipulations. De plus, les deepfakes représentent une menace sérieuse pour les personnalités politiques et les figures publiques, pouvant être utilisés pour les discréditer et ainsi impacter gravement la stabilité politique et sociale du pays.

Ces enjeux sécuritaires ne sont cependant qu'une facette du problème. La manipulation de contenus visuels a également des répercussions profondes sur la société dans son ensemble, affectant la vie privée des individus et les structures sociales de manière significative :

- **Atteinte à la réputation** : Des contenus manipulés peuvent être utilisés pour nuire à la réputation d'une personne, causant des dommages psychologiques, professionnels et sociaux irréparables.
- **Chantage et extorsion** : Les deepfakes peuvent être utilisés comme outil de chantage, menaçant la vie privée et la sécurité financière des individus.
- **Erosion de la confiance sociale** : La prolifération de contenus falsifiés peut conduire à une méfiance généralisée envers les médias et les institutions, fragilisant le tissu social et la cohésion nationale.
- **Impact sur les entreprises et les institutions** : Les organisations peuvent être victimes de campagnes de désinformation basées sur des contenus manipulés, affectant leur réputation, leur valeur boursière et leur stabilité.
- **Manipulation de l'opinion publique** : Les contenus falsifiés peuvent être utilisés pour influencer les processus démocratiques, comme les élections, menaçant ainsi les fondements mêmes de la démocratie.

Face à tous ces défis, la Gendarmerie Nationale se trouve confrontée à deux problématiques majeures :

1. **Absence d'outils adaptés** : Il n'existe pas d'outil informatique complet d'aide à la décision, spécifiquement dédié à l'analyse et à la détection des falsifications dans les médias visuels. Cette lacune entrave considérablement la capacité des autorités à identifier et à contrer efficacement ces menaces, laissant la société vulnérable face à ces nouvelles formes de manipulation.

2. **Exploitation sous-optimale des données** : Malgré la collecte de données sur les falsifications par divers organismes tels que la police et la gendarmerie nationale, ces informations ne sont pas exploitées de manière optimale. Cette situation crée une ambiguïté préjudiciable qui limite la compréhension des modalités et des circonstances des falsifications, et par conséquent, la capacité à prendre des décisions rapides et éclairées pour lutter contre ce phénomène. Cette sous-exploitation des données empêche également le développement de stratégies préventives efficaces et l'adaptation rapide aux nouvelles techniques de manipulation.

I.2.2 Objectifs du projet

La Gendarmerie Nationale (GN), en tant qu'institution chargée de la sécurité publique et de l'application de la loi, se trouve au cœur de cette problématique. Confrontée à une augmentation des cas impliquant des preuves visuelles potentiellement manipulées, elle doit constamment adapter ses méthodes et outils d'investigation pour maintenir son efficacité dans la lutte contre la criminalité.

Dans ce contexte, la GN a identifié un besoin crucial : disposer d'un outil intelligent d'aide à la décision pour la détection des manipulations dans les images et les vidéos. Cet outil vise à renforcer les capacités des enquêteurs en leur fournissant des moyens techniques avancés pour analyser et authentifier les contenus visuels.

Le projet s'inscrit dans une démarche plus large de modernisation des pratiques d'investigation de la GN. Il répond à plusieurs enjeux :

- Améliorer l'efficacité et la précision des analyses forensiques des contenus visuels.
- Réduire le temps nécessaire à l'authentification des preuves numériques.
- Renforcer la crédibilité des preuves présentées lors des procédures judiciaires.
- Anticiper l'évolution des techniques de falsification et s'y adapter rapidement.

Pour relever ces défis complexes et atteindre les objectifs ambitieux fixés par la Gendarmerie Nationale, il est essentiel de s'appuyer sur une structure dotée de l'expertise et des ressources nécessaires. C'est dans ce cadre que l'organisme d'accueil joue un rôle crucial dans la réalisation de ce projet innovant. Sa capacité à combiner recherche de pointe et applications pratiques en fait un partenaire idéal pour développer l'outil d'aide à la décision dont la Gendarmerie Nationale a besoin. Examinons maintenant en détail la structure et l'organisation de cet organisme qui le positionnent comme un acteur clé

dans la lutte contre la manipulation des médias visuels et dans le renforcement de la sécurité numérique.

I.3 Organisme d'accueil

La Gendarmerie Nationale est un corps de sécurité placé sous la tutelle du Ministère de la Défense Nationale. Elle a été créée par les autorités algériennes au lendemain de l'indépendance du pays (le 23 août 1962). Elle est chargée des missions de surveillance générale et continue du territoire national, de renseignement et d'information des autorités publiques et l'exercice de l'action préventive et répressive. Trois principales missions sont exercées par les services de la gendarmerie nationale :

- Mission de police judiciaire.
- Mission de police administrative.
- Mission de police militaire.

La figure I-1 montre l'organisation interne de la GN à travers un organigramme regroupant les descriptions du site officiel de la GN :

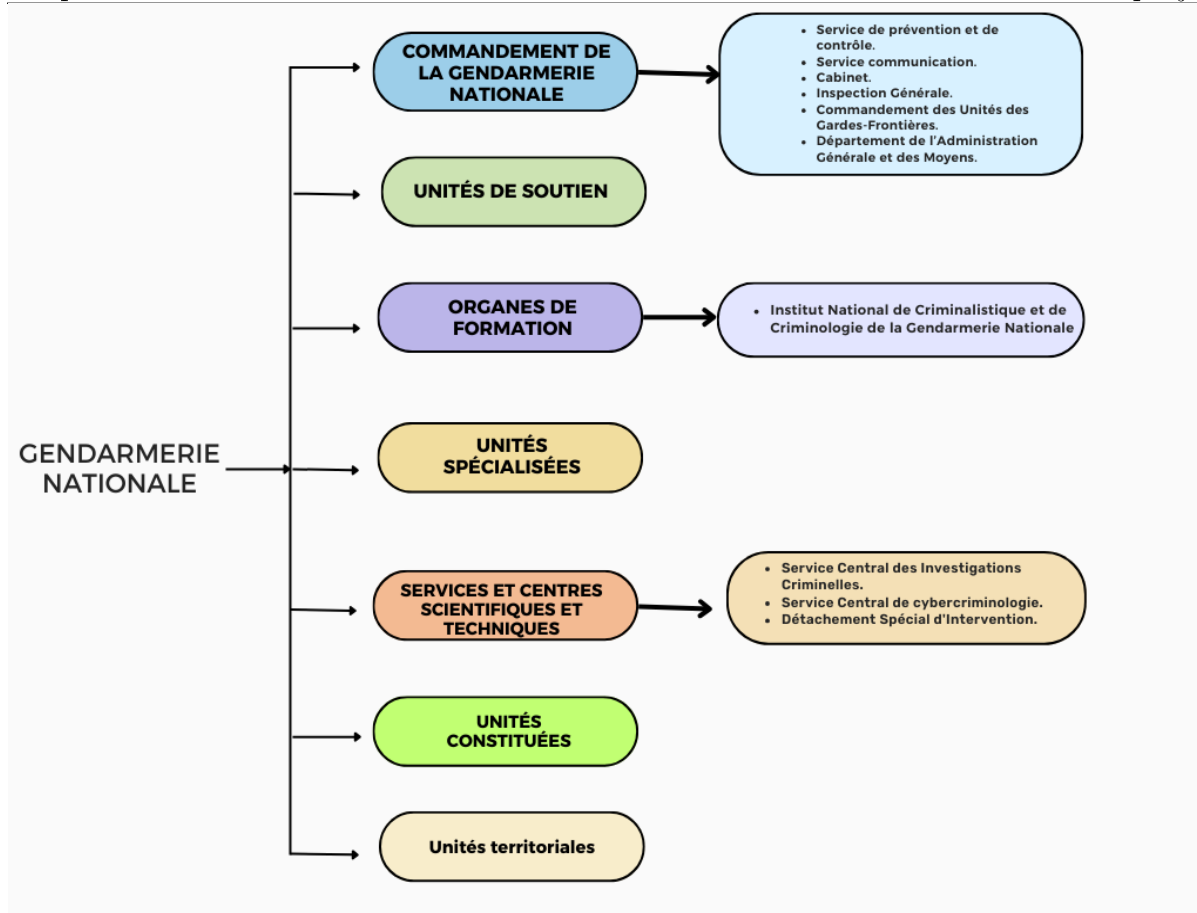


FIGURE I-1: Organisation interne de la gendarmerie nationale

I.3.1 L'organisation du CRD-GN

Le Centre de recherche et développement de la gendarmerie Nationale (CRD-GN) est un établissement militaire à caractère scientifique et technique (EMST) créé en 2006 pour soutenir et accompagner la Gendarmerie Nationale dans divers domaines d'intérêt technologique, scientifique et environnemental. Il conçoit et réalise des projets de recherche seul ou avec des partenaires socio-économiques et des établissements scientifiques nationaux militaires et civils. Il effectue des études en s'appuyant sur les expertises internes et externes dans des thématiques variées, relatives aux divers domaines de la sécurité publique. [1]

Le CRD-GN est organisé en départements et laboratoires pour détenir les compétences en adéquation avec les activités de la Gendarmerie Nationale, dans différents domaines scientifiques et techniques pour assurer une meilleure qualité des ressources humaines travaillant sur des technologies de pointe, à savoir :

- Électronique, Informatique et Télécommunications.

-
- Mécanique, Énergétique et Chimique.
 - Essais et Évaluations des Équipements et Matériels.
 - Informations Scientifiques et Veille Technologique.
 - Criminalistique, Sociologique et Criminologique.
 - Catastrophes Industrielles et Naturelles.

Notre projet a été proposé par le département de Criminalistique, Sociologique et Criminologique.

I.4 Étude de l'existant

Pour des raisons évidentes de confidentialité exigée par le CRD-GN, nous n'allons pas faire une étude de l'existant pour ce projet. Le CRD-GN a donc proposé de traiter le projet comme une problématique nouvelle à laquelle nous devons trouver une solution sans tenir compte de ce qui existe déjà chez eux.

Chapitre II.
Généralités

II.1 Introduction

Les images et les vidéos sont omniprésentes dans notre quotidien, servant de moyens de communication et d'expression dans une multitude de contextes. Historiquement perçues comme des témoignages objectifs de la réalité, l'avènement de la photographie numérique et des logiciels de retouche sophistiqués a remis en question leur fiabilité et leur intégrité.

Face au défi croissant des falsifications visuelles convaincantes, de nombreux efforts ont été déployés dans le domaine de l'analyse d'images numériques pour détecter les manipulations.

Ce chapitre explore les sujets essentiels dans le domaine de la forensique numérique appliquée aux images et aux vidéos. Il examine la création d'images numériques. De plus, il offre un aperçu de la littérature actuelle sur la détection de falsifications d'images numériques, incluant les types de manipulations et les méthodes de détection.

II.2 Concepts généraux sur les images

L'intégration d'images joue un rôle crucial dans notre projet, servant à illustrer les concepts et à valider nos résultats. Cette section se concentre sur les méthodes d'acquisition et les caractéristiques techniques de ces éléments visuels, soulignant leur importance dans notre analyse.

Afin de bien comprendre la détection des falsifications et manipulations dans les médias visuels on doit cerner la nature même d'**une image**. Le terme « image » revêt diverses significations, allant de la simple représentation visuelle d'un objet à une représentation mentale [2]. Dans le contexte **numérique**, une image prend une forme spécifique : numérique. Une image numérique, se réfère à **toute image qui a été acquise, traitée et sauvegardée sous forme codée** représentable par des nombres, ou valeurs numériques. Ce processus, appelé **numérisation**, permet de convertir une image physique, telle qu'une photographie ou une illustration, en une forme numérique, caractérisée par des valeurs discrètes [2]. Cette représentation offre la possibilité d'une exploitation ultérieure par des outils logiciels sur ordinateur. La figure II-1 illustre le processus de numérisation d'une image :

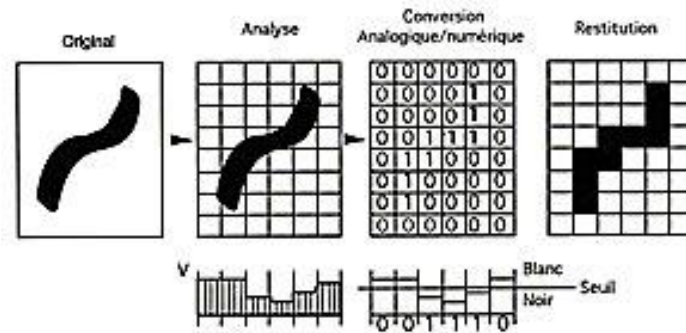


Figure III-1 : Processus de numérisation d'une image [3]

Commencant par l'image originale, suivie de l'analyse en grille, la conversion analogique/numérique en valeurs binaires, et se terminant par la restitution numérique de l'image.

(i) Images numériques entre types et caractéristiques

Les types d'images les plus courants incluent les **images bitmap (ou raster)** et les **images vectorielles**. Les images bitmap sont composées de pixels individuels, chacun ayant une valeur de couleur spécifique, ce qui les rend adaptées pour les photographies et les images réalistes. En revanche, les images vectorielles sont basées sur des formules mathématiques qui décrivent les traits et les formes, ce qui leur permet d'être redimensionnées sans perte de qualité, les rendant idéales pour les logos et les illustrations [2]. La figure II-2 nous permet de mieux saisir les différences entre ces deux types d'images. Elle permet de visualiser concrètement comment une image bitmap se dégrade lorsqu'elle est agrandie, tandis qu'une image vectorielle conserve sa netteté quelle que soit sa taille.

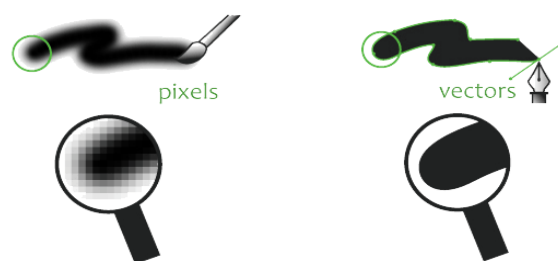


Figure III-2 : Différences entre les types d'images [4]

Les images numériques ont également des **caractéristiques** spécifiques qui peuvent influencer leur qualité et leur utilisation. Ces caractéristiques comprennent la **résolution**, qui détermine le niveau de détail de l'image, mesurée en pixels par pouce (ppp) ou en mégapixels (MP). Une résolution plus élevée offre une meilleure qualité

d'image mais peut également entraîner des fichiers plus volumineux. La figure II-3 montre les différentes résolutions courantes pour les images numériques :

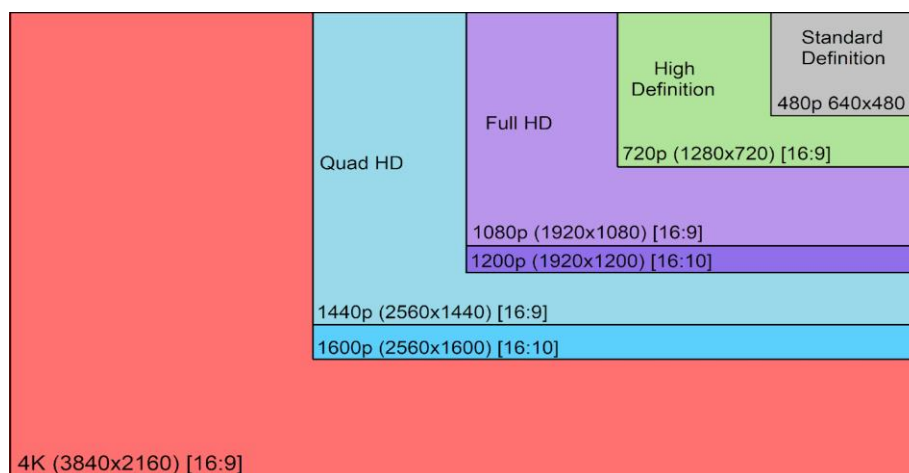


Figure III-3 : Différentes résolutions courantes pour les images [5]

Où par exemple *1080p (1920x1080) [16:9]* signifie une résolution d'écran de 1920 pixels de large par 1080 pixels de haut, avec un rapport d'aspect de 16:9.

La profondeur de couleur, quant à elle, indique le nombre de bits utilisés pour représenter la couleur de chaque pixel, ce qui affecte la précision des couleurs. Une plus grande profondeur de couleur permet une reproduction plus fidèle, mais nécessite également des fichiers de taille plus importante. La figure II-4 illustre ce concept en représentant deux différentes profondeurs de couleur, mettant en lumière comment ce paramètre affecte la qualité visuelle et la fidélité des couleurs.

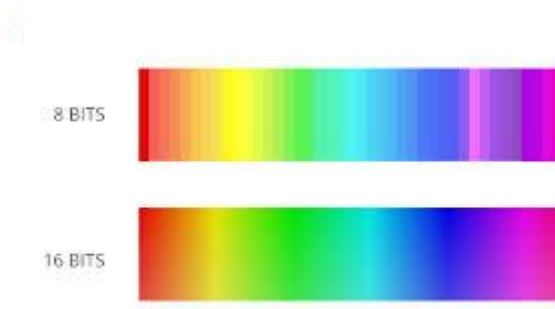


Figure III-4 : Deux différentes profondeurs de couleur [6]

Enfin, le **format de fichier** est un aspect crucial des images numériques, définissant une structure de données standardisée qui détermine comment les informations de l'image sont encodées, compressées et stockées. Il spécifie l'organisation des métadonnées (telles que les dimensions, la profondeur de couleur et les paramètres de compression) ainsi que les données de pixels elles-mêmes. Des formats courants comme

JPEG, PNG, GIF et TIFF sont largement utilisés, chacun offrant ses propres avantages et limitations (comme indiquée dans la figure II-5) en termes de qualité d'image, efficacité de compression et prise en charge de fonctionnalités avancées comme la transparence ou l'animation. [7]

FORMAT	NOM	EXTENSION	NBRE COULEURS	UTILISATION	AFFICHAGE
JPEG	Joint Photographic Experts Group	.jpg	16 millions	Ecran - Web	Oui
PNG	Portable Network Graphics	.png	16 millions	Ecran - Web	Oui
TIFF	Tagged Image File Format	.tif	2 à 16 millions	Impression papier	Non
GIF	Graphics Interchange Format	.gif	256 max.	Ecran - Web	Oui
PDF	Portable Document Format	.pdf	16 millions	Impression papier	-
SVG	Scalable Vector Graphics	.svg	16 millions	Ecran - Web	-
EPS	Encapsulated PostScript	.eps	16 millions	Impression papier	-

Figure III-5 : Comparaison entre les différents formats d’images [8]

Cette compréhension approfondie des caractéristiques de l'image nous conduit naturellement à explorer le pipeline de formation d'une image.

(ii) Pipeline pour former une image

Cette section présente le pipeline de formation des images. Il peut être vu dans sa forme la plus simple sur la figure II-6. Ce pipeline [9] décrit les différentes étapes que traverse une image depuis la capture initiale par le capteur jusqu'à son stockage final :

Phase 1 : Dispositif d'acquisition : Ce matériel se compose principalement de trois éléments clés : l'objectif ou Lens, le réseau de filtres de couleur (CFA), et le capteur, comme illustré dans la figure II-6.

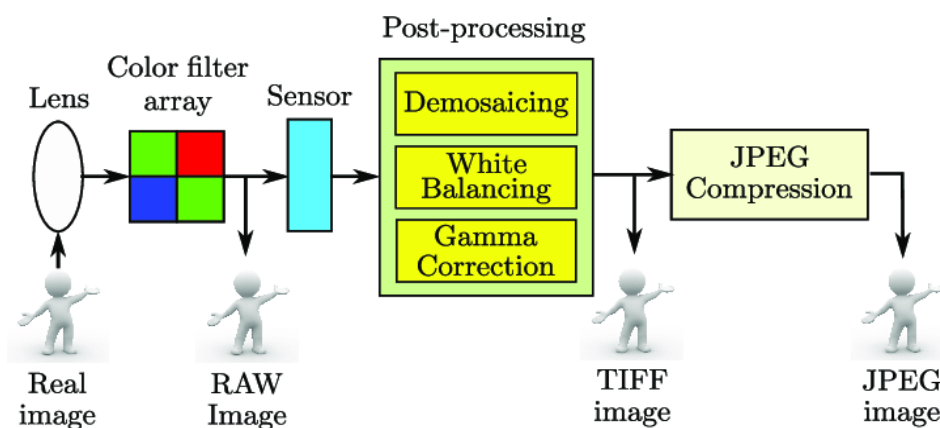


Figure III-6 : Pipeline de traitement d'image numérique [10]

Lorsqu'un appareil photo numérique capture une scène, la lumière passe d'abord à travers l'**objectif (Lens)** vers le capteur, où elle est filtrée par le **réseau de filtres de couleur (CFA : Color Filter Array)**. Ce processus est sujet à plusieurs perturbations : le bruit de fond et la non-uniformité de la réponse photo (**PRNU : Photo Response Non-Uniformity**), issus respectivement des fluctuations de lumière et des imperfections du filtre de couleur. Une fois convertie en signal électrique, l'amplification du signal introduit également du bruit, notamment le bruit du courant d'obscurité et le bruit de lecture. Ces sources de bruit peuvent être utilisées pour l'authentification des images numériques comme nous le verrons ultérieurement.

Phase 2 : Post-traitements : Une fois que les données brutes de l'image sont acquises, elles subissent plusieurs étapes de post-traitement pour être affichées en tant qu'image RGB normale. Initialement, l'image est mosaïquée selon un modèle comme celui de Bayer¹, où chaque pixel enregistre une couleur spécifique (rouge, vert, bleu). Un algorithme de dé-mosaïquage est ensuite appliqué pour reconstruire trois canaux de couleur distincts à partir de cette image brute unique. Ces canaux sont interpolés pour former une image RGB complète. Ensuite, l'image est équilibrée en blanc pour corriger toute dominante de couleur indésirable due aux conditions d'éclairage. Un algorithme d'équilibrage des blancs est utilisé pour trouver et corriger les pixels supposés gris neutre. Enfin, la correction gamma (1.1) est appliquée, une opération non linéaire qui ajuste la luminosité des pixels pour mieux correspondre à la perception humaine, compressant les hautes lumières pour donner plus d'espace à la représentation des teintes sombres.

$$I_{out} = I_{in}^{gamma} \quad (1.1)$$

Phase 3 : Compression d'image : Les images brutes, une fois post-traitées, génèrent des fichiers volumineux. Pour réduire la taille des fichiers, la compression d'images est couramment utilisée dès le processus d'acquisition. Un exemple répandu est la compression JPEG.

Après avoir examiné les fondamentaux des images et leur acquisition, tournons-nous vers la forensique numérique. Ce domaine est crucial pour identifier et détecter les manipulations d'images, un enjeu majeur à l'ère du numérique. Nous allons maintenant

¹ Le modèle de Bayer est une grille 2x2 de filtres colorés (rouge, vert, bleu) sur un capteur d'image, avec deux filtres verts pour une meilleure sensibilité à cette couleur. Le demosaicing est ensuite utilisé pour interpoler les couleurs manquantes.

explorer les types de manipulations, les méthodes de détection, et les outils utilisés dans ce domaine

II.3 Concepts généraux sur la forensique numérique

Nous commençons par définir quelques concepts spécifiques aux forensiques :

Forensique : La forensique, également connue sous le nom de science forensique, est une science interdisciplinaire qui combine les principes de la science et de la technologie pour analyser et interpréter des preuves matérielles dans le cadre d'enquêtes criminelles ou civiles. Elle vise à établir des faits, à identifier des preuves et à présenter des conclusions objectives devant les tribunaux ou d'autres instances judiciaires [11]. Parmi les nombreuses branches de la forensique, une discipline en particulier a émergé pour répondre aux défis uniques posés par l'ère numérique : **la forensique numérique**.

La forensique numérique : représente une discipline cruciale pour examiner et comprendre les artefacts numériques, y compris les images et vidéos altérées. Il englobe un ensemble de techniques et de méthodes utilisées pour identifier, collecter, analyser et présenter des preuves numériques, non seulement dans le contexte juridique, mais aussi dans des domaines de sécurité informatique et de lutte contre la cybercriminalité. [11]

Au cœur de cette discipline se trouve **l'analyse forensique** : également connue sous le terme **d'investigation numérique** ou « **digital forensics** » en anglais, est un domaine scientifique qui se concentre sur les enquêtes liées aux crimes informatiques. Son objectif principal est de récupérer et d'analyser des supports numériques potentiellement suspects afin de fournir des preuves tangibles pour étayer ou contredire des hypothèses, notamment lors de procédures judiciaires. [12]

Dans le domaine de l'analyse forensique numérique, il est essentiel de faire la distinction entre deux types d'images : les images falsifiées et les images vierges. Une **image falsifiée**, également connue sous le nom **d'image altérée**, est une image numérique qui a été intentionnellement modifiée par rapport à son état d'origine en utilisant diverses techniques telles que l'ajout, la suppression ou la modification d'objets ou de caractéristiques dans l'image. D'autre part, **une image vierge** est une image numérique qui n'a pas été manipulée ou modifiée de quelque manière que ce soit par rapport à son état d'origine. Elle représente la version authentique et non éditée de la scène ou des données capturées. La figure III-7 illustre la différence entre une image vierge et une image falsifiée.

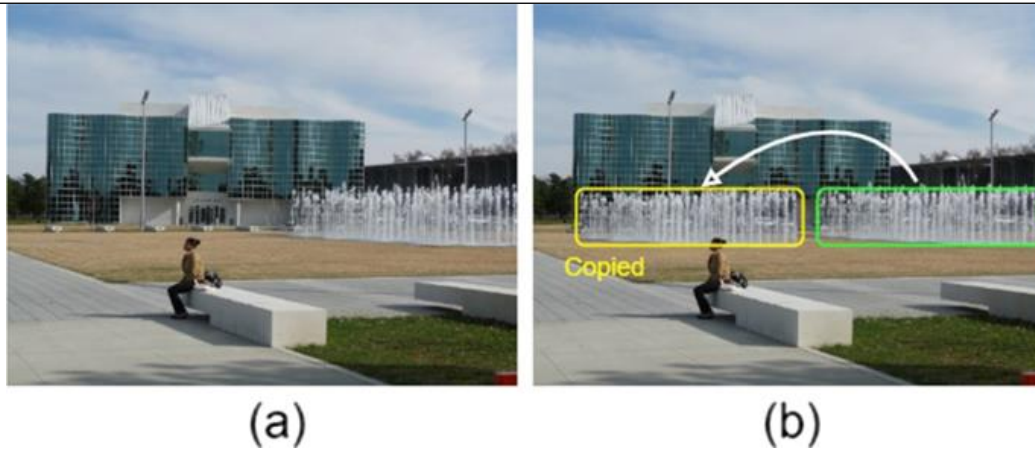


Figure III-7 : Comparaison entre une image originale et une autre falsifiée. [13]

a) l'image vierge, et b) représente l'image falsifiée et intentionnellement altérée en ajoutant une zone copiée à partir d'une autre partie de l'image a.

Passons maintenant vers une revue de la littérature pour nous éclairer sur les différents types de falsifications, les approches existantes pour les détecter et **les outils déjà développés pour y remédier.**

II.3.1 Types de falsification dans les images

La pratique consistant à modifier ou à manipuler des images numériques à l'aide de différentes méthodes et de différents logiciels est appelée manipulation d'images. Elle consiste à modifier la composition, la substance ou l'apparence d'une image afin d'atteindre un certain objectif. On a plusieurs types qu'on peut classer comme :

(a) Copier Déplacer (Copy Move) : Pour réaliser une falsification par copy-move, il faut dupliquer une ou plusieurs parties de l'image et les coller à différents endroits de l'image originale. Les falsifications par copy-move modifient considérablement le contenu sémantique de l'image cible en cachant des informations ou en dupliquant des objets ou des personnes [14]. Comme les sections reproduites semblent être des répliques exactes de l'image originale, il est particulièrement difficile de distinguer les zones authentiques des zones manipulées dans ce type de manipulation (pour rendre la partie clonée moins visible, Il peut utiliser des méthodes, comme le lissage des bords (edge smoothing) ou le flou (blurring)). La figure II-8 démontre un exemple concret illustrant cette technique de falsification.



Figure III-8 : Exemple de copy-move. [15]

Où l'une des tours de départ est une copie subreptice.

(b) **Epissage (Splicing)** : Cette technique diffère du copy-move en ce sens que les objets ou régions collés proviennent d'une ou plusieurs autres photos et que l'image résultante peut présenter des signes évidents de falsification, comme un éclairage, un shading (ombrage) ou une texture, afin de dissimuler ou de supprimer des objets ou des détails indésirables. La falsification par splicing peut être utilisée pour cacher des éléments particuliers ou montrer une scène imaginaire [14]. La figure II-9 met en évidence de manière claire comment le splicing peut être utilisé pour modifier le contenu d'une image. Dans l'image de gauche, nous voyons la scène originale composée d'un objet orange sur un damier. L'image de droite montre le résultat du splicing : une sphère jaune a été ajoutée, modifiant ainsi le contenu et potentiellement le sens de l'image.

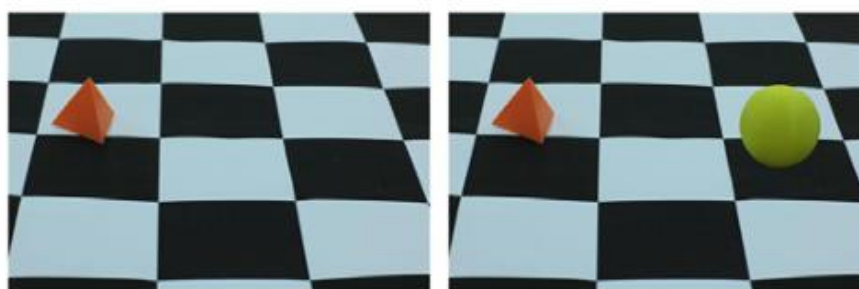


Figure III-9 : Démonstration de l'altération d'image par splicing. [16]

(c) **Inpainting (Restauration)** : Ce type d'attaque consiste à remplir une région ou un « trou » dans l'image avec un contenu plausible. L'inpainting est généralement utilisé pour restaurer des zones endommagées dans les images. Certains algorithmes, comme un réseau GAN (Generative Adversarial Network), peuvent être utilisés pour synthétiser l'inpainting. Il analyse le contexte de l'image et tente de produire des informations crédibles qui se fondent parfaitement dans l'environnement [14]. D'autre

part, les attaquants potentiels peuvent également l'utiliser à des fins malveillantes pour effacer ou dissimuler des informations dans une image. (Exemple dans la figure II-10)



Figure II-10 : Exemple d'inpainting généré avec la démo interactive de Nvidia [15]

Le masque d'inpainting (c'est-à-dire la région supprimée qui doit être remplie par l'algorithme) est représenté en jaune dans l'image de gauche. L'image de droite montre le résultat après l'inpainting.

(d) Suppression d'image (Image Removal) : Ce type de contrefaçon consiste à retirer des personnes, des objets ou du texte d'une image afin d'en modifier le contexte ou l'interprétation. La complexité des procédures de suppression peut varier, allant des capacités d'éradication de base en fonction du contenu dans les logiciels d'édition d'images à des approches plus complexes.

Pour comprendre ce concept, examinons la figure II-11 qui montre la suppression d'un élément d'une image (Image Removal). L'image originale (à gauche) montre deux cahiers étiquetés, tandis que l'image manipulée (à droite) présente la suppression de l'étiquette sur l'un des cahiers.



Figure III-11 : Exemple de suppression d'image : Effacement d'un élément [16]

Examinons maintenant les techniques concrètes utilisées pour détecter les falsifications d'images.

II.3.2 Catégorisation des techniques de détection de la falsification

Il existe plusieurs catégories différentes de méthodes de détection de la falsification d'images et de vidéos. Un aperçu de toutes ces catégories peut être vu dans la figure II-12. Nous pouvons distinguer deux grandes familles de détection de la falsification d'images : **l'approche passive et l'approche active**.

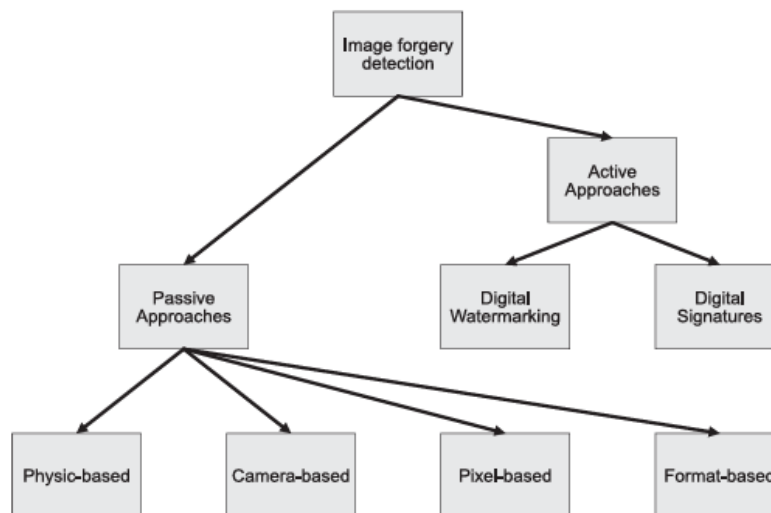


Figure III-12 : Classification des techniques de détection de la falsification d'images

Les techniques de détection active de la falsification comprennent deux grandes familles : **le filigrane (tatouage numérique ou watermarking) et les méthodes basées sur la cryptographie (signature numérique)**. Elles jouent un rôle crucial dans **l'authentification de la propriété du contenu et la prévention des violations des droits d'auteur**. Ces techniques prennent en charge diverses applications au-delà de la protection des droits d'auteur, notamment la détection des falsifications et la dissimulation des erreurs. Bien que leur concept de base soit identique (**l'ajout d'éléments inviolables aux supports numériques**), leurs méthodes de mise en œuvre diffèrent considérablement. Les filigranes numériques modifient les informations des pixels pour insérer des marques visibles ou invisibles, classées en fonction de leur robustesse. Les filigranes fragiles, couramment utilisés pour la détection de la falsification, sont facilement détruits par des modifications, même mineures, du support. À l'inverse, les méthodes basées sur la cryptographie cryptent les supports à l'aide d'algorithmes de chiffrement ou les signent numériquement, ce qui permet de détecter toute modification en cas de non-concordance des signatures. [9]

Cependant, cette **approche présente des limites**, car elle nécessite l'intégration de signatures ou de filigranes lors de l'acquisition ou ultérieurement par un individu. Cette dépendance à l'égard d'un matériel spécifique, tel que des caméras spécialisées,

limite la généralisation de l'application. En outre, des facteurs tels que la compression, la mise à l'échelle et le bruit peuvent affecter la robustesse de ces techniques.

D'autre part, **les approches passives** de détection de la falsification d'images et de vidéos sont basées sur la connaissance de la formation de l'image ou sur des hypothèses concernant le processus de falsification. Contrairement à l'approche active, **ces techniques ne nécessitent pas de matériel spécialisé ni d'informations de première main sur le contenu visuel**. Ainsi, elles sont également appelées **approches passives-aveugles**. L'idée fondamentale derrière cette approche est que les images et les vidéos ont des propriétés ou des caractéristiques inhérentes qui sont constantes dans les originaux. **Lorsqu'une image ou une vidéo est falsifiée, ces motifs sont altérés**. Les approches passives extraient ces caractéristiques des images et des vidéos et les analysent dans le but de détecter différentes falsifications. Par conséquent, pour surmonter les inefficacités rencontrées dans l'approche active, l'utilisation de l'approche passive pour la détection de la falsification est privilégiée.

Nous pouvons distinguer quatre catégories de détection passive :

- **Physic Based Methods** : c'est des méthodes qui reposent sur des principes physiques pour détecter les falsifications. En connaissant le type de scène acquise, il est possible de modéliser certaines propriétés physiques que les éléments de la scène doivent respecter. De nombreuses méthodes se basent sur l'analyse de l'environnement lumineux. Par exemple, pour une image capturée en extérieur par une journée ensoleillée, la direction de la lumière devrait être cohérente sur l'ensemble de l'image car la lumière du soleil provient d'un point infiniment éloigné. De même, pour une image d'intérieur avec des personnes, des incohérences dans le reflet des yeux pourraient indiquer l'ajout de personnes à la scène. Pour la détection des deepfakes, certains auteurs ont proposé d'examiner la pose incohérente du visage due à l'imprécision du processus de génération. Ces méthodes sont extrêmement spécifiques car elles modélisent des propriétés physiques pour des scénarios bien définis. L'un des avantages de ces approches est qu'elles sont totalement indépendantes de la caméra, du type de support et de la taille de l'image, ainsi que de la compression utilisée. Elles reposent sur les erreurs commises par le manipulateur, qui peuvent être extrêmement difficiles à éviter, telles qu'un éclairage incohérent lors du splicing.
- **Camera-Based Methods** : se concentrent sur le dispositif d'acquisition. Elles tentent soit de modéliser l'ensemble du pipeline de formation d'images, soit de se concentrer sur des artefacts très spécifiques. Ces méthodes modélisent la façon

dont une image est normalement formée et recherchent des anomalies par rapport à ce modèle. Par exemple, certaines méthodes ont modélisé la manière dont l'aberration chromatique due à l'objectif devrait être orientée et ont mis en évidence la falsification en recherchant une orientation incohérente. D'autres approches supposent que l'estimation de la fonction de réponse de la caméra (CRF) doit être unique pour chaque sous-partie d'une image et tentent de détecter l'existence d'une CRF estimée significativement différente afin de mettre en évidence les traces de falsification. De nombreuses méthodes tentent de déduire la grille de l'auto-corrélation du bruit (PRNU), ce qui leur permet d'authentifier des images ultérieures provenant du même appareil ou de détecter des contrefaçons. Enfin, d'autres méthodes tentent d'exposer les incohérences dans le bruit en tant que preuves de falsification. Les performances de ces méthodes varient considérablement en fonction de l'artefact étudié. En général, ces méthodes fonctionnent mieux lorsque la qualité de l'image est suffisante. L'avantage de ces approches est qu'elles fournissent des preuves importantes des falsifications qui sont facilement interprétables.

- **Pixel-Based Methods** : Contrairement aux deux catégories précédentes, les méthodes basées sur les pixels ne se basent pas sur le support d'acquisition ni sur les propriétés physiques de l'objet acquis. Au lieu de cela, elles émettent des hypothèses sur le processus d'altération que le manipulateur pourrait utiliser. Par exemple, les méthodes de détection Copy-Move partent du principe que la partie dupliquée ne sera pas beaucoup modifiée après avoir été collée, et recherchent donc des éléments anormalement similaires dans l'image. Certaines méthodes supposent que le contrefacteur peut brouiller les limites d'un objet dupliqué ou même de l'objet entier pour le mélanger, et recherchent donc des traces de flou artificiel. De même, d'autres méthodes recherchent une balance des blancs incohérente en supposant que le faussaire essaiera de faire correspondre les couleurs de l'objet collé. Par définition, ces méthodes sont généralement spécifiques à une manipulation donnée. Comme les méthodes basées sur la physique, il n'y a pas d'hypothèse sur l'appareil ou la compression.
- **Format-Based Methods** : Ces méthodes se concentrent sur la compression appliquée au média. Elles doivent donc supposer que le support était authentique avant la première compression. De même que l'approche basée sur la caméra, elles modélisent soit l'ensemble du processus de compression, soit seulement une partie spécifique. À partir de là, elles recherchent des incohérences dans certains artefacts JPEG, des traces de double compression, ou détectent des traces antérieures de compression dans des images non compressées. En général, ces méthodes sont

extrêmement efficaces. L'un de leurs principaux inconvénients est qu'elles ne peuvent pas détecter les falsifications qui se sont produites avant la première compression. Bien que cela puisse être pratique, ces méthodes peuvent être utilisées de manière plus active en limitant les paramètres de compression à des valeurs spécifiques et en authentifiant le support sur la base de cette connaissance.

Dans le contexte actuel, il est essentiel de se familiariser avec les outils permettant de détecter et de prévenir les falsifications d'images et de vidéos. Cette dernière partie présentera un aperçu de quelques logiciels utilisés pour la détection de manipulations. En comprenant les capacités de ces outils, nous pourrions mieux cerner comment lutter contre les enjeux liés à la vérification et à l'authenticité des contenus médiatiques. Cependant, il est également essentiel de comprendre les logiciels de manipulation tels que Lightworks et Adobe Premiere pour mieux relever les défis de la vérification et de l'authenticité des médias numériques. (Voir annexe A-6)

II.3.3 Outils de Détection de Contenus Falsifiés

Ces logiciels ne sont pas des outils automatiques permettant de décider si une image est falsifiée ou non, mais plutôt des outils d'aide destinés à expérimenter divers algorithmes découverts dans les derniers documents de recherche et ateliers.

(a) Fake Image Detector :

Il effectue une analyse des niveaux d'erreur (ELA) pour détecter les incohérences dans les niveaux de compression de l'image téléchargée. Il prend également en compte les métadonnées de l'image pour vérifier toute manipulation éventuelle. [17]

Son interface est simple et conviviale. Il suffit de télécharger l'image et de cliquer sur « Scan Now ». Les résultats sont générés en quelques secondes. Toutefois, il est à noter que selon les développeurs de l'outil, sa précision est entre 60 et 70 %.

(b) Sherloq :

C'est un projet visant à développer un environnement entièrement intégré dédié à la forensique des images numériques. Lancé en 2015, ce logiciel a connu plusieurs mises à jour jusqu'à sa dernière version établie en 2020. Il propose une multitude de fonctionnalités, notamment l'extraction d'informations générales sur une image telles que les métadonnées et la distribution des pixels. De plus, Sherloq offre la possibilité de parcourir les services de recherche en ligne pour trouver des images visuellement similaires, ainsi que d'améliorer les images grâce à des ajustements de contraste et de luminosité. Pour la détection de manipulations sur les images, Sherloq propose une gamme de solutions simples telles que la détection de la compression JPEG et l'analyse

du bruit, ainsi que des solutions plus traditionnelles comme le Splicing, le copy-move, et le rééchantillonnage d'images. [18]

(c) InVid :

C'est un outil de vérification développé dans le cadre du projet européen InVID (In Video Veritas). Il utilise des algorithmes avancés de traitement d'image et de vidéo pour analyser les contenus. Il emploie des techniques de détection de keyframes pour extraire les images clés, puis applique des méthodes de hachage perceptuel pour la recherche d'images inversée. L'outil intègre également des analyses forensiques, comme la détection d'artefacts de compression et l'analyse des motifs de bruit, pour identifier les manipulations potentielles. De plus, InVid exploite des API (application programming interface) de réseaux sociaux pour contextualiser le contenu, en extrayant et analysant les métadonnées associées aux vidéos. Cet outil est largement utilisé par les fact-checkers professionnels dans les grandes rédactions et organisations médiatiques pour vérifier l'authenticité des vidéos virales et combattre la désinformation en ligne. [19]

1.4 Conclusion

Dans ce chapitre, nous avons exploré un ensemble de concepts clés liés à notre domaine d'intérêt. Nous avons débuté par la définition de l'image et son processus d'acquisition, suivi d'une exploration du domaine de la forensique et de la forensique numérique. Nous avons ensuite examiné les différents types de manipulation d'images et les approches proposées dans la littérature pour traiter ce problème. Enfin, nous avons présenté deux outils puissants utilisés dans la détection de manipulation d'images.

Chapitre III.

Etat de l'Art

III.1 Introduction

La détection des falsifications d'images et de vidéos en forensique numérique nécessite une approche multidisciplinaire. Ce chapitre explorera l'état de l'art des techniques essentielles : le traitement d'images comme fondement, l'apprentissage profond guidé par le processus CRISP-DM, et la vision par ordinateur pour la détection. Nous examinerons également les avancées en classification, notamment le Transfer Learning et l'Ensemble Learning, ainsi que l'architecture U-Net pour la segmentation. Cette synergie de méthodes offre une perspective complète sur les défis actuels de la forensique numérique visuelle.

III.2 Traitement d'images (Image processing)

C'est une branche du traitement du signal dédiée aux images et vidéos. Elle englobe des opérations visant à améliorer la lisibilité et l'interprétation des images. Cela inclut des processus tels que le rehaussement du contraste et la réduction du bruit. Les opérations de traitement peuvent être divisées en deux niveaux : le **traitement bas-niveau**, qui se concentre sur l'analyse des caractéristiques de l'image sans interprétation, et le **traitement haut-niveau**, qui interprète ces caractéristiques pour des applications telles que la prise de décision [20]. Cette définition du traitement d'images peut être illustrée par la figure III-1 :

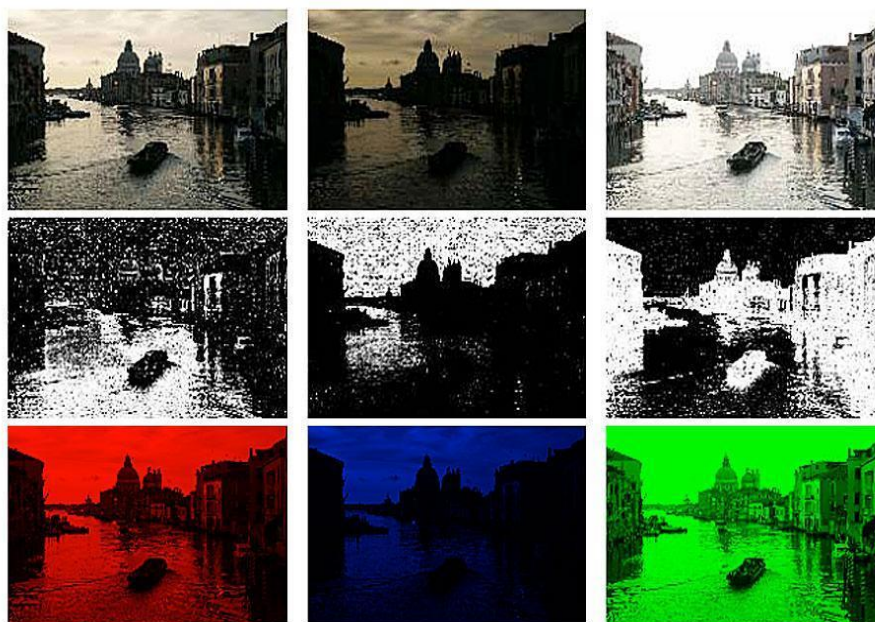


Figure III-1 : Variations de traitement appliquées à une vue de Venise [21]

Version originale, puis version soumise à des ajustements de luminosité et de contraste, seuillage (binarisation) et colorisation (rouge, bleu et vert).

Alors que le traitement d'images repose sur des règles prédéfinies, **l'apprentissage profond** offre une approche plus flexible et puissante.

III.3 Apprentissage Profond (Deep Learning)

C'est une branche avancée de l'intelligence artificielle utilisant des réseaux de neurones artificiels multicouches. Ces réseaux imitent le cerveau humain, permettant aux systèmes d'apprendre automatiquement à partir de grandes quantités de données. Ils excellent dans le traitement de données complexes comme les images ou les vidéos, en extrayant des caractéristiques de plus en plus abstraites à chaque couche. Cette approche permet une adaptation continue et une haute précision dans des tâches telles que la détection d'objets et la classification. [22]

Il est crucial maintenant de comprendre comment structurer les applications du Deep Learning dans des projets concrets. C'est ici qu'intervient le processus **CRISP-DM (Cross-Industry Standard Process for Data Mining)**, une méthodologie éprouvée qui guide l'implémentation des techniques d'apprentissage profond de manière systématique et efficace.

III.3.1 Processus CRISP-DM

Ce modèle se compose de six étapes principales (comme indiqué dans la figure III-2), chacune ayant un rôle clé dans la réussite du projet.

Tout d'abord, **la phase de compréhension du problème, ou Business Understanding**, permet de définir les objectifs du projet en fonction des besoins métiers identifiés.

Ensuite, **la phase de compréhension des données, ou Data Understanding**, nous permet d'acquérir des connaissances initiales sur les données.

La troisième étape, **la préparation des données, ou Data Preparation**, consiste à sélectionner, nettoyer, construire, intégrer et formater les données en vue de leur modélisation.

La quatrième phase, **la modélisation**, implique la sélection et l'application des techniques de modélisation appropriées, ainsi que le calibrage des paramètres du modèle pour obtenir des résultats optimaux.

Par la suite, l'**étape d'évaluation** permet d'évaluer de manière approfondie le ou les modèles construits afin d'en vérifier la qualité et la validité par rapport aux objectifs initiaux.

Enfin, la dernière étape, le **déploiement**, consiste à mettre en œuvre le ou les modèles retenus au sein des processus métiers opérationnels. [23]

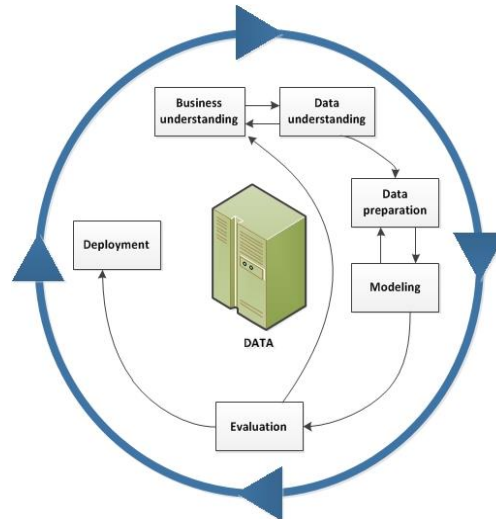


Figure III-2 : Processus CRISP-DM [24]

Note : Le processus CRISP-DM est itératif et les phases peuvent interagir entre elles. Il favorise une approche structurée et méthodique tout en restant suffisamment flexible pour s'adapter à différents types de projets d'analyse de données.

Après avoir exploré l'apprentissage profond et sa méthodologie d'application, il est essentiel de nous concentrer sur le domaine spécifique dans lequel s'inscrit notre projet : **la vision par ordinateur**.

III.4 Vision par ordinateur (Computer Vision)

Cette discipline, au cœur de notre étude sur la détection des falsifications d'images, est un domaine de l'intelligence artificielle qui permet aux ordinateurs d'analyser des images et des vidéos pour en extraire des informations significatives. Cette technologie permet aux machines de « voir » et de comprendre leur environnement visuel [19]. Contrairement à la vision humaine, qui repose sur l'interprétation cognitive et subjective des informations visuelles, la vision par ordinateur traite ces informations de manière systématique et objective. Cette distinction est illustrée dans la figure III-3.

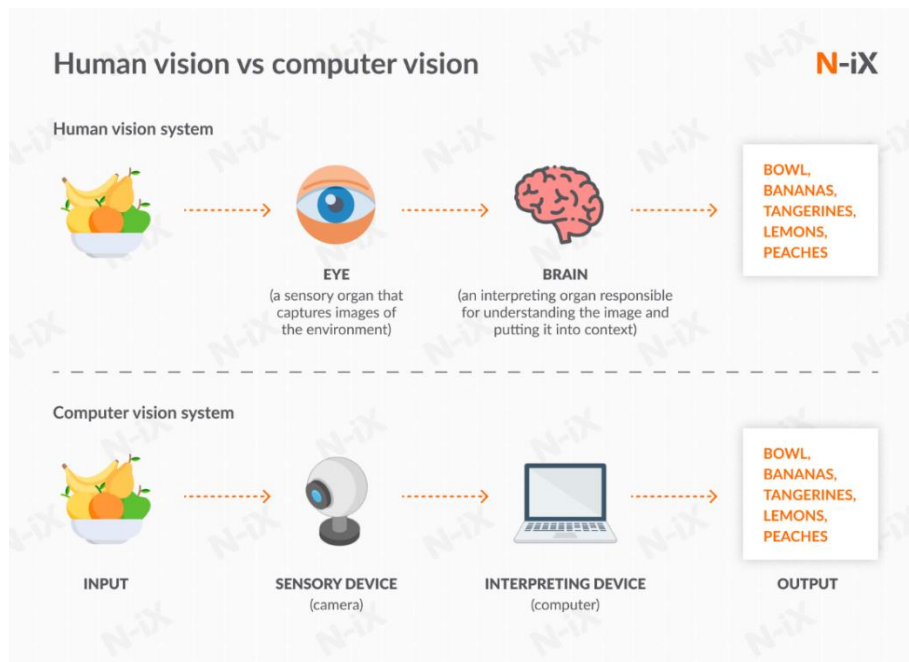


Figure III-3 : Comparaison entre le processus de traitement visuel entre la vision humaine et la vision par ordinateur [25]

La vision par ordinateur englobe plusieurs tâches clés. Parmi celles-ci, nous nous concentrerons sur trois des plus importantes : la détection, la classification et la segmentation d'objets. Examinons plus en détail les aspects essentiels mentionnés dans le tableau suivant III-1 :

Tableau III-1: Segmentation d'images vs. Détection d'objets vs. Classification d'images

Aspect	Classification d'images	Détection d'objets	Segmentation d'images
Objectif	Attribuer une étiquette ou une catégorie à l'ensemble de l'image	Identifie et localise plusieurs objets	Diviser l'image en régions significatives
Sortie	Un seul label ou une seule catégorie	Boîtes de délimitation autour des objets détectés	Masques de segmentation par pixel
Focus	Classification de haut niveau de l'ensemble de l'image	Détection d'objets avec localisation	Segmentation détaillée des objets et de l'arrière-plan

Complexité	Plus simple et plus rapide	Complexité modérée	Plus complexes et plus intensifs en termes de calculs
Exemples	Détermine si une image a été manipulée, sans préciser où.	Identifie/localise les zones falsifiées potentiellement.	Isole précisément les régions falsifiées du reste de l'image.

III.4.1 Détection d'objets

La détection d'objets consiste à détecter, localiser et classer des objets d'intérêt dans des images numériques ou vidéo. Elle consiste à identifier la présence d'objets cibles, à déterminer leur emplacement précis en fournissant des boîtes de délimitation « bounding boxes » ou des coordonnées de pixels, et à les classer dans des catégories prédéfinies [26]. La figure III-4 illustre une scène où on peut voir que le modèle a pu détecter une personne et d'autres objets dans l'image à travers des bounding boxes.

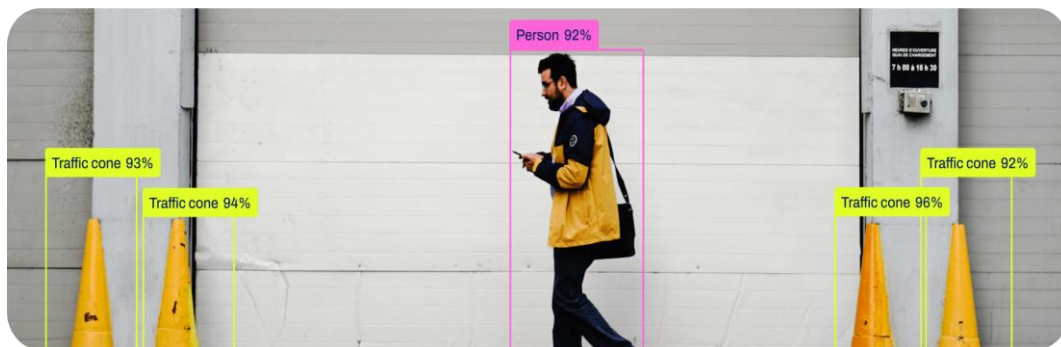


Figure III-4 : Exemple illustratif de détection d'objets. [27]

Après avoir examiné la détection d'objets, concentrons-nous maintenant sur les deux approches cruciales pour notre problématique : la classification et la segmentation.

III.4.2 Approches Avancées en Classification : Transfer Learning et Ensemble Learning

La classification d'image consiste à étiqueter les images sur la base d'une analyse à une image entière sur la base de données d'entraînement préexistantes d'images déjà étiquetées. Bien que ce processus puisse sembler simple, il nécessite une analyse au niveau du pixel afin de déterminer l'étiquette la plus appropriée pour l'image. Cette analyse détaillée fournit des données et des informations qui permettent de prendre de bonnes décisions et d'obtenir des résultats exploitables. [26]

La classification d'images, bien qu'efficace, peut être optimisée davantage pour améliorer ses performances et sa précision. Deux techniques avancées se distinguent particulièrement dans ce domaine : **le transfert learning et l'ensemble learning**. Ces approches permettent non seulement d'améliorer la précision de la classification, mais aussi de réduire le temps d'entraînement et d'augmenter la robustesse des modèles. Examinons comment ces méthodes peuvent être appliquées pour renforcer la capacité à détecter les falsifications d'images.

(i) **Transfer Learning (Apprentissage par transfert)**

C'est une technique d'apprentissage automatique qui permet de mutualiser les connaissances acquises par un modèle sur une tâche donnée pour améliorer les performances d'un autre modèle sur une tâche similaire [28]. Cette approche est particulièrement utile (à détailler dans les avantages du Transfer Learning qui suit) dans le domaine du Deep Learning, où l'entraînement de réseaux de neurones artificiels nécessite souvent un grand nombre de données. Voyons comment fonctionne cette méthode :

(a) **Fonctionnement du Transfer Learning :**

1. **Sélection d'un Modèle Pré-entraîné** : Choisir un modèle qui a été pré-entraîné sur une tâche similaire ou sur un large ensemble de données. Les modèles populaires incluent EfficientNet, Inception et MobileNet pour les tâches de la vision par ordinateur.
2. **Architecture et Adaptation** :
 - **Couches Convolutives** : Conserver les couches convolutives préentraînées qui capturent des caractéristiques générales.
 - **Couches Fully Connected** : Remplacer ou ajouter des couches fully connected adaptées à la nouvelle tâche (par exemple, changer le nombre de neurones en fonction du nombre de classes dans la nouvelle tâche).
3. **Configuration des Hyperparamètres** : Ajuster les hyperparamètres pour l'entraînement, tels que le taux d'apprentissage, l'optimiseur, et la taille de lot. Cela peut inclure la régularisation pour éviter le surapprentissage.

4. Stratégie d'Entraînement :

- **Phase de Gel** : Geler les poids des couches pré-entraînées pour préserver les caractéristiques générales. Entraîner uniquement les nouvelles couches.
- **Fine-Tuning** : Décongeler progressivement certaines des couches pré-entraînées pour affiner les poids en fonction des spécificités de la nouvelle tâche, améliorant ainsi la performance.

(b) Avantages du Transfer Learning

- **Efficacité des Données** : Utilisation efficace des données limitées pour la nouvelle tâche en exploitant les connaissances générales acquises par le modèle pré-entraîné.
- **Réduction du Temps d'Entraînement** : Moins de temps et de ressources nécessaires pour entraîner le modèle comparé à un entraînement from scratch.
- **Meilleure Performance** : Amélioration des performances grâce aux caractéristiques robustes déjà apprises par le modèle pré-entraîné.
- **Accessibilité** : Disponibilité de nombreux modèles pré-entraînés en open source, facilitant le développement de nouveaux modèles.

Bien que le transfer learning soit une approche efficace pour optimiser les modèles de vision par ordinateur, il existe d'autres méthodes d'optimisation qui peuvent également améliorer les performances. Parmi celles-ci, on a **les méthodes ensemblistes**.

(ii) Technique d'ensemble pour la classification

C'est des méthodes qui permettent de combiner plusieurs modèles pour obtenir des résultats plus précis et plus robustes. Cette méthode est basée sur l'idée que différents modèles peuvent capturer divers aspects du problème et que leur combinaison peut produire des résultats plus robustes et précis qu'un seul modèle. [29]

Il existe plusieurs façons de combiner les modèles dans l'ensemble learning, notamment **le bagging, le boosting et le stacking**. Le bagging (Bootstrapped AGGregatING) consiste à créer plusieurs sous-ensembles de données à partir du jeu de données d'origine, à entraîner un modèle sur chacun de ces sous-ensembles, puis à combiner les prédictions de chaque modèle pour obtenir une prédiction finale. Le boosting, quant à lui, consiste à entraîner des modèles séquentiellement (comme le montre la figure III-5), en donnant plus de poids aux échantillons mal classés par les

modèles précédents. Le stacking, une technique plus avancée, utilise les prédictions des modèles de base comme entrées pour un méta-modèle qui apprend à les combiner de manière optimale.

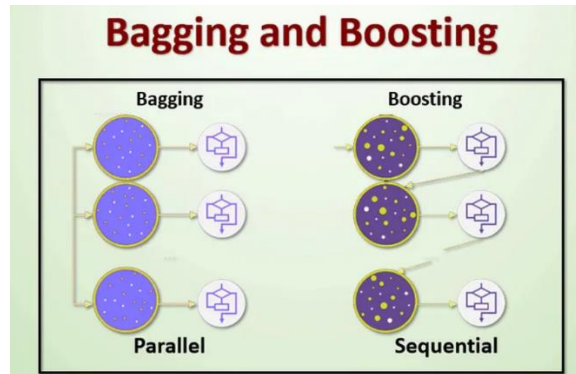


Figure III-5 : principe de travail pour le bagging et le boosting [30]

L'ensemble Learning présente plusieurs avantages par rapport à l'utilisation d'un seul modèle. Tout d'abord, il **réduit la variance** et **améliore la précision des prédictions** en combinant les résultats de plusieurs modèles. De plus, il permet de **réduire le risque de surapprentissage**, car chaque modèle est entraîné sur un sous-ensemble différent des données. Enfin, l'ensemble learning peut être appliqué à n'importe quel type de modèle, ce qui en fait une approche très flexible et polyvalente.

Dans le cadre de notre étude, nous avons choisi d'utiliser la méthode de **bagging** pour plusieurs raisons. Premièrement, le bagging est particulièrement efficace pour réduire la variance des modèles, ce qui est crucial pour **améliorer la stabilité et la généralisation de nos prédictions**. Deuxièmement, contrairement au boosting qui peut être sensible au bruit dans les données, le **bagging est plus robuste face aux outliers et aux erreurs d'étiquetage**. Enfin, le **bagging est plus simple à implémenter et à paralléliser que le stacking**, ce qui le rend plus adapté à notre contexte de traitement de données à grande échelle.

Abordons maintenant la dernière tâche de la vision par ordinateur : la **segmentation**. Dans cette section, nous expliquerons ce qu'est la segmentation, **en quoi consiste UNET**, et les **avantages** de l'utilisation d'UNET.

III.4.3 UNET et ses Applications en Segmentation d'images

La segmentation d'image est une technique qui divise une image numérique en plusieurs segments ou régions sur la base des caractéristiques des pixels. Cette opération

définit les contours précis des objets, ce qui simplifie et améliore l'analyse de l'image en la divisant en composants significatifs (par exemple les personnes dans une image comme l'indique la figure III-6). Elle implique des tâches telles que la séparation de l'avant-plan « foreground » et de l'arrière-plan « background », le regroupement de régions sur la base de similitudes de couleur ou de forme, et la création d'un masque d'objets au niveau des pixels. [31]

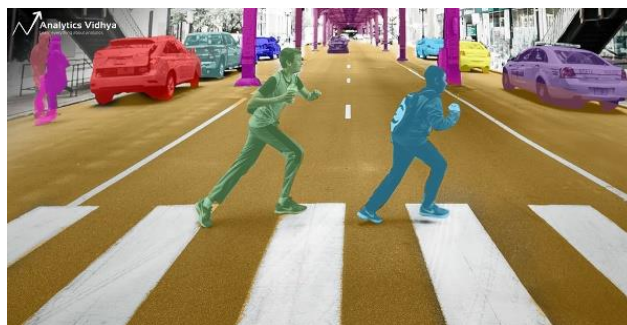


Figure III-6 : Exemple de segmentation d'image pour la détection de piétons et de véhicules dans une scène urbaine [32]

(i) Structure du réseau U-NET

Le réseau U-Net tire son nom de sa forme distinctive en « U ». Cette architecture comporte deux chemins : un chemin contractant (ou de descente) qui capture le contexte et un chemin symétrique d'expansion (ou de montée) qui permet une localisation précise.

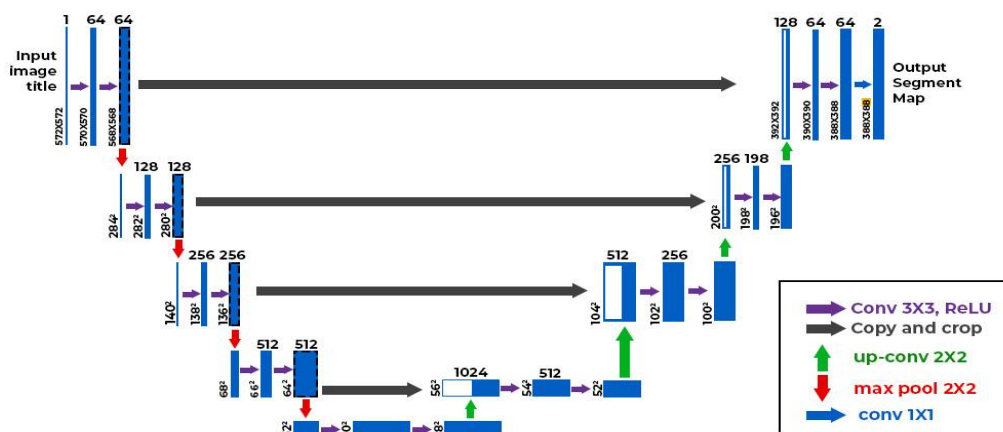


Figure III-7 : Architecture U-Net [33]

(ii) Description détaillée de l'architecture U-Net

- **Chemin de contraction (encodeur) :** L'entrée de l'U-Net est une image avec une résolution spatiale donnée et un certain nombre de canaux. Le chemin de contraction se compose de blocs successifs, chacun contenant deux couches de

convolution (illustrée en figure III-7) suivies d'une couche d'activation ReLU et d'une opération de max pooling. Chaque couche convolutive utilise des filtres de petite taille (typiquement 3x3) pour capturer les motifs locaux dans l'image. À chaque étape de contraction, la résolution spatiale de l'image est réduite (à cause du max pooling), mais la profondeur des cartes de caractéristiques augmente (à cause de l'ajout de filtres). Cela permet de capturer des informations de plus en plus abstraites et globales de l'image.

- **Partie la plus basse** : Au fond de l'U, il y a un bloc de convolution supplémentaire sans max pooling suivant. Cela représente le point de transition entre la contraction et l'expansion.
- **Chemin d'expansion (décodeur)** : Le chemin d'expansion inverse le processus de contraction. Il commence par des opérations d'up-convolution (ou déconvolution illustrée en figure III-8), qui augmentent la résolution spatiale des caractéristiques. Après chaque up-convolution, une opération de « copie et rognage » est effectuée, où les cartes de caractéristiques correspondantes du chemin de contraction sont copiées et fusionnées avec la sortie actuelle. Cela permet de restaurer les informations spatiales perdues lors de la contraction. Le chemin d'expansion se poursuit avec des couches de convolution pour raffiner les caractéristiques et reconstruire l'image à sa résolution originale.
- **Sortie** : La dernière couche du réseau est convolutive avec un nombre de filtres égal au nombre de classes de segmentation désirées. Elle produit une carte de segmentation où chaque pixel est classé dans une catégorie spécifique.

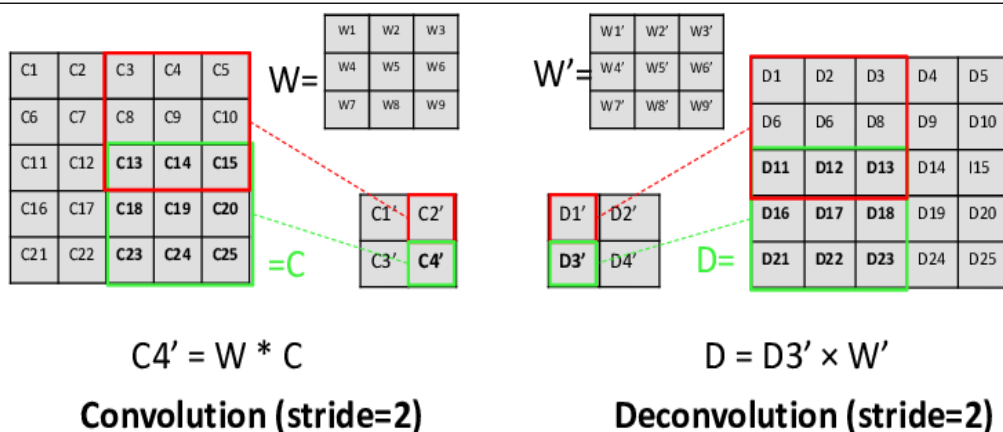


Figure III-8 : Principe de convolution et déconvolution [34]

La particularité d'U-Net réside dans sa capacité à fusionner les caractéristiques de contexte à partir du chemin contractant avec des informations de localisation plus détaillées à partir du chemin d'expansion. Cela se fait par des connexions de saut (skip connections) qui copient des caractéristiques de la partie contractante et les combinent avec les caractéristiques dans la partie d'expansion correspondante.

Ces connexions skip fonctionnent par concaténation : les cartes de caractéristiques de l'encodeur sont directement liées et « empilées » avec les cartes correspondantes du décodeur le long de la dimension des canaux. Par exemple, si l'encodeur produit une carte de caractéristiques de taille (H, W, C1) et le décodeur une carte de taille (H, W, C2) après upsampling, la concaténation résulte en une carte de taille (H, W, C1+C2). Cette combinaison est ensuite traitée par des convolutions supplémentaires pour « fusionner » ces informations.

(iii) Avantages du Choix de l'architecture UNET pour la segmentation

L'architecture UNET est devenue une référence incontournable pour la **segmentation d'images**, offrant de nombreux avantages par rapport aux autres approches. Tout d'abord, sa **structure symétrique** en forme de U permet une extraction et une reconstruction efficaces des caractéristiques à différentes échelles, grâce aux **connexions skip** qui facilitent la propagation des informations spatiales détaillées. En outre, **UNET est particulièrement performant lorsque les données d'entraînement sont limitées**, ce qui le rend particulièrement utile dans les domaines où les données étiquetées sont rares ou coûteuses à obtenir. L'une des forces d'UNET réside également dans sa capacité à **capturer les détails fins et les contours précis des objets**.

UNET offre également **une rapidité d'inférence**, grâce à sa structure permettant des prédictions relativement rapides une fois le modèle entraîné, ce qui est important pour les applications en temps réel ou nécessitant un traitement rapide. En outre, UNET est une architecture polyvalente, initialement conçue pour l'imagerie biomédicale, mais qui s'est avérée efficace dans divers domaines y compris la détection de manipulations d'images. UNET est également **facile à entraîner**, grâce à sa **convergence rapide et à sa faible sensibilité au problème de disparition du gradient**, grâce aux connexions skip. Enfin, UNET est **robuste aux transformations**, ce qui signifie qu'il peut gérer efficacement les rotations et les déformations dans les images, ce qui est utile pour les applications où les objets peuvent apparaître dans différentes orientations. Après avoir exploré en détail U-Net, nous allons maintenant nous pencher sur les différentes stratégies les plus employées pour optimiser les performances des modèles.

III.5 Conclusion

Dans ce chapitre, nous avons exploré les concepts clés de l'intelligence artificielle dans le domaine de la détection de falsification dans les images, ce qui nous a permis de développer une base solide et une compréhension approfondie de la problématique. Cela nous aidera à aborder la résolution de manière plus efficace.

Chapitre IV.

Méthodes statistiques de détection de falsifications d'images

IV.1 Introduction

Avant d'entamer la présentation de notre solution, il est à noter que notre approche a débuté par l'**exploration des techniques de prétraitement et d'amélioration d'images** (voir annexe B).

Nous nous concentrons désormais sur les **méthodes statistiques** de détection de falsifications, qui bien que plus anciennes, restent essentielles pour les experts en analyse forensique. Nous aborderons le phénomène du **JPEG Ghost**, l'**analyse des incohérences de bruit**, le **filtrage médian**, l'**ELA (Error Level Analysis)** et l'**analyse des artefacts de démosaïquage**. Bien que simples, ces méthodes restent efficaces pour détecter certaines falsifications d'images et constituent un premier niveau d'analyse incontournable. Ce chapitre détaille le fonctionnement, les avantages, les limites et des exemples d'application de chaque méthode, aidant ainsi les investigateurs à utiliser ces outils efficacement. Pour mieux comprendre ces méthodes et leur application, nous allons structurer ces techniques sous forme d'un organigramme présenté dans la figure IV-1

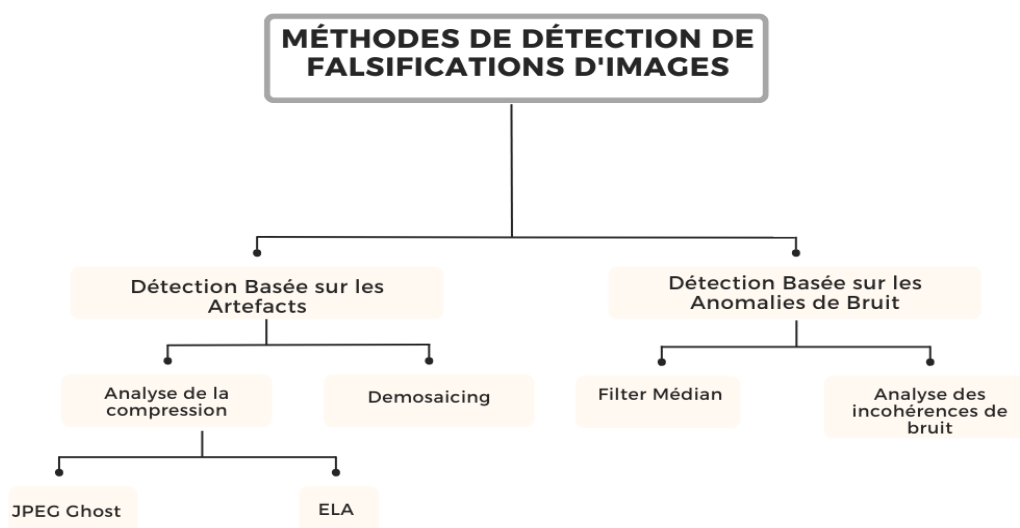


FIGURE IV-1: Organigramme des méthodes statistiques de détection de falsifications d'images et vidéos

L'organigramme présente une vue d'ensemble des méthodes statistiques utilisées pour détecter les falsifications dans les images numériques et les vidéos. L'organigramme est réparti en deux catégories principales : la détection basée sur les artefacts et la détection basée sur les anomalies de bruit.

La première catégorie, la détection basée sur les artefacts, est subdivisée en deux branches : l'analyse de la compression et le démosaïquage. L'analyse de la compression

comprend deux techniques : le JPEG Ghost, qui détecte les traces résiduelles laissées par les opérations de compression/décompression JPEG, et l'ELA (Error Level Analysis), qui révèle les niveaux de compression différents au sein d'une image.

La seconde catégorie, la détection basée sur les anomalies de bruit, se divise en deux techniques : le filtrage médian et l'analyse des incohérences de bruit. Le filtrage médian est une méthode de lissage d'image robuste qui permet de mettre en évidence les contours et les zones potentiellement retouchées ou insérées dans une image. L'analyse des incohérences de bruit, quant à elle, révèle des zones ayant subi des traitements différents, potentiellement liés à des manipulations.

IV.2 Détection Basée sur les Anomalies de Bruit

C'est une méthode de détection des falsifications d'images en identifiant les anomalies dans les propriétés de bruit d'une image. Le bruit est un motif aléatoire de variation des valeurs de pixels d'une image. Il peut être causé par divers facteurs, tels que le bruit du capteur et le bruit de quantification. La falsification peut introduire des anomalies dans les propriétés de bruit d'une image. Par exemple, si une partie d'une image est copiée et collée à partir d'une autre image (copy-move), le bruit de la région collée peut être différent du bruit de la zone environnante.

IV.2.1 Analyse des incohérences de bruit

La motivation derrière la méthode d'analyse des incohérences de bruit provient de la prise de conscience que **les images manipulées présentent souvent des irrégularités dans la distribution du bruit. Alors que les images authentiques affichent généralement des caractéristiques de bruit uniformes sur l'ensemble de l'image, l'ajout de bruit aléatoire localisé sur les régions falsifiées peut perturber cette uniformité.** Détecter ces incohérences dans le motif de bruit peut fournir des informations précieuses sur l'authenticité d'une image et aider à identifier les zones susceptibles d'avoir été manipulées. [35]

L'objectif principal de cette méthode est de **segmenter une image en fonction de ses caractéristiques de bruit.** Étant donné une image avec des régions présentant des niveaux de bruit variables, on vise à détecter et localiser précisément ces régions. Cela implique d'estimer la variabilité spatiale du bruit au sein de l'image et de la partitionner en segments homogènes basés sur ces variations. La méthode repose sur l'hypothèse d'un modèle de bruit non stationnaire, où la variance du bruit peut varier spatialement au sein de l'image. Ce modèle représente l'image comme une combinaison de signal non corrompu $S(x, y)$ et de bruit gaussien blanc $N(x, y)$, où :

$$I(x, y) = S(x, y) + N(x, y) \quad (4.1)$$

Le bruit $N(x, y)$ est supposé suivre une distribution gaussienne avec une variance $\sigma_N^2(x, y)$ qui varie spatialement, c'est-à-dire :

$$N(x, y) \sim \mathcal{N}(0, \sigma_N^2(x, y)) \quad (4.2)$$

Pour estimer la variance du bruit à chaque position (x, y) , une fenêtre de taille $\omega \times \omega$ est utilisée, et la variance locale estimée $\widehat{\sigma}_N^2(x, y)$ est calculée par :

$$\widehat{\sigma}_N^2(x, y) = \frac{1}{\omega^2} \cdot \sum_{(i,j) \in fenetre} \{I(i, j) - \bar{I}(x, y)\}^2 \quad (4.3)$$

Où : $\bar{I}(x, y)$ est la moyenne des intensités dans la fenêtre centrée à (x, y) . En adoptant ce modèle, la méthode prend en compte la nature complexe de la distribution du bruit dans les images réelles et adapte son analyse en conséquence.

Méthodologie :

L'analyse des incohérences de bruit repose sur une méthodologie rigoureuse impliquant plusieurs étapes clés. La première consiste à décomposer l'image à l'aide d'une **transformée en ondelettes** afin d'isoler les différentes composantes fréquentielles. Cette décomposition permet ensuite de se concentrer sur les détails hautes fréquences diagonaux, particulièrement sensibles aux variations de bruit. L'image est alors partitionnée en blocs non chevauchants sur lesquels les niveaux de bruit sont estimés individuellement par des techniques d'estimation basées sur les ondelettes. Enfin, un algorithme de fusion des blocs regroupe les zones présentant des caractéristiques de bruit similaires en régions homogènes, mettant en évidence les zones potentiellement manipulées. Examinons les différentes étapes de cette méthodologie :

Transformée en ondelettes : La transformée en ondelettes est un outil mathématique permettant de décomposer un signal ou une image en différentes composantes fréquentielles. Dans notre contexte, elle décompose l'image en ses bandes de fréquences constituantes, révélant les détails des basses (approximations) et hautes (détails) fréquences. Cette décomposition est cruciale pour analyser les caractéristiques du bruit, car elle permet d'isoler et d'examiner les motifs de bruit à travers différentes échelles et orientations au sein de l'image (coefficients d'approximation (LL), coefficients horizontaux (HL), coefficients verticaux (LH) et coefficients diagonaux (HH)). En appliquant l'analyse par ondelettes, la méthode acquiert la distribution spatiale du bruit et prépare l'image pour les étapes de traitement ultérieures.

La figure IV-2 explique comment l'image est décomposée en utilisant des filtres passe-haut h_{high} et passe-bas h_{low} le long des colonnes et des lignes, suivis d'un sous-échantillonnage par 2. Les résultats sont les coefficients diagonaux D (HH), verticaux V (LH), horizontaux H (HL) et d'approximation A (LL), permettant d'analyser les composantes fréquentielles et les motifs de bruit à différentes échelles et orientations. La figure IV-3 montre les différents coefficients de la Transformée discrète en ondelettes (Discrete wavelet transform : DWT)

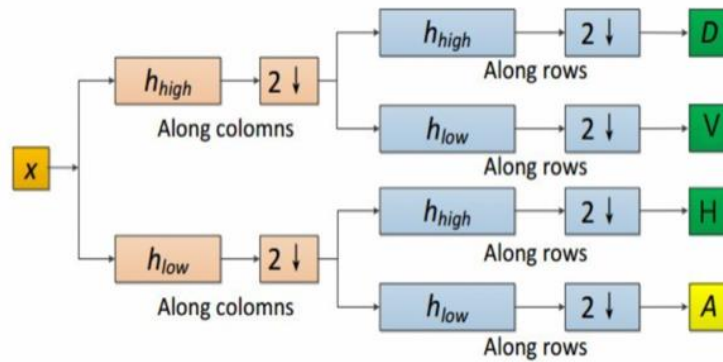


FIGURE IV-2: Illustration de la transformée en ondelettes appliquée à une image x [36]

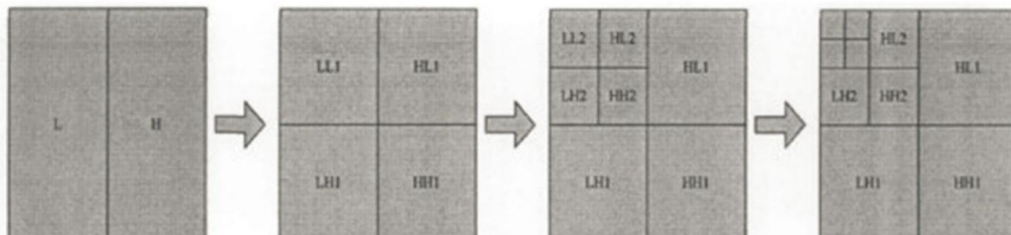


FIGURE IV-3: Les coefficients diagonaux D (HH), verticaux V (LH), horizontaux H (HL) et d'approximation A (LL) de la transformée discrète en ondelettes. [37]

Blocs non chevauchants : Après la transformée en ondelettes, la méthode se concentre sur les **détails diagonaux hautes fréquences** de l'image, représentés par la sous-bande HH1 qui contient les caractéristiques d'image à petite échelle et est particulièrement sensible aux variations de bruit. La méthode **subdivise la sous-bande HH1 du premier niveau de décomposition en blocs Bi** non chevauchants de taille prédéfinie (généralement des blocs carrés de $R \times R$ pixels). En décomposant l'image en plus petits blocs, la méthode assure une granularité dans son analyse et améliore sa capacité à détecter les variations subtiles des niveaux de bruit dans les régions de l'image. On aura alors, pour une image de taille $M \times N$, r blocs avec :

$$r = \left\lfloor \frac{M}{R} \right\rfloor \times \left\lfloor \frac{N}{R} \right\rfloor \quad (4.4)$$

Estimation du niveau de bruit : Une fois l'image partitionnée en blocs, la méthode estime l'écart-type du bruit au sein de chaque bloc. La méthode emploie des techniques basées sur les ondelettes, en se concentrant spécifiquement sur la sous-bande HH1. En utilisant un estimateur basé sur la médiane (MAD : median absolute deviation), la méthode estime robustement l'écart-type du bruit, même en présence de valeurs aberrantes ou de motifs de bruit complexes. Cette étape fournit des informations essentielles sur les caractéristiques du bruit de chaque bloc. L'équation utilisée pour estimer l'écart-type du bruit est la suivante :

$$\hat{\sigma} = \frac{\text{median}(|HH1|)}{0.6745} \quad (4.5)$$

Dans cette équation, la médiane des coefficients de la sous-bande diagonale HH1 est divisée par 0.6745, un **facteur de normalisation** qui permet de convertir la déviation absolue médiane en une estimation de l'écart-type. **Ce facteur est utilisé car, pour une distribution normale (ou gaussienne), la déviation absolue médiane est environ 0.6745 fois l'écart-type.** Ainsi, cette méthode permet une estimation robuste et fiable de l'écart-type du bruit dans chaque bloc, même en présence de valeurs aberrantes ou de motifs de bruit complexes.

Fusion des blocs : Une fois les estimations des niveaux de bruit obtenues pour chaque bloc, la méthode procède au partitionnement de l'image en sous-régions homogènes connexes basées sur ces caractéristiques de bruit. Ce partitionnement est réalisé par un processus de fusion de blocs ayant des niveaux de bruit similaires qui sont regroupés pour former des régions contiguës. L'algorithme de fusion combine itérativement les blocs voisins en se basant sur un **seuil de similarité T** prédéfini :

$$|\hat{\sigma}_i - \hat{\sigma}_j| < T \quad (4.6)$$

Assurant que les régions ayant des niveaux de bruit cohérents soient regroupées (vérifiant l'équation 4.6). Cela résulte en une carte partitionnée de l'image, mettant en évidence les régions ayant des caractéristiques de bruit distinctes et fournissant un aperçu de la variabilité spatiale du bruit au sein de l'image.

La figure IV-4 nous présente une application intéressante d'une méthode d'analyse des inconsistances de bruit, utilisée pour détecter des manipulations dans une image. Sur la partie gauche de la figure, nous pouvons observer un paysage naturel dans lequel un oiseau a été ajouté. À première vue, cette insertion pourrait passer inaperçue, mais c'est sans compter sur l'analyse des niveaux de bruit, présentée sur la partie droite de la figure. Cette analyse révèle une inconsistance de bruit dans une région de l'image, encerclée en

rouge. En effet, il est possible de noter une différence de couleur entre l'oiseau et le fond de l'image, ce qui indique clairement que l'oiseau a été ajouté.

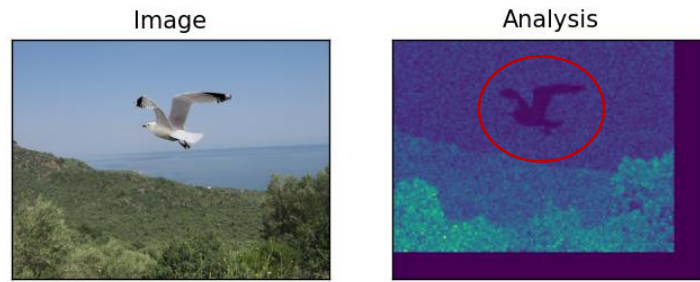


FIGURE IV-4: Application de la méthode d'analyse des inconsistances de bruit

Bien que l'analyse des incohérences de bruit soit une méthode puissante pour détecter les falsifications d'images, il existe une autre approche complémentaire basée sur **l'étude des résidus de bruit après filtrage**. Cette technique, en tirant également parti des propriétés inhérentes du bruit dans les images numériques, offre une perspective différente et complémentaire pour identifier les régions manipulées.

IV.2.2 Analyse des résidus de bruit post-filtrage par filtre médian

Une approche efficace pour exploiter ce principe consiste à appliquer un filtre de suppression du bruit à l'image et à analyser les résidus de bruit restants. Les filtres de suppression de bruit sont conçus pour supprimer ou éliminer le bruit tout en préservant le contenu sous-jacent de l'image. Cependant, lorsqu'ils sont appliqués aux régions falsifiées, ces filtres ne peuvent pas efficacement supprimer le bruit en raison des caractéristiques de bruit incohérentes introduites par la manipulation.

Bien que divers filtres de suppression de bruit puissent être utilisés, le **filtre médian** est un choix populaire en raison de sa simplicité et de son efficacité pour supprimer le bruit tout en préservant les contours image I . Il fonctionne en remplaçant chaque pixel $I(x,y)$ par la médiane des valeurs des pixels dans un voisinage défini par un noyau K . Mathématiquement, cela peut être exprimé par :

$$I_{\text{filtre médian}}(x,y) = \text{median} \{I(x+i,y+j) \mid (i,j) \in K\} \quad (4.7)$$

Où : $(x+i,y+j)$ parcourt tous les pixels voisins inclus dans le noyau K , et *median* représente la médiane des valeurs.

La méthodologie d'analyse des résidus de bruit impliquant le filtre médian peut être résumée comme suit :

1. Appliquer le filtre médian à l'ensemble de l'image, ce qui donne une version lissée et débruitée de l'image.
2. Calculer l'image résiduelle de bruit $R(x, y)$ en soustrayant l'image filtrée $I_{\text{filtre médian}}(x, y)$ de l'image originale $I(x, y)$, puis l'amplifier pour obtenir la carte de bruit amplifiée $R_{\text{amplifiée}}(x, y)$. Mathématiquement, cela se définit par :

$$R(x, y) = I(x, y) - I_{\text{filtre médian}}(x, y) \quad (4.8)$$

$$R_{\text{amplifiée}}(x, y) = \alpha \cdot R(x, y) \quad (4.9)$$

Où : α est un facteur d'amplification.

L'image résiduelle de bruit $R(x, y)$ révèle le bruit dans les zones authentiques, tandis que les zones falsifiées montrent des motifs de bruit distincts. Cette carte de bruit est cruciale en forensique. Elle identifie les zones suspectes où le bruit est anormalement élevé. Une inspection visuelle permet de localiser et d'étudier ces parties altérées, offrant des indices sur l'authenticité de l'image. Cependant, cette méthode nécessite d'être combinée à d'autres techniques et analysée rigoureusement en tenant compte des caractéristiques du bruit et des limitations du filtre utilisé. Dans l'exemple illustré dans la figure IV-5, nous appliquons cette méthode à une image contenant une falsification par l'ajout d'un oiseau. L'image falsifiée est à droite (a), l'image filtrée au centre (b), et l'image des résidus amplifiée est à gauche (c). Dans cette dernière image, le cercle rouge indique que l'oiseau est plus sombre que le reste de l'image, montrant ainsi un niveau de bruit différent introduit par la manipulation.



(a) Image falsifiée

(b) Image filtrée

(c) Résidus de bruits

FIGURE IV-5: Détection de Manipulation par Analyse des Résidus de Bruit

IV.3 Détection basée sur les artefacts

Ce sont les différentes approches qui s'attaquent aux traces et irrégularités introduits dans les images par le processus de manipulation, comme les bordures floues, les incohérences de texture et les artefacts de compression. Il existe plusieurs causes à l'apparition de ces artefacts dans les images manipulées. La **compression JPEG** est l'une des causes les plus courantes. Cet algorithme de compression avec perte peut introduire des artefacts dans les images, tels que le **quadrillage (blocking)** et les **sonneries (ringing)**. Le **rééchantillonnage** est un autre processus qui peut introduire des artefacts. Il s'agit de la modification de la taille d'une image, ce qui peut entraîner l'apparition de **flou et de crénelage (Aliasing)** illustré dans la figure IV-6.



FIGURE IV-6: Exemple de crénelage (Aliasing) [38]

L'aliasing se manifeste comme des bords irréguliers ou des motifs de moiré qui n'existent pas dans la scène réelle (à droite) mais apparaissent à cause de l'échantillonnage.

Du nœud père (la détection basée sur les artefacts), nous passons vers le nœud suivant : **la détection basée sur la compression**.

IV.3.1 Analyse de la compression

La compression est un processus utilisé pour réduire la taille d'un fichier, en l'occurrence une image, en éliminant ou en réduisant les données redondantes ou inutiles. Il existe deux types de compression : **sans et avec perte**.

Dans la **compression sans perte**, les données d'origine peuvent être parfaitement reconstruites à partir des données compressées. Aucune information n'est perdue pendant le processus de compression. Les méthodes de compression sans perte sont utilisées lorsqu'il est essentiel de maintenir l'intégrité de l'image, comme dans l'imagerie médicale ou à des fins d'archivage. Les formats PNG, GIF représentent des exemples de compression sans perte.

En revanche, dans la **compression avec perte**, certaines données sont éliminées lors du processus de compression pour obtenir des taux de compression plus élevés. Cela signifie que la qualité de l'image est réduite, car certaines informations visuelles sont perdues. Les méthodes de compression avec perte sont souvent utilisées lorsqu'une taille de fichier plus petite est plus importante que la préservation de tous les détails de l'image. Le format JPEG est l'exemple type de compression avec perte. [39]

Le format JPEG (Joint Photographic Experts Group) utilise une compression avec perte basée sur la transformée en cosinus discrète (DCT) et le codage entropique comme illustré dans la figure IV-7 :

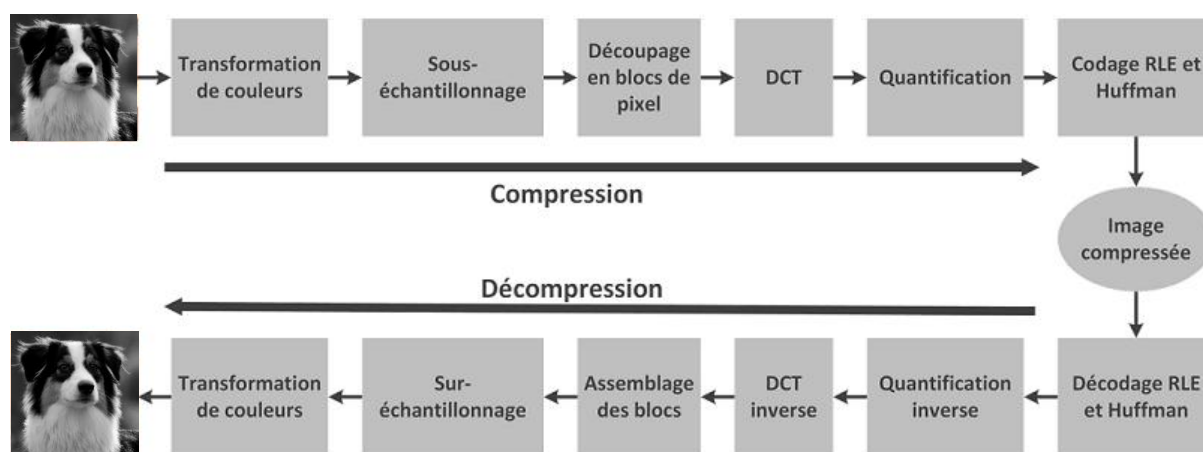


FIGURE IV-7: Processus de compression d'une image au format JPEG

Le processus de compression JPEG débute par **une transformation de couleurs**, où l'image est convertie du modèle initial (généralement RGB) vers un modèle luminance/chrominance (YCrCb) séparant la composante de luminance (intensité) des composantes de chrominance (teinte et saturation). Une étape de **sous-échantillonnage** est ensuite appliquée pour réduire les informations de chrominance, exploitant la moindre sensibilité de l'œil humain aux variations de couleur par rapport à la luminance. Il existe trois modes de sous-échantillonnage dans la norme JPEG : **4:4:4**, **4:2:2** et **4:2:0**. Le premier nombre représente la largeur d'un bloc pour le traitement de la chrominance. En JPEG cette largeur est toujours de quatre. Le deuxième nombre représente le nombre d'échantillons de chrominance dans la première ligne et le deuxième nombre dans la deuxième ligne. La figure III-8 représente les différents types de sous-échantillonnage :

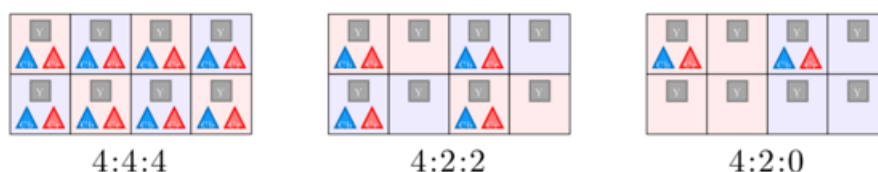


FIGURE IV-8: Différents modes de sous échantillonnage de la chrominance [40]

Où on a :

- **Le mode 4:4:4** représente une image d'origine sans altération, contenant des informations complètes de luminance (Y) et de chrominances (Cr et Cb), offrant une meilleure qualité visuelle mais un taux de compression faible.
- **En mode 4:2:2**, toutes les informations de luminance sont conservées, mais seulement deux informations de Cr et deux de Cb par ligne. Les pixels sans information de chrominance sont décodés par interpolation, ce qui réduit la finesse des couleurs mais augmente considérablement le taux de compression.
- **Le mode 4:2:0** supprime toutes les informations de chrominance de la deuxième ligne. Lors du décodage, une approximation est faite par moyenne pour retrouver ces informations, ce qui améliore le taux de compression mais peut affecter la qualité visuelle.

L'image est ensuite **découpée en blocs de pixels de taille fixe**, typiquement 8x8, qui seront traités indépendamment. Sur chaque bloc, **une transformée en cosinus discrète (DCT)** (Equation 4.10) est appliquée pour représenter l'information en termes de fréquences spatiales, distinguant les basses des hautes fréquences.

$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2x+1)v\pi}{16} \right] \quad (4.10)$$

Où :

- v est l'espace de fréquence vertical pour les entiers $0 \leq v < 8$
- u est l'espace de fréquence horizontal pour les entiers $0 \leq u < 8$
- $\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } u = 0 \\ 1, & \text{sinon} \end{cases}$ est un facteur d'échelle de normalisation pour rendre la transformation orthonormée
- $g_{x,y}$ est la valeur du pixel à la position (x, y)
- $G_{u,v}$ est le coefficient de la DCT à la position (u, v)

Une étape cruciale de **quantification** suit, où les coefficients de hautes fréquences sont davantage quantifiés, entraînant une perte de précision et l'apparition d'artefacts caractéristiques de la compression JPEG (Il suffit de diviser chaque coefficient de notre

matrice des fréquences : résultat après DCT, par une constante de la matrice de quantification : figure IV-9, pour obtenir notre matrice quantifiée.)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	92
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

FIGURE IV-9: Exemple de matrice de quantification pour une compression plus rapide [41]

Enfin, le bloc quantifié est encodé par un codage par plages uniquement sur le nombre 0 (RLE : codage simple qui consiste à remplacer toutes les suites de caractères identiques par le nombre de caractères suivit du dit caractère) suivi d'un codage de Huffman (codage sans perte avec longueur variable à base d'un arbre) afin de réduire encore la taille des données. Nous devons transformer la matrice en chaîne de nombres avant de l'encoder, pour cela, nous la parcourons avec un ZIG-ZAG comme indiquée dans la figure IV-10 :

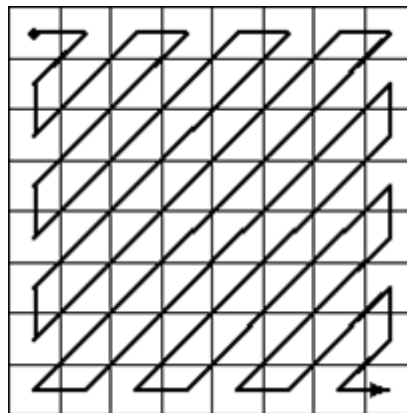


FIGURE IV-10: Parcours pour transformer la matrice en chaîne de nombre [41]

Munis de ces connaissances sur la compression et son processus, nous sommes maintenant prêts à explorer la méthode de détection des falsifications d'images basée sur l'analyse de **JPEG Ghost**.

a) JPEG Ghost :

Lorsqu'une image numérique est falsifiée par composition, comme en combinant la tête d'une personne avec le corps d'une autre ou en insérant des éléments provenant d'autres sources, des traces subtiles des compressions initiales peuvent subsister. La technique JPEG Ghost exploite les propriétés de la compression JPEG pour détecter si une partie de l'image a été compressée à une qualité inférieure au reste. [42]

JPEG introduit des artefacts caractéristiques. Si une région de l'image a subi une compression différente, ces niveaux distincts laisseront des patterns d'artefacts subtils mais détectables. L'analyse minutieuse de ces patterns permet de mettre en évidence les zones potentiellement manipulées ou insérées. JPEG Ghost exploite ces différences de compression pour révéler les régions suspectes d'une image. Les JPEG Ghost dévoilés par cette méthode peuvent fournir des preuves de manipulations (insertion ou suppression d'éléments ou le remplacement de parties d'une image).

Pour bien comprendre la méthode JPEG Ghost, nous allons nous concentrer sur **la quantification** dans le processus de compression JPEG. Chaque coefficient de la transformée en cosinus discrète (DCT) est divisé par une valeur de quantification (q), comme indiquée dans l'équation 4.11 :

$$\hat{c} = \text{round} \left(\frac{c}{q} \right) \quad (4.11)$$

Déterminée en fonction de la fréquence spatiale et du canal de l'image (ces valeurs sont généralement plus élevées dans les canaux de chrominance et les hautes fréquences spatiales, car elles sont moins perceptibles par le système visuel humain). Cette étape **réduit la précision des coefficients** pour une compression efficace, mais des valeurs de quantification plus élevées entraînent une compression plus agressive avec une dégradation de la qualité.

Double Compression : Lorsque des coefficients de l'image sont quantifiés à un niveau donné q_1 et qu'ils subissent ensuite une seconde quantification à un niveau inférieur q_2 , **des différences apparaissent entre les ensembles de coefficients**. On aura alors :

$$- \quad \text{Première Quantification } (q_1) : \hat{c}_1 = \text{round} \left(\frac{c}{q_1} \right) \quad (4.12)$$

$$- \quad \text{Seconde Quantification } (q_2) : \hat{c}_2 = \text{round} \left(\frac{\hat{c}_1}{q_2} \right) \quad (4.13)$$

Cette re-quantification peut se produire si l'image est compressée une deuxième fois. On aura donc 3 cas à vérifier :

- **Niveaux Égaux ($q_2=q_1$)** : Les différences entre les coefficients quantifiés sont minimales car la précision reste constante.
- **Niveau Inférieur à la Première Quantification ($q_2 < q_1$)** : La seconde quantification est moins agressive. Cependant, les erreurs introduites par la première quantification q_1 persistent, ce qui cause un décalage des valeurs des coefficients et crée des différences.
- **Niveau Supérieur à la Première Quantification ($q_2 > q_1$)** : La seconde quantification est plus agressive, réduisant encore plus la précision des coefficients. Cela augmente les différences entre les ensembles de coefficients quantifiés.

Identification des régions de compression différentielle

Ces différences entre les coefficients quantifiés à différents niveaux sont calculées pour **identifier les régions où la compression a été appliquée de manière distincte ou variable à différentes parties de l'image**. Les « fantômes JPEG » ou JPEG Ghosts se manifestent par des **minima locaux** dans ces différences, indiquant des zones où les coefficients ont été précédemment quantifiés avec une valeur de quantification plus élevée (qualité inférieure).

La détection des fantômes JPEG implique également une analyse des différences entre les valeurs de pixels directement (Equation 4.14), plutôt que de simplement comparer les coefficients quantifiés.

$$d(x, y, q) = \frac{1}{3} \sum_{i=1}^3 [f(x, y, i) - f_q(x, y, i)]^2 \quad (4.14)$$

Où : $f(x, y, i)$, $i = 1, 2, 3$, représente chacun des canaux RGB, et $f_q(\cdot)$ est le résultat de la compression de $f(\cdot)$ A la qualité q .

Étant donné que la différence d'image est calculée sur l'ensemble des fréquences spatiales, une région avec de faibles quantités de contenu à haute fréquence spatiale (par exemple, un ciel principalement uniforme) présentera une différence plus faible par rapport à une région fortement texturée (par exemple, de l'herbe). Afin de compenser ces différences, nous considérons une mesure de différence spatialement moyennée et normalisée. L'image de différence est d'abord moyennée sur une région de pixels de taille $b \times b$ comme suit :

$$\delta(x, y, q) = \frac{1}{3} \sum_{i=1}^3 \frac{1}{b^2} \sum_{b_x=0}^{b-1} \sum_{b_y=0}^{b-1} [f(x + b_x, y + b_y, i) - f_q(x + b_x, y + b_y, i)]^2 \quad (4.15)$$

Et puis on la normalise, pour que la moyenne des différences pour chaque position (x, y) soit dans l'intervalle $[0,1]$:

$$d(x, y, q) = \frac{\delta(x, y, q) - \min_q[\delta(x, y, q)]}{\max_q[\delta(x, y, q)] - \min_q[\delta(x, y, q)]} \quad (4.16)$$

Cette approche mathématique permet de **minimiser les erreurs dues aux variations dans les tables de quantification et les qualités JPEG entre différents appareils photo et logiciels**. Pour compenser d'éventuels désalignements causés par la re-compression, **l'image est décalée à plusieurs positions avant d'être enregistrée à chaque qualité JPEG**. Cette approche garantit une détection précise des fantômes JPEG en tenant compte de toutes les variations possibles dans l'alignement des blocs de pixels lors de la re-compression. C'est une technique **simple mais potentiellement puissante pour détecter les altérations dans les images JPEG de faible qualité**. Cette approche détecte explicitement si une partie de l'image a été compressée à une qualité inférieure à la qualité JPEG enregistrée de l'ensemble de l'image. **Une telle région est détectée en re-sauvegardant simplement l'image à une multitude de qualités JPEG et en détectant les minima locaux spatialement localisés dans la différence entre l'image et sa contrepartie compressée en JPEG**. Dans de nombreuses situations, ces minima (fantômes JPEG), sont très saillants et facilement détectés.

Les graphiques présentés dans la figure IV-11 illustrent le concept de JPEG Ghost. L'axe horizontal représente différents niveaux de qualité JPEG auxquels l'image est re-sauvegardée. L'axe vertical montre la différence normalisée entre l'image originale et sa version re-compressée à chaque niveau de qualité. **Le graphique (a) présente une courbe relativement lisse avec un minimum distinct autour du niveau de quantification 17, indiquant la qualité de compression originale de l'image entière. La forme régulière suggère une compression uniforme sur toute l'image**. En revanche, **le graphique (b) montre une courbe plus irrégulière avec plusieurs minima locaux, suggérant la présence des JPEG Ghost c-à-d. : des régions de l'image ayant subi des compressions différentes**. D'un point de vue forensique, (a) représente **probablement** une image non manipulée. En revanche, le graphique (b) indique une image potentiellement manipulée, où certaines parties ont été compressées différemment ou proviennent de sources distinctes. Les minima locaux dans (b) correspondent à des JPEG Ghost, révélant des zones de l'image ayant subi des compressions antérieures différentes.

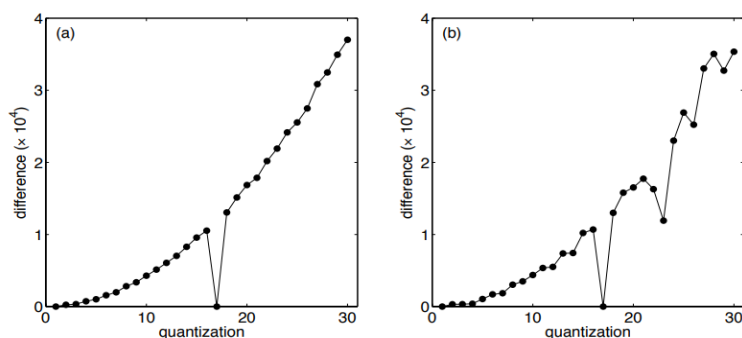


FIGURE IV-11: Différences normalisées vs. Niveaux de re-quantification JPEG [43]

(a) Cas d'une image non manipulée. (b) Cas d'une image suspectée de manipulation.

Après avoir examiné les graphiques de quantification qui nous ont fourni une perspective numérique sur la méthode JPEG Ghost, passons maintenant à une représentation visuelle plus concrète. La figure IV-12 représente une série d'images qui illustrent l'efficacité de la technique JPEG Ghost pour détecter une manipulation spécifique : l'ajout d'un oiseau à une image de paysage. L'image originale en couleur (en haut à gauche) montre un oiseau au-dessus d'un paysage verdoyant, mais cette scène est en réalité le résultat d'une manipulation.

Les images suivantes, numérotées de 60 à 96, représentent les différences entre l'image manipulée et ses versions recompressées à différents niveaux de qualité JPEG. Dans ces visualisations en niveaux de gris, les zones plus claires indiquent des différences plus importantes. Il est clairement visible que l'oiseau apparaît comme une zone distinctement différente, plus sombre, dans presque toutes les versions de l'image, en particulier dans les compressions de niveau moyen (70-90). Cette différence marquée suggère que l'oiseau a été ajouté à l'image originale et a subi une histoire de compression différente du reste de l'image.

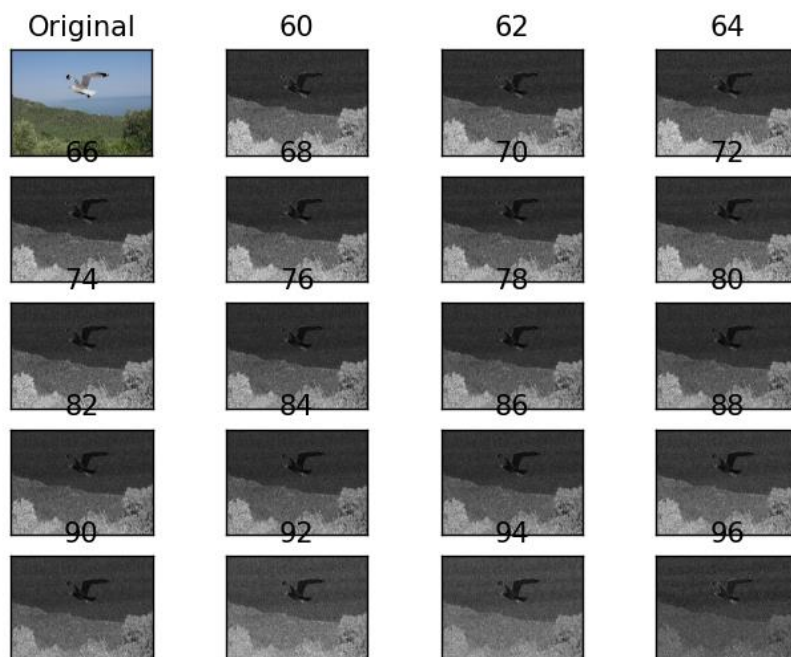


FIGURE IV-12: Effet JPEG Ghost, une analyse multi-niveau de compression

b) Analyse du niveau d'erreur :

L'analyse du niveau d'erreur, ou **Error Level Analysis (ELA)** en anglais, est une technique avancée utilisée dans la forensique numérique et l'analyse d'images. Cette méthode se concentre principalement sur l'examen des images compressées au format JPEG pour détecter d'éventuelles altérations ou manipulations. [45]

Le principe de l'ELA repose sur l'étude des caractéristiques de compression avec perte des images JPEG. La technique fonctionne en recompressant l'image à un niveau de qualité connu, puis en comparant cette version avec l'original. Les zones de l'image présentant des différences significatives de niveau de compression sont mises en évidence, ce qui peut indiquer des modifications. En pratique, l'ELA permet aux experts en forensique de :

- Identifier les parties d'une image modifiées, ajoutées ou supprimées.
- Analyser la cohérence de la compression dans l'ensemble de l'image.
- Obtenir des indices sur l'historique de manipulation d'une image.

L'algorithme global de l'Analyse du Niveau d'Erreur peut être résumé en une série d'étapes principales. La figure IV-13 illustre ces étapes de manière simplifiée, offrant une vue d'ensemble du flux de travail de l'ELA.

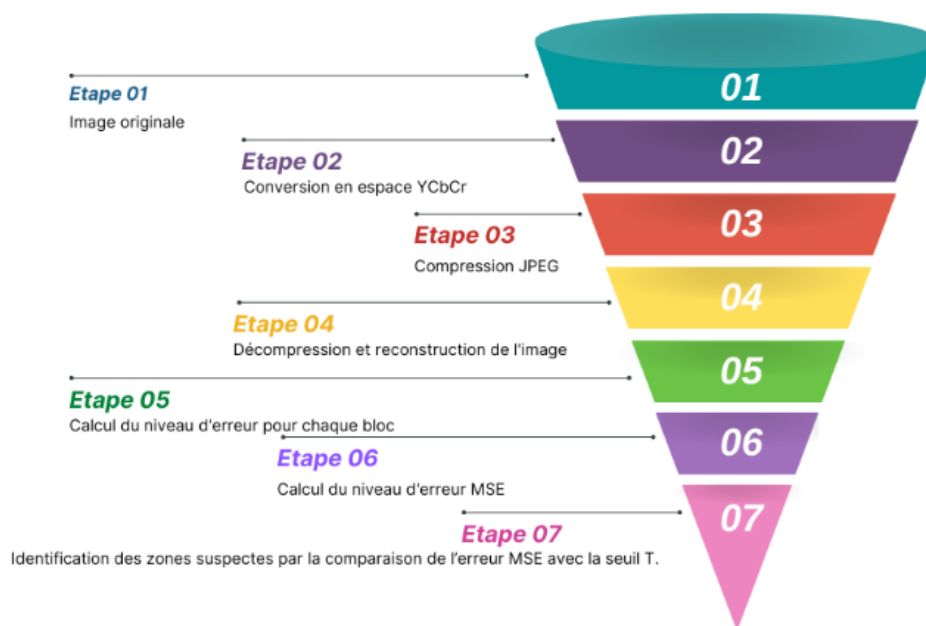


FIGURE IV-13: Processus en entonnoir de l'Analyse du Niveau d'Erreur (ELA) pour la détection de falsifications d'images

(1) **Conversion de l'image en espace colorimétrique YCbCr, division en blocs et compression de chaque bloc à l'aide de l'algorithme JPEG :**

L'image est d'abord convertie en espace colorimétrique YCbCr, qui sépare les composantes de luminance (Y) et de chrominance (CbCr). L'image est ensuite divisée en blocs B de 8x8, et chaque bloc est compressé indépendamment à l'aide de l'algorithme JPEG à un facteur de qualité prédéfini, tel que 95% (Comme il est déjà expliqué avant dans le processus de compression).

$$B_{compressed} = JPEG(B, \text{qualité}) \quad (4.17)$$

Où : qualité représente le facteur de qualité, par exemple 95%.

Le processus de compression introduit des artéfacts, en particulier dans les zones où l'image a été altérée ou modifiée comme illustré dans la figure IV-14.



FIGURE IV-14: Exemple d'artefacts de compression dans une image JPEG [45]

(2) **Décompression des blocs compressés et reconstruction de l'image enregistrée :**

Après la compression de chaque bloc, l'algorithme décompresse les blocs compressés $B_{compressed}$:

$$B_{decompressed} = JPEG^{-1}(B, \text{qualité}) \quad (4.18)$$

Ensuite on combine ces blocs compressés pour reconstruire la version enregistrée (compressée) de l'image I^{\sim} .

(3) **Calcul du niveau d'erreur pour chaque bloc en comparant les blocs d'origine et enregistrés :**

L'algorithme calcule le niveau d'erreur pour chaque bloc en mesurant les différences entre le bloc d'origine non compressé B et le bloc enregistré décompressé $B_{decompressed}$.

$$E = B - B_{decompressed} \quad (4.19)$$

Après cela on calcule l'erreur quadratique moyenne (MSE) pour chaque bloc B_i :

$$MSE(B_i) = \frac{1}{N} \sum_{j=1}^N (B_{i,j} - B_{i,j}^{decompressed})^2 \quad (4.20)$$

Où : N est le nombre de pixels dans le bloc, dans notre cas $N = 8 \times 8$.

Les zones présentant des différences significatives indiquent un potentiel de manipulation ou d'altération.

(4) **Définition d'une valeur de seuil pour identifier les zones de niveau d'erreur élevé :**

Une valeur de seuil T est définie pour déterminer les zones de l'image présentant un niveau d'erreur élevé. Cette valeur de seuil est généralement déterminée empiriquement ou en fonction des connaissances préalables de l'algorithme de compression et des artéfacts de compression attendus. Pour identifier les blocs avec une MSE qui dépasse T on suit cette équation :

$$si: MSE(B_i) > T \quad (4.21)$$

Donc les zones dépassant ce seuil sont considérées comme suspectes et peuvent nécessiter une enquête plus approfondie.

(5) Mise en évidence des zones de l'image dépassant la valeur de seuil :

Enfin, l'algorithme met en évidence ou signale les zones de l'image qui dépassent la valeur de seuil prédéfinie. En peut illustrer cela par cette équation :

$$I_{\tilde{highlighted}} = \{B_i \mid MSE - B_i) > T\} \quad (4.22)$$

Ces zones mises en évidence sont considérées comme suspectes et peuvent indiquer un potentiel de manipulation ou d'altération.

La figure IV-15 montre l'application de l'Analyse du Niveau d'Erreur (ELA) sur une image. À gauche, on voit l'image avec un oiseau ajouté. À droite, on peut observer le résultat de l'ELA, où l'oiseau apparaît distinctement et se démarque du fond. Cette différence de couleur dans l'analyse ELA met en évidence l'oiseau comme un élément potentiellement suspect, suggérant une possible manipulation de cette partie de l'image.



FIGURE IV-15: Exemple de la détection de falsification à l'aide de la méthode ELA

IV.3.2 Dé-mosaïquage

La capture d'une image en couleurs RGB avec un appareil photo numérique nécessite, en théorie, trois capteurs (CCD ou CMOS) : un pour la lumière rouge, un pour la lumière verte et un pour la lumière bleue. Cependant, pour réduire les coûts, la plupart des appareils photo grand public n'utilisent qu'un seul capteur équipé d'un filtre de

Bayer. Ce filtre organise la détection des couleurs en mosaïque, capturant des informations incomplètes sur chaque pixel. Pour transformer cette image en mosaïque en une image RGB complète, différents algorithmes de dé-mosaïquage sont utilisés pour interpoler les couleurs manquantes. Le choix de l'algorithme de dé-mosaïquage influence considérablement la qualité de l'image résultante. Il est ainsi possible d'utiliser plusieurs algorithmes de dé-mosaïquage pour générer des images de qualité variable à partir de la mosaïque originale capturée par le filtre de Bayer. [46]

Nous allons maintenant expliquer la procédure qui permet de générer une image RGB complète à partir des données originales du capteur (image en mosaïque). Pour cela, nous allons expliquer le concept d'interpolation dans le dé-mosaïquage. L'interpolation dans le dé-mosaïquage fait référence à la technique utilisée pour générer une image couleur complète à partir des échantillons de couleurs partiels obtenus par le capteur d'une caméra utilisant un filtre de couleurs (CFA). Étant donné que chaque pixel d'un CFA ne capture qu'une seule couleur primaire (rouge, vert ou bleu), les algorithmes d'interpolation prédisent les deux couleurs manquantes à chaque emplacement de pixel en analysant les valeurs des pixels voisins.

Il existe deux types d'interpolation dans le dé-mosaïquage : **L'interpolation bilinéaire** et **L'interpolation linéaire de haute qualité (High-Quality Linear Interpolation)**.

L'interpolation bilinéaire : C'est la méthode la plus élémentaire utilisée pour le dé-mosaïquage d'une image de Bayer. Cette approche se base sur le principe selon lequel la couleur d'un pixel absent est vraisemblablement similaire aux couleurs des pixels qui l'entourent. Ainsi, pour estimer les valeurs manquantes dans chaque canal de couleur, nous déterminons la moyenne des valeurs des pixels voisins. En théorie, nous pouvons débiter par n'importe lequel des trois canaux (rouge, vert ou bleu). Par exemple, si nous commençons par le canal rouge, nous inspectons les pixels environnants pour repérer les valeurs manquantes, nous calculons leur moyenne et nous attribuons cette moyenne au pixel absent.

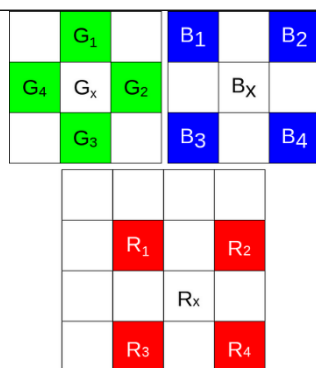


FIGURE IV-16: Algorithme d'interpolation bilinéaire [47]

D'après la figure IV-16 nous pouvons utiliser les équations 4.23, 4.24 et 4.25 pour interpoler les valeurs de G_x , B_x et R_x :

$$R_x = \frac{1}{4}(R_1 + R_2 + R_3 + R_4) \quad (4.23)$$

$$B_x = \frac{1}{4}(B_1 + B_2 + B_3 + B_4) \quad (4.24)$$

$$G_x = \frac{1}{4}(G_1 + G_2 + G_3 + G_4) \quad (4.25)$$

Pour l'interpolation linéaire de haute qualité (**High-Quality Linear Interpolation**), l'algorithme d'interpolation bilinéaire produit fréquemment des résultats sous-optimaux, notamment au niveau des bords. Les méthodes d'interpolation de haute qualité reconnaissent qu'il n'est pas toujours suffisant de se baser uniquement sur les pixels adjacents du même canal pour interpoler les pixels manquants dans chaque canal. Par exemple, pour interpoler un pixel vert, il est essentiel d'utiliser non seulement les valeurs des pixels verts voisins, mais aussi les valeurs du canal existant à l'emplacement de ce pixel. Cette approche, appelée interpolation par correction de gradient, intègre plusieurs valeurs de couleur pour obtenir des résultats plus précis. Il existe huit filtres 5x5 distincts dans le but d'améliorer le processus d'interpolation comme indiqué dans la figure IV-17.

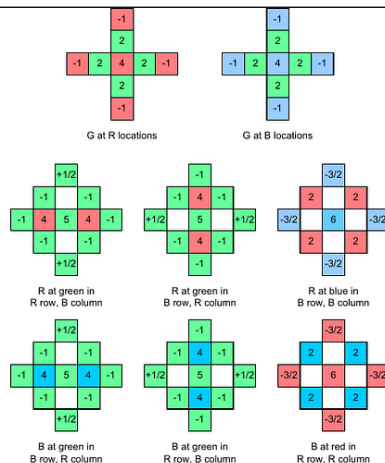


FIGURE IV-17: Coefficients des 8 filtres pour une interpolation linéaire de haute qualité [47]

Pour estimer la valeur d'un pixel en utilisant ces filtres, il est nécessaire de convoluer le pixel souhaité avec le filtre approprié. Par exemple, pour déterminer la valeur d'un pixel vert en présence d'une valeur bleue, il faut utiliser le deuxième filtre. Les valeurs des canaux concernés sont pondérées en fonction du filtre, additionnées, puis moyennées en divisant par 8 (la somme totale des poids des filtres est égale à 8).

Passons maintenant au but de cette méthode, à savoir comment l'utiliser comme outil de détection des manipulations.

Dé-mosaïquage pour la détection des falsifications : Cette méthode repose surtout sur l'évaluation du modèle d'interpolation CFA (Color Filter Array) de l'appareil photo numérique utilisé au départ. Pour identifier le modèle CFA d'une image, on se sert de plusieurs modèles CFA potentiels pour ré-interpoler l'image. Ensuite, on calcule l'erreur quadratique moyenne (MSE) entre l'image d'origine et chaque image ré-interpolée en utilisant ces modèles potentiels. Bien qu'il y ait 36 possibilités de configurations de filtres pour une cellule CFA 2×2 (ce sont les configurations couvrant toutes les permutations possibles des couleurs rouge, vert et bleu dans une cellule CFA 2×2), les appareils photo numériques se servent habituellement d'un seul des quatre modèles CFA standard de Bayer parmi ces possibilités (illustré dans la figure IV-18). Ainsi, les calculs de MSE ne sont réalisés qu'avec ces quatre modèles de Bayer comme candidats potentiels.

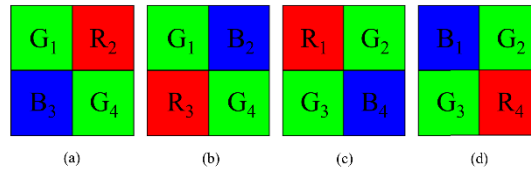


FIGURE IV-18: Bayer CFA patterns [48]

La technique de détection de manipulation repose sur une observation cruciale : l'erreur quadratique moyenne associée au modèle CFA réellement utilisé dans l'image devrait être considérablement inférieure à celles des trois autres modèles. Une valeur MSE plus faible indique une correspondance plus étroite avec le modèle CFA original.

Si aucune des valeurs MSE n'est significativement inférieure aux autres, cela suggère que l'image a peut-être été modifiée de manière à masquer le modèle CFA, comme le redimensionnement, la re-compression ou le filtrage. Ces opérations peuvent altérer suffisamment les valeurs de pixels pour que le modèle CFA original ne fournisse plus une correspondance significativement meilleure que les alternatives. De plus, des incohérences dans les modèles CFA entre différents sous-blocs de l'image peuvent indiquer une modification ou une retouche localisée.

Considérons l'intensité de l'image de la chaîne de couleur $L_c(x, y)$ au niveau de la position spatiale (x, y) avec $c \in R, V, B$. Soit $\Psi_{k,c}$ le jeu d'emplacements du filtre de couleur de la chaîne c pour un type particulier de motif CFA désigné par k , où $k = 1, 2, 3, 4$. Le masque de filtre de couleur $\theta_{k,c}(x, y)$ correspondant à $\Psi_{k,c}$ est défini comme suit :

$$\theta_{k,c}(x, y) = f(x) = \begin{cases} 1, & (x, y) \in \Psi_{k,c} \\ 0, & \text{sinon} \end{cases} \quad (4.26)$$

La technique utilise des blocs de taille $W \times W$, où $W = 8$ pixels, pour diviser l'image en prenant en compte uniquement les blocs non lisses. Les blocs non lisses sont les parties de l'image où il y a des changements significatifs dans les valeurs de pixels, indiquant des bords, des textures ou des détails. Ces zones sont moins uniformes et ont une variance et une complexité plus élevées dans leurs valeurs de pixels. Chaque bloc non lisse est désigné par B_i , où $i = 1, \dots, N$, N étant le nombre de blocs non lisses contenus dans l'image. Les blocs réinterpolés avec le filtre k sont désignés par $\hat{B}_{i,k}$. Ces blocs sont calculés en effectuant une convolution entre le noyau bilinéaire et le bloc B_i réaffiché avec le motif CFA k -ième défini comme suit :

$$\hat{B}_{i,k} = f(b_i, \theta_k) \quad k = 1, \dots, 4 \quad (4.27)$$

Ensuite, nous calculons l'erreur MSE entre les blocs de B et \hat{B} dans les régions non lisses sur l'ensemble de l'image à l'aide de l'équation suivante :

$$E_i(k, c) = \frac{1}{w \times w} \sum_{x=1}^w \sum_{y=1}^w \left(B_i(x, y, c) - \hat{B}_{i,k}(x, y, c) \right)^2 \quad (4.28)$$

Où E_i est une matrice 4×3 contenant les erreurs quadratiques moyennes pour chaque canal de couleur.

Pour détecter les distances relatives entre les canaux de couleur, une nouvelle matrice d'erreur $E_i^{(2)}$ est créée. La normalisation de toutes les lignes du tableau E_i est effectuée à l'aide de l'équation suivante :

$$E_i^{(2)}(k, c) = 100 \times \frac{E_i(k, c)}{\sum_{l=1}^3 E_i(k, l)}, c = 1, \dots, 3 \quad (4.29)$$

En raison du nombre réduit de pixels interpolés dans le canal vert, les valeurs minimales du MSE sont observées dans ce canal. Cela s'explique par le fait que dans les filtres CFA de Bayer, le canal vert est sur-échantillonné (il y a plus de pixels verts que de pixels rouges ou bleus), ce qui permet une meilleure interpolation et des valeurs MSE plus faibles. Nous nous concentrons sur le canal vert. Nous utilisons l'erreur normalisée pour ce canal, ce qui nous donne un vecteur $V_i(k)$ défini comme suit :

$$V_i(k) = 100 \times \frac{E_i^{(2)}(k, 2)}{\sum_{l=1}^4 E_i^{(2)}(l, 2)} \quad (4.30)$$

- $E_i^{(2)}(k, 2)$ Représente l'erreur normalisée pour le canal vert et le modèle CFA k .
- $\sum_{l=1}^4 E_i^{(2)}(l, 2)$ Est utilisée pour normaliser l'erreur de chaque modèle par rapport à la somme totale des erreurs des quatre modèles CFA pour le canal vert.

Cette formule calcule un pourcentage qui représente la proportion de l'erreur du modèle k par rapport à la somme des erreurs des quatre modèles pour le canal vert. Si le modèle k correspond bien au CFA utilisé pour capturer l'image, $V_i(k)$ aura une valeur relativement élevée par rapport aux autres modèles. En présence de dé-mosaïquage du filtre Bayer, il est attendu que l'une des erreurs d'estimation du modèle CFA dans le vecteur V_i soit significativement plus petite que les autres. Par conséquent, l'uniformité du vecteur V_i peut être utilisée comme indicateur d'une éventuelle opération de dé-mosaïquage. Ainsi, une nouvelle métrique pour estimer l'uniformité de V_i est définie comme suit :

$$U(i) = \sum_{l=1}^4 |V_i(l) - 25| \quad (4.31)$$

Pour une décision globale pour l'image donnée, toutes les valeurs de $U(i)$ sont calculées pour tous les blocs d'image non lisses. Enfin, la valeur médiane du vecteur U est calculé comme métrique CFA finale, $F1 = \text{médiane}(U)$. Plus la métrique $F1$ de la CFA est élevée, plus est probable l'interpolation de l'image avec un démosaïquage CFA. Par conséquent, il peut être déduit qu'aucun traitement ou altération significatif n'a eu

lieu. Le chiffre 25 provient de la normalisation du vecteur V_i . Pour chaque bloc non lisse, V_i est un vecteur de quatre valeurs (une pour chaque modèle CFA possible), et chaque valeur dans V_i est exprimée en pourcentage. La somme des quatre valeurs de V_i est toujours égale à 100, car $V_i(k)$ est défini comme suit :

$$V_i(k) = 100 \times \frac{E_i^{(2)}(k,2)}{\sum_{l=1}^4 E_i^{(2)}(l,2)} \quad (4.32)$$

Puisque la somme des $V_i(k)$ pour k de 1 à 4 est toujours 100, si les quatre valeurs étaient égales, chacune serait 25 :

$$V_i(1) + V_i(2) + V_i(3) + V_i(4) = 100 \quad (4.33)$$

$$\text{Si : } V_i(1) = V_i(2) = V_i(3) = V_i(4) \quad (4.34)$$

$$\text{Alors : } V_i(1) = V_i(2) = V_i(3) = V_i(4) = 25 \quad (4.35)$$

Le chiffre 25 représente donc une distribution uniforme des valeurs normalisées. Si le vecteur V_i était parfaitement uniforme, chaque $V_i(k)$ serait de 25. L'équation de $U(i)$ permet de mesurer à quel point les valeurs de V_i s'écartent de cette distribution uniforme idéale. Une déviation importante indique une non-uniformité, ce qui peut être un indicateur d'altérations. La figure IV-19 présente l'application de l'algorithme de démosaïquage sur une image. À gauche, avec un oiseau ajouté. À droite, l'oiseau est nettement visible et se détache du fond. Cette différence de couleur souligne l'oiseau comme un élément potentiellement suspect, qui suppose une possible manipulation.

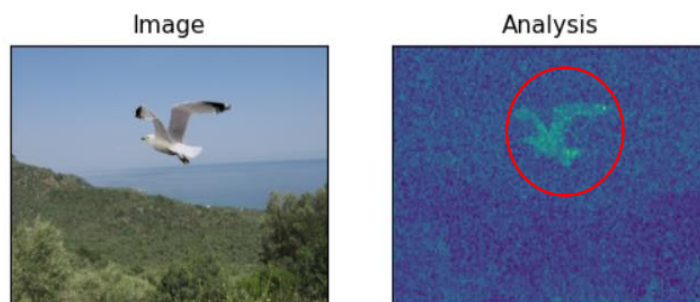


FIGURE IV-19: Exemple de la détection de falsification à l'aide de du dé-mosaïquage.

IV.4 Conclusion

Dans cette partie, nous avons présenté les méthodes statistiques de détection de falsifications d'images, en nous concentrant sur la détection basée sur les artefacts et la détection basée sur les anomalies de bruit. Nous avons exploré en détail les techniques telles que l'analyse de la compression, le dé-mosaïquage, le filtrage médian et l'analyse des incohérences de bruit. Ces méthodes ont montré leur efficacité pour détecter certaines falsifications d'images, en particulier celles qui laissent des traces visibles dans l'image.

Cependant, nous avons également constaté que ces méthodes présentent certaines limites. Par exemple, l'analyse de la compression peut ne pas être efficace pour détecter les falsifications dans les images qui n'ont pas été compressées ou qui ont été compressées à l'aide de méthodes sans perte. De même, l'analyse des incohérences de bruit peut ne pas être en mesure de détecter les manipulations subtiles qui n'affectent pas la distribution du bruit dans l'image. Face à ces limites, il est nécessaire d'explorer des méthodes plus avancées pour la détection de falsifications d'images.

Chapitre V.
Méthodes basées sur
le Deep Learning

V.1 Introduction

Les méthodes statistiques abordées dans le chapitre précédent ont prouvé leur efficacité, mais présentent des limites face à la complexité croissante des techniques de manipulation d'images. C'est pourquoi les approches basées sur l'apprentissage profond (Deep Learning) ont gagné en popularité ces dernières années. Leur capacité à extraire automatiquement des caractéristiques discriminantes des données et à modéliser des relations complexes en font des outils puissants pour la détection de falsifications.

Dans ce chapitre, nous explorerons différentes approches du Deep Learning appliquées à la tâche de détection de falsifications d'images, nous allons mettre l'accent sur **l'architecture U-Net pour la segmentation d'images**, ainsi que sur diverses techniques **d'augmentation de données** telles que les transformations géométriques et les ajouts de bruit. Nous avons également exploré la segmentation d'images en grilles pour capturer des détails fins et localiser précisément les manipulations. En outre, nous avons utilisé des méthodes telles que le **Transfer Learning et l'ensemble Learning** pour améliorer les performances des modèles pour la tâche de **classification**.

Dans ce qui suit, nous retracerons le déroulement complet de notre projet selon les différentes phases du processus.

V.2 Compréhension du problème (Business Understanding)

Dans cette phase de business understanding de notre projet nous visons à comprendre les enjeux de la détection des falsifications dans les images et les vidéos, les défis actuels, et les solutions potentielles et à identifier la meilleure approche technique. En analysant le problème à résoudre, nous avons identifié plusieurs approches potentielles dans le domaine du computer vision : la classification, la détection d'objets et la segmentation. Après une évaluation approfondie, nous avons opté pour la **segmentation** comme solution la plus adaptée à notre cas d'usage.

V.2.1 Pourquoi la segmentation

La segmentation d'images s'est avérée être la méthode la plus efficace pour détecter les falsifications dans les images et les vidéos, dépassant les méthodes de classification et de détection. Cette approche offre une précision spatiale au niveau du pixel, ce qui permet de localiser et de visualiser les zones modifiées grâce à un masque. De plus, la segmentation est plus polyvalente car elle est capable de détecter divers types de manipulations. Elle fournit également une **granularité** d'information supérieure, ce qui facilite la quantification des modifications et réduit les faux positifs. En outre, la

segmentation s'adapte à différentes résolutions et constitue une base solide pour des analyses forensiques approfondies.

V.3 Compréhension des données (Data Understanding)

Avant de plonger dans les techniques de modélisation avancées, il est primordial de bien comprendre les données avec lesquelles nous allons travailler. Cependant, dans notre projet, nous avons rencontré une contrainte majeure : **par souci de confidentialité, nous n'avons pas pu avoir accès aux données du CRD-GN**. C'est pourquoi l'étape de « Compréhension des données » est d'autant plus importante. Elle nous permet d'explorer en profondeur les caractéristiques, la qualité et les éventuels problèmes liés aux données dont nous disposons.

Pour pallier cette limitation, nous avons **consacré beaucoup de temps et d'énergie à la recherche de données représentatives de notre problème**. Cette recherche approfondie nous a amenés à **combiner** plusieurs ensembles de données pour former un dataset global, qui servira de base à notre entraînement. Nous avons utilisé différents ensembles de données, notamment IMD 20, REALISTIC TAMPERING, CASIA-2, DEFACTO, CoMoFoD et Benchmark Data du Labo Pattern Recognition.

En prenant le temps de bien comprendre nos données dès le départ, nous pouvons anticiper les obstacles et planifier les actions nécessaires pour les préparer au mieux pour la modélisation, tout en respectant les contraintes de confidentialité.

Dans cette section, nous allons vous présenter les différentes bases de données ainsi que les différentes tâches que nous avons entreprises pour parvenir à une compréhension globale de notre jeu de données.

Datasets :

a. IMD 20 (Real-Life Manipulated Images):

L'ensemble de données IMD 20 [49] contient 2 010 images manipulées dans la vie réelle, téléchargées depuis Internet. Les versions réelles correspondantes de ces images sont également fournies. De plus, il existe un masque binaire manuellement créé qui localise la zone manipulée de chaque image manipulée. Ce masque permet d'identifier précisément les zones altérées de l'image, ce qui est utile pour entraîner notre modèle à détecter les modifications apportées aux images. Les images de l'ensemble de données IMD 20 sont variées et représentent différents types de manipulations.

b. REALISTIC TAMPERING :

Cet ensemble de données contient 220 images manipulées créées manuellement à l'aide de logiciels modernes de retouche photo tels que GIMP et Affinity Photo. Cet ensemble de données couvre divers scénarios de manipulation difficiles, impliquant à la fois l'insertion et la suppression d'objets. Les images ont été capturées à l'aide de quatre appareils photo différents, notamment Sony alpha57, Canon 60D, Nikon D7000 et Nikon D90. Toutes les images sont des bitmaps RVB uint8 de 1920 x 1080 px stockées au format TIFF. Les images ont été recadrées au milieu de bitmaps TIFF plein cadre obtenus en convertissant les fichiers bruts de l'appareil photo en TIFF à l'aide de ddraw avec les paramètres par défaut pour Sony A57, en utilisant le logiciel Canon d'origine pour Canon 60D et en acquérant directement les images TIFF à partir de l'ensemble de données RAISE pour Nikon D90 et D7000. L'ensemble de données contient des images trafiquées, des images originales, des cartes de référence à 3 niveaux et des signatures PRNU obtenues avec l'estimateur MLE standard du laboratoire DDE. Les cartes de référence contiennent trois étiquettes : zones authentiques, modifications sémantiques importantes du contenu et dommages collatéraux, qui correspondent aux pixels modifiés par rapport à l'image d'origine mais représentant encore à peu près le même contenu. Les cartes de référence ont été obtenues par raffinement manuel de cartes de différence automatiquement générées, ce qui peut rendre la distinction entre les zones trafiquées et les dommages collatéraux quelque peu approximative. Pour l'évaluation standard, les zones grises et blanches peuvent être considérées comme trafiquées. Dans les cas d'insertion d'objet, les ombres et les reflets sont traités comme des dommages collatéraux, malgré une déviation potentiellement importante par rapport aux pixels d'origine. [50]

c. CASIA-2.0 :

Le CASIA 2.0 [51] est un ensemble de données publiques de grande envergure, comprenant des images manipulées et authentiques, spécialement conçu pour l'entraînement et l'évaluation de modèles d'apprentissage profond pour la détection de falsifications d'images. Cet ensemble de données contient un total de 12,613 images RGB, réparties en deux catégories principales : les images authentiques (Au) et les images falsifiées (Tp).

Les images authentiques sont au nombre de 7 491 et n'ont subi aucune manipulation. Les images falsifiées, quant à elles, comptent 5 123 images dans lesquelles des objets, du texte ou des personnes ont été ajoutés ou supprimés à l'aide de techniques de manipulation sophistiquées. Les images falsifiées couvrent sept types de manipulations différents, à savoir l'ajout d'objet, la suppression d'objet, l'ajout de texte, l'ajout de personne, la suppression de personne, le remplacement d'arrière-plan et le mélange de différentes images.

Les images de l'ensemble de données CASIA 2.0 sont de résolutions variées, allant de 240x160 à 900x600 pixels, et proviennent de sources diverses telles que des caméras, des Smartphones et internet. De plus, l'ensemble de données fournit des masques de segmentation binaires pour les régions manipulées dans les images T_p , permettant ainsi une évaluation précise des performances de détection au niveau des pixels. Le CASIA 2.0 est l'un des ensembles de données de référence les plus utilisés dans la recherche sur la détection de falsifications d'images, apprécié pour sa grande taille, sa diversité et son réalisme.

d. CoMoFoD:

Cette base de données est une collection d'images manipulées qui comprend 260 ensembles d'images, répartis en deux catégories : petites images (512x512) et grandes images (3000x2000). Dans chaque catégorie, plusieurs transformations ont été appliquées aux images, notamment la translation, la rotation, le redimensionnement, la distorsion et une combinaison de plusieurs transformations. [52]

Chaque ensemble d'images contient quatre éléments : l'image originale, un masque de couleur, un masque binaire et l'image falsifiée. Le masque de couleur indique les régions originales et falsifiées de l'image, avec le noir représentant l'arrière-plan et les couleurs représentant les régions falsifiées. Le masque binaire est utilisé pour l'évaluation de la détection, avec le noir représentant l'arrière-plan et le blanc représentant les régions falsifiées. L'image falsifiée est l'image originale avec une ou plusieurs régions copiées et collées à un nouvel emplacement.

En plus des transformations appliquées aux images, des méthodes de post-traitement ont été appliquées à toutes les images originales et falsifiées de la base de données CoMoFoD. Ces méthodes comprennent la compression JPEG, l'ajout de bruit, le flou d'image, la modification de la luminosité, la réduction de la couleur et les ajustements de contraste. Ces méthodes de post-traitement ont été appliquées pour rendre les images plus réalistes et pour simuler les manipulations d'images réelles.

e. DEFACTO :

Il contient environ 19 000 falsifications par copier-coller, disponibles dans le répertoire `copymove_img`. Chaque falsification est accompagnée de deux masques binaires. L'un, situé dans le sous-répertoire `probe_mask`, indique l'emplacement de la falsification, et l'autre, dans le sous-répertoire `donor_mask`, indique l'emplacement de la source dans l'image. Ce jeu de données permet non seulement de qualifier les méthodes automatiques d'authentification et de détection de manipulations, mais aussi de les

entraîner. Il constitue donc une base solide pour le développement de méthodes de détection de falsifications d'images. [53]

f. Benchmark Data du Labo Pattern Recognition :

Il s'agit d'une collection d'images utilisée pour évaluer les performances des algorithmes de détection de manipulation d'images. Elle contient 96 images à différents stades, allant de l'original à la version manipulée, ainsi que les parties altérées correspondantes. Quatre appareils différents ont été utilisés pour capturer ces images.

Le tableau V-1 résume les caractéristiques des différents datasets utilisés pour l'évaluation des algorithmes de détection de manipulation d'images.

TABLEAU V-1: Récapitulatif des données utilisées

Dataset name	mask	Image Format	Post-processing	Forgery types	Real/Forged Images
IMD 20	Oui	JPG		Splicing, copy-move, removal	/2010
REALISTIC TAMPERING	Oui	TIFF		object insertion, removal.	
CASIA-2.0	Oui	TIFF, JPEG, BMP	Oui	Splicing, copy-move, removal	7491/5123
CoMoFoD	Oui	JPEG, PNG	Oui	Copy-move	260/260
DEFACTO	Oui	TIFF		Copy-move	/18194
Benchmark Data	Oui	JPEG, PNG		Splicing, copy-move, removal	64/320



FIGURE V-1: Exemples d'images extraites des datasets

En explorant en profondeur notre ensemble de données (**EDA : Exploratory Data Analysis**), nous avons rapidement constaté une grande diversité parmi les images présentes, issues de multiples sources différentes. Des variations notables sont apparues en termes de résolution, de dimensions, de formats, de qualité d'image et même de contenu représenté. De telles différences risquent d'impacter significativement les performances des algorithmes de détection des zones manipulées, en introduisant des biais indésirables et en complexifiant la tâche d'apprentissage.

Il devenait donc essentiel de préparer minutieusement notre dataset afin d'homogénéiser au maximum l'ensemble des images. Pour ce faire, nous avons entrepris une série d'étapes de prétraitement et de normalisation visant à ramener toutes les images à une échelle commune, tout en réduisant les variations superflues liées aux conditions d'acquisition. Notre objectif est d'obtenir un dataset parfaitement préparé, garant de résultats fiables et précis pour notre système de détection de manipulations.

V.4 Préparation des données (Data Preparation)

V.4.1 Organisation des données

Nous avons mis en place plusieurs étapes clés permettant d'harmoniser et de préparer efficacement notre dataset.

(a) Normalisation du nommage des fichiers

La première étape a consisté à normaliser le nommage de l'ensemble des fichiers d'images et de masques afin d'instaurer une convention cohérente. À l'aide d'expressions régulières (Regex : Regular Expressions), nous avons renommé systématiquement chaque image ajoutant le suffixe « `_image_image` » et chaque masque avec le suffixe « `_image_mask` ». Par exemple, « `img001.jpg` » est devenu « `img001_image_image.png` » et son masque associé « `img001_image_mask.png` ».

(b) Conversion au format PNG

Nous avons ensuite converti tous les fichiers d'images au format **PNG**. Ce choix repose sur plusieurs avantages du PNG : il s'agit d'un format sans perte permettant de conserver l'intégrité des données d'origine, tout en offrant une compression efficace et en supportant la transparence (canaux alpha), utile pour certaines tâches.

(c) Tri dans les dossiers « fake » et « pristine »

Une étape cruciale a été l'organisation physique des fichiers en deux dossiers distincts : « fake » regroupant toutes les images manipulées ainsi que leurs masques correspondants, et « pristine » contenant les images authentiques non manipulées. Cette séparation claire facilite grandement la gestion des deux catégories d'images lors des phases d'entraînement et d'évaluation de nos modèles.

(d) Génération de masques pour les images pristines

Notons que pour les images pristines, le dataset initial ne fournissait pas de masques associés. Afin de conserver une structure homogène, nous avons généré des masques vides (constitués uniquement de pixels noirs illustré dans la figure V-2) de dimensions **512 x 512**, indépendamment des dimensions des images authentiques correspondantes. Cette étape permet d'assurer une cohérence dans le traitement des données, étant donné que toutes les images seront redimensionnées par la suite.

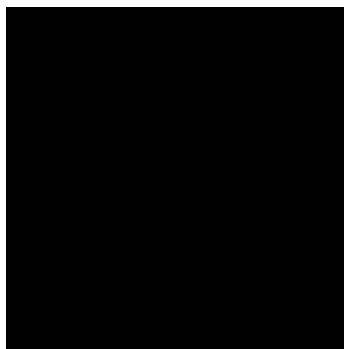


FIGURE V-2: Mask généré

(e) Binarisation des masques

Enfin, nous avons adopté la convention de travailler avec des masques binaires en noir et blanc, où les pixels blancs (valeur 1) représentent les zones manipulées, tandis que les pixels noirs (valeur 0) correspondent aux zones inchangées. Cette binarisation simplifie grandement le traitement et l'analyse des images par nos algorithmes.

Grâce à cette organisation rigoureuse des données (comme montré dans la figure V-3), nous disposons désormais d'un dataset structuré, normalisé et prêt pour les étapes suivantes de prétraitement et de modélisation. Une base solide est ainsi posée pour garantir la qualité et la fiabilité de nos résultats.

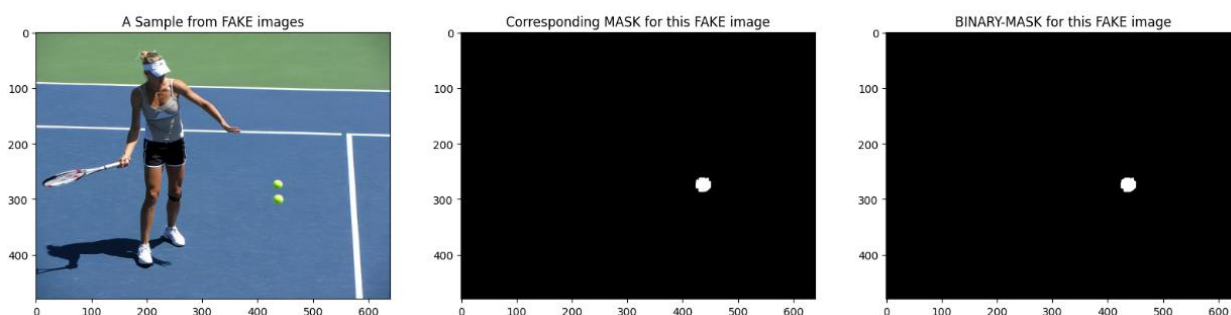


FIGURE V-3: Mask binaire généré

V.4.2 Analyse approfondie des caractéristiques

Après avoir organisé rigoureusement notre dataset, nous avons entrepris une analyse approfondie des caractéristiques intrinsèques des images et de leurs masques associés. Cette étape visait à mieux cerner la diversité et la distribution de certains attributs clés, comme les dimensions, les nombres de canaux, etc.

(a) Un rapide rappel

Avant d'aller plus loin, citons quelques caractéristiques d'images importantes qu'on a analysées :

- **Hauteur (Height)** : Nombre de pixels le long de l'axe vertical de l'image.
- **Largeur (Width)** : Nombre de pixels le long de l'axe horizontal de l'image.
- **Canaux (Channels)** : Nombre de composantes de couleur formant l'image. Par exemple, une image en niveaux de gris possède 1 canal, tandis qu'une image couleur RGB en possède 3 (rouge, vert, bleu).

(b) Extraction et visualisations

Dans un premier temps, nous avons extrait et stocké ces différentes caractéristiques dans des dataframes dédiés. Pour chaque image, nous avons ainsi répertorié sa hauteur, sa largeur, son nombre de canaux, mais aussi les mêmes attributs pour son masque correspondant. Cette structuration des données a grandement facilité leur exploration et leur visualisation.

Nous avons généré plusieurs représentations graphiques. Des histogrammes ont permis de visualiser les ratios d'images manipulées « fake » et authentiques « pristine ». Des diagrammes en bâtons ont révélé la répartition par nombre de canaux. Mais nous sommes allés plus loin en traçant les densités de probabilité (PDF) des dimensions, séparément pour les fakes et pristines. Ces visualisations informatives nous ont non seulement aidés à mieux comprendre nos données, mais aussi à guider nos efforts de normalisation pour mettre toutes les images à la même échelle.

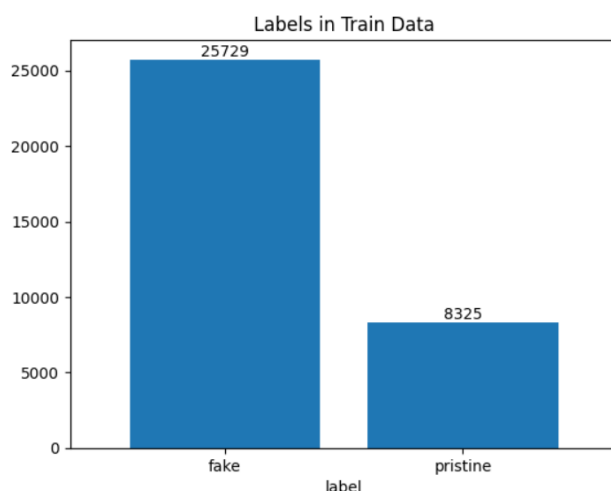


FIGURE V-4: Diagramme en bâtons pour le nombre d'images par classe

Le diagramme en bâtons de la figure V-4 montre la répartition des images entre les deux classes dans les données d'entraînement. 25,729 images fake et 8,325 pristine. Cela indique un déséquilibre notable, avec une proportion beaucoup plus élevée d'images « fake ». Nous avons délibérément laissé ce déséquilibre dans les classes afin d'avoir plus de données de type « fake », car la classification se fait au niveau des pixels.

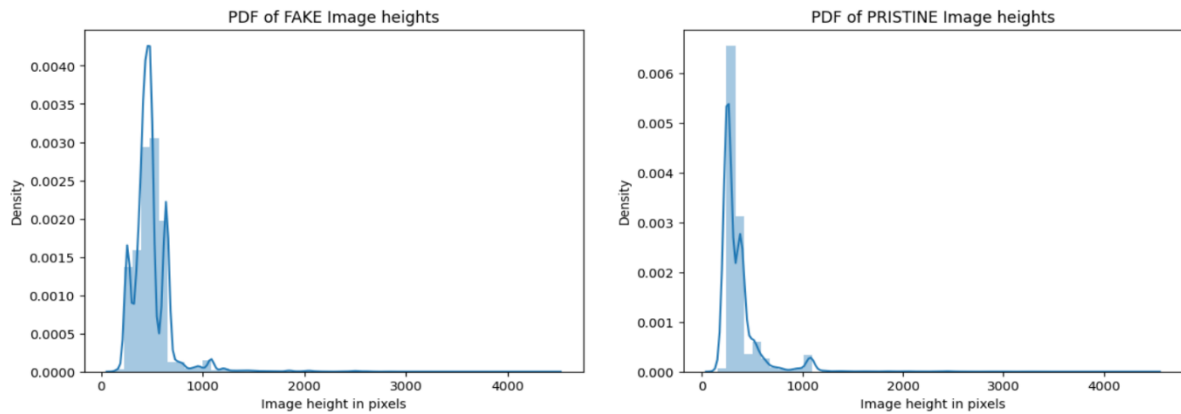


FIGURE V-5: PDF pour la hauteur des images

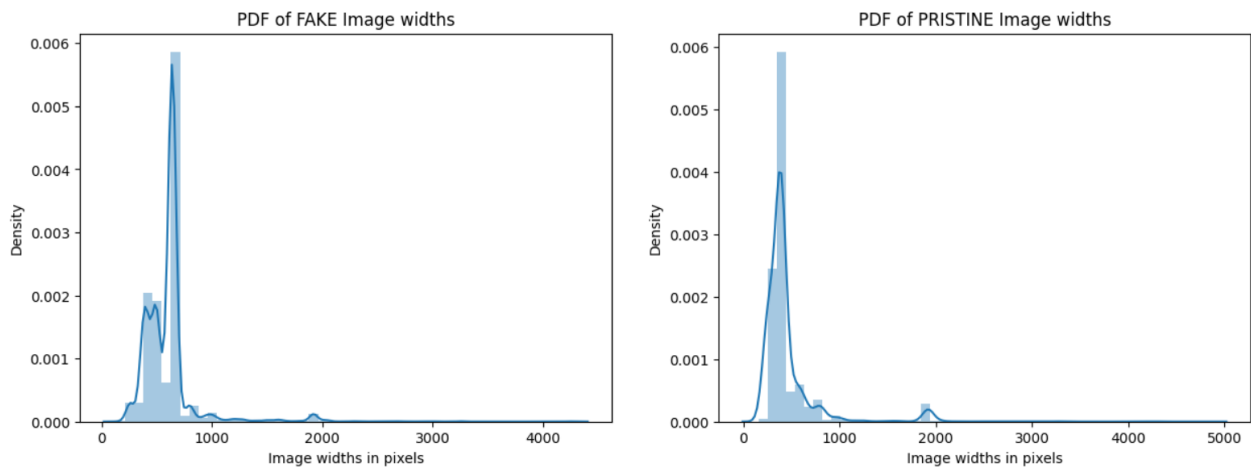


FIGURE V-6: PDF pour la largeur des images

Les graphiques V-5 et V-6 montrent la distribution des hauteurs et largeurs des images pour les ensembles « Fake » et « Pristine », où :

- La majorité des images, ont une hauteur entre 0 et 1000 pixels.
- Quelques images dépassent 1000 pixels, mais elles sont rares.

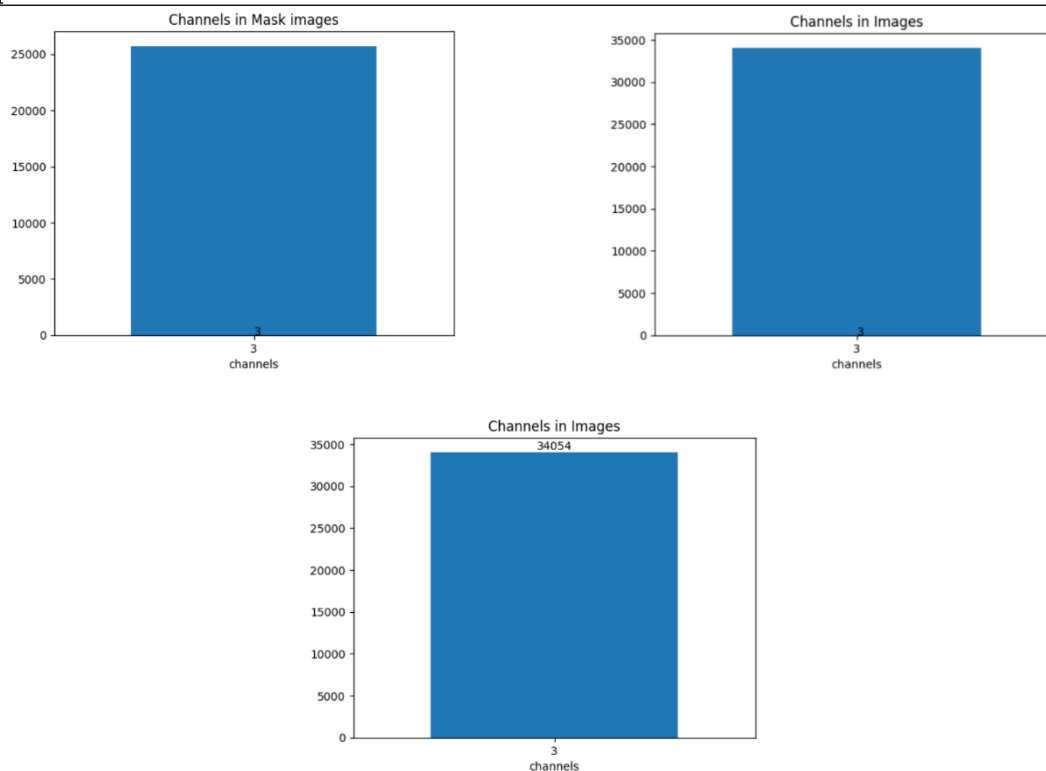


FIGURE V-7: Nombre de canaux pour les images et les masques

Les diagrammes en bâtons de la figure V-7 montrent que toutes les images et les masques ont 3 canaux.

- **Images (fake et pristine)** : 34,054 images avec 3 canaux.
- **Masques** : 25,729 masques avec 3 canaux.

Cela assure une uniformité dans les données.

Cette analyse détaillée des caractéristiques a été riche en enseignements. Elle nous a permis de mieux comprendre la structure sous-jacente de nos données, d'identifier d'éventuels problèmes ou biais à corriger, et surtout de mieux préparer le terrain pour les étapes ultérieures de prétraitement et de normalisation. Les visualisations réalisées nous ont guidés dans le processus de mise à l'échelle des images et des masques, essentiel pour assurer une qualité optimale de nos données avant modélisation.

Fort de ces connaissances, nous avons pu passer à l'étape suivante : la préparation minutieuse de nos données pour l'entraînement du modèle.

V.4.3 Préparation et traitement des données

Cette phase va mettre toutes nos données dans la même échelle en passant par plusieurs étapes :

(a) Nettoyage des données redondantes et manquantes

Lors de l'inspection initiale des données, nous avons identifié des cas de **redondance et d'incohérence**. Par exemple, il y avait des instances où une image n'avait pas de masque correspondant ou inversement. Dans d'autres cas, nous avons trouvé des données en double, où la même image ou le même masque apparaissait plus d'une fois dans le dataset. Pour assurer la cohérence et l'intégrité de notre dataset, nous avons décidé de supprimer ces données redondantes et incohérentes. Plus précisément, nous avons supprimé toutes les images manquantes ou les masques manquants, ainsi que toutes les données en double. Cela nous a permis de garantir que notre dataset ne contenait que des données uniques et cohérentes, prêtes pour les étapes ultérieures de prétraitement et de modélisation.

(b) Conversion en masks binaires

Pour convertir les masques en format binaire, nous avons d'abord appliqué un léger flou gaussien aux masques d'origine à trois canaux, afin de réduire le bruit éventuel et d'adoucir les contours des régions d'intérêt. Bien que cette étape de lissage soit facultative, elle aide à obtenir des masques binaires de meilleure qualité. Ensuite, nous avons converti les valeurs de pixels en 0 (noir) ou 255 (blanc), c-à-d une binarisation où les pixels blancs représentent les zones d'intérêt (potentiellement manipulées) et les pixels noirs représentent l'arrière-plan, ce qui a permis d'obtenir des **masques binaires à un canal unique**, facilitant ainsi leur traitement ultérieur.

(c) Normalisation des données

Afin de préparer nos données pour la modélisation, nous avons effectué une normalisation des images. Cette étape consistait à ramener les valeurs des pixels dans la plage $[0, 1]$ en divisant les images par 255. Cette normalisation nous a permis d'éviter les problèmes numériques lors de l'entraînement de nos modèles et à faciliter la convergence des algorithmes d'optimisation.

(d) Répartition des Données en Entraînement, Test et Validation

Après la préparation des données, nous les avons organisées sous forme de deux colonnes : une pour les chemins vers les images et une autre pour les chemins vers les masques correspondants. Cette structure simplifie non seulement le chargement et le traitement des données lors de l'entraînement, mais réduit également l'utilisation de la

mémoire en évitant de charger toutes les données en une seule fois. Cela permet une utilisation plus efficace de nos ressources informatiques.

Nous avons également créé des ensembles similaires pour les données de test et de validation. Pour structurer les données, nous avons construit un DataFrame avec deux colonnes :

- **Image_path** : les chemins vers les images.
- **Masl_path** : les chemins vers les masques.

(e) **Techniques de prétraitement des images : Augmentation et segmentation en grilles**

Dans le cadre de notre étude, nous avons exploré diverses approches pour enrichir jeu de données et d'affiner notre analyse. Parmi les techniques essentielles que nous avons mises en œuvre, deux se sont révélées particulièrement pertinentes : l'augmentation des données et la segmentation des images en grilles. Examinons en détail ces techniques et leur application dans notre recherche.

- **Augmentation d'images :**

Pour l'augmentation d'images, nous avons opté à l'utilisation de la bibliothèque **Albumentations** [54] qui est une bibliothèque d'augmentation d'images à la fois rapide et flexible. Elle est très prisée pour les projets de vision par ordinateur parce qu'elle permet d'appliquer une multitude de transformations de manière efficace. Nous avons choisi d'utiliser Albumentations pour sa performance et sa flexibilité. Nous avons pu appliquer diverses transformations à nos images (Padding, Flipping, Transformation Élastique, etc. détaillée dans l'annexe). La figure V-8 représente les résultats de l'augmentation de nos données avec le redimensionnement et le retournement horizontal.

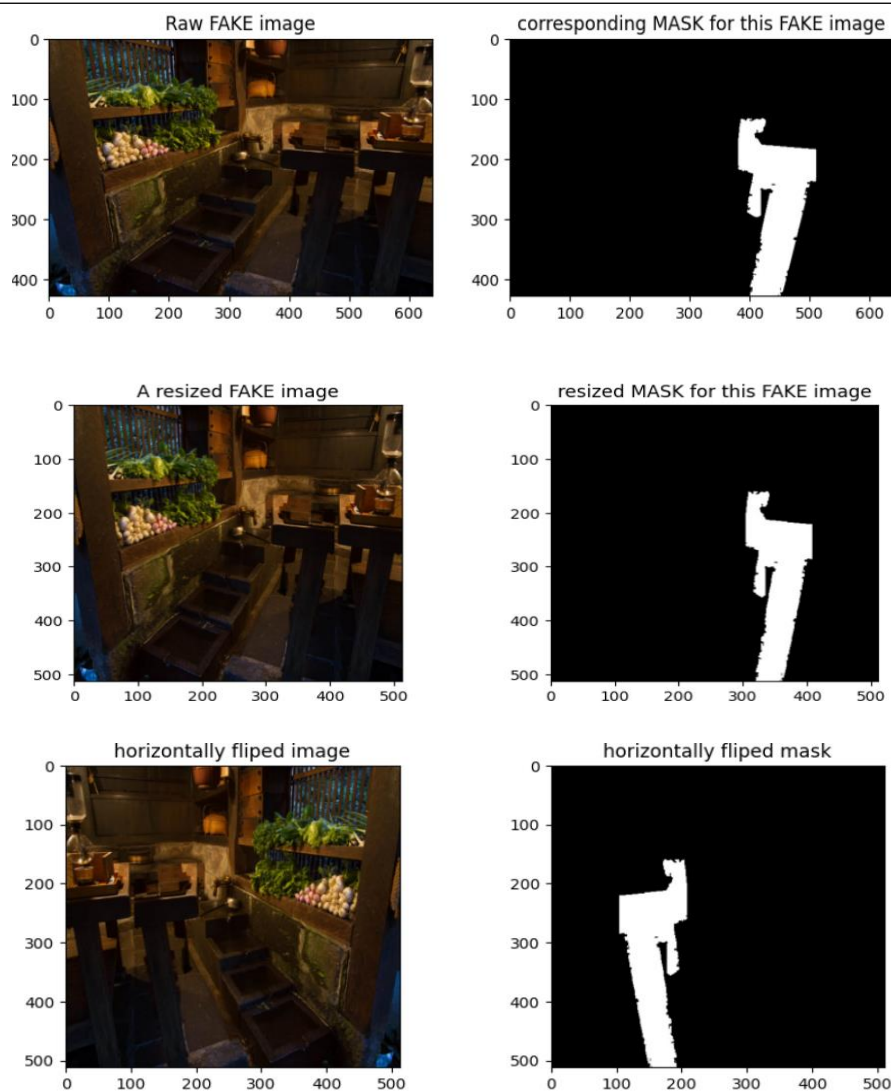


FIGURE V-8: Quelques exemples de l'augmentation des images.

- **Segmentation d'images en grilles**

Pour examiner les images de manière plus détaillée, nous avons adopté une technique appelée segmentation en grilles. Cette méthode consiste à diviser les images en petits segments, ce qui permet une analyse plus précise et repose sur deux paramètres principaux :

- **Taille de la grille** : Nous avons utilisé des segments de 128x128 pixels. Cette taille permet de capturer suffisamment de détails tout en restant gérable en termes de mémoire et de calcul.
- **Stride** : Nous avons utilisé un pas de 32 pixels, permettant un chevauchement significatif qui améliore la robustesse des caractéristiques extraites.

Cette approche nous a permis de :

- **Analyser des images de grande taille** : En les décomposant en segments plus petits, nous évitons les limitations de mémoire et de calcul.
- **Capturer des détails locaux** : Les petits segments permettent de détecter des caractéristiques fines qui pourraient être perdues dans une analyse globale.
- **Augmenter effectivement notre jeu de données** : En créant plusieurs échantillons à partir d'une seule image, nous augmentons la quantité de données d'entraînement sans nécessiter de nouvelles images.

Nous avons mis en place une méthode spécifique pour segmenter les images manipulées afin d'assurer que chaque segment contienne au moins 25% de pixels manipulés :

- **Segmentation de l'image** : Chaque image manipulée est découpée en patches de 128x128 pixels avec une taille de noyau de 128 pixels.
- **Déplacement du noyau** : Le noyau est déplacé de 32 pixels à chaque itération pour couvrir toute l'image.
- **Filtrage des patches** : Chaque patch est sauvegardé si au moins 25% de ses pixels sont blancs (manipulés).

Cette approche a optimisé l'équilibre entre la précision de l'analyse et la charge de calcul, garantissant ainsi que le modèle dispose de suffisamment d'informations pour une classification précise tout en maintenant des performances de calcul acceptables.

La figure V-9 montre quelques grilles (patches) comme résultat de la segmentation d'une image en segments

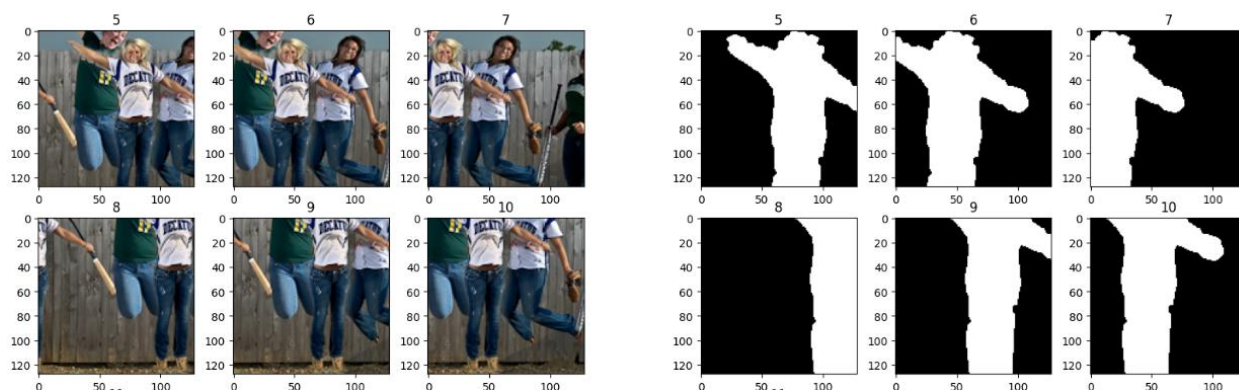


FIGURE V-9: Quelques exemples de la segmentation d'une image

(f) Générateurs de Données : L'étape finale de préparation

Avant d'entamer la phase d'entraînement, nous avons mis en place des générateurs de données, une étape cruciale pour optimiser notre processus. Ces générateurs agissent

comme des assistants efficaces, préparant les données, juste à temps pour nos modèles en appliquant le même traitement sur chaque lot de données. Leur fonctionnement peut être résumé en quatre étapes clés :

- **Chargement** : Ils récupèrent les images et les masques à partir des chemins spécifiés.
- **Conversion** : Transformation des données en tableaux NumPy, un format adapté à nos modèles.
- **Normalisation** : Application du processus de normalisation décrit précédemment.
- **Création de lots** : Assemblage des données en paquets prêts à l'emploi pour l'entraînement.

Cette approche présente plusieurs avantages notables. Elle permet **une gestion efficace de la mémoire**, comparable à un approvisionnement « juste à temps » plutôt qu'à un stockage massif. De plus, elle **accélère considérablement le processus d'entraînement**, évitant les goulets d'étranglement liés au chargement des données. Grâce à cette préparation minutieuse, nous avons pu optimiser l'utilisation de nos ressources informatiques, un aspect crucial pour les phases d'entraînement intensives qui allaient suivre.

V.5 La modélisation (modeling)

Cette étape cruciale consiste à concevoir, entraîner et affiner des modèles capables d'interpréter nos images et d'en extraire des informations pertinentes.

Pour notre tâche de segmentation d'images, nous avons opté pour l'architecture UNET, déjà détaillée dans II.4.3 Notre approche s'est déclinée en plusieurs variantes, chacune explorant une facette différente de l'optimisation du modèle :

- UNET avec augmentation de données
- UNET entraîné uniquement sur des données synthétiques (fake data)
- UNET avec données augmentées et segmentées en grilles

Dans cette section, nous examinerons en détail chacune de ces approches et leur mise en œuvre. Nous explorerons les choix de paramètres, et les stratégies d'optimisation employées pour maximiser les performances.

V.5.1 Approches Spécifiques

Le tableau V-2 présente une comparaison détaillée des trois approches utilisées pour la segmentation.

TABLEAU V-2: Comparaison détaillée des trois approches utilisées pour la segmentation.

Approche	Motivation	Modifications apportées à l'architecture	Processus d'entraînement	Défis rencontrés et solutions	Inconvénients
U-Net avec Augmentation de Données Classique	Multiplier les exemples d'entraînement en appliquant des transformations aléatoires aux images originales, améliorant la robustesse et la généralisation du modèle en simulant diverses conditions de prise de vue.	Aucune modification n'a été apportée à l'architecture de base du U-Net.	Les images augmentées ont été générées en temps réel à l'aide de data generators d'Albumentations, appliquant des transformations comme la rotation, les retournements, les changements de contraste et de luminosité, et l'ajout de bruit.	Maintenir un bon équilibre entre la diversité des données et le surapprentissage. Nous avons ajusté les paramètres d'augmentation pour obtenir un jeu de données diversifié tout en préservant les caractéristiques essentielles des images.	Les images augmentées peuvent ne pas refléter toutes les variations possibles des données réelles, limitant la capacité du modèle à généraliser efficacement.
U-Net Entraîné Uniquement sur des Données Synthétiques (fake).	Générer un grand volume d'images annotées, particulièrement utile lorsque les données annotées réelles sont rares ou difficiles à obtenir.	Aucune modification n'a été apportée à l'architecture de base du U-Net.	Les images synthétiques ont été générées en utilisant des algorithmes spécifiques pour simuler des conditions réelles.	Assurer que les images synthétiques soient suffisamment réalistes pour que le modèle puisse généraliser aux données réelles. Nous avons itéré sur plusieurs versions de nos algorithmes de génération de données synthétiques pour atteindre un niveau de réalisme satisfaisant.	Les images synthétiques peuvent ne pas capturer toutes les complexités des données réelles, réduisant ainsi l'efficacité du modèle sur des images réelles non vues.
U-Net avec Données	La segmentation en grilles permet	L'architecture de base du U-Net n'a	Les data generators ont été configurés	La gestion de la redondance due	Le traitement de segments

Augmentées et Segmentées en Grilles	d'analyser des images à une échelle plus fine, capturant ainsi des détails locaux importants qui pourraient être perdus sur des images de grande taille.	pas été modifiée, mais les images ont été segmentées en grilles de 128x128 pixels avec un stride de 32 pixels.	pour générer des segments d'images et leurs masques correspondants, en appliquant des transformations d'augmentation en temps réel.	aux chevauchements des segments. Nous avons utilisé des techniques de suppression de doublons et de fusion des prédictions pour obtenir des résultats cohérents.	d'images peut entraîner une augmentation de la redondance et des coûts de calcul, nécessitant des techniques supplémentaires pour la gestion des chevauchements et la fusion des résultats.
--	--	--	---	--	---

V.5.2 Paramètres de modélisation

Nous avons maintenu une architecture U-Net constante à travers toutes nos approches expérimentales. Afin d'assurer une base de comparaison cohérente, nous avons fixé certains paramètres clés pour l'ensemble de nos expériences :

- **Taille de lot (batch size²) de 8** : La taille de lot détermine le nombre d'exemples d'entraînement utilisés dans une itération avant la mise à jour des poids du modèle. Une taille de lot de 8 permet de trouver un bon équilibre entre stabilité de l'entraînement et utilisation de la mémoire GPU.

- **16 étapes par époque (steps per epoch³)** : Les étapes par époque correspondent au nombre d'itérations que le modèle effectue à chaque époque. En utilisant 16 étapes par époque, nous avons assuré que chaque époque comprenne un nombre suffisant de mises à jour des poids pour une convergence stable.

- **Taux de dropout de 0,05** : Le dropout est une technique de régularisation utilisée pour prévenir le surapprentissage en désactivant de manière aléatoire une fraction des neurones pendant l'entraînement [55]. Un taux de 0,05 signifie que 5% des neurones sont désactivés à chaque itération, ce qui aide à généraliser le modèle sans trop perturber

² Le batch size représente le nombre d'échantillons d'entraînement utilisés dans une seule itération pour mettre à jour les poids d'un modèle.

³ Représente le nombre d'itérations nécessaires pour parcourir une fois l'ensemble complet des données d'entraînement lors de l'entraînement d'un modèle

l'apprentissage. La figure V-10 illustre comment le Dropout modifie les connexions dans un réseau, en désactivant aléatoirement des neurones pendant l'entraînement.

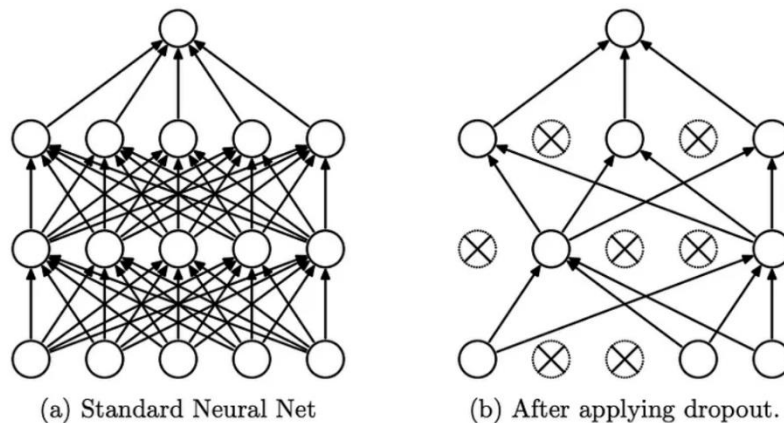


FIGURE V-10: Exemple de Dropout [56]

- **Optimiseur Adam** : L'optimiseur Adam (Adaptive Moment Estimation) combine les avantages des algorithmes de descente de gradient avec momentum et de descente de gradient adaptative [57]. Il ajuste les taux d'apprentissage pour chaque paramètre et converge plus rapidement que les autres optimisateurs, ce qui le rend particulièrement efficace pour les architectures complexes comme U-Net.

Il est à noter que pour l'optimisation, nous n'avons pas effectué d'ajustements spécifiques supplémentaires car le modèle U-Net est réputé pour **sa rapidité et son efficacité** dans la segmentation d'images. Cette caractéristique nous a permis de nous concentrer sur l'amélioration de la qualité des données d'entrée et des techniques d'entraînement sans compromettre les performances de calcul.

V.5.3 Outils, Technologies et Optimisations Clés

Dans le cadre de notre projet, nous avons intégré une variété d'outils et de technologies clés pour développer et optimiser nos modèles d'apprentissage profond, ainsi que pour gérer efficacement les données et les images. Ces outils ont joué un rôle crucial dans l'amélioration des performances de nos modèles et dans la facilitation du processus de développement.

Pour l'exploration interactive des données et l'expérimentation de code, nous avons utilisé **Jupyter Notebook**. **Visual Studio Code (VSCode)** a servi d'un **IDE principal pour le développement, le débogage et la gestion de nos projets**.

En ce qui concerne les bibliothèques principales, nous avons employé **NumPy** [58] et **Pandas** [59] pour la manipulation des données structurées, les calculs numériques et

la préparation des données avant l'entraînement des modèles. **OpenCV** [60] a été utilisé pour le prétraitement des images, la vision par ordinateur, ainsi que la manipulation et l'analyse d'images et de vidéos. On a utilisé aussi **Albumentations** [54] pour augmenter nos données. **Matplotlib** [61] et **Seaborn** [62] ont été utilisés pour la visualisation des données, permettant de générer des graphiques, des histogrammes et d'autres représentations visuelles pour l'analyse des résultats.

TensorFlow [63] avec **TensorFlow.keras** [64] a été notre framework principal pour la construction et l'entraînement des modèles d'apprentissage profond. Nous avons exploité des architectures pré-entraînées telles qu'EfficientNet et InceptionV3, bénéficiant ainsi de modèles éprouvés pour différentes tâches.

Pour tirer pleinement parti de la puissance de calcul des GPU, nous avons utilisé le **CUDA Toolkit** [65] avec **TensorFlow**, spécifiquement les modèles de **GTX 1660 Ti** et **GTX 1650**, accélérant ainsi le processus d'entraînement des modèles grâce au parallélisme massif offert par les GPU.

En outre, nous avons utilisé **Shutil** [66] pour la gestion des fichiers et des répertoires, tandis que **TQDM** [67] a fourni des barres de progression pendant le traitement des données et l'entraînement des modèles. **Scikit-Learn** [68] a été intégré pour l'évaluation des modèles avec des métriques telles que les rapports de classification, la matrice de confusion, et d'autres indicateurs essentiels pour l'évaluation de la performance des modèles. **TensorFlow Addons (TFA)** a été intégré pour ajouter des fonctionnalités supplémentaires à TensorFlow, enrichissant ainsi les capacités de notre modèle avec des outils comme la métrique F1 Score.

V.6 Évaluation Comparative des Approches et Résultats

Dans le cadre de notre tâche de segmentation et de génération de masques pour les zones manipulées dans les images, nous avons exploré et implémenté plusieurs approches. Pour évaluer rigoureusement leurs performances respectives, nous avons utilisé plusieurs métriques calculées sur trois ensembles distincts : l'ensemble d'entraînement, l'ensemble de validation et l'ensemble de test. Cette approche nous permet non seulement d'évaluer la performance brute des modèles, mais aussi leur capacité de généralisation. En présentant une analyse comparative détaillée de nos différentes méthodes, nous pouvons déterminer leurs forces, leurs faiblesses et leur efficacité globale dans notre contexte spécifique.

V.6.1 Métriques d'Évaluation

- **Accuracy** : Cette mesure permet de quantifier la proportion globale de pixels correctement classifiés par le modèle. L'accuracy est définie comme le ratio des pixels correctement classés sur le total des pixels. Mathématiquement, elle peut être exprimée de la manière suivante :

$$Accuracy = \frac{VP+VN}{VP+VN+FP+FN} \quad (5.1)$$

Où VP : représente les Vrais Positifs, VN : les Vrais Négatifs, FP : les Faux Positifs et FN : les Faux Négatifs.

Cette métrique est importante car elle donne une indication globale de la performance du modèle, mais elle peut ne pas refléter pleinement l'efficacité du modèle.

- **Logloss (Perte logarithmique)** : C'est une métrique utilisée pour évaluer la qualité des prédictions probabilistes d'un modèle. Elle mesure la divergence entre les probabilités prédites et les vraies étiquettes pour chaque pixel. La formule mathématique de la logloss est la suivante :

$$LogLoss = -\frac{1}{n} \sum_{i=0}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (5.2)$$

Où N : est le nombre total de pixels, y_i : est la vraie étiquette (0 ou 1) et \hat{y}_i : est la probabilité prédite pour chaque pixel.

Cette métrique est particulièrement pertinente pour la segmentation car elle prend en compte la certitude des prédictions pour chaque pixel, pénalisant fortement les erreurs confiantes. En d'autres termes, la logloss permet de mesurer la précision des prédictions du modèle tout en tenant compte de la probabilité de chaque pixel.

- **F1-Score** : cette métrique offre une mesure équilibrée entre la précision et le rappel. La précision mesure la proportion de pixels correctement prédits comme manipulés parmi tous les pixels prédits comme manipulés, tandis que le rappel mesure la proportion de pixels manipulés correctement détectés parmi tous les pixels manipulés réels. Le F1-score est défini comme la moyenne harmonique de la précision et du rappel. La moyenne harmonique est utilisée car elle donne plus de poids aux valeurs les plus faibles, ce qui signifie que le F1-score sera plus faible si la précision ou le rappel est faible, et est exprimé mathématiquement comme suit :

$$F1 \text{ Score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (5.3)$$

Où :

$$Precision = \frac{TP}{TP+FP} \text{ et } Recall = \frac{TP}{TP+FN} \quad (5.4)$$

Dans le contexte de la segmentation des zones manipulées, le F1-Score est crucial car il équilibre la nécessité de détecter correctement les pixels manipulés (rappel élevé) tout en minimisant les fausses détections (précision élevée). Il varie entre 0 et 1, où 1 indique une précision et un rappel parfaits et 0 indique une précision et un rappel nuls. En général, un F1-score élevé indique que le modèle a une bonne capacité à détecter les pixels manipulés tout en minimisant les erreurs de classification.

V.6.2 Résultats

Nous avons procédé au calcul systématique des métriques sur les ensembles d'entraînement (Évaluer la performance brute des modèles), de validation (Vérifier la capacité de généralisation sur l'ensemble de validation pendant l'entraînement) et de test (Mesurer la performance finale et non biaisée sur l'ensemble de test). Cette approche nous permet de comparer ses performances sur différents jeux de données et d'identifier d'éventuels problèmes de surapprentissage ou de sous-apprentissage.

Dans la section suivante, nous exposerons notre protocole expérimental, incluant une description du dataset de test et notre méthodologie d'évaluation. Nous présenterons ensuite un tableau comparatif des performances de chaque approche, accompagné de graphiques illustrant les performances relatives. Cette analyse quantitative sera suivie d'une évaluation qualitative, où nous examinerons des exemples visuels de détections réussies et d'échecs pour chaque approche, tout en discutant des patterns observés.

De plus, nous évaluerons la robustesse et la capacité de généralisation de nos modèles en testant leur performance sur des images de qualités variées et sur des types de manipulations non vus pendant l'entraînement. Enfin, nous aborderons l'efficacité computationnelle de chaque approche, en comparant les temps d'inférence et en discutant des besoins en ressources. Cette analyse approfondie nous permet de dresser un tableau complet (Tableau V-3) qui récapitule les métriques de chaque méthode dans le contexte de la détection de zones manipulées dans les images et vidéos.

TABLEAU V-3: Tableau récapitulatif des scores obtenus lors du training pour les différentes approches

Approches	Phase	Accuracy	Log-loss	F1-Score
U-Net avec Augmentation de Données Classique	Train	0.9532	0.1887	-
	Val	0.9598	0.1672	-
	Test	0.9558	0.1811	-
U-Net Entraîné Uniquement sur des Données manipulées (fake).	Train	0.9582	0.1889	-
	Val	0.9578	0.1846	-
	Test	0.9580	0.1885	-
U-Net avec Données Augmentées et Segmentées en Grilles	Train	0.6284	0.6348	0.5946
	Val	0.6360	0.6347	0.5983
	Test	0.5902	0.7102	0.5305

Pour mieux visualiser et comparer les performances de notre modèle, nous avons représenté ces résultats sous forme de bar chart dans la figure V-11.

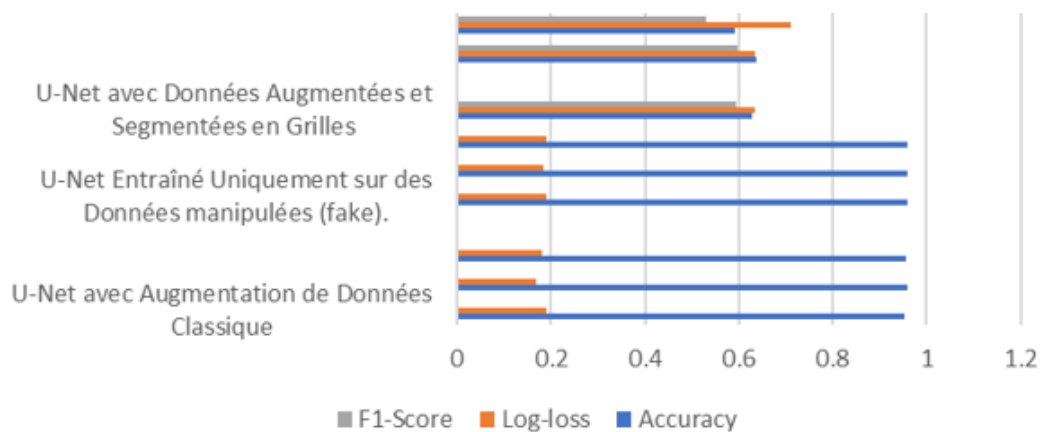


FIGURE V-11: Bar Chart récapitulatif des scores obtenus lors du training des différents modèles.

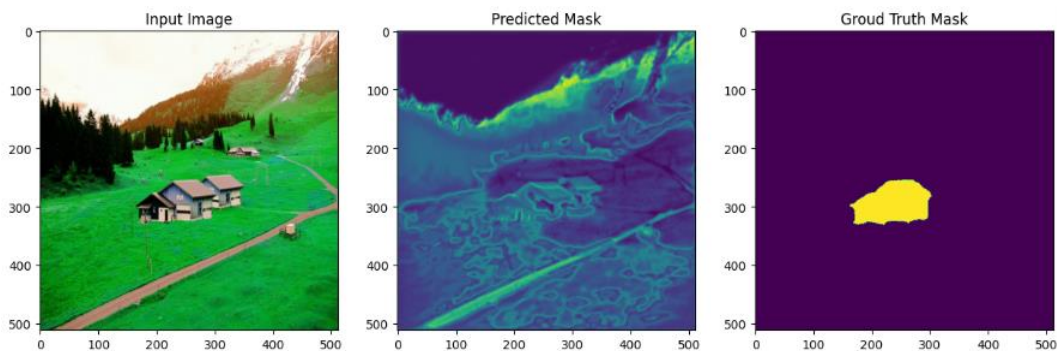
On y compare les performances de différentes variantes de l'U-Net sur les métriques **Log-loss**, **Accuracy**, et **F1-score**. Les modèles incluent U-Net avec Données Augmentées et Segmentées en Grilles, U-Net Entraîné Uniquement sur des Données manipulées (fake), et U-Net avec Augmentation de Données Classique.

Les résultats montrent que le U-Net Entraîné Uniquement sur des Données manipulées (fake) et l'U-Net avec augmentation de données classique affichent les meilleures performances en termes de métriques quantitatives, avec des F1-scores et Accuracy élevés, ainsi que des Log-loss bas.

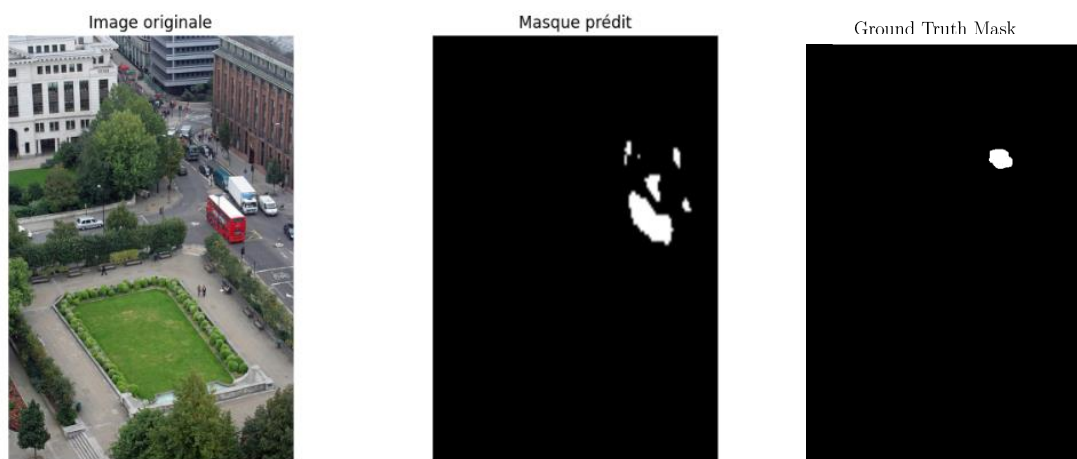
Il est à noter que les « tirets : - » pour l'approche U-Net avec Augmentation de Données Classique indiquent des résultats insatisfaisants, de l'ordre de 5 à 10%.

Cependant, il est crucial de souligner que malgré des scores quantitatifs similaires pour les trois meilleures approches, l'évaluation visuelle des résultats révèle une différence significative. **En termes de qualité visuelle des prédictions, la méthode utilisant la segmentation en grilles s'est avérée nettement supérieure.** Cette observation met en lumière l'importance de ne pas se fier uniquement aux métriques quantitatives, mais de prendre également en compte la qualité visuelle des résultats, particulièrement dans les tâches de segmentation d'images.

Afin d'illustrer cette différence, nous avons inclus des exemples de prédictions pour chaque méthode dans les figures V-12.



(a)



(b)

FIGURE V-12: La comparaison des deux approches de segmentation

La méthode (a), utilisant le modèle U-Net avec augmentation, n'a pas produit de résultats satisfaisants. Le masque généré diffère considérablement du masque attendu, le modèle se contentant de segmenter l'image en zones similaires. En revanche, l'approche (b), basée sur la segmentation en grille, s'est montrée nettement plus efficace. Elle a permis de générer un masque de bonne qualité, fournissant des indications pertinentes sur les zones où des manipulations ont été détectées.

Comme nous pouvons le constater, la méthode de segmentation en grilles produit des résultats visuellement plus satisfaisants, bien que ses métriques quantitatives ne soient pas nécessairement supérieures aux autres méthodes. Cette discordance entre les métriques quantitatives et la qualité visuelle souligne la complexité de l'évaluation des modèles de segmentation et l'importance d'une approche d'évaluation multi-facettes.

En effet, dans notre première approche d'utilisation de U-Net avec données augmentées, nous avons rencontré un problème de déséquilibre des classes, où la majorité des pixels étant noirs, notre modèle a réussi à les prédire correctement, ce qui a donné une accuracy élevée mais n'a pas amélioré la détection des zones manipulées. Pour résoudre ce problème, nous avons adopté une deuxième approche en entraînant le modèle uniquement sur des données manipulées (fake), mais cela n'a pas permis de résoudre le problème de déséquilibre des classes, car les zones manipulées ne représentent généralement que 5% de l'image.

C'est alors que nous avons introduit l'approche de segmentation en grilles, où nous avons choisi de ne garder que les fenêtres ayant un pourcentage de pixels blancs (fake) d'au moins 25%. Cette condition a permis d'équilibrer un peu nos classes et de mieux

apprendre les caractéristiques des zones manipulées. Cette approche a donc permis d'améliorer la qualité visuelle des résultats comme le montre la figure V-12.

Bien que les métriques quantitatives de notre modèle de segmentation aient affiché des scores prometteurs, l'inspection visuelle des résultats a révélé des lacunes significatives dans la détection et la localisation précise des falsifications. Cette disparité entre les performances mesurées et les résultats pratiques nous a conduits à reconsidérer notre approche.

Afin d'améliorer la fiabilité globale de notre système et de réduire le taux de faux négatifs, nous avons décidé d'explorer une stratégie alternative. Cette nouvelle méthode introduit une étape préliminaire de classification, visant à identifier les images potentiellement falsifiées avant de procéder à leur segmentation détaillée. Cette transition vers une approche en deux temps nous permet de concentrer nos efforts de segmentation sur les images les plus susceptibles de contenir des manipulations, tout en minimisant le risque de manquer des falsifications subtiles.

V.7 Retour à la compréhension du problème, approche de classification

La classification préalable des images a pour but de catégoriser chaque image en deux classes distinctes : **fake (falsifiée) ou pristine (authentique)**. Pour développer ce classifieur, nous avons recours à des techniques d'apprentissage supervisé, en utilisant un vaste ensemble de données étiquetées. Cela a nécessité quelques modifications sur notre dataset. **Dans le cas de la segmentation, notre label était un masque, tandis que dans le cas de la classification, notre label est une classe.** Nous avons donc dû réadapter notre dataset pour qu'il soit en adéquation avec notre tâche. L'objectif principal est de créer un modèle capable de distinguer avec précision les images manipulées des images originales, en se basant sur diverses caractéristiques extraites des images.

Cette stratégie de classification préalable présente plusieurs avantages notables. Tout d'abord, elle permet de **réduire les faux négatifs** en identifiant d'abord les images potentiellement falsifiées, ce qui réduit le risque de négliger des manipulations subtiles. De plus, elle **optimise les ressources** en n'appliquant la segmentation, qui est une tâche plus complexe et gourmande en ressources, qu'aux images classées comme suspectes, ce qui améliore l'efficacité globale du système. En outre, la classification préalable permet **d'améliorer la précision** en agissant comme un filtre initial, ce qui permet à l'algorithme de segmentation de se concentrer uniquement sur les images les plus

susceptibles d'être falsifiées. Enfin, cette approche en deux temps **offre une flexibilité accrue** en permettant d'ajuster séparément les seuils de sensibilité pour la classification et la segmentation, offrant ainsi un meilleur contrôle sur l'équilibre entre précision et rappel.

Pour optimiser notre système de détection de falsifications, nous avons implémenté une **technique d'ensemble (Ensemble Learning)** combinant plusieurs modèles de classification avancés. Cette approche vise à exploiter les forces de différentes architectures de réseaux neuronaux tout en minimisant leurs faiblesses individuelles. **En utilisant multiple modèles en parallèle**, nous cherchons à améliorer la précision, la robustesse et la généralisation de nos prédictions. Dans les sections suivantes, nous détaillerons notre méthode d'ensemble learning, le mécanisme de vote majoritaire, et présenterons les modèles de classification intégrés dans notre système.

En outre, nous avons choisi la **technique de bagging** pour répondre à d'autres exigences techniques telles que les ressources de traitement (le traitement parallèle étant plus rapide qu'un traitement séquentiel) et le fait que notre **dataset était équilibré**, ce qui justifie notre choix.

Examinons maintenant comment cette mise en concurrence des modèles s'articule avec le mécanisme de vote majoritaire pour produire des résultats robustes et fiables.

V.7.1 Mise en concurrence des modèles et mécanisme de vote majoritaire

Notre approche d'ensemble learning consiste à mettre en concurrence plusieurs architectures de réseaux neuronaux. Nous avons opté pour des modèles de **transfer learning** et avons sélectionné une variété de modèles de **la famille EfficientNet** [69] ainsi que le modèle **InceptionV3** [70], tous reconnus pour leurs performances exceptionnelles en classification d'images.

Voici un tableau comparatif (Tableau V-4) des modèles mentionnés ci-dessus, présentant leurs caractéristiques clés, telles que la profondeur, le nombre de paramètres, la taille d'entrée et les performances en termes de précision (sur image) et de FLOPs (FLOating-point Operations Per Second). Ce tableau donne des détails concernant les différents modèles que nous allons mettre en concurrence.

TABLEAU V-4: Comparaison entre les modèles pré-entraînés

Modèle	Profondeur	Paramètres	Taille d'entrée	FLOPs	Top-1 Accuracy	Avantages
EfficientNet B0	237	5.3M	224x224	0.39B	76.3%	Bon compromis entre performance et efficacité computationnelle
EfficientNet B4	456	19M	380x380	4.2B	82.9%	Niveau de complexité intermédiaire avec une profondeur et une largeur accrue
EfficientNet B7	813	66M	600x600	37.0B	84.3%	Le plus grand et potentiellement le plus performant de la première génération
EfficientNet V2B3	337	12M	300x300	1.8B	81.5%	Performance améliorée par rapport à EfficientNetB3 de la première génération
EfficientNet V2M	456	21M	384x384	4.0B	83.1%	Équilibre entre performance et coût computationnel
EfficientNet V2L	576	48M	512x512	9.9B	84.6%	Le plus complexe et potentiellement le plus précis de la deuxième génération
InceptionV3	159	23.8M	299x299	5.7B	77.3%	Capable de capturer des caractéristiques à différentes échelles grâce à des modules de convolution parallèles

Dans le cadre de notre approche, nous avons non seulement sélectionné des modèles pré-entraînés sur de vastes ensembles de données d'images générales, tels que **ImageNet** (le score de l'accuracy dans le tableau V-4 est le résultat de ce pré-entraînement), et affinés sur notre dataset spécifique de détection de falsifications, mais nous avons également mis en place un mécanisme de **vote majoritaire** pour combiner les prédictions de ces modèles. Chaque modèle **vote** pour une classe (**fake ou pristine**), et **la classe qui reçoit le plus de votes est considérée comme la prédiction finale de l'ensemble**. Cette approche simple et efficace nous permet de tirer parti de la diversité

des modèles et de leurs architectures complémentaires, tout en améliorant la précision globale de notre système de détection de falsifications.

En plus du vote majoritaire, nous avons également implémenté le **calcul de la probabilité moyenne pour la classe prédite**. Cela se fait en moyennant les probabilités prédites par chaque modèle pour la classe gagnante, ce qui fournit une mesure de confiance pour la prédiction finale. Cette mesure de confiance est importante car elle nous permet d'évaluer la fiabilité de nos prédictions et d'identifier les cas où une attention particulière est nécessaire. En combinant le vote majoritaire et la probabilité moyenne, nous sommes en mesure de maximiser les avantages de notre approche et de fournir des résultats précis et fiables.

V.7.2 Evaluation et résultats

Dans cette section, nous présentons les résultats obtenus par notre approche (Tableau V-5). Nous avons évalué les performances de notre modèle en utilisant les mêmes métriques que celles utilisées pour la segmentation, à savoir le **Log-Loss**, l'**accuracy** et le **F1-score**.

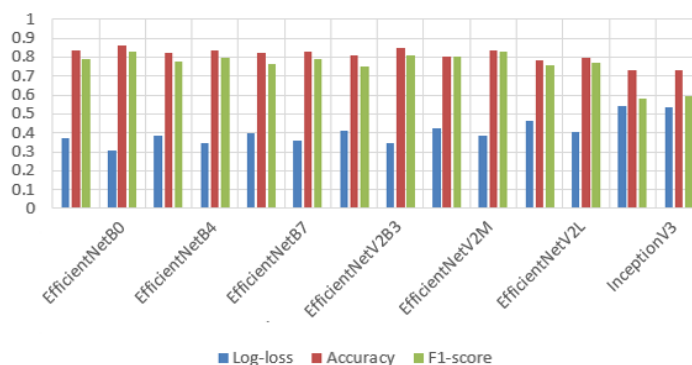
TABLEAU V-5: récapitulatif des scores obtenus lors du training des différents modèles

	Phase	Log-loss	Accuracy	F1-score
EfficientNetB0	Train	0.3726	0.8346	0.7879
	Val	0.3094	0.8592	0.8286
EfficientNetB4	Train	0.3863	0.8245	0.7736
	Val	0.3475	0.8386	0.7944
EfficientNetB7	Train	0.3961	0.8201	0.7612
	Val	0.3560	0.8273	0.7903
EfficientNetV2B3	Train	0.4109	0.8102	0.7527
	Val	0.3460	0.8482	0.8103
EfficientNetV2M	Train	0.4211	0.8010	0.8018
	Val	0.3845	0.8334	0.8304
	Train	0.4608	0.7818	0.7583

EfficientNetV2L	Val	0.4015	0.7995	0.7702
InceptionV3	Train	0.5390	0.7341	0.5798
	Val	0.5379	0.7308	0.5935

Pour mieux visualiser et comparer les performances de notre modèle sur les ensembles d'entraînement et de validation, nous avons représenté ces résultats sous forme de bar chart dans la figure V-13, ou on compare les performances de différents modèles sur les métriques **Log-loss**, **Accuracy**, et **F1-score** pour les **dataset de l'entraînement (à gauche) et de validation (à droite)**. Les modèles incluent diverses variantes d'EfficientNet (B0, B4, B7, V2B3, V2M, V2L) ainsi qu'InceptionV3. Les résultats montrent qu'EfficientNetB0 affiche de bons résultats en précision et F1-score, malgré un Log-loss plus élevé. EfficientNetB4 et B7 améliorent ces performances, avec des Log-loss plus faibles et des métriques de précision et F1-score plus élevées. Les versions plus récentes, EfficientNetV2B3, V2M, et V2L, continuent cette tendance d'amélioration. InceptionV3, bien qu'efficace, présente un Log-loss légèrement plus élevé que les variantes EfficientNet, mais sa précision et son F1-score restent compétitifs.

On peut dire que les variantes plus récentes d'EfficientNet (notamment les versions V2) offrent des gains notables en précision et F1-score, tout en réduisant le Log-loss, démontrant ainsi une meilleure performance globale dans les tâches de classification d'images. Les différences de performance peuvent être attribuées à l'architecture avancée d'EfficientNet (l'utilisation des blocs **MBCConv⁴** et **Fusion-MBCConv⁵**).



⁴ Mobile Inverted Bottleneck Convolution, est un bloc de convolution utilisé dans les réseaux neuronaux mobiles, caractérisé par une structure de bottleneck inversé pour améliorer l'efficacité computationnelle et la performance du modèle.

⁵ La fusion MBCConv combine plusieurs blocs MBCConv de différentes résolutions spatiales pour capturer des caractéristiques à différentes échelles et optimiser la performance du modèle.

FIGURE V-13: Comparaison entre les performances des différents modèles

Remarque :

Pendant notre entraînement, nous avons conservé les mêmes hyperparamètres et paramètres que ceux utilisés dans nos modèles de segmentation. En plus de cela, nous avons également essayé d'ajouter d'autres techniques telles que l'Early Stopping et l'Adaptive Learning Rate.

L'**Early Stopping** [71] est une technique de régularisation pour les réseaux neuronaux profonds qui arrête l'entraînement lorsque les performances sur l'ensemble de validation commencent à se dégrader, indiquant un début de surapprentissage. Cette méthode sélectionne les paramètres du modèle au point optimal, avant que la généralisation ne se détériore, améliorant ainsi les performances globales du modèle. Le nombre d'itérations avant l'arrêt peut être traité comme un hyperparamètre à optimiser pour obtenir les meilleures performances du modèle d'apprentissage. Dans notre cas, nous avons fixé le nombre d'époques à 100, mais lors de l'entraînement, certains modèles ont terminé leur entraînement avec un nombre d'époques inférieur à 100. Par exemple, l'entraînement du modèle **EfficientNetB7** s'est arrêté à l'époque 74. Cela nous a permis d'obtenir une meilleure capacité de généralisation de la part de nos modèles dans un temps d'entraînement plus court.

La figure V-14 montre deux courbes : l'erreur d'entraînement (rouge) qui diminue continuellement, et l'erreur de développement (grise) qui atteint un minimum avant de remonter. Le point **Early Stopping** indique le moment optimal pour arrêter l'entraînement, où le modèle offre la meilleure généralisation avant de commencer à surapprendre. Cette visualisation démontre l'efficacité de l'Early Stopping pour éviter le surapprentissage tout en optimisant les performances du modèle.

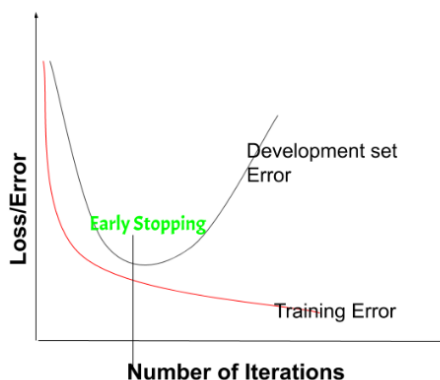


FIGURE V-14: illustration du principe de l'Early Stopping. [72]

Pour la technique d'**Adaptive Learning Rate** [73], il s'agit d'une méthode d'optimisation des modèles d'apprentissage profond qui ajuste automatiquement les taux d'apprentissage pendant l'entraînement. Pour notre cas, nous avons opté pour l'utilisation de la technique : réduction du taux d'apprentissage sur plateau. Cette technique est utilisée pour améliorer la convergence et éviter les stagnations de performance. Lorsque la performance du modèle, mesurée par une métrique spécifique comme la perte de validation, cesse de s'améliorer pendant un certain nombre d'époques (par exemple : 5 époques), cette technique réduit le taux d'apprentissage. En réduisant le taux d'apprentissage de moitié, elle permet au modèle de faire des mises à jour des poids plus petites et plus précises, favorisant ainsi une meilleure exploration de l'espace des paramètres. Cette méthode est particulièrement utile pour surmonter les plateaux dans la courbe de performance et peut conduire à une meilleure convergence globale du modèle. Elle doit être utilisée en complément d'autres techniques comme l'arrêt anticipé, qui arrête l'entraînement lorsqu'il n'y a plus d'amélioration après un certain temps, pour optimiser l'efficacité de l'entraînement.

Comparaison de la performance de notre modèle avec des annotations humaines pour la détection de zones manipulées dans les images

Pour évaluer la performance de notre modèle de manière plus concrète et subjective, nous avons mené une expérience impliquant des humains. Nous avons sélectionné **100 images et les avons distribuées à 10 personnes** pour qu'elles annotent les zones manipulées qu'elles pouvaient détecter. En parallèle, nous avons utilisé notre classifieur de vote majoritaire et le bloc de segmentation pour générer le masque des images classées comme manipulées. Les résultats ont révélé des différences significatives entre les performances humaines et celles de notre modèle. Les participants humains ont réussi à **classifier correctement les images avec un taux de 40%**, mais ont montré de faibles résultats dans la phase de segmentation, avec seulement **10% de réussite pour identifier correctement les zones manipulées** (soit 4 images sur 100 où ils ont fourni des indices corrects). En revanche, notre modèle a démontré des performances supérieures, avec **une précision de classification de 65%** et une capacité à **identifier correctement les zones de manipulation dans 24,61% des cas** (soit 16 images sur 100 où les zones identifiées correspondaient étroitement aux régions effectivement manipulées). **Ces résultats que montre la figure V-15 soulignent l'efficacité de notre approche automatisée par rapport à l'analyse humaine pour cette tâche complexe.**

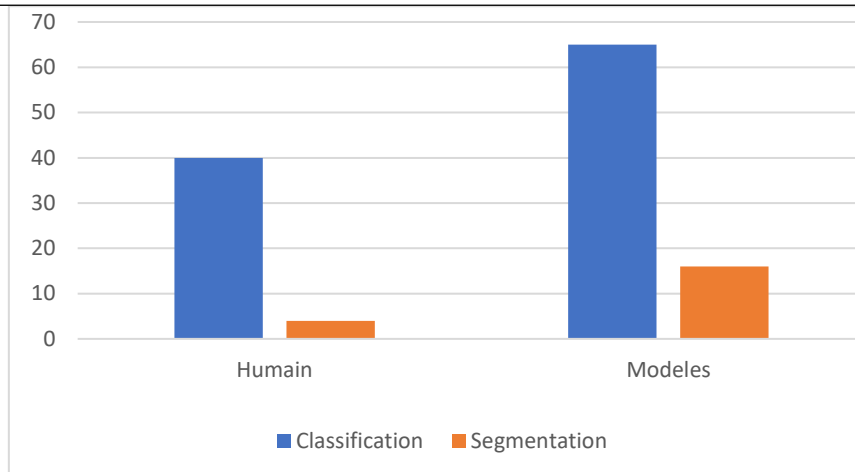


FIGURE V-15: Comparaison de résultats de l'approche automatisée par rapport à l'analyse humaine

Dans la figure V-16, le modèle de détection de manipulation a systématiquement identifié des zones manipulées dans chaque image, produisant des masques blancs de tailles variables dans la deuxième colonne. Cependant, les masques de vérité terrain dans la troisième colonne ont montré que les zones manipulées étaient généralement plus petites et plus précises que celles prédites par le modèle. En comparaison, les résultats de **l'analyse humaine dans la quatrième colonne étaient moins cohérents**. Dans certains cas, aucune manipulation n'a été détectée même si elle est présente (cas de la 2eme image), résultant en des masques noirs, tandis que dans d'autres cas, les zones identifiées étaient plus grandes que la vérité terrain. **Ces résultats suggèrent que le modèle a tendance à surévaluer la taille des zones manipulées, tandis que l'analyse humaine peut être plus variable et moins fiable dans la détection des manipulations.**

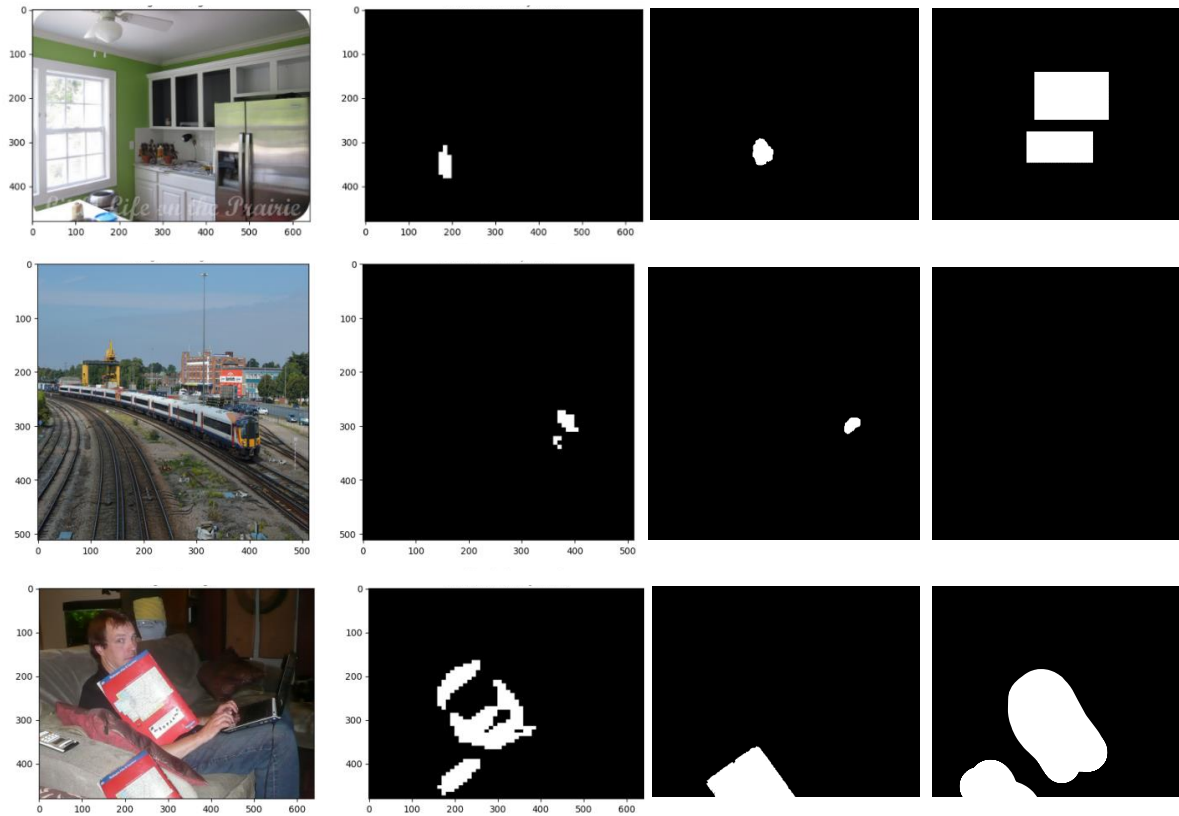


FIGURE V-16: Comparaison entre les résultats obtenus par l'analyse humaine et l'analyse automatique.

V.7.3 Pipeline de prédiction des falsifications : Analyse adaptative d'images et de vidéos

Notre pipeline de prédiction, illustré dans la figure V-17, qui représente l'ensemble des techniques et processus appliqués séquentiellement aux entrées de notre modèle lors des phases de test et d'utilisation, présente une flexibilité remarquable. Ce pipeline se caractérise par sa capacité à accepter deux formats distincts d'entrée : images statiques et séquences vidéo. Cette dualité dans le traitement des données d'entrée permet une application étendue de notre système, le rendant adaptable à diverses situations d'analyse de contenu visuel.

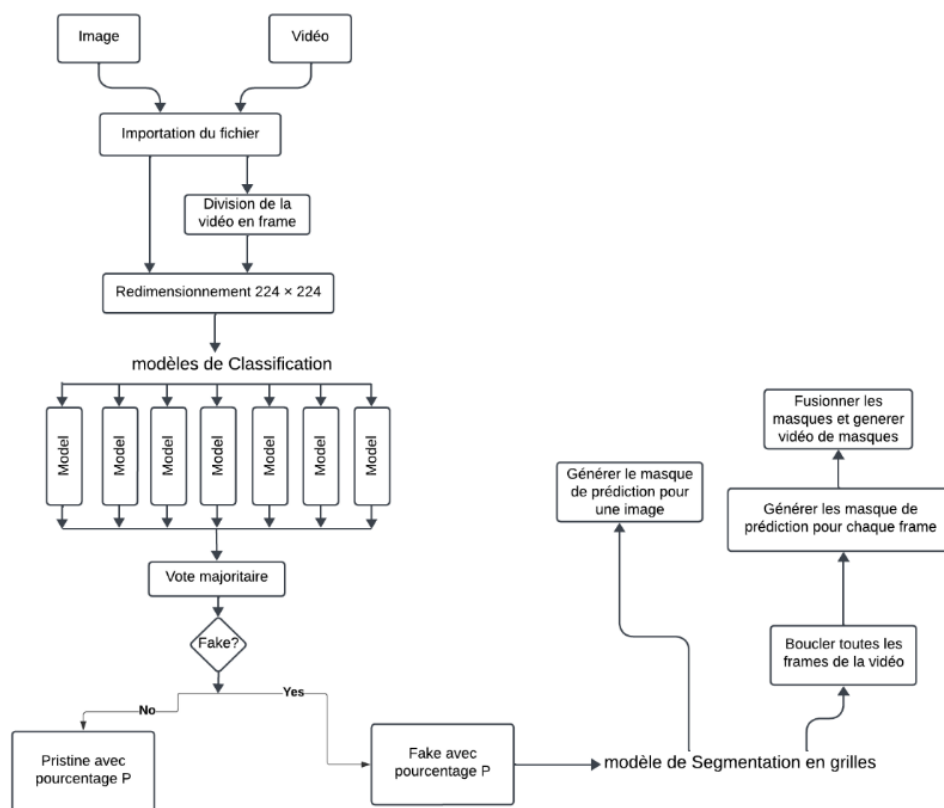


FIGURE V-17: Pipeline de prédiction

Pour les images statiques et les vidéos, le processus suit les étapes suivantes :

(i) **Pré-traitement :**

1. **Importation du fichier :** Le pipeline accepte des images individuelles ou des séquences vidéo.
2. **Division de la vidéo en frames (pour les vidéos) :** La vidéo est décomposée en une série de frames individuelles.
3. **Redimensionnement :** Chaque image ou frame est redimensionnée à une résolution standard de 224x224 pixels, garantissant une cohérence dans le traitement et optimisant les performances du modèle.

(ii) **Classification :**

1. **Modèles de Classification :** Une série de modèles de classification sont utilisés pour analyser chaque image ou frame. Ces modèles votent ensuite de manière majoritaire pour déterminer si l'image ou le frame est « Fake » (falsifiée) ou « Pristine » (authentique).

2. **Vote majoritaire** : Les résultats des différents modèles de classification sont agrégés pour prendre une décision finale sur la nature de l'image ou de la frame (si un frame d'une vidéo est détectée comme « Fake », toute la vidéo sera considérée comme manipulée).

Le résultat final de classification indique si l'image est « Pristine » avec un pourcentage de confiance P_p , ou « Fake » avec un pourcentage de confiance P_f . Si l'image est considérée comme pristine, le processus se termine. Sinon, si l'image ou le frame est classifiée comme fake, un autre processus sera introduit : la segmentation.

(iii) Segmentation :

Pour le traitement des images, l'algorithme essaye de générer un masque de prédiction, mettant en évidence les zones potentiellement falsifiées au niveau des pixels, tandis que pour les vidéos, notre approche est plus complexe :

1. La séquence vidéo est d'abord décomposée en une série de frames individuelles.
2. Chaque frame est traité de manière indépendante, suivant le même processus que pour une image statique.
3. Une fois l'analyse de tout le frame est effectuée, nous procédons à la reconstruction de la vidéo, intégrant les masques de prédiction pour chaque frame.

Il est crucial de souligner que notre méthodologie, conformément aux spécifications de le GN, se focalise exclusivement sur la détection des falsifications intra-frame (cas des vidéos). Cette approche cible les manipulations au niveau des pixels au sein de chaque image individuelle. Nous n'avons pas intégré d'analyse des falsifications inter-frame dans ce pipeline. Ce type d'altération, qui concerne les modifications affectant la continuité temporelle entre les frames successives, n'entre pas dans le cadre de notre étude actuelle. Cette limitation est importante à noter car certaines techniques de falsification vidéo sophistiquées peuvent exploiter ces aspects temporels (addition ou suppression de frames par exemple).

V.8 Déploiement

Conformément aux exigences spécifiques de la GN, nous avons opté pour le développement d'une **application desktop** pour déployer notre solution de détection de falsifications. Cette approche offre plusieurs avantages, notamment en termes de performance et de sécurité des données traitées localement.

L'application a été développée en utilisant les technologies **Electron** et **RestfulAPI**, dont les détails techniques sont fournis dans l'annexe C-4.

Cette synergie technologique nous a permis de concevoir une interface utilisateur intuitive et fonctionnelle, qui traduit efficacement la puissance de nos algorithmes en une expérience utilisateur fluide et accessible. L'interface de notre application, fruit de cette approche intégrée, se caractérise par une organisation logique et une facilité d'utilisation qui masque la complexité des opérations sous-jacentes.

Voici une description détaillée des composantes principales de notre interface, illustrant comment nous avons mis en œuvre ces technologies pour créer une expérience utilisateur optimale :

V.8.1 Frontend (Interface utilisateur)

L'interface utilisateur de notre application desktop se caractérise par une conception épurée et intuitive, visant à faciliter l'utilisation pour tous les types d'utilisateurs. L'écran principal est organisé de manière logique, divisant l'espace en plusieurs sections distinctes pour une navigation aisée entre les différentes fonctionnalités comme indique la figure V-18 :

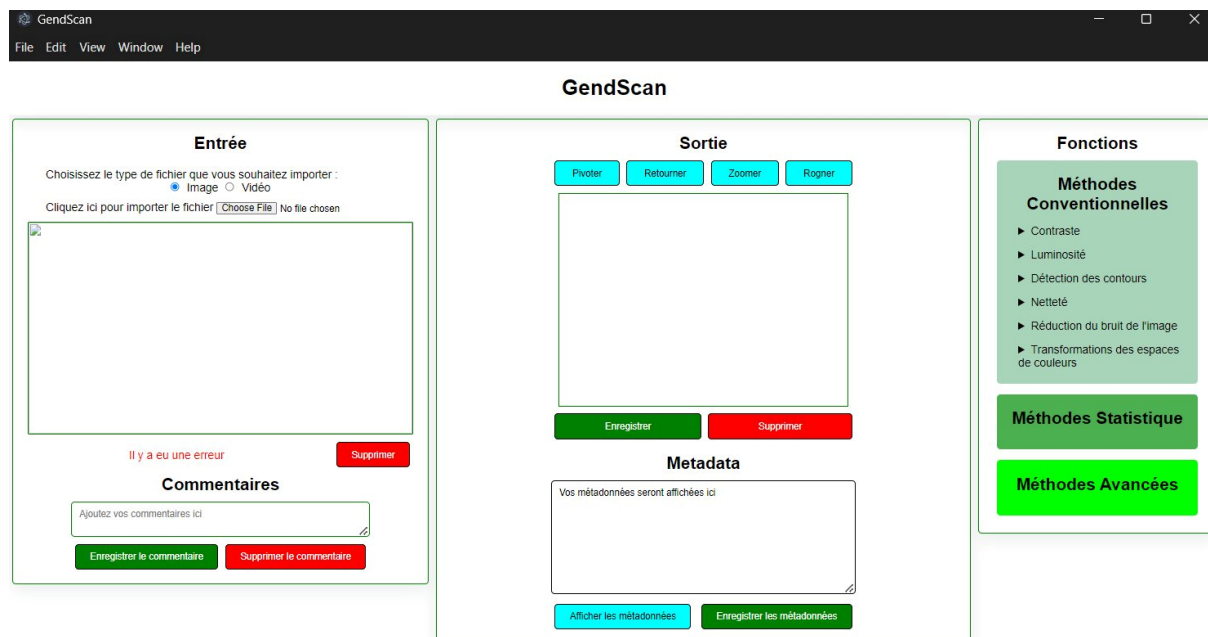


FIGURE V-18: Interface implémentée

Structure et Fonctionnalités de l'Interface Utilisateur

Barre de menu : En haut de l'interface, une barre de menu classique offre un accès rapide aux fonctionnalités essentielles :

- File : Gestion des fichiers (ouverture, sauvegarde)
- Edit: Options d'édition standard
- View: Contrôle de l'affichage
- Window : Gestion de la fenêtre d'application
- Help : Accès à l'aide et aux informations

Section d'Importation « Entrée » : Au cœur de l'application, la section « Entrée » facilite l'importation et la gestion initiale des fichiers à analyser :

- Sélection du type de fichier (image ou vidéo) via des boutons radio
- Bouton « Choose File » pour l'importation de fichiers
- Zone d'affichage pour la prévisualisation du fichier importé
- Affichage de messages d'erreur en cas de problème d'importation
- Option « Supprimer » pour retirer le fichier et réinitialiser la section

Cette conception permet une prise en main rapide du processus d'importation, première étape cruciale de l'analyse.

Zone de Traitement « Sortie » : Adjacente à la section d'entrée, la zone « Sortie » offre des outils de manipulation et de visualisation des résultats :

- Outils de manipulation (Pivoter, Retourner, Zoomer, Rogner)
- Large zone d'affichage pour les résultats des opérations
- Options « Enregistrer » et « Supprimer » pour la gestion des modifications

Cette section permet aux utilisateurs d'ajuster et d'analyser visuellement les fichiers traités, offrant une flexibilité essentielle dans le processus de détection.

Espace « Commentaires » pour l'Investigateur : Cette section est conçue pour faciliter le travail de documentation de l'investigateur :

- Zone de texte pour la saisie détaillée des observations et analyses

- Fonction de sauvegarde des commentaires en fichier texte
- Option de suppression pour réviser les notes

Cet espace permet à l'investigateur de consigner ses observations en temps réel, de documenter sa méthodologie, et de noter ses conclusions. La possibilité de sauvegarder ces notes en fichier texte facilite leur intégration dans des rapports officiels et permet un partage aisé entre collègues, renforçant ainsi l'efficacité du processus d'investigation.

Gestion des « Metadata » : Une section spécifique est consacrée à l'affichage et à la gestion des métadonnées :

- Zone de texte pour l'affichage des informations
- Options pour afficher et sauvegarder les métadonnées

Cet élément permet une analyse approfondie des caractéristiques techniques des fichiers, complémentaire à l'analyse visuelle.

Panel « Fonctions » de Détection : Au cœur de l'application, le panel « Fonctions » regroupe les différentes méthodes de détection de falsifications :

- Méthodes simples : Ajustements basiques (contraste, luminosité...)
- Méthodes traditionnelles : Techniques éprouvées de détection (ELA, CFA...)
- Méthodes complexes : Algorithmes avancés pour une analyse approfondie (Classification et Segmentation)

Chaque catégorie est accessible via un bouton dédié, permettant une application ciblée des méthodes de détection.

L'interface principale suit une disposition logique avec la section « Entrée » en haut à gauche, suivie de la section « Sortie » à droite. Les sections « Commentaires » et « Metadata » sont placées en dessous, tandis que la section « Fonctions » occupe une place importante sur le côté droit de l'écran. L'interface utilisateur repose sur un backend robuste et sophistiqué qui effectue le travail complexe de traitement et d'analyse des images et vidéos. Dans les coulisses de notre application, les composants essentiels du backend comprennent les algorithmes de traitement d'image, les modèles de détection de falsifications, et l'infrastructure nécessaire pour gérer efficacement les flux de données entre le frontend et les processus d'analyse.

V.8.2 Backend (logique de traitement)

(i) Fonctionnalités Clés du Backend

Notre backend, véritable cœur opérationnel de l'application, intègre un ensemble de fonctionnalités sophistiquées pour assurer un traitement efficace et précis des fichiers multimédias :

1. Traitement Avancé des Fichiers

- Redimensionnement automatique des images et vidéos pour une analyse optimale
- Conversion intelligente des formats pour garantir la compatibilité avec nos algorithmes
- Gestion minutieuse des métadonnées pour préserver l'intégrité des informations
- Stockage sécurisé et temporaire des résultats de traitement

2. Synergie avec le Modèle de Prédiction

- Intégration transparente du modèle via une interface Python robuste
- Prétraitement pointu des données pour les aligner aux exigences du modèle
- Post-traitement élaboré des résultats pour une interprétation visuelle claire et précise

3. Gestion Rigoureuse des Flux de Données

- Validation méticuleuse des fichiers entrants (format, dimensions, intégrité)
- Génération de masques de prédiction haute précision pour identifier les zones suspectes

Ces fonctionnalités backend travaillent de concert pour transformer les entrées brutes en analyses approfondies et fiables. Chaque étape, du traitement initial à la génération des rapports, est optimisée pour garantir non seulement la précision des détections, mais aussi une expérience utilisateur fluide et intuitive.

L'interaction harmonieuse entre ces composants backend et l'interface utilisateur crée un écosystème puissant, capable de relever les défis complexes de la détection de falsifications dans les médias numériques.

(ii) Défis techniques et solutions :

- Gestion de la mémoire pour les fichiers volumineux : Implémentation d'un système de traitement par lots
- Optimisation des performances : Parallélisation de certains traitements

Après avoir exploré les fonctionnalités clés du backend, il est essentiel de comprendre comment ces éléments s'articulent avec le frontend pour offrir une expérience utilisateur cohérente et efficace. Dans la section suivante, nous allons examiner en détail l'interaction cruciale entre le frontend et le backend, mettant en lumière les mécanismes qui permettent une communication fluide et une intégration transparente entre ces deux composants essentiels de notre application.

V.8.3 Synergie Frontend-Backend : Orchestration du Flux de Données

L'efficacité de notre application repose sur une communication fluide et bien structurée entre l'interface utilisateur (frontend) et le cœur de traitement (backend). Cette synergie est rendue possible grâce à l'implémentation d'une API RESTful robuste, qui sert de pont entre ces deux composants essentiels.

Notre architecture de communication s'articule autour de trois end-points principaux :

- /upload : pour la transmission sécurisée des fichiers à analyser
- /analyze : pour déclencher et gérer le processus d'analyse
- /results : pour récupérer et afficher les résultats de détection

L'échange de données entre le frontend et le backend s'effectue via le format JSON, choisi pour sa légèreté et sa flexibilité. Cette approche standardisée assure une interopérabilité optimale et facilite le traitement des informations de part et d'autre.

Cette infrastructure de communication constitue la colonne vertébrale de notre application, permettant une orchestration harmonieuse du flux de données tout au long du processus de détection de falsifications. Examinons maintenant en détail comment les données circulent à travers notre système, de l'upload initial jusqu'à l'affichage des résultats finaux.

Flux de Données : De l'Importation à l'Analyse Finale

Notre application assure une interaction transparente entre l'interface utilisateur et le backend en gérant un flux de données comme démontre le tableau V-6 :

TABLEAU V-6: Flux de Données

Étape	Interface Utilisateur	Action Frontend	Traitement Backend	Retour Frontend
1. Importation et Validation Initiale	L'utilisateur choisit le type de fichier (image/vidéo) via des boutons radio. Sélection du fichier via le bouton « Choose File ».	Envoi du fichier au backend (requête POST à /upload).	Validation du format, des dimensions et de l'intégrité du fichier. Redimensionnement et conversion si nécessaire. Extraction et vérification des métadonnées.	Affichage du fichier validé dans la section « Entrée ».
2. Prétraitement et Analyse Manipulation du fichier (Pivoter, Retourner, Zoomer, Rogner).	Manipulation du fichier (Pivoter, Retourner, Zoomer, Rogner).	Transmission des commandes de manipulation au backend (ex : POST /rotate).	Application des manipulations demandées. Préparation des données pour l'analyse. Exécution des algorithmes de détection de falsifications. Post-traitement des résultats pour une interprétation visuelle claire.	-
3. Affichage et Interprétation des Résultats	-	-	-	Présentation des résultats dans la section « Sortie ». Affichage des masques de prédiction pour les zones suspectes. Mise à jour de la section « Metadata » avec les

				informations extraites.
4. Annotations et Gestion des Commentaires	Saisie de commentaires dans la zone dédiée.	Envoi des commentaires au backend (POST /comments).	Enregistrement ou suppression des commentaires selon l'action utilisateur.	Actualisation de l'affichage des commentaires.
5. Stockage et Sécurité des Données	-	-	Stockage temporaire sécurisé des fichiers et résultats. Application de mécanismes d'authentification et d'autorisation	Affichage des statuts de stockage et de sécurité.
6. Finalisation et Exportation	Demande d'enregistrement final via le bouton « Enregistrer ».	Envoi de la requête d'enregistrement (POST /save).	Stockage permanent et sécurisé des résultats par le backend.	Affichage d'un message de confirmation à l'utilisateur.

Ce tableau résume les différentes étapes du flux de données, les actions de l'interface utilisateur, du frontend et du backend, ainsi que les retours attendus en frontend. L'orchestration minutieuse de ce flux de données permet une analyse approfondie et fiable des falsifications, tout en offrant à l'utilisateur un contrôle total sur le processus d'investigation.

Passant de l'orchestration minutieuse du flux de données à la phase suivante de notre projet, nous nous concentrons maintenant sur les tests et la validation future de notre application.

V.8.4 Tests et Validation Future

Pour la version 1.0 de notre application, nous avons prévu une phase exhaustive de tests et de validation qui sera mise en œuvre après la livraison du logiciel au client (exemples : Tests d'interface utilisateur pour valider l'expérience utilisateur dans divers scénarios, Tests de performance pour mesurer les temps de traitement sur différents types de fichiers, etc.). Cette approche nous permettra de recueillir des retours d'expérience réels et d'affiner notre solution en conséquence.

V.9 Comparaison entre les différentes approches

Dans le but de faciliter la prise de décision, nous avons élaboré un tableau comparatif détaillé des différentes approches étudiées. Cette synthèse met en lumière les caractéristiques clés, les avantages et les limitations de chaque méthode, permettant ainsi

une évaluation objective et approfondie. Ce tableau servira d'outil essentiel au décideur, lui offrant une vue d'ensemble claire et structurée pour choisir la méthode la plus appropriée à ses besoins spécifiques et aux objectifs du projet. En présentant les informations de manière concise et comparative, nous visons à optimiser le processus de sélection et à garantir une décision éclairée et stratégique.

Tableau V-7 : Comparaison entre les différentes approches

Catégorie	Méthode	Description	Avantages	Inconvénients
Méthodes Statistiques	JPEG Ghost	Détection des traces résiduelles laissées par les opérations de compression/décompression JPEG.	Simple à mettre en œuvre, efficace pour les images compressées	Sensible aux variations de compression, peut générer des faux positifs
	Error Level Analysis (ELA)	Analyse des niveaux de compression différents au sein d'une image pour révéler les falsifications.	Efficace pour détecter les zones retouchées	Peut être affectée par la qualité de l'image d'origine
	Analyse des Incohérences de Bruit	Identification des anomalies dans les propriétés de bruit d'une image, notamment les variations de bruit entre différentes régions de l'image.	Utile pour détecter les falsifications subtiles	Sensible au type et à la qualité de l'image
	Filtrage Médian	Méthode de lissage d'image robuste permettant de mettre en évidence les contours et les zones potentiellement retouchées ou insérées dans une image.	Efficace pour révéler les falsifications dans les images bruitées	Peut générer des faux positifs dans les images avec des textures complexes
	Analyse des Artefacts de Démosaïquage	Détection des artefacts introduits par le processus de démosaïquage dans les images falsifiées.	Efficace pour les images issues de capteurs numériques	Peut ne pas être applicable à tous les types d'images
Méthodes Avancées	Deep Learning (CNN, Transfer Learning)	Utilisation des réseaux de neurones convolutifs (CNN) et des techniques d'apprentissage par	Haute précision, adaptable à différents types de falsifications	Nécessite de grandes quantités de données pour l'entraînement,

		transfert pour améliorer la précision de la détection des falsifications.		computationnellement coûteux
	Segmentation avec UNET	Utilisation de l'architecture UNET pour la segmentation d'images et l'identification fine des manipulations.	Précision élevée pour la segmentation des images	Complexité élevée, nécessite des ressources computationnelles importantes
Méthodes Conventio nnelles	Techniques de Prétraitement	Ajustements basiques tels que le contraste, la luminosité, etc., pour améliorer la qualité visuelle des images avant l'analyse.	Simple à mettre en œuvre, améliore la perception visuelle	Limitée à l'amélioration visuelle, ne détecte pas directement les falsifications
	Extraction des Métadonnées EXIF	Analyse des métadonnées EXIF pour obtenir des informations sur l'historique de l'image et détecter des anomalies potentielles.	Utile pour obtenir des informations contextuelles sur l'image	Peut être contournée par les falsificateurs expérimentés

V.10 Conclusion

Dans cette partie, nous avons étudié et intégré des méthodes de pointe pour la détection de falsifications d'images et de vidéos, en utilisant des techniques avancées telles que UNET et des modèles pré-entraînés. Notre solution a démontré une grande robustesse et efficacité, avec une amélioration significative de la précision et de la fiabilité par rapport aux méthodes traditionnelles. Cependant, ces méthodes présentent des défis en termes de complexité computationnelle et de besoins en ressources et nécessitent toujours des vérifications par les analystes car ça reste toujours un outil d'aide à la décision.

Il est essentiel de poursuivre les efforts pour optimiser et adapter ces techniques afin de faire face aux évolutions technologiques et aux nouvelles menaces de falsification numérique.

Conclusion Générale

Ce projet de fin d'études a abouti à la création d'un outil innovant d'aide à la décision pour la détection de falsifications dans les images et vidéos. En exploitant les dernières avancées en intelligence artificielle et vision par ordinateur, notre travail a embrassé un large éventail de domaines, des fondements théoriques aux applications pratiques les plus récentes.

Notre démarche a débuté par l'établissement d'une base théorique solide, essentielle pour comprendre les enjeux complexes de la détection de falsifications visuelles. Nous avons exploré en profondeur les concepts liés aux images numériques, leurs mécanismes d'acquisition, et les diverses techniques de manipulation.

L'état de l'art que nous avons dressé offre une vue d'ensemble des technologies de pointe en matière de traitement d'images et d'apprentissage profond. Nous avons particulièrement mis en avant l'importance du transfer learning, de l'ensemble learning, et l'utilisation de modèles sophistiqués comme UNET pour la segmentation d'images.

L'analyse approfondie du contexte du projet au sein du CRD-GN nous a permis d'identifier avec précision les besoins spécifiques et les lacunes existantes, renforçant ainsi la pertinence et l'utilité de notre approche.

Notre méthodologie a judicieusement allié des méthodes statistiques robustes à des techniques avancées de deep learning. Les approches statistiques, centrées sur l'analyse des anomalies de bruit et des artefacts de compression, ont fourni une première couche de détection efficace. Ces méthodes ont été complétées par des algorithmes de deep learning sophistiqués, impliquant une préparation méticuleuse des données et une modélisation optimisée. Le déploiement réussi de notre solution, comprenant une interface utilisateur conviviale et un backend performant, témoigne de la viabilité opérationnelle de notre approche.

En conclusion, ce projet démontre avec succès la faisabilité et l'efficacité d'un outil intelligent pour la détection de falsifications visuelles. En fusionnant des méthodes statistiques éprouvées avec des techniques de deep learning à la pointe de la technologie, nous avons posé les bases d'une solution robuste et évolutive. Les résultats obtenus sont très prometteurs et ouvrent la voie à des améliorations futures qui augmenteront encore la précision et la fiabilité de la détection. Notre travail apporte ainsi une contribution significative aux efforts visant à préserver l'intégrité des contenus visuels dans l'ère

numérique actuelle, offrant un outil précieux aux professionnels de la justice, des médias et de la sécurité.

Perspectives d'amélioration

Les perspectives d'amélioration de notre outil intelligent d'aide à la décision pour la détection de manipulations dans les images et les vidéos sont vastes et ambitieuses. Notre objectif principal est d'évoluer vers une solution **plus complète et performante**, capable de répondre aux défis croissants dans le domaine de la sécurité numérique.

L'extension des capacités de détection est une priorité majeure. Nous **visons à intégrer de nouvelles fonctionnalités pour identifier un large éventail de manipulations numériques** avancées, incluant les deepfakes, le face swapping et les manipulations inter-frames. Cette évolution permettra à notre outil de rester pertinent face aux techniques de falsification en constante évolution.

La transformation de notre application en **un outil multimodal** constitue une avancée significative. Outre l'analyse des images et des vidéos, nous prévoyons d'étendre nos capacités à **l'examen des fichiers audio et des documents textuels**. Cette approche holistique renforcera considérablement l'efficacité de notre solution dans la détection des diverses formes de contenu manipulé.

L'amélioration des performances techniques est au cœur de notre stratégie. Nous **envisageons d'implémenter de nouvelles techniques d'intelligence artificielle et d'optimiser nos modèles existants**. Ces efforts viseront à améliorer la précision de nos détections tout en permettant le **traitement efficace de grands volumes de données**, un aspect crucial pour l'utilisation à grande échelle de notre outil.

La création d'une base de données algérienne annotée par des experts du domaine représente un projet ambitieux et essentiel. Cette initiative répondra au besoin critique de données représentatives du contexte local, améliorant ainsi la précision et la pertinence de nos détections dans l'environnement spécifique de l'Algérie. Cette base de données pourrait également devenir une ressource précieuse pour la recherche et le développement dans le pays.

L'interconnexion sécurisée avec divers organismes, notamment les forces de l'ordre, est un aspect important de notre développement futur. Nous prévoyons de mettre en place une infrastructure permettant un accès contrôlé à notre outil, tout en assurant la sécurité et la traçabilité des données. Cette fonctionnalité facilitera l'utilisation de notre solution dans un cadre officiel et professionnel, renforçant son utilité pratique dans la lutte contre la désinformation et la fraude numérique.

Bibliographie

- [1] https://www.mdn.dz/site_cgn/sommaire/presentation/unit_spe/crd/crd_fr.php
- [2] <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>
- [3] <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>
- [4] https://www.google.com/url?sa=i&url=https%3A%2F%2Ffilecamp.com%2Fblog%2Fvector-vs-bitmap-images-explained%2F&psig=AOvVaw0KV-yfNUdjgEicXc_NXOyn&ust=1720517312877000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCLCLys2Ql4cDFQAAAAAdAAAAABAE
- [5] <https://www.slideshare.net/slideshow/study-on-fundamentals-of-raster-scan-graphics/267980567>
- [6] <https://www.winlight.fr/blog/la-profondeur-de-couleur-en-bits-dossier-complet-pour-un-ecran-led-a-limage-eclatante/>
- [7] <https://www.noiise.com/definition/formats-dimage/>
- [8] https://www.canva.com/fr_fr/decouvrir/tous-les-formats-images/
- [9] Thèse gael_mahfoudi-2021, partie 2.5.1 image formation pipeline.
- [10] https://www.researchgate.net/figure/Illustration-of-a-typical-acquisition-pipeline-in-a-digital-camera_fig1_335500278
- [11] <http://criminologie.site.koumbit.net/article/science-forensique>
- [12] <https://www.algosecure.fr/conseil/investigation-forensic>
- [13] <https://link.springer.com/article/10.1007/s11042-023-16455-x>
- [14] <https://link.springer.com/article/10.1007/s11042-023-15609-1#Sec3%E2%80%8E>
- [15] <https://link.springer.com/article/10.1007/s11042-022-14016-2>
- [16] Papier : Enhanced features in image manipulation detection
- [17] <https://www.fakeimagedetector.com/>
- [18] <https://github.com/GuidoBartoli/sherlock>
- [19] <https://www.invid-project.eu/>
- [20] Digital image processing book: 19 chapitre 1 p18

- [21] <https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcT2PxZ7vdYRu1W8lSR4yRwsnO0bEyzivDiATXRq1hG7tGQ4oqa>
- [22] Hands on ML with tensorflow2.0 Part II. Neural Networks and Deep Learning, p393
- [23] IBM SPSS Modeler - Guide CRISP-DM, chapitre 1, p9
- [24] https://www.ibm.com/docs/fr/SS3RA7_18.3.0/pdf/ModelerCRISPDM.pdf chapitre 01 page 09
- [25] <https://www.labellerr.com/blog/a-guide-on-how-to-achieve-automated-data-labeling-with-machine-learning/>
- [26] Classification, Object Detection, Segmentation: **25** Khan, Asharul. (2020). Machine Learning in Computer Vision. Procedia Computer Science. 167. 10.1016/j.procs.2020.03.355.
- [27] <https://docs.ultralytics.com/fr/tasks/detect/>
- [28] Hands on ML with tensorflow2.0 chapitre 11, p494
- [29] Hands on ML with tensorflow2.0 Ensemble Learning and Random Forests, p286
- [30] <https://www.mdpi.com/2079-9292/13/6/1095>
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." arXiv preprint arXiv:1505.04597 (2015). Available at: <https://arxiv.org/abs/1505.04597>.
- [32] <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/>
- [33] <https://datasciencetoday.net/index.php/en-us/deep-learning>
- [34] https://www.researchgate.net/figure/Illustration-of-convolution-and-deconvolution-operations-convolution-operation_fig1_332591458
- [35] Mahdian, Babak & Saic, Stanislav. Using noise inconsistencies for blind image forensics. Image and Vision Computing
- [36] https://blog.51cto.com/u_13527/7957811
- [37] <https://www.cnblogs.com/wxl845235800/p/9965256.html>
- [38] <https://www.dpreview.com/forums/post/50795076>

- [39] <https://zestedesavoir.com/articles/1532/la-compression-jpeg/>
- [40] [https://fr.wikipedia.org/wiki/Sous-%C3%A9chantillonnage de la chrominance](https://fr.wikipedia.org/wiki/Sous-%C3%A9chantillonnage_de_la_chrominance)
- [41] https://www-igm.univ-mlv.fr/~dr/XPOSE2002/compression/default_fichiers/page0037.htm
- [42] H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," in IEEE Transactions on Information Forensics and Security, vol. 4, no. 1, pp. 154-160, March 2009, doi: 10.1109/TIFS.2008.2012215.
- [43] <https://www.semanticscholar.org/paper/Exposing-Digital-Forgeries-From-JPEG-Ghosts-Farid/6345039a2e574987a7fc4bb9032878812de598a7>
- [44] N. B. A. Warif, M. Y. I. Idris, A. W. A. Wahab and R. Salleh, "An evaluation of Error Level Analysis in image forensics).
- [45] https://www.researchgate.net/figure/An-example-of-artefacts-created-during-lossy-JPEG-compression-with-quality-q-60-Each_fig1_334360686
- [46] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts,"
- [47] <https://medium.com/swlh/image-demosaicing-bilinear-interpolation-vs-high-quality-linear-interpolation-5fd2268c4c7a>
- [48] https://www.researchgate.net/figure/Four-2-2-Bayer-CFA-patterns-a-Pat-1-G-1-R-2-B-3-G-4-b-Pat-2-G-1-B_fig1_365087695
- [49] <https://staff.utia.cas.cz/novozada/db/>
- [50] <https://pkorus.pl/downloads/dataset-realistic-tampering>
- [51] <https://drive.google.com/file/d/1IDUgcoUeonBxx2rASX-QwV9fhbtqdY8/view>
- [52] <https://www.vcl.fer.hr/comofod/download.html>
- [53] <https://defactodataset.github.io/>
- [54] <https://alumentations.ai/docs/>
- [55] Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. "Improving Neural Networks by Preventing Co-Adaptation of Feature
- [56] <https://www.semanticscholar.org/paper/R-%C3%A9seaux-de-neurones-convolutifs-pour-la-s-%C3%A9mantique-Fourure/2df1c3894781c44122247a68d1288fdb89317b7b>

- [57] Hands on ML with tensorflow2.0, CHAPITRE 11, p506
- [58] <https://numpy.org/devdocs/user/index.html#user>
- [59] https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- [60] https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [61] <https://matplotlib.org/stable/users/index.html>
- [62] <https://seaborn.pydata.org/tutorial/introduction.html>
- [63] <https://www.tensorflow.org/guide/basics?hl=fr>
- [64] <https://keras.io/guides/>
- [65] <https://docs.nvidia.com/cuda/>
- [66] <https://docs.python.org/3/library/shutil.html>
- [67] <https://tqdm.github.io/>
- [68] https://scikit-learn.org/stable/user_guide.html
- [69] Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946. Retrieved from <https://arxiv.org/abs/1905.11946>
- [70] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going Deeper with Convolutions. arXiv preprint arXiv:1409.4842. Retrieved from <https://arxiv.org/abs/1409.4842>
- [71] Hands on ML with tensorflow2.0, CHAPITRE 4, p235
- [72] <https://medium.com/@fraidoonomarzai99/overfitting-and-underfitting-regularization-l1-l2-drop-out-1e41a90265bd>
- [73] Hands on ML with tensorflow2.0, CHAPITRE 11, p506

ANNEXES

Concepts clés sur les Vidéos

1. Introduction

Cette annexe offre un aperçu technique de la vidéo, couvrant sa définition, ses caractéristiques et formats. Elle explore les types de falsifications vidéo (temporelles, spatiales, spatio-temporelles) et les outils de création de contenus falsifiés. Ces informations complètent l'analyse principale en fournissant un contexte technique approfondi.

2. C'est quoi une vidéo

Une vidéo est un format qui utilise des techniques pour capturer et représenter des images (frames) animées sous forme d'un signal électrique, ainsi que pour restituer ces images à l'aide de ce signal. Ce signal vidéo se caractérise par divers paramètres standardisés tels que la **cadence d'image**, la **résolution**, le **standard de couleur**, la **composition analogique ou numérique**, le **format d'image** et les données associées. Fondamentalement, une vidéo consiste en une succession de frames affichées à une certaine cadence, exploitant la capacité de l'œil humain à distinguer environ 20 images par seconde. En affichant plus de 20 images par seconde, la vidéo est capable de créer l'illusion d'une image animée, trompant ainsi l'œil humain.

Après avoir bien compris ce qu'est une vidéo, passons maintenant à une exploration plus approfondie de ses différentes **caractéristiques**. Nous examinerons le conteneur, les codecs, la fréquence d'images, la taille de l'image, le ratio d'aspect, et enfin, le débit binaire.

3. Caractéristique d'une vidéo

Pour comprendre pleinement les particularités d'une vidéo, il est crucial de se familiariser avec ses caractéristiques techniques fondamentales. Voici un aperçu des éléments clés qui déterminent la qualité et la performance d'une vidéo :

Conteneur : Le conteneur vidéo est l'enveloppe englobant tous les aspects d'une vidéo, incluant l'audio, la vidéo, les métadonnées, etc. Il organise ces éléments pour une lecture synchronisée. Tous les conteneurs peuvent stocker des données audio et vidéo, mais leur capacité à inclure des éléments supplémentaires comme les métadonnées ou les sous-titres peut varier. Ils sont distingués par des extensions de fichier telles que .mp4,

.mov et .avi, certains permettant des fonctionnalités additionnelles tandis que d'autres se limitent aux composants audios et vidéo.

Codecs : les codecs déterminent comment les vidéos et l'audio sont compressés, ce qui affecte la taille et la qualité du fichier. Les codecs sont différents pour la compression audio et la compression vidéo. Les codecs vidéo réduisent l'espace de stockage en compressant les fichiers, en fusionnant les données similaires, en minimisant les couleurs et en réduisant la résolution. Ils peuvent être « avec perte », ce qui entraîne une diminution de qualité, ou « sans perte », préservant toutes les informations, mais augmentant la taille du fichier.

Fréquence d'images : la fréquence d'images, affichée en images par seconde, décrit la fréquence à laquelle une nouvelle image s'affiche. Une fréquence d'images élevée produit généralement une vidéo plus fluide.

Taille de l'image : c'est la taille de chaque image individuelle mesurée en pixels. Elle est liée à la résolution et au ratio d'aspect.

Ratio d'aspect : Le ratio d'aspect fait référence à la largeur par rapport à la hauteur d'une vidéo. Les ratios d'aspect courants comprennent 4:3, 16:9 et 21:9.

Débit binaire : le débit binaire fait essentiellement référence à la qualité d'une vidéo, un débit binaire élevé produisant une vidéo de meilleure qualité.

Maintenant, passons à un autre élément clé des vidéos : **le format du vidéo**. La prochaine section donnera un aperçu complet des différents formats vidéo couramment utilisés.

4. Format du vidéo

Les formats vidéo sont des enveloppes numériques qui stockent et organisent les données audio et vidéo pour une lecture optimale. Chaque format a ses avantages et ses inconvénients en termes de compatibilité, de qualité et de taille de fichier. Le tableau A-1 compare entre les formats MP4, MOV et AVI :

Tableau A-1 : Comparaison des formats vidéo courants

Format	Description	Avantages	Inconvénients
MP4	Format vidéo le plus répandu, réputé pour sa compatibilité et sa capacité à stocker de l'audio, de la vidéo, des sous-titres, du texte et des images.	- Universellement pris en charge par les appareils numériques et les lecteurs multimédias. - Offre une vidéo de haute qualité avec une perte	- Facilité d'accès susceptible de favoriser le piratage et l'édition peut s'avérer difficile. - Problèmes de synchronisation audio possibles dus à des problèmes de codec.

		minimale lors de la compression.	
MOV	Format de fichier conçu par Apple, ressemblant beaucoup au format MP4, mais excelle sur les appareils Apple.	Admirés dans l'industrie cinématographique pour leur capacité multipiste, offrant une haute qualité et un montage facile, servant de norme professionnelle.	- Lisibilité limitée entre les appareils, les programmes Windows et taille de fichier importante en raison de leur haute qualité. - La compression entraîne une certaine perte de données qui affecte la qualité vidéo.
AVI	La réponse de Microsoft au MOV, introduit en 1995, fait partie des formats vidéo les plus anciens.	Bien adaptés aux vidéos courtes et à la télévision, compatibles avec les principaux systèmes d'exploitation et navigateurs Web, s'adaptent à d'autres formats et offrent une qualité audio-vidéo exceptionnelle, supportant même l'enregistrement sur DVD.	Nature obsolète et absence de données de sous-titres l'empêchent d'être pratique pour les applications vidéo modernes. Perte de qualité lors de la compression, ce qui le rend moins souhaitable pour les besoins contemporains.

Après avoir examiné en détail les vidéos, leurs caractéristiques et leurs formats courants, nous allons maintenant explorer un aspect crucial : **les différents types de falsifications vidéo**. Celles-ci se divisent principalement en **falsification temporelle (inter-frame)**, **falsification spatiale (intra-frame)**, et **falsification spatio-temporelle**.

5. Types de falsifications

Les types de manipulation vidéo se divisent en deux catégories principales : **Intra-frame et Inter-frame**. Les notions d'inter- et d'intra-frame distinguent principalement la falsification temporelle de la falsification spatiale. La falsification inter-frame est effectuée au niveau de la séquence : les pixels des frames individuelles ne sont pas modifiés, mais la séquence dans son ensemble est changée. La falsification intra-frame s'effectue au niveau du pixel : certaines régions spatiales sont modifiées, mais les modifications sont corrélées dans le temps pour former une région falsifiée convaincante.

Dans notre cas, nous avons deux types de fichiers différents : image et vidéo. Les types de manipulation dans les vidéos se divisent en trois grands blocs : **Falsification spatiale (intra-frame)**, **Falsification temporelle (inter-frame)** et **Falsification spatio-temporelle**. Pour les images, les types de falsifications se divisent également en deux blocs : **Active** et **Passive**. Il est à noter que les types de manipulation présents

dans la **Falsification spatio-temporelle** des vidéos sont les mêmes que ceux présents dans les falsifications **Active** des images.

La figure A-1 présente une classification des différents types de falsifications numériques pour les fichiers image et vidéo.

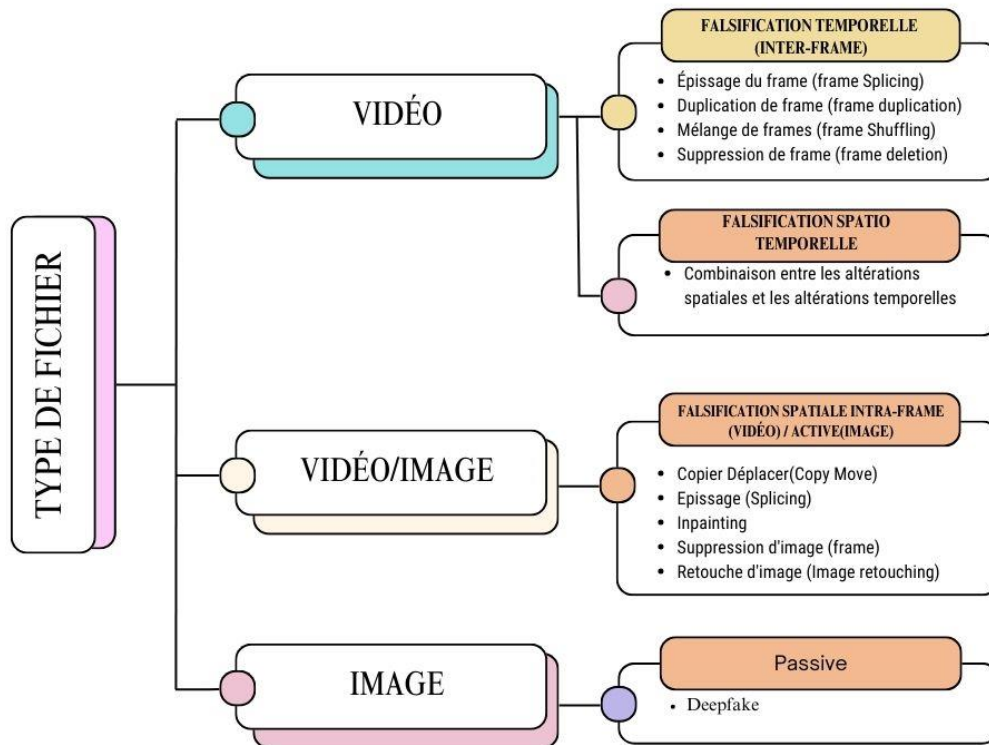


FIGURE A-1 : Types de manipulation pour les images et les vidéos

5.1 Falsification temporelle (inter-frame)

Épissage du frame (frame Splicing) : est une forme de manipulation vidéo, consiste à extraire des frames spécifiques, souvent centrées sur des éléments de premier plan (foreground), d'une vidéo et à les insérer dans une autre. Cette pratique vise à modifier le flux, la durée ou le contenu de la vidéo et constitue une méthode courante d'altération intra-image dans la vidéo numérique.

Duplication de frame : La falsification par duplication d'images est une forme courante de falsification dans le domaine des vidéos numériques, caractérisée par une manipulation inter-images se produisant dans le domaine temporel. Cette technique consiste à reproduire des images au sein d'une séquence vidéo et à les disperser dans différentes positions temporelles, soit pour masquer, soit pour reproduire certains événements décrits dans la séquence. Ses applications vont de la modification à la reproduction d'événements au sein d'une même séquence vidéo.

Pour illustrer concrètement cette technique de falsification, examinons deux séquences d'images extraites d'une vidéo de surveillance montrées dans la figure A-2. Ces exemples mettent en lumière comment la duplication de frames peut être utilisée pour altérer subtilement le contenu d'une séquence vidéo.

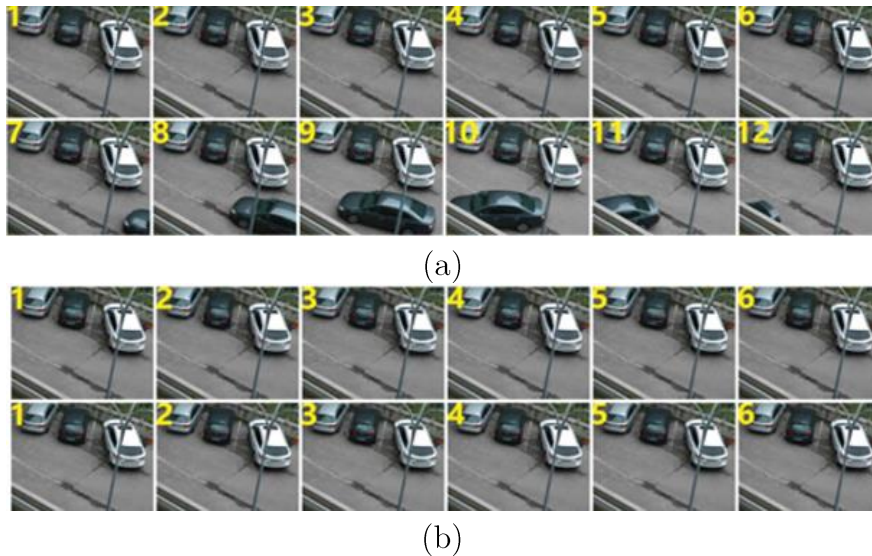


FIGURE A-2 : Duplication de frame

a) Séquence vidéo authentique b) Séquence vidéo manipulée. Cette figure montre une duplication de frames où les images 1 à 6 ont été copiées pour remplacer les images 7 à 12. Cette manipulation efface le passage d'une voiture dans le parking.

Mélange de frames (frame Shuffling) : est une technique couramment utilisée par les faussaires dans la manipulation de contenus vidéo numériques, dans le but de masquer ou d'accentuer certaines activités représentées dans les séquences. Cette pratique trompeuse consiste à mélanger ou à modifier l'ordre séquentiel des frames du vidéo, conférant ainsi une signification différente à la vidéo originale. Il s'agit essentiellement d'une variante de la duplication d'images, où les images répliquées sont temporairement réorganisées avant d'être réintégrées dans la séquence.

La figure A-3 illustre concrètement la définition du Mélange de frames en présentant un exemple de manipulation de l'ordre des frames dans une vidéo.

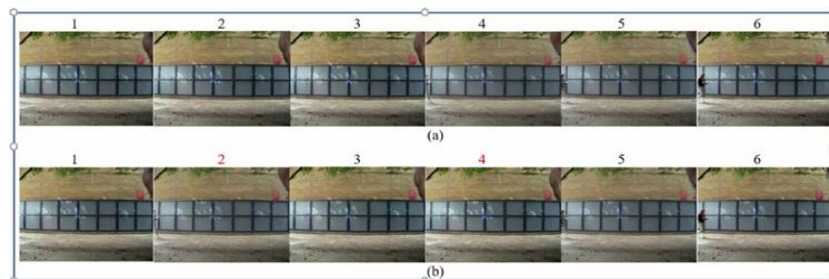


Figure A-3 : Falsification par mélange de frames

a) Séquence vidéo originale b) Séquence vidéo falsifiée après avoir effectué une falsification par répliation (la quatrième image est mélangée à la deuxième).

Suppression de frame (frame deletion) : Cette méthode consiste à supprimer certains frames d'une vidéo, souvent dans le but de fabriquer des preuves trompeuses concernant des activités illicites. Cette manipulation peut servir à cacher ou à simuler un contenu présent dans la séquence originale. Les frames peuvent être extraites de plusieurs segments ou concentrées dans une seule zone lors de ces attaques. Au fur et à mesure que les frames sont supprimés de la vidéo originale, le nombre total de frames diminue. Il existe plusieurs méthodes pour supprimer les images, y compris la suppression manuelle à l'aide d'un logiciel d'édition vidéo.

La figure A-4 illustre visuellement le concept de suppression de frames décrit dans la définition.

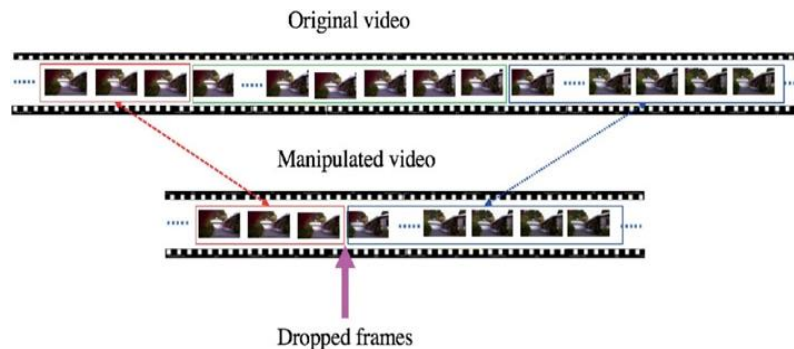


FIGURE A-4 : Illustration de la suppression de frames

Comparaison entre une vidéo originale et une vidéo manipulée, où la vidéo originale se compose de trois parties chaque partie contenant plusieurs frames, tandis que la vidéo manipulée est privée de la deuxième partie.

5.2 Falsification spatiale (Intra-frame) pour les vidéos

Comme spécifié précédemment, l'approche intra-frame implique la division d'une vidéo en plusieurs frames (images). Par conséquent, les types de manipulations effectuées sur les frames individuelles sont les mêmes que ceux appliqués aux images isolées.

5.2. Falsification spatio-temporelle :

La falsification spatio-temporelle est une technique sophistiquée de falsification vidéo, qui combine les altérations spatiales et temporelles d'un signal vidéo pour créer des manipulations convaincantes. Cette méthode combine la falsification inter-frame, qui implique des changements entre les frames, et la falsification intra-frame, qui cible des frames individuels. En combinant ces techniques, la séquence des frames est réorganisée et le contenu graphique de chaque frame est également manipulé dans la même vidéo.

Cette fusion permet de créer des altérations transparentes mais trompeuses, ce qui constitue un formidable défi pour la détection et l'analyse.

La figure A-5 montre clairement le concept de falsification spatio-temporelle dans une vidéo. À gauche, la séquence de frames représente une vidéo originale (A) où chaque frame capture un instant précis dans le temps et montre un personnage en mouvement (en bleu). À droite, on voit une version falsifiée (D) de cette vidéo. Par exemple, la frame 4 de la vidéo originale est altérée, le personnage est transformé (de bleu à rouge) et des objets peuvent être changés. Les axes X et Y représentent les dimensions spatiales de chaque frame, tandis que l'axe Temps indique la séquence des frames. Les altérations incluent des changements intra-frame et inter-frame. En réorganisant la séquence des frames et en manipulant le contenu graphique de chaque frame, la figure illustre comment la falsification spatio-temporelle peut créer des altérations subtiles mais trompeuses, rendant leur détection et leur analyse particulièrement difficiles.

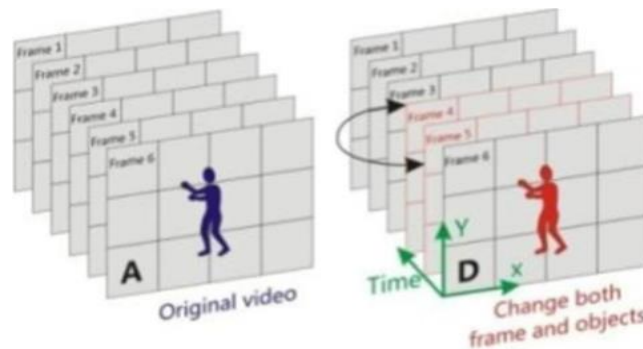


FIGURE A-5 : Un exemple d'une falsification spatio-temporelle

Après avoir identifié les types de falsifications, examinons maintenant les outils qui permettent de les créer. **Les outils de création de contenus falsifiés** se sont considérablement développés ces dernières années, rendant la production de faux de plus en plus accessible et sophistiquée. Examinons maintenant les principaux outils utilisés pour fabriquer ces contenus falsifiés.

6. Outils de Création de Contenus Falsifiés

Ces logiciels sont des outils puissants permettant de modifier, retoucher et manipuler des images et des vidéos, bien qu'ils aient des utilisations légitimes dans les domaines artistiques et professionnels, ils peuvent également être utilisés pour créer des contenus trompeurs ou falsifiés, il est important de comprendre leurs capacités pour mieux appréhender les défis de la détection de manipulations.

Lightworks : C'est un logiciel d'édition vidéo professionnel, reconnu pour sa puissance et sa précision dans le montage, ses caractéristiques principales sont le montage multi-caméras pour synchroniser et éditer des séquences de plusieurs sources, des outils de correction colorimétrique avancés, une large bibliothèque d'effets visuels et de

transitions, des capacités de mixage audio intégrées et le support de multiples formats vidéo, y compris la 4K, Il est capable de manipuler finement les séquences vidéo permet de créer des montages complexes qui peuvent altérer significativement le contexte ou le contenu original d'une vidéo.

Adobe Premiere Pro : C'est un autre logiciel d'édition vidéo de niveau professionnel, largement utilisé dans l'industrie du cinéma et de la télévision, ses fonctionnalités clés comprennent le montage non-linéaire avancé, l'intégration poussée avec d'autres logiciels Adobe, des outils d'étalonnage colorimétrique professionnels, des capacités de titrage et d'animation de texte sophistiquées et le support pour l'édition de contenus VR et 360°. Il est capable de manipuler des séquences vidéo de haute qualité, combinée à son intégration avec d'autres outils Adobe, en fait un outil puissant pour la création de contenu vidéo hautement modifié ou même entièrement fabriqué.

Ces logiciels, bien que conçus pour des usages professionnels légitimes, peuvent être détournés pour créer des contenus trompeurs d'une qualité exceptionnelle, leur sophistication croissante souligne l'importance de développer des outils de détection de manipulation tout aussi avancés (le but de notre projet) et de promouvoir une littératie médiatique critique auprès du grand public.

Annexe B

Méthodes conventionnelles à base de traitement d'image

1. Introduction

Dans le cadre d'un projet de fin d'études sur la détection de falsifications dans les images et vidéos, le CRD-GN a demandé l'intégration de méthodes classiques de traitement d'images. Ces méthodes sont essentielles pour permettre aux investigateurs d'effectuer une analyse visuelle approfondie avant d'utiliser des techniques plus avancées et automatisées.

L'objectif principal est de fournir aux investigateurs des outils puissants pour optimiser leur perception et leur compréhension des images à examiner. Ces méthodes couvrent un large éventail de fonctionnalités, allant de l'amélioration de la qualité visuelle à la transformation des informations de couleur, en passant par des outils de visualisation, de filtrage et de correction d'artefacts.

Les investigateurs pourront ajuster des paramètres tels que le contraste, la luminosité et la netteté, tout en explorant diverses représentations des données d'image. Des fonctionnalités comme le zoom, le rognage et la rotation faciliteront l'inspection approfondie de régions spécifiques. L'extraction des métadonnées EXIF (Exchangeable image file format) apportera également des informations précieuses sur l'historique de l'image.

Ces techniques permettront de détecter des indices ou anomalies subtiles nécessitant une inspection plus poussée. Elles constituent ainsi un ensemble d'outils préliminaires mais cruciaux, répondant aux exigences du CRD-GN et préparant le terrain pour des analyses plus complexes de détection de falsifications.

2. Différentes méthodes d'amélioration d'image

L'amélioration d'image dépend des caractéristiques spécifiques de l'image traitée et des objectifs visés par l'investigateur. Il existe une multitude de techniques, chacune adaptée à des besoins particuliers et à des types d'images spécifiques. Examinons à présent les techniques d'amélioration d'image les plus pertinentes pour notre étude, en nous concentrant sur leur application dans le domaine de l'analyse forensique d'images.

2.1 Amélioration du contraste

Commençons par l'ajustement du **contraste** qui se définit comme la différence de luminance ou de couleur entre les différentes parties d'une image. Il joue un rôle crucial dans la perception visuelle et la lisibilité des détails. Le contraste détermine la gamme dynamique de l'image, c'est-à-dire l'écart entre ses zones les plus claires et les plus sombres. Son réglage permet d'accentuer ou d'atténuer ces différences, modifiant ainsi significativement l'apparence et l'impact visuel de l'image, sans nécessairement altérer sa luminosité globale.

Cette définition nous amène à distinguer différents types de contraste, chacun influençant de manière unique l'aspect final de l'image. **Le contraste tonal**, applicable aux photos en couleur comme en noir et blanc, vise à capturer une large gamme de tons pour obtenir une image équilibrée. À l'opposé, on trouve **le fort contraste**, caractérisé par une vaste gamme allant des noirs profonds aux blancs éclatants, produisant des images aux couleurs vibrantes et aux textures marquées. Enfin, **le faible contraste** se distingue par une différenciation minimale entre les zones claires et sombres, créant des images douces où les objets se fondent davantage dans l'arrière-plan.

Ayant compris l'importance et les principes fondamentaux du contraste, il est temps d'aborder les différentes techniques utilisées pour l'améliorer. L'amélioration du contraste dans le traitement d'images vise à optimiser la visibilité des détails et des objets en ajustant les niveaux de luminosité des pixels. Ce processus implique une manipulation des valeurs d'intensité et leur redistribution sur une gamme plus étendue, accentuant ainsi les écarts de luminosité entre les éléments de l'image et leur environnement. Examinons maintenant les diverses méthodes employées pour parvenir à ce résultat et augmenter l'efficacité visuelle des images.

2.1.1 Histogramme d'une image

Un histogramme d'une image est une représentation graphique de la distribution des valeurs d'intensité des pixels de l'image. Il s'agit d'un outil essentiel en traitement d'images, permettant d'analyser la fréquence de chaque niveau d'intensité sur une échelle de 0 à 255 (pour une image en échelle de gris, comme indiqué dans la figure B-1) ou pour chaque canal de couleur dans une image couleur.

Pour une image RGB, l'histogramme peut être vu comme un tableau tridimensionnel, où chaque dimension correspond à l'une des composantes de couleur : rouge, vert et bleu. Concrètement, si une image a des dimensions a par b avec 3 canaux de couleur, elle peut être représentée comme un tableau 3D de taille $a \times b \times 3$. Chaque pixel est ainsi défini par trois valeurs d'intensité, une pour chaque composante de couleur. Cela signifie qu'une image de taille $a \times b$ contient un total de $a \times b$ pixels, chacun étant décrit par trois valeurs d'intensité correspondant aux trames rouge, verte et bleue.

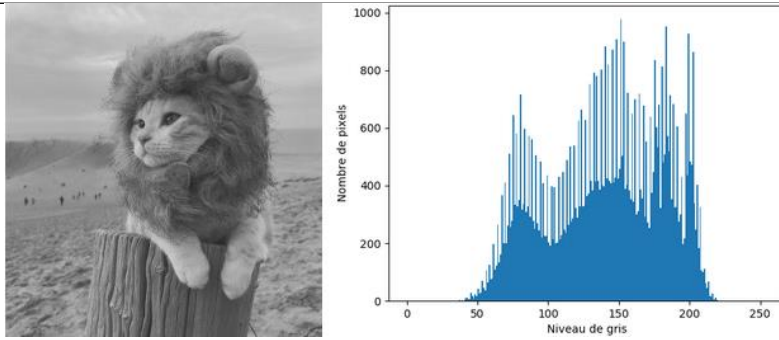


Figure B-1 : Histogramme d'une image en niveaux de gris

2.1.2 Égalisation d'histogramme

L'égalisation d'histogramme est une technique simple qui vise à améliorer le contraste des images en redistribuant les valeurs d'intensité les plus courantes dans l'histogramme. Cette méthode a généralement pour effet d'augmenter le contraste global des images lorsque les données utilisables sont représentées par des valeurs de contraste similaires, surtout lorsque l'image est caractérisée par une gamme étroite de valeurs d'intensité. En ajustant ainsi les intensités, on parvient à une répartition plus uniforme sur l'histogramme, exploitant ainsi toute la gamme des intensités de manière équilibrée. Ce processus permet aux zones de faible contraste local de bénéficier d'un contraste accru.

L'égalisation d'histogramme s'avère particulièrement efficace pour améliorer la qualité visuelle des images en niveaux de gris, ainsi que des images dont les arrière-plans et les premiers plans présentent à la fois des zones claires et sombres. Cependant, il convient de noter qu'elle peut entraîner une diminution du signal utilisable (la redistribution des intensités, comme illustré dans la figure B-2, de manière plus uniforme, peut parfois amplifier le bruit présent dans l'image ou réduire certains détails subtils).

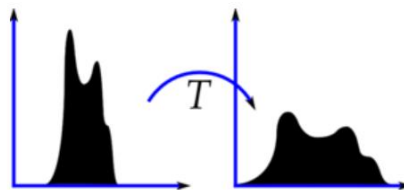


Figure B-2 : Histogramme d'une image avant et après l'égalisation [80]

L'histogramme d'une image est transformé de manière à répartir plus uniformément les intensités des pixels sur toute la plage des valeurs possibles

Algorithme d'égalisation :

L'algorithme d'égalisation d'histogramme suit une série d'étapes précises pour améliorer le contraste d'une image. Il commence par analyser la fréquence de chaque

niveau d'intensité dans l'image, établissant ainsi l'histogramme initial. Ensuite, il calcule la fonction de densité de probabilité (PDF) pour chaque valeur d'intensité, représentant la probabilité d'occurrence de chaque niveau. À partir de la PDF, l'algorithme dérive la fonction de densité cumulative (CDF), qui correspond à l'histogramme normalisé cumulé. Cette CDF est ensuite mise à l'échelle pour correspondre à la plage complète des intensités d'image (généralement de 0 à 255 sur 8-bits), et les valeurs résultantes sont arrondies à l'entier le plus proche [19]. Mathématiquement, pour une image en niveaux de gris, la PDF est exprimée par :

$$P_x(i) = P(x = i) = \frac{n_i}{n}; 0 \leq i < L \quad (\text{B.1})$$

Où : n_i est le nombre d'occurrences du niveau de gris i et n le nombre total de pixels.

La CDF est alors calculée en sommant ces probabilités comme suit :

$$cdf_x(i) = \sum_{j=0}^i P_x(x = j) \quad (\text{B.2})$$

Cette approche permet une redistribution des intensités qui exploite toute la gamme dynamique disponible, aboutissant à une amélioration significative du contraste global de l'image.

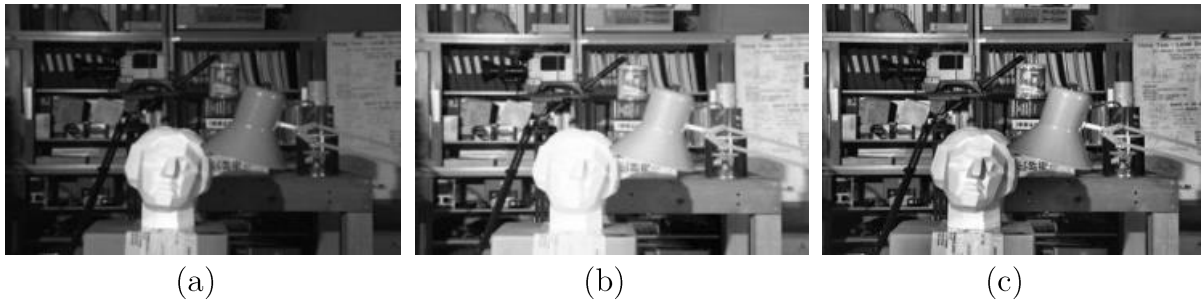
Passons maintenant à une méthode plus sophistiquée de l'égalisation :

2.1.3 Contrast Limited Adaptive Histogram Equalization (CLAHE)

C'est une technique avancée d'amélioration d'image qui combine deux concepts clés : **l'égalisation adaptative d'histogramme (AHE)** et **la limitation de contraste**. L'AHE divise l'image en tuiles rectangulaires, calculant un histogramme distinct pour chacune. Cette approche permet d'améliorer le contraste local plutôt que global, en adaptant la redistribution des valeurs de luminosité à chaque section de l'image. Pour chaque tuile, une fonction de distribution cumulative (CDF) et une fonction de transformation sont calculées, s'appliquant principalement aux pixels centraux. Cette méthode est particulièrement efficace pour améliorer la définition des contours et le contraste local dans diverses régions de l'image.

La composante de limitation de contraste du CLAHE vient compléter l'AHE en atténuant le problème d'amplification du bruit. CLAHE limite cette amplification en recadrant (coupant) l'histogramme à une valeur prédéfinie avant de calculer la fonction de distribution cumulative (CDF). La valeur à laquelle l'histogramme est recadré, le soi-disant limiteur de recadrage (clip limit), dépend de la normalisation de l'histogramme et donc de la taille de la région de voisinage. Si un compartiment d'histogramme est au-dessus de la limite de contraste spécifiée (contrast limit), ces pixels sont recadrés et distribués uniformément vers d'autres compartiments avant d'appliquer l'égalisation de

l'histogramme. Après l'égalisation, pour supprimer les artefacts dans les bords des tuiles, une interpolation bilinéaire est appliquée. Voici une figure B-3 comparative entre les résultats de l'application de ces 2 techniques pour l'amélioration du contraste :



FigureB-3 : a) Image originale b) Image avec égalisation d'histogramme c) Image avec CLAHE.

Le contraste de l'arrière-plan a été amélioré dans la figure (b) après l'égalisation de l'histogramme. Cependant, le visage de la statue a perdu des détails à cause de la surbrillance. Pour remédier à cela, nous avons utilisé l'égalisation adaptative de l'histogramme (c).

Une fois le contraste ajusté, nous pouvons nous attaquer à un autre aspect clé : **la luminosité**.

2.2 Ajustement de la luminosité

L'ajustement de la luminosité ou bien « Brightness » est une technique fondamentale qui permet de corriger les niveaux de brillance d'une image trop sombre ou trop claire. Une luminosité adéquate est cruciale pour révéler les détails. D'abord, on va commencer par définir la luminosité.

La luminosité d'une image fait référence à l'intensité globale de la lumière présente dans l'image. Elle détermine le degré de clarté ou d'obscurité de l'image dans son ensemble. Le réglage de la luminosité consiste à modifier le niveau global de luminance de l'image sans altérer de manière significative le contraste ou l'équilibre des couleurs. On se réfère à l'intensité perçue de la lumière dans une image. Il s'agit d'un attribut subjectif influencé par le niveau de luminance moyen de l'ensemble de l'image. Pour illustrer l'impact de la luminosité sur une image, considérons l'exemple suivant. La figure B-4 montre une image originale (gauche) et une version modifiée avec un réglage de luminosité (droite).

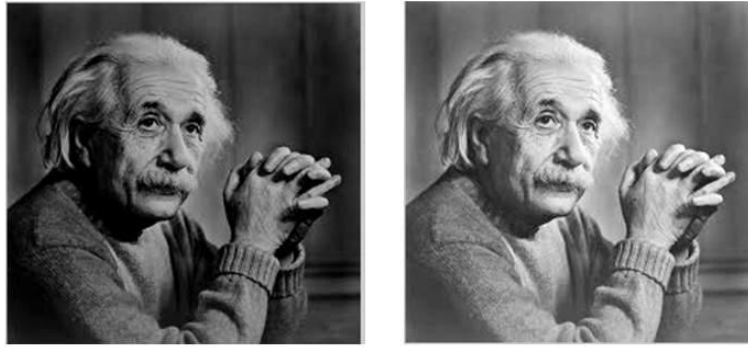


Figure B-4 : L'image de droite est plus claire que l'image de gauche (en termes de luminosité)

La luminosité peut être simplement augmentée ou diminuée par des simples opérations mathématiques appliquées à la matrice de l'image :

- **Ajustement additif** : Augmenter ou diminuer les valeurs des pixels uniformément sur tous les canaux pour ajuster la luminosité. Cette méthode est simple mais peut entraîner une perte de détails dans les hautes lumières ou les ombres si elle est appliquée de manière excessive.
- **Ajustement multiplicatif**: Mettre à l'échelle les valeurs des pixels en fonction d'un facteur pour ajuster la luminosité. L'ajustement multiplicatif préserve le contraste relatif de l'image tout en modifiant son niveau de luminosité global.
- **En combinant judicieusement ces deux techniques**, il est possible d'obtenir un contrôle fin sur la luminosité d'une image, tout en minimisant les artefacts indésirables. Dans notre cas, nous les combinerons en une seule formule automatisée. Cette méthode utilise deux paramètres clés : **alpha (α) pour le contraste (gain) et bêta (β) pour la luminosité (biais)**. L'expression globale peut s'écrire :

$$g(i,j) = \alpha \cdot f(i,j) + \beta \quad (\text{B.3})$$

Où : $f(i,j)$ représente l'intensité du pixel original et $g(i,j)$ l'intensité du pixel de sortie ajusté.

Pour calculer automatiquement les valeurs optimales de **alpha (α) et bêta (β)**, nous analysons l'histogramme de l'image. L'optimisation vise à ce que la plage de sortie se situe dans l'intervalle [0...255]. Nous calculons la distribution cumulée pour déterminer les seuils inférieurs et supérieurs en ignorant les fréquences de couleurs inférieures à un certain pourcentage (disons 1%). Cela nous donne les limites minimales et maximales ajustées de l'histogramme. Alpha est alors calculée en divisant l'écart souhaité (255) par l'écart réel des niveaux de gris après écrêtage :

$$\alpha = \frac{255}{\text{maximum} - \text{minimum}} \quad (\text{B.4})$$

Bêta est ensuite obtenu en résolvant l'équation pour :

$$g(i, j) = 0 \text{ avec } f(i, j) = \text{minimum} : \beta = -\text{minimum} \cdot \alpha \quad (\text{B.5})$$

Cette méthode améliore automatiquement le contraste et la luminosité en étirant la plage des niveaux d'intensité tout en préservant les détails importants de l'image comme indique la figure B-5.

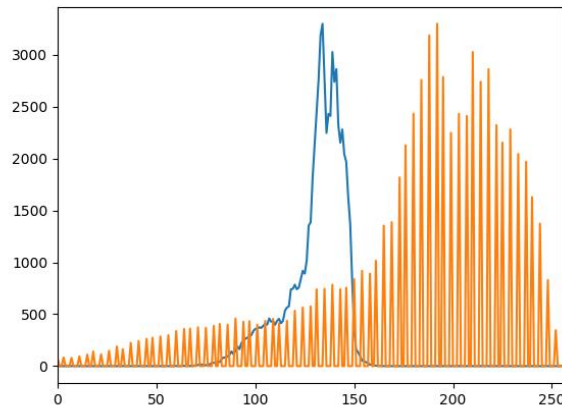


Figure B-5 : En orange la plage des niveaux d'intensité après ajustement en bleu

- **Correction gamma** : ça consiste à Ajuster la valeur gamma de l'image pour contrôler la luminosité des tons moyens. Les valeurs gamma inférieures (< 1) augmentent la luminosité, tandis que les valeurs supérieures (> 1) la diminuent. Cette méthode permet un réglage non linéaire de la luminosité. La formule est la suivante:

$$V_{out} = A \cdot V_{in}^{\gamma} \quad (\text{B.5})$$

Où : la valeur d'entrée réelle non négative V_{in} est élevée à la puissance γ et multipliée par la constante A pour obtenir la valeur de sortie V_{out} . Dans le cas courant de $A = 1$, les entrées et les sorties sont généralement comprises entre 0 et 1.

Le tableau B-1 résume les avantages de ces méthodes et leurs inconvénients :

Tableau B-1 : Les avantages et les inconvénients des méthodes d'ajustement de la luminosité.

Méthode	Avantages	Inconvénients
---------	-----------	---------------

Ajustement additif	<ul style="list-style-type: none"> - Simple à mettre en œuvre. - Rapide et efficace. - Permet d'ajuster uniformément la luminosité de l'image. 	<ul style="list-style-type: none"> - Peut entraîner une perte de détails dans les hautes lumières ou les ombres si appliqué de manière excessive. - Les résultats peuvent sembler moins naturels par rapport aux méthodes plus complexes.
Ajustement multiplicatif	<ul style="list-style-type: none"> - Préserve le contraste relatif de l'image. - Offre un contrôle plus fin sur le niveau de luminosité global. 	<ul style="list-style-type: none"> - - Peut modifier le contraste global de manière inattendue si le facteur est mal choisi. - Peut nécessiter un ajustement minutieux pour obtenir les résultats souhaités.
Combinaison des deux méthodes	<ul style="list-style-type: none"> - Offre un contrôle plus précis sur la luminosité et le contraste. - Permet de réduire les effets indésirables tels que les pertes de détails dans les zones de hautes lumières et d'ombres. 	<ul style="list-style-type: none"> - Requiert un réglage minutieux des paramètres α et β pour éviter les artefacts et préserver les détails de l'image. - Peut être plus complexe à implémenter par rapport aux méthodes individuelles
Correction gamma	<ul style="list-style-type: none"> - Permet un réglage non linéaire de la luminosité. - Contrôle efficace de la luminosité des tons moyens. - Peut améliorer le contraste dans certaines plages de luminosité. 	<ul style="list-style-type: none"> - Peut introduire des distorsions ou des artefacts si la valeur gamma est mal choisie. - Nécessite une compréhension approfondie de la relation entre la valeur gamma et la luminosité des tons dans l'image. - Les modifications peuvent ne pas être intuitives pour les utilisateurs non familiers avec le concept de valeur gamma.

A titre d'exemple, des figures B-7 qui illustrent vraiment les différences entre les méthodes implémentées et appliquées sur la figure B-6.



Figure B-6: Image originale avec faible luminosité [81]

La figure ci-dessous présente quatre images résultant de l'application de différentes techniques d'amélioration. En haut à gauche, on peut observer le résultat de la correction gamma. En haut à droite, on voit le résultat de l'ajustement multiplicatif. En bas à gauche, on aperçoit le résultat de l'ajustement additif. Enfin, en bas à droite, on constate le résultat de l'égalisation d'histogramme. Cette dernière méthode semble offrir le meilleur résultat dans ce cas particulier, car elle cherche toujours à trouver un équilibre entre le contraste et la luminosité. Toutefois, il est important de noter que la méthode qui donne le meilleur résultat peut varier en fonction de l'image d'origine et des préférences personnelles en matière d'apparence visuelle.

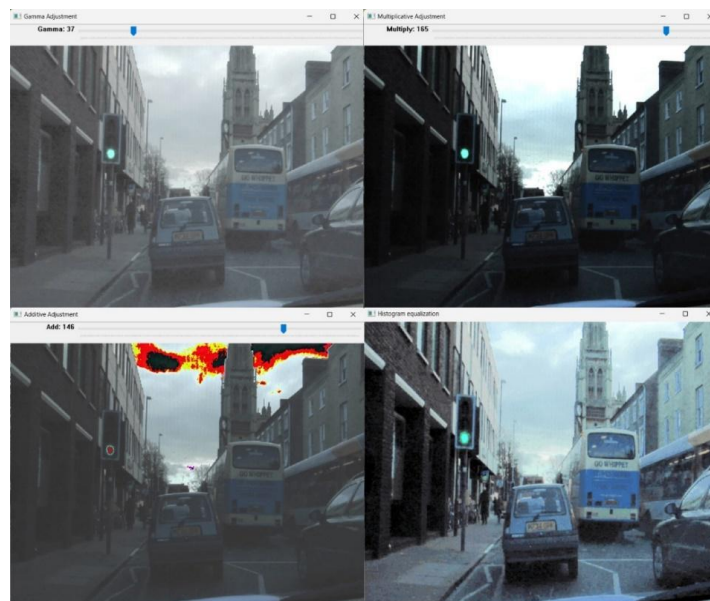


Figure B-7 : Images résultantes après application des différentes méthodes

Après avoir examiné les méthodes d'amélioration du contraste et de la luminosité, nous pivotons maintenant vers une analyse plus approfondie des techniques de détection des contours et de renforcement de la netteté. Cette transition marque un passage de l'ajustement global de l'image à une exploration plus ciblée et détaillée, où chaque bord et chaque détail sont scrutés avec une précision accrue.

2.3 Détection des contours (Edge detection)

Dans le cadre de l'analyse forensique d'images, la détection des contours revêt une importance cruciale. **Un contour, ou edge en anglais**, représente un changement soudain ou une discontinuité marquée dans l'intensité des pixels d'une image. Ces transitions significatives peuvent être considérées comme des frontières visuelles délimitant différentes régions ou objets au sein de l'image.

Nous pouvons distinguer plusieurs types de contours, notamment **les contours horizontaux, verticaux et diagonaux** (illustrés dans la figure B-8). Ces changements brusques d'intensité dans une direction particulière permettent de mettre en évidence les formes, les structures et les détails essentiels d'une scène.

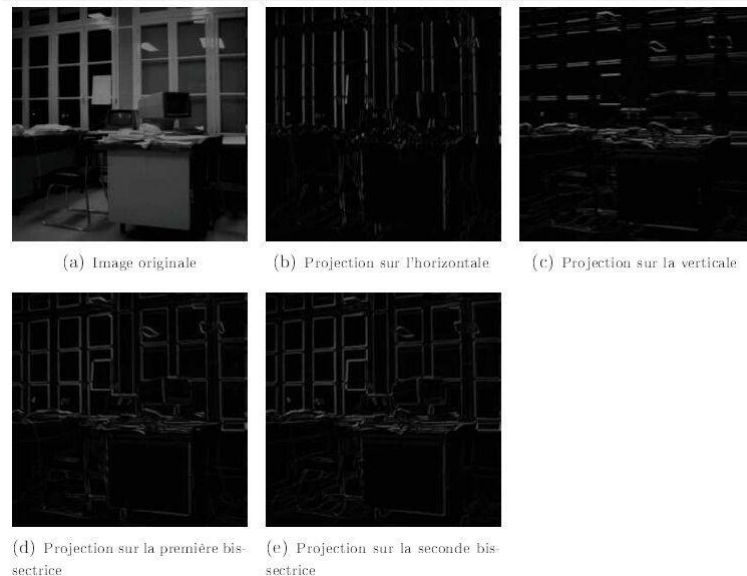


Figure B-8: Différents types de contours

Pour un analyste forensique, la détection efficace des contours est cruciale pour plusieurs raisons.

- Tout d'abord, elle facilite **l'identification et l'extraction d'éléments d'intérêt**, tels que des objets, des textures ou des motifs spécifiques. En isolant ces caractéristiques, l'analyste peut mieux les examiner et les comparer à d'autres éléments de preuve.
- De plus, la détection des contours peut **révéler des informations cachées ou subtiles** qui pourraient passer inaperçues lors d'une simple inspection visuelle. Par exemple, des incohérences dans les contours pourraient indiquer une manipulation ou une altération de l'image.
- Enfin, l'analyse des contours peut fournir des indices précieux sur la source, la provenance ou l'authenticité d'une image, en permettant **d'identifier des signatures ou des artefacts spécifiques liés à certains dispositifs d'imagerie ou techniques de traitement**.

Après avoir souligné l'importance cruciale de la détection des contours dans le cadre de l'analyse forensique d'images, il est essentiel d'explorer les différentes approches et techniques permettant d'effectuer cette tâche de manière efficace. Il existe plusieurs méthodes et masques pour la détection des contours, chacun ayant ses propres

caractéristiques et avantages. Voici quelques-uns des masques les plus courants que nous allons utiliser et comparer :

2.3.1 Opérateur de Prewitt

Cet opérateur utilise deux masques de convolution distincts pour détecter les contours horizontaux et verticaux d'une image (figure B-9). Il calcule les dérivées dans ces deux directions afin de mettre en évidence les changements brusques d'intensité caractéristiques des contours.

<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td style="background-color: #cccccc;">0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table> <p>(a) M_x</p>	1	0	-1	1	0	-1	1	0	-1	<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td style="background-color: #cccccc;">0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table> <p>(b) M_y</p>	1	1	1	0	0	0	-1	-1	-1
1	0	-1																	
1	0	-1																	
1	0	-1																	
1	1	1																	
0	0	0																	
-1	-1	-1																	

Figure B-9 : Masques de détections de contours horizontaux et verticaux de Prewitt

2.3.2 Opérateur de Sobel

Très similaire à Prewitt, l'opérateur de Sobel est également un masque dérivé utilisé pour la détection de contours. Il calcule lui aussi les contours horizontaux et verticaux, mais introduit un léger lissage en pondérant davantage les pixels centraux (figure B-10), le rendant ainsi un peu plus robuste au bruit que Prewitt.

-1	0	+1		+1	+2	+1
-2	0	+2		0	0	0
-1	0	+1		-1	-2	-1
G_x				G_y		

Figure B-10 : Masques de détections de contours horizontaux et verticaux de Sobel. On peut bien remarquer la pondération des pixel centraux

2.3.3 Masques de boussole de Robinson

Aussi connus sous le nom de masques directionnels, ces masques consistent à appliquer un seul et même masque en le faisant tourner dans les huit directions principales de la boussole afin de calculer les contours dans chacune de ces directions.

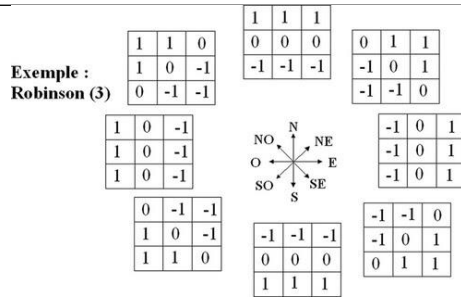


Figure B-11 : Les 8 masques de détections de contours de Robinson

2.3.4 Masques de Kirsch

Les masques de Kirsch fonctionnent sur le même principe que les masques de Robinson. Ce sont également des masques dérivés utilisés pour détecter les contours, mais dans toutes les directions cette fois.

$$\begin{aligned}
 \mathbf{K1} &= \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \mathbf{K2} &= \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \mathbf{K3} &= \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \mathbf{K4} &= \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 \mathbf{K5} &= \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \mathbf{K6} &= \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \mathbf{K7} &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \mathbf{K8} &= \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
 \end{aligned}$$

Figure B-12 : Les 8 masques de détections de contours de Kirsch

2.3.5 Opérateur Laplacien:

Le filtre Laplacien se distingue en tant que masque dérivé de second ordre. Ce filtre a pour objectif de détecter les contours dans les directions horizontale et verticale simultanément, sans nécessiter d'applications séparées pour chaque direction. La dérivée de second ordre permet de distinguer si les changements observés dans l'image résultent d'une variation de pixel dans des régions continues ou s'ils proviennent d'un contour.

Le noyau Laplacien général (L) se caractérise par une valeur positive au centre entourée de valeurs négatives disposées en forme de croix (On distingue le Laplacien positif du négatif selon le signe des coefficients du masque.), avec une somme de toutes les valeurs du filtre égale à zéro. Il peut être représenté par la matrice suivante présenté dans la figure B-13 :

0	1	0
1	-4	1
0	1	0

Figure B-13 : Le masque du laplacien

En considérant l'image comme une fonction de deux variables, $f(x,y)$, nous devons prendre en compte les dérivées partielles secondes le long des deux axes spatiaux. Le Laplacien de l'image est donné par :

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (\text{B.6})$$

Avec :

$$\nabla f = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y} \quad (\text{B.7})$$

Où : $\partial^2 f / \partial x^2$, $\partial^2 f / \partial y^2$, $\partial f / \partial x$ et $\partial f / \partial y$ représentent les dérivées partielles secondes et premières de f par rapport à x et y , respectivement. Après développement de cette formule pour le cas discret (cas d'images), on aura :

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (\text{B.8})$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (\text{B.9})$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)] \quad (\text{B.10})$$

Où : $f(x,y)$ représente la valeur du pixel central.

2.3.6 Méthode de Canny

C'est une méthode très populaire et efficace pour la détection de contours. C'est un processus multi-étape comprenant: **un lissage** (par exemple avec un filtre gaussien) pour réduire le bruit et éviter les faux contours, **le calcul des gradients à chaque pixel** pour mettre en évidence les transitions, une **suppression des non-maximums** pour éliminer les arêtes qui ne sont pas de vrais contours, et **un double seuillage** conservant uniquement les contours forts et les contours faibles connectés aux forts (comme indiqué dans la figure B-14). Bien que plus complexe, Canny offre d'excellents résultats en détectant précisément les vrais contours tout en supprimant la plupart des contours indésirables.

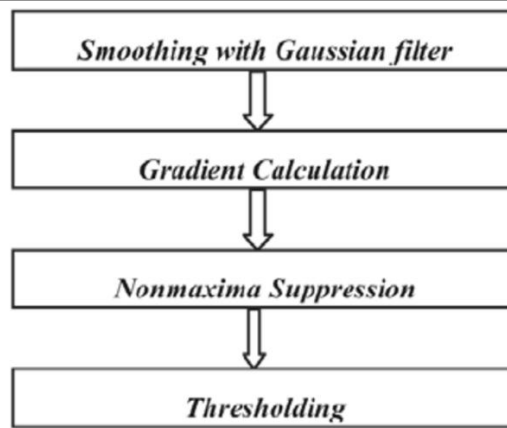


Figure B-14 : Le pipeline de la méthode de Canny

Après avoir appliqué différentes méthodes de détection de contours sur notre image d'intérêt, nous obtenons des résultats variés en fonction des approches utilisées. La figure B-15 illustre les résultats obtenus pour chaque méthode.

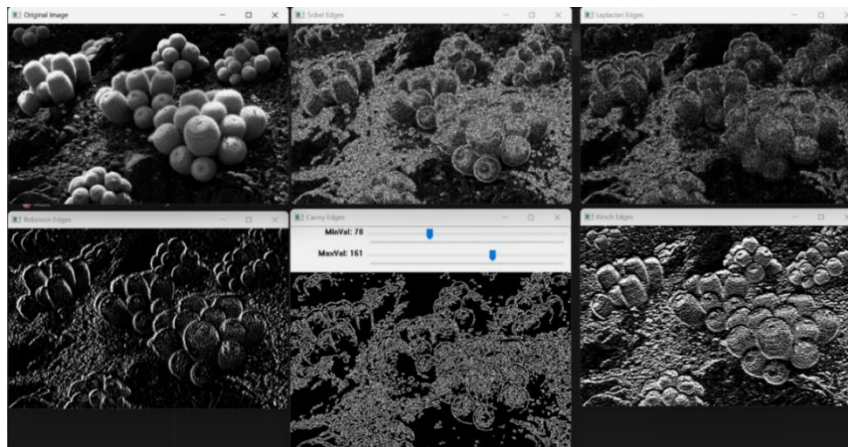


Figure B-15 : Comparaison des techniques de détection de contours sur une image de structures fongiques

Les résultats incluent de gauche à droite et de haut en bas: image originale, contours Sobel, Laplacien, Robinson, Canny et Krisch. Chaque méthode révèle des aspects différents. Sobel et Laplacien accentuent les bords, Krisch préserve mieux la texture, et Canny offre des contours nets et précis

Afin de mieux comprendre les performances et les caractéristiques de chaque méthode de détection de contours, le tableau B-2 comparatif est présenté ci-dessous. Ce tableau met en évidence les avantages et les inconvénients de chaque approche, ainsi que leurs principales applications.

Tableau B-2 : Les avantages et les inconvénients des méthodes d'ajustement de la luminosité

Méthode	Avantages	Inconvénients
Opérateur de Prewitt	<ul style="list-style-type: none"> - Simplicité de mise en œuvre. - Identifie les contours horizontaux et verticaux. - Utile pour des applications nécessitant une détection rapide des contours 	<ul style="list-style-type: none"> - Sensible au bruit, peut produire des contours indésirables. - Moins efficace pour les contours obliques ou diagonaux.
Opérateur de Sobel	<ul style="list-style-type: none"> - Légère réduction du bruit grâce à un léger lissage. - Identifie les contours horizontaux et verticaux. - Implementation relativement simple. 	<ul style="list-style-type: none"> - Sensible au bruit et peut également produire des contours indésirables. - Moins efficace pour les contours obliques ou diagonaux.
Masques de boussole de Robinson	<ul style="list-style-type: none"> - Détection des contours dans huit directions principales. - Simple à mettre en œuvre. - Utile pour une détection initiale des contours 	<ul style="list-style-type: none"> - Détection moins précise que d'autres méthodes plus complexes. - Risque de détecter des contours indésirables
Masques de Kirsch	<ul style="list-style-type: none"> - Détection des contours dans toutes les directions. - Utilisation pratique pour une détection générale des contours. 	<ul style="list-style-type: none"> - Risque de fausses détections de contours.
Opérateur Laplacien	<ul style="list-style-type: none"> - Détection des contours grâce aux discontinuités d'intensité. - Sensible aux variations brusques d'intensité. - Peut être utilisé pour des détections générales de contours 	<ul style="list-style-type: none"> - Risque de faux positifs en présence de bruit ou de faibles contrastes
Méthode de Canny	<ul style="list-style-type: none"> - Excellente suppression du bruit et des faux contours grâce à un processus multi-étape. - Précision élevée dans la détection des vrais contours. - Suppression efficace des contours indésirables. 	<ul style="list-style-type: none"> - Plus complexe à mettre en œuvre. - Peut nécessiter un ajustement minutieux des paramètres de seuillage pour obtenir les meilleurs résultats.

	- Sélection précise des contours forts et connectés.	
--	--	--

Remarque :

L'application d'un filtre à une image implique une opération de **convolution** avec un noyau de taille $n \times n$.

La convolution est une opération mathématique fondamentale dans le traitement d'images numériques. Elle consiste à appliquer un filtre, également appelé noyau ou masque, à chaque pixel de l'image pour produire une nouvelle image transformée.

Mathématiquement, pour une image 2D $f(m,n)$ et un noyau $h(m,n)$, la convolution est définie comme :

$$y[m, n] = x[m, n] * h[m, n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i, j] \cdot h[m - i, n - j] \quad (\text{B.11})$$

Où : représente l'opération de convolution, et les sommes sont effectuées sur tous les pixels (i,j) du voisinage défini par le noyau.

Pratiquement, le processus de convolution en traitement d'images peut être visualisé comme un noyau qui **glisse** méthodiquement sur l'image, se centrant sur chaque pixel tour à tour. À chaque position, une opération précise s'effectue : les valeurs du noyau sont multipliées avec les valeurs correspondantes des pixels de l'image environnants. Ces produits sont ensuite additionnés, générant une nouvelle valeur pour le pixel central. Cette séquence d'opérations est répétée systématiquement pour chaque pixel de l'image, transformant ainsi l'intégralité de celle-ci selon les caractéristiques du noyau utilisé.

Alors que la détection des contours révèle les contours et les délimitations, la Netteté vient affiner et accentuer ces détails, apportant une clarté supplémentaire à l'image.

2.4 Netteté (Sharpening)

L'accentuation de la netteté, ou image sharpening, est une technique d'amélioration des images numériques qui vise à rendre les détails plus nets et les contours plus précis. Cette méthode permet de mettre en valeur des éléments spécifiques tels que du texte, des logos ou des visages, et peut également contribuer à la réparation d'images anciennes ou endommagées en supprimant la décoloration, la poussière et les rayures. L'accentuation de la netteté améliore le réalisme et l'attrait visuel en ajoutant du contraste, de la profondeur et de la texture, et peut faciliter d'autres opérations de traitement d'images comme la reconnaissance, la classification et la segmentation.

Le fonctionnement de l'accentuation repose sur l'**augmentation de la différence entre les pixels proches des contours ou des frontières d'une image et les pixels des zones environnantes**, créant ainsi l'illusion de contours plus nets. Il existe plusieurs méthodes d'ajustement de la netteté comme le sharpening Laplacien (qu'on a détaillé dans la partie de détection de contours), qui applique un filtre Laplacien calculant la dérivée de second ordre de l'image et ajoute le résultat à l'image originale. Un exemplaire dans la figure B-16 illustre bien la différence :

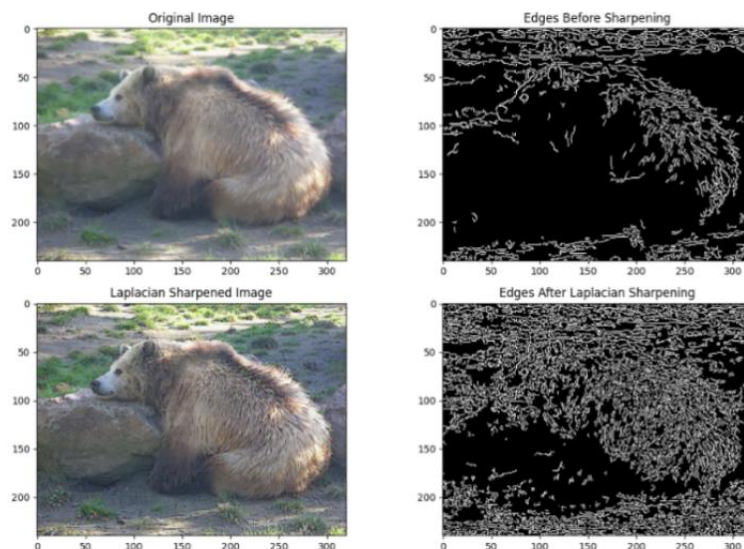


Figure B-16 : Comparaison de la netteté d'une image avant et après l'application du filtre de Laplace

Les contours sont plus clairs après l'application du filtre de Laplace.

Après avoir exploré diverses techniques d'amélioration de l'image, il est essentiel d'aborder une autre étape cruciale dans le traitement des images : la réduction du bruit. Le « dé bruitage joue un rôle fondamental pour éliminer les interférences et les

imperfections visuelles. Cette technique permet d'obtenir des images plus claires et nettes, améliorant ainsi leur qualité et leur utilité pour des analyses plus précises.

2.5 Dé bruitage d'images (Image Denoising)

Le dé bruitage d'images est une tâche qui consiste à supprimer le bruit présent dans une image. Le bruit peut être introduit lors de l'acquisition ou du traitement de l'image, et peut dégrader sa qualité visuelle en la rendant difficile à interpréter [19]. Les techniques de débruitage visent à restaurer la qualité d'origine de l'image en réduisant ou éliminant le bruit, tout en préservant les détails importants. Pour réduire le bruit sans affecter les détails visuels, trois types de signaux doivent être ciblés par le dé bruitage :

- La lumière diffuse réfléchie dans toutes les directions.
- Les réflexions spéculaires dans une direction particulière.
- Les ombres de sources lumineuses infinies comme la lumière du soleil.

Un dé bruitage efficace permettra d'améliorer nettement la qualité visuelle d'une image bruitée, facilitant son analyse et son interprétation.

Il existe deux grandes catégories de bruit affectant les images numériques : **le bruit additif et le bruit multiplicatif**. Le bruit additif suit la règle :

$$W(x, y) = s(x, y) + n(x, y) \quad (\text{B.12})$$

Où : $s(x, y)$ représente le signal ou l'image d'origine, $n(x, y)$ le bruit introduit, et $W(x, y)$ l'image corrompue.

Des exemples de bruit additif incluent le bruit gaussien, uniforme ou « sel et poivre ». Le bruit multiplicatif, quant à lui, suit la règle :

$$W(x, y) = s(x, y) \cdot n(x, y) \quad (\text{B.13})$$

Comme dans le cas du bruit de mode (« speckle noise »).

Le bruit gaussien est un bruit aléatoire uniformément distribué sur le signal. Chaque pixel de l'image bruitée est la somme de la valeur réelle du pixel et d'une valeur aléatoire suivant une distribution gaussienne. Cette distribution gaussienne a une forme de cloche caractéristique, définie mathématiquement par sa fonction de densité de probabilité. Le bruit gaussien peut être causé par de nombreux phénomènes naturels intervenant lors de la capture et de la numérisation de l'image.

La figure B-17 illustre à gauche diverses courbes de distribution gaussienne, et à droite, une représentation d'une distribution uniforme de probabilité.

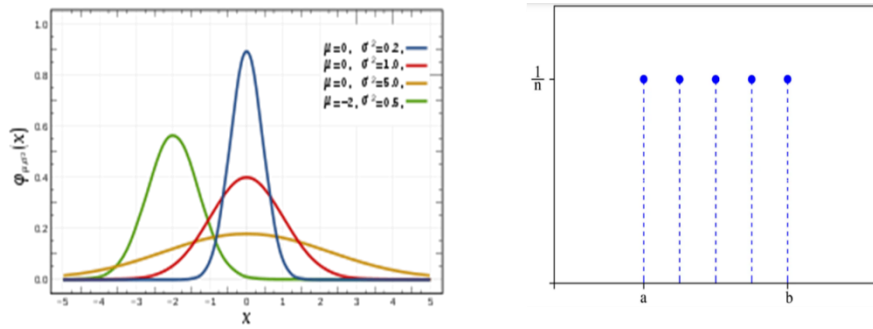


Figure B-17 : distributions de probabilités, gaussienne et uniforme

Le bruit « poivre et sel » est quant à lui un bruit impulsif n'ayant que deux valeurs possibles, correspondant à des erreurs de transmission ou de défauts dans les capteurs. Les pixels corrompus sont alternativement remplacés par la valeur minimale (poivre) ou maximale (sel), donnant à l'image un aspect granuleux caractéristique. Sa distribution de probabilité présente des pics marqués aux valeurs extrêmes, conformément à sa nature impulsive.

La figure B-18 représente les différents types de bruit que l'on peut avoir.

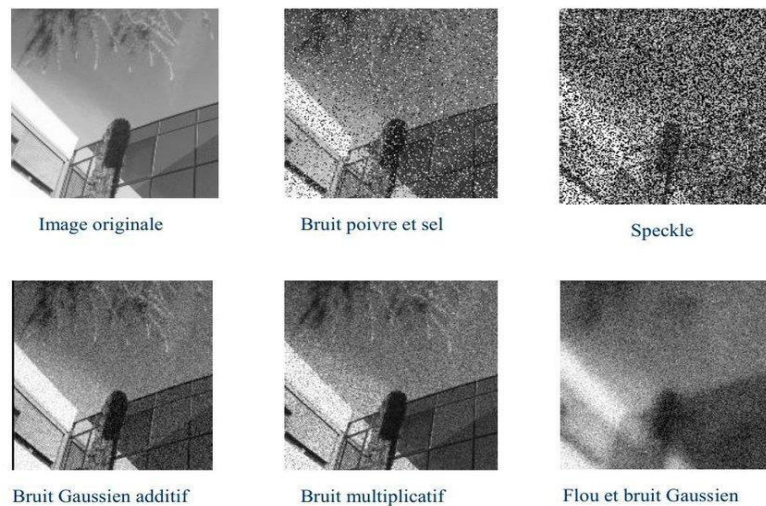


Figure B-18 : différents types de bruit

D'autres types de bruits existent, comme le bruit brownien ou le bruit de scène, chacun avec ses propres caractéristiques. La compréhension de ces différentes formes de bruit est cruciale pour pouvoir appliquer les techniques de réduction de bruit les plus adaptées lors de l'analyse d'images numériques.

Plusieurs approches et méthodes ont été développées pour relever ce défi du débruitage d'images. Parmi les plus courantes, on retrouve **les filtres linéaires** tels que les filtres **moyenneurs** ou **gaussiens**, qui lissent l'image en moyennant les valeurs des pixels voisins. D'autres techniques **non-linéaires** comme les **filtres médians** ou **bilatéraux** permettent de mieux préserver les contours. Les **méthodes d'analyse**

multi résolution comme les **ondelettes** offrent également des outils puissants pour séparer le signal du bruit. Enfin, **des approches plus avancées basées sur l'apprentissage profond** ont récemment montré d'excellentes performances en matière de dé bruitage tout en préservant fidèlement les détails de l'image.

Plusieurs approches et méthodes ont été développées pour réduire le bruit dans les images numériques. Parmi les plus courantes, on retrouve les filtres linéaires tels que les filtres moyenneurs ou gaussiens.

- **Les filtres moyenneurs** fonctionnent en remplaçant chaque pixel par la moyenne des valeurs de ses voisins, selon l'équation

$$g(x, y) = \frac{1}{M \cdot N} \cdot \sum_i \sum_j f(x + i, y + j) \cdot w(i, j) \quad (\text{B.14})$$

Ce filtre agit en lissant l'image, c'est-à-dire en réduisant les variations d'intensité entre pixels voisins. Il s'implémente par convolution linéaire, où le pixel central est remplacé par une somme pondérée des valeurs de son voisinage. Le masque ou noyau utilisé est généralement carré. Si la somme des coefficients du masque vaut 1, la luminosité moyenne de l'image n'est pas altérée. Si elle vaut 0, la luminosité moyenne est perdue et l'image résultante sera assombrie. Il fonctionne sur le principe de décalage-multiplication-somme. Il est efficace et bénéfique lorsque le bruit dans l'image est de type « sel et poivre ». En effet, il agit comme un filtre passe-bas qui ne laisse pas passer les hautes fréquences présentes dans le bruit. Des noyaux de plus grande taille (5x5, 7x7) produisent un dé bruitage plus important mais rendent l'image plus floue. Un compromis doit être fait entre la taille du noyau et l'intensité du dé bruitage. Ce filtre moyenneur est également connu sous le nom de filtre à coefficients constants, car la matrice de poids reste inchangée durant tout le processus. Il est particulièrement utile lorsque le bruit n'affecte que certaines régions de l'image et doit être éliminé localement.

- **Les filtres gaussiens**, quant à eux, appliquent un noyau de convolution pondéré suivant une distribution gaussienne, définie par :

$$w(i, j) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (\text{B.15})$$

Où : σ représente l'écart-type du noyau. Cela permet un lissage plus doux, tout en conservant mieux les contours de l'image que les filtres moyenneurs. Le degré de lissage peut être ajusté en modifiant la valeur de σ . Ces filtres gaussiens sont également très efficaces contre le bruit additif.

Bien que simples à mettre en œuvre, ces approches linéaires présentent cependant certaines limites. Elles peuvent en effet **altérer la netteté de l'image et gommer**

certains détails fins, en plus de ne pas être toujours optimales face à des types de bruit plus complexes comme le bruit impulsionnel ou multiplicatif.

C'est pourquoi d'autres techniques non-linéaires ont été développées pour tenter de mieux préserver les structures de l'image tout en réduisant le bruit de manière plus efficace. Parmi celles-ci, on peut citer les filtres médians et bilatéraux, qui offrent de meilleures performances dans certains cas.

1. **Le filtre médian** suit le même principe de fenêtre glissante que le filtre moyeneur, mais applique une opération non-linéaire. Pour chaque pixel, les valeurs des pixels voisins sont triées par ordre numérique, et la valeur médiane est affectée au pixel central. En cas de masque 3 x 3 la formule mathématique est la suivante :

$$g(p) = \text{median}\{f(p), f(N8(p))\} \quad (\text{B.16})$$

Où : $f(p)$ représente la valeur du pixel considéré et $N8(p)$ son voisinage 8-connexe.

Le filtre médian se montre particulièrement efficace pour réduire le bruit impulsionnel aléatoire, tout en préservant mieux les contours que les filtres linéaires. Cependant, il peut s'avérer moins performant pour lisser des bruits non-impulsifs comme le bruit gaussien. De plus, étant non localisé, il peut aussi altérer des pixels non affectés par le bruit.

Le tableau B-3 résume les Types de Filtres et Leurs Caractéristiques :

Tableau B-3 : Avantages et inconvénients des méthodes de dé-bruitage

Type de Filtre	Avantages	Inconvénients
Filtre Moyeneur (Linéaire)	<ul style="list-style-type: none"> - Facile à implémenter. - Efficace pour le bruit « sel et poivre ». - Lissage uniforme. - Opération de convolution simple. 	<ul style="list-style-type: none"> - Altère la netteté de l'image - Moins efficace pour d'autres types de bruit - Effet de flou prononcé sur les détails fins - Altère la luminosité moyenne si somme des poids $\neq 1$
Filtre Gaussien (Linéaire)	<ul style="list-style-type: none"> - Lissage doux de l'image. - Efficace contre le bruit additif. - Réduction significative du bruit. 	<ul style="list-style-type: none"> - Altère les contours et détails fins - Moins efficace pour certains types de bruit - Contrôle du lissage avec σ

	- Opère selon une distribution gaussienne pondérée.	
Filtre Médian (Non linéaire)	<ul style="list-style-type: none"> - Préserve les contours. - Efficace pour le bruit impulsionnel. - Adapté aux images avec bruit impulsionnel. - Opération non linéaire basée sur la médiane. 	<ul style="list-style-type: none"> - Moins efficace pour le bruit gaussien - Altère les détails non affectés par le bruit - Sensible à la taille du noyau

Pour comparer objectivement l'impact de ces filtres, il est nécessaire d'utiliser des métriques d'évaluation fiables. Deux métriques couramment employées dans ce contexte sont le PSNR (Peak Signal-to-Noise Ratio) et le SSIM (Structural Similarity Index).

Les métriques PSNR et SSIM sont couramment utilisées pour évaluer quantitativement la qualité d'une image dégradée ou filtrée par rapport à une image de référence. Ces métriques sont basées sur la mesure de différence entre l'image de référence notée x et l'image dégradée notée x_b .

Le PSNR compare la valeur maximale possible d'un pixel dans l'image à l'énergie de la différence normalisée entre les deux images. Mathématiquement, le PSNR entre x et x_b est calculé comme suit :

$$PSNR(x, x_b) = \log_{20} \left(\frac{255^2}{\sqrt{MSE(x, x_b)}} \right) \quad (B.17)$$

$$MSE(x, x_b) = \frac{1}{N} \sum_{i,j,c=0}^{H,W,C} (x^{i,j,c} - x_b^{i,j,c})^2 \quad (B.16)$$

Où : MSE est l'erreur quadratique moyenne entre les pixels des deux images. Cette métrique se concentre principalement sur les aspects locaux de l'image et permet une comparaison pixel par pixel.

H : Nombre de pixels de hauteur de l'image. W : Nombre de pixels de largeur de l'image. C : Nombre de canaux (1 pour l'image en nuances de gris, 3 pour l'image couleur)

D'autre part, le SSIM se concentre davantage sur les caractéristiques globales de l'image en considérant les valeurs statistiques de celle-ci. La formule du SSIM prend en compte la luminance, le contraste et la structure des images, calculés respectivement par les moyennes, les écarts-types et la covariance des valeurs de pixels. Le SSIM varie entre 0 et 1, avec 1 représentant une similitude parfaite entre les deux images.

$$SSIM(x, x_b) = \frac{(2\mu_{x_b}\mu_x + c_1)(2\sigma_{x_b}\sigma_x + c_2)(\text{cov}_{x_b x} + c_3)}{(\mu_{x_b}^2 + \mu_x^2 + c_1)(\sigma_{x_b}^2 + \sigma_x^2 + c_2)(\sigma_{x_b}\sigma_x + c_3)} \quad (B.17)$$

μ_{image} : Espérance de l'image.

σ_{image} : Ecart type de l'image.

$\text{Cov}_{\text{image1image2}}$: Covariance entre deux images

$c1 = (k1L)^2$, $c2 = (k2L)^2$ et $c3 = \frac{c2}{2}$: Constantes destinées à stabiliser la division.

Valeurs par défaut : $k_1 = 0.01$

$k_2 = 0.01$, L étant la valeur maximale d'un pixel (ici 255).

Bien que le PSNR soit une métrique largement utilisée, elle ne capture pas toujours avec précision la perception humaine de la qualité visuelle. C'est pourquoi le SSIM est souvent préféré, car il tient compte de la structure globale de l'image et se rapproche davantage de l'évaluation subjective de la qualité par l'œil humain. Cependant, les deux métriques restent complémentaires et sont souvent utilisées conjointement pour évaluer de manière plus complète l'effet des filtres sur les images.

La figure B-19 présente une comparaison visuelle des effets de trois filtres différents - gaussien, médian et moyeneur - appliqués à une même image. L'image originale est située en haut à gauche. En haut à droite, nous pouvons observer le résultat obtenu après l'application du filtre gaussien. En bas à gauche, se trouve le résultat du filtre médian. Enfin, en bas à droite, nous avons le résultat du filtre moyeneur.

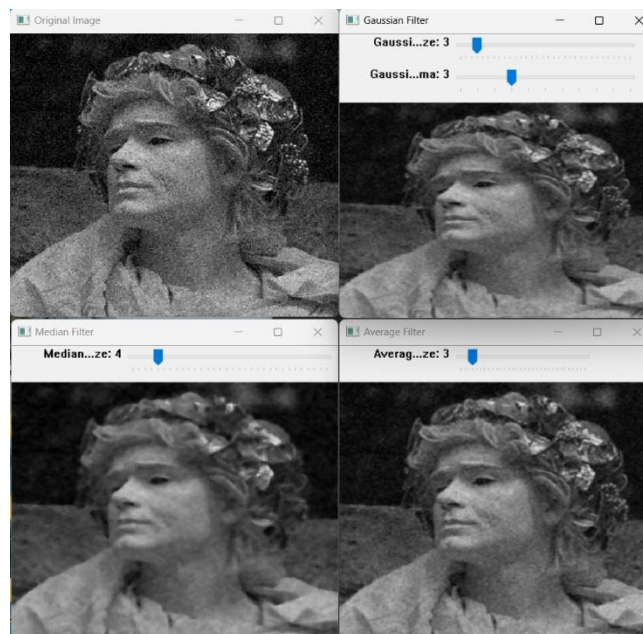


Figure B-19 : Comparaison des effets des filtres gaussien, médian et moyeneur sur le dé bruitage d'une image

Les espaces colorimétriques et les transitions entre ces espaces revêtent une importance capitale dans notre projet. Comprendre comment les couleurs sont

représentées et modifiées dans différents espaces colorimétriques est crucial pour détecter les altérations visuelles susceptibles de tromper ou manipuler les informations visuelles. Dans cette étude, nous explorerons les principaux espaces colorimétriques utilisés dans la représentation des images et des vidéos, en mettant l'accent sur leur rôle dans la détection des falsifications. Nous aborderons également les défis associés à l'identification des altérations de couleur dans ces différents espaces.

3. Les espaces colorimétriques

La couleur est une perception visuelle qui se crée lorsque la lumière tombe sur un objet et se reflète dans nos yeux. La couleur que nous voyons dépend de la longueur d'onde de la lumière réfléchiée par l'objet. L'œil humain est capable de voir une large gamme de couleurs créées par le spectre visible de la lumière, qui va du rouge au violet.

3.1 La science des espaces colorés

Dans le domaine du traitement d'images, les espaces colorimétriques jouent un rôle essentiel permettant de représenter les couleurs sous forme de valeurs numériques manipulables. Chaque espace colorimétrique est un modèle mathématique définissant une gamme spécifique de couleurs pouvant être affichées ou imprimées. Ces espaces sont fondamentaux pour la manipulation précise des couleurs et la reproduction fidèle des images.

Chaque espace colorimétrique possède un ensemble distinct de couleurs primaires qui servent de base pour créer toutes les autres couleurs de l'espace. Ces couleurs primaires sont soigneusement choisies pour maximiser la couverture spectrale et garantir une représentation précise des couleurs.

La diversité des espaces colorimétriques existe afin de répondre à des besoins spécifiques. Par exemple, l'espace colorimétrique RGB (rouge, vert, bleu) définit une couleur en termes de pourcentages de ces trois couleurs primaires mélangées ensemble. D'autres modèles, comme le modèle HSL (teinte, saturation, luminosité), décrivent les couleurs en termes de leur teinte (nuance de couleur), de leur saturation (quantité de gris ou de couleur pure) et de leur luminance (intensité ou luminosité globale). Chaque espace colorimétrique offre ainsi une manière unique et souvent plus pratique de représenter et de manipuler les couleurs dans le traitement d'images.

3.1.1 RGB (Red, Green, Blue)

Dans le domaine du traitement d'images, les images sont représentées sous forme de tableaux numériques $m \times n \times 3$, où chaque élément spécifie les valeurs d'intensité des canaux de couleur rouge, vert et bleu (RGB). Dans l'espace colorimétrique RGB, chaque

couleur est définie par une combinaison de valeurs de rouge, de vert et de bleu, variant de 0 à 255. Ce modèle de couleur est additif, ce qui signifie que différentes combinaisons de ces trois couleurs primaires permettent de créer un large spectre de couleurs.

L'espace colorimétrique RGB est couramment utilisé pour afficher des couleurs sur des écrans électroniques, tels que les moniteurs d'ordinateurs et les téléviseurs. Ce choix est dû à sa compatibilité avec la technologie d'affichage, ainsi qu'à sa capacité à représenter une gamme étendue de couleurs avec une grande précision. En utilisant des valeurs spécifiques pour chaque canal de couleur, le modèle RGB permet de reproduire une vaste palette de teintes, de nuances et de saturations, offrant ainsi une expérience visuelle riche et dynamique sur les dispositifs d'affichage modernes.

3.1.2 HSV (Hue, Saturation, Value)

C'est les trois paramètres de description d'une couleur. HSV Représente les couleurs en termes de teinte (le type de couleur), de saturation (l'intensité de la couleur) et de valeur (la luminosité). Il est souvent utilisé pour la sélection et la manipulation des couleurs, car il sépare les informations relatives à l'intensité des informations relatives à la couleur. HSV est souvent utilisé dans le traitement d'images et les tâches de vision par ordinateur telles que la segmentation des couleurs, le suivi d'objets et l'extraction de caractéristiques basées sur la couleur. Il sépare les informations relatives à l'intensité des informations relatives à la couleur, ce qui facilite l'utilisation indépendante des propriétés de la couleur.

La figure B-20 représente le modèle de couleur HSV. Le cône illustre comment la teinte, la saturation et la valeur se combinent pour créer une couleur spécifique.

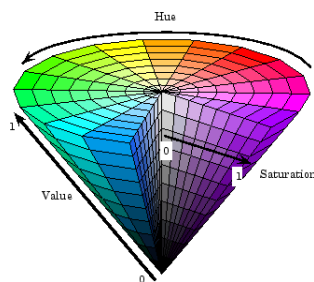


Figure B-20 : Illustration de l'espace couleur HSV

Maintenant on va expliquer comment transformer une image dans l'espace RGB à une image dans l'espace HSV.

Algorithm de Conversion RGB vers HSV :

1. *Diviser r, g, b par 255*
2. *Calculer $cmax, cmin, différence(diff)$*

$$cmax = \max(r, g, b) \quad (B.18)$$

$$cmin = \min(r, g, b) \quad (B.19)$$

$$diff = cmax - cmin \quad (B.20)$$

3. Calcul de la teinte (H) :

$$H = 0; \text{ si } cmax = cmin \quad (B.21)$$

$$H = \begin{cases} \left(60 \cdot \left(\frac{g-b}{diff}\right) + 360\right) \% 360; \text{ si } cmax = r \\ \left(60 \cdot \left(\frac{b-r}{diff}\right) + 120\right) \% 360; \text{ si } cmax = g \\ \left(60 \cdot \left(\frac{r-g}{diff}\right) + 240\right) \% 360; \text{ si } cmax = b \end{cases} \quad (B.22)$$

4. Calcul de la saturation (S) :

$$S = \begin{cases} 0; \text{ si } cmax = 0 \\ \frac{diff}{cmax} \cdot 100; \text{ si non} \end{cases} \quad (B.23)$$

5. Calcul de la valeur (V) :

$$V = cmax \cdot 100 \quad (B.24)$$

La figure B-21 démontre la conversion d'une image de l'espace RGB à l'espace HSV.

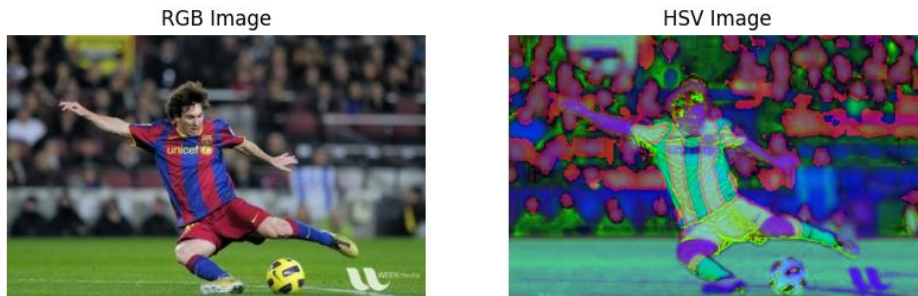


Figure B-21 : Conversion d'une image de l'espace RGB à l'espace HSV

3.1.3 YCbCr

Espace colorimétrique utilisé dans la compression et la transmission vidéo tels que MPEG et JPEG. Y est la composante luma (la luminosité de la couleur) et C_B et C_R sont les composantes chroma de différence bleue et de différence rouge (informations sur les couleurs).

Pour convertir une image RGB à une image YCbCr on utilise ces équations :

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (B.25)$$

$$Cr \leftarrow (R - Y) \cdot 0.713 + \delta \quad (B.26)$$

$$Cb \leftarrow (B - Y) \cdot 0.564 + \text{delta} \quad (\text{B.27})$$

$$R \leftarrow Y + 1.403 \cdot (Cr - \text{delta}) \quad (\text{B.28})$$

$$G \leftarrow Y - 0.714 \cdot (Cr - \text{delta}) - 0.344 \cdot (Cb - \text{delta}) \quad (\text{B.29})$$

$$B \leftarrow Y + 1.773 \cdot (Cb - \text{delta}) \quad (\text{B.30})$$

$$\text{delta} = \begin{cases} 128 & \text{for 8-bit images} \\ 32768 & \text{for 16-bit images} \\ 0.5 & \text{for floating point images} \end{cases} \quad (\text{B.31})$$

La figure B-22 présente le processus de décomposition d'une image RGB en ses composantes YCbCr.

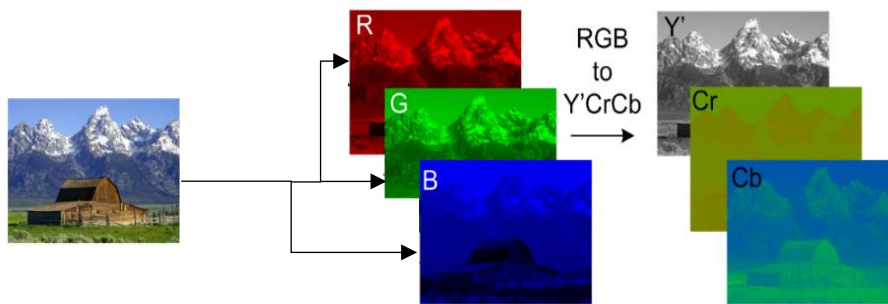


Figure B-22 : Décomposition d'une image RGB en composantes YCbCr

3.1.4 LAB

Conçu pour se rapprocher davantage de la vision humaine que les autres espaces colorimétriques. Il exprime la couleur en trois valeurs : L pour la luminosité et les canaux a et b représentent respectivement les composantes vert-rouge et bleu-jaune de la couleur. Ces 4 couleurs se sont les couleurs uniques de la vision humaine (l'information sur les couleurs). LAB est souvent utilisé dans la correction des couleurs, la correspondance des couleurs et les calculs de différence de couleur, où l'uniformité et la précision de la perception sont importantes.

La figure B-23 illustre la structure tridimensionnelle de l'espace couleur LAB.

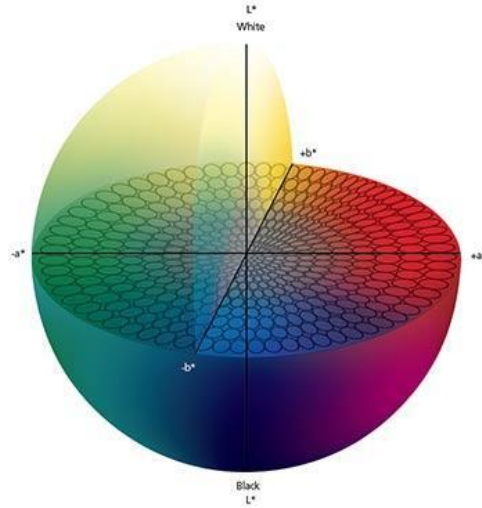


Figure B-23 : Illustration de l'espace couleur LAB

Nous allons maintenant expliquer les étapes pour convertir une image de l'espace de couleur RVB en une image dans l'espace de couleur LAB.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{B.32})$$

$$X \leftarrow \frac{X}{X_n}, \text{ où } X_n = 0.950456 \quad (\text{B.33})$$

$$Z \leftarrow Z/Z_n, \text{ où } Z_n = 1.088754 \quad (\text{B.34})$$

$$L \leftarrow \begin{cases} 116 \cdot Y^{\frac{1}{3}} - 16 & \text{pour } Y > 0.008856 \\ 903.3 \cdot Y & \text{pour } Y \leq 0.008856 \end{cases} \quad (\text{B.35})$$

$$a \leftarrow 500(f(X) - f(Y)) + \text{delta} \quad (\text{B.36})$$

$$b \leftarrow 200(f(Y) - f(Z)) + \text{delta} \quad (\text{B.37})$$

$$\text{où } f(t) = \begin{cases} t^{\frac{1}{3}} & \text{for } t > 0.008856 \\ 7.767t + \frac{16}{116} & \text{pour } t \leq 0.008856 \end{cases} \quad (\text{B.38})$$

et

$$\text{delta} = \begin{cases} 128 & \text{for 8-bit images} \\ 0 & \text{for floatng-point images} \end{cases} \quad (\text{B.39})$$

Cette sortie $0 \leq L \leq 100, -127 \leq a \leq 127, -127 \leq b \leq 127$.

- Les valeurs sont converties en destination type de données.
- 8-bit images : $L \leftarrow L * \frac{255}{100}, a \leftarrow a + 128, b \leftarrow b + 128$
- 16-bit images : (actuellement non pris en charge)

-
- 32 – bit images : L, a et b sont laissés tels quels

La conversion d'une image de l'espace de couleur RGB vers l'espace de couleur LAB est illustrée visuellement dans la figure B-24 :



Figure B-24 : Image RGB (à gauche) ; Image dans l'espace de couleur LAB (à droite)

Après avoir exploré en profondeur les espaces colorimétriques et leur impact significatif sur la représentation visuelle, nous allons maintenant aborder un autre aspect crucial de la manipulation d'images : les transformations géométriques et spatiales. Ces transformations impliquent la modification de la position, de la forme ou de l'orientation des objets dans l'espace. Elles permettent non seulement d'adapter les images à nos besoins spécifiques, mais aussi de créer des représentations plus précises et plus dynamiques du monde qui nous entoure.

4. Transformations géométriques et spatiales

Les transformations géométriques, telles que la rotation, le retournement, le zoom et le recadrage, constituent un ensemble fondamental de techniques dans le domaine de la manipulation d'images et de vidéos. Leur application permet de remodeler la structure spatiale des médias visuels, offrant ainsi une multitude de possibilités créatives tout en soulevant des questions cruciales quant à l'intégrité et à l'authenticité des contenus visuels. Cette étude se propose d'examiner en profondeur ces transformations géométriques, leur incidence sur l'apparence visuelle des médias numériques, ainsi que les stratégies de détection élaborées pour identifier les altérations géométriques et garantir l'intégrité des données visuelles.

Maintenant, nous allons expliquer quelques-uns des concepts de base de la transformation géométrique et spatiales dans le traitement des images :

- **Fonctions de transformation** : Les fonctions de transformation sont des fonctions mathématiques utilisées pour modifier les propriétés géométriques d'une image. Ces fonctions transforment les coordonnées de chaque pixel d'une image en de nouvelles coordonnées sur la base d'une règle de transformation spécifiée.

Parmi les fonctions de transformation couramment utilisées, citons la mise à l'échelle, la rotation, la translation et le cisaillement.

- **Système de coordonnées** : Un système de coordonnées est un système de référence utilisé pour définir la position spatiale des pixels dans une image. Dans le traitement des images numériques, le système de coordonnées le plus couramment utilisé est le système de coordonnées cartésiennes, qui utilise deux axes perpendiculaires (x et y) pour représenter les positions horizontale et verticale des pixels d'une image.
- **Interpolation** : L'interpolation est le processus d'estimation des valeurs des pixels d'une image à des endroits qui ne sont pas explicitement définis. Cette opération est nécessaire lors de la transformation d'une image, car les nouvelles coordonnées peuvent ne pas coïncider avec les positions des pixels d'origine. Les algorithmes d'interpolation sont utilisés pour estimer les valeurs des pixels aux nouveaux emplacements sur la base des valeurs des pixels environnants.

Nous parlerons de deux types de transformations dont nous avons le plus besoin : la rotation et le retournement (flipping).

4.1 Rotation

La rotation est une transformation fondamentale dans le traitement des images qui consiste à modifier l'orientation d'une image d'un certain angle autour d'un point ou d'un axe spécifié. Elle est couramment utilisée pour corriger les erreurs d'orientation, aligner les images ou obtenir les compositions souhaitées. Lorsqu'une image est tournée, ses pixels sont repositionnés en fonction de l'angle de rotation, tout en préservant la taille et la forme de l'image.

4.1.1 Paramètres

Angle : L'angle de rotation indique l'ampleur de la rotation en degrés. Les angles positifs indiquent une rotation dans le sens inverse des aiguilles d'une montre, tandis que les angles négatifs indiquent une rotation dans le sens des aiguilles d'une montre.

Centre de rotation : La rotation est effectuée autour d'un point ou d'un centre spécifié. Par défaut, le centre de rotation est le centre de l'image.

4.1.2 Applications

- Correction des erreurs d'orientation dans les documents numérisés ou les photographies.
- Alignement d'images pour l'assemblage ou l'enregistrement de panoramas.

-
- Transformation d'images pour des tâches de détection ou de reconnaissance d'objets.

La figure B-25 présente une image pivotée avec un angle de 90 degrés par rapport à l'image originale.



Figure B-25 : Image pivotée avec un angle de 90 degrés par rapport à l'image originale

4.2 Zoom :

Le zoom est le processus d'agrandissement ou de grossissement d'une région ou d'une zone spécifique à l'intérieur d'une image. Cela permet aux utilisateurs de se concentrer sur des détails ou des zones d'intérêt particuliers en augmentant leur taille et leur visibilité. En effet, zoomer simplement signifie agrandir une image dans le sens où les détails présents deviennent plus visibles et clairs, comme illustré dans la figure B-26

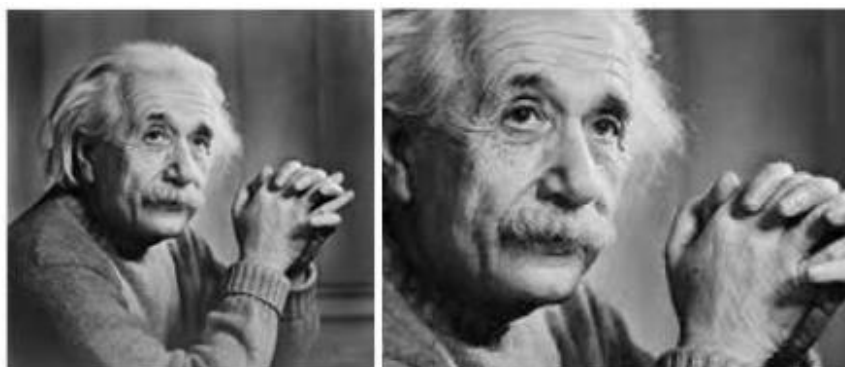


Figure B-26 : Un exemple de l'application du zoom sur une image

Cependant, il existe deux types de zoom distincts, ayant chacun leurs propres caractéristiques et niveaux de performance. Une transition naturelle nous amène à explorer ces deux modes de zoom :

Zoom optique vs Zoom numérique

- **Zoom optique :** Le zoom optique est réalisé grâce au mouvement des lentilles de l'appareil photo. Un zoom optique est en fait un véritable zoom. Le résultat du zoom optique est bien meilleur que celui du zoom numérique. Dans un zoom optique, une image est agrandie par l'objectif de telle sorte que les objets dans

l'image semblent plus proches de l'appareil photo. Dans un zoom optique, l'objectif s'allonge physiquement pour zoomer ou agrandir un objet.

- **Zoom numérique** : Le zoom numérique est essentiellement un traitement d'image au sein de l'appareil photo. Lors d'un zoom numérique, le centre de l'image est agrandi et les bords de l'image sont rognés. En raison de l'agrandissement du centre, il semble que l'objet est plus proche de vous. Pendant un zoom numérique, les pixels sont élargis, ce qui compromet la qualité de l'image.

Nous allons seulement nous intéresser au **zoom numérique**, qui peut être effectué à l'aide de plusieurs techniques de traitement d'image. Parmi la multitude de méthodes existantes pour réaliser cette tâche, nous allons aborder une seule méthode que nous avons implémenter.

La réplication de pixels, également connue sous le nom d'interpolation du plus proche voisin, est une méthode simple et intuitive pour effectuer un zoom numérique sur une image. Comme son nom l'indique, cette approche consiste à répliquer les pixels voisins en les dupliquant horizontalement et verticalement selon le facteur de zoom souhaité. Son fonctionnement est très direct : pour chaque pixel de l'image originale, on crée n pixels identiques dans l'image zoomée, où n correspond au facteur de zoom. Cette opération est effectuée ligne par ligne, puis colonne par colonne. Par exemple, pour zoomer une image 2×2 d'un facteur 2, chaque pixel est répliqué 4 fois, résultant en une image finale de dimensions 4×4 .

La méthode **Zero Order Hold**, également connue sous le nom de **Zoom deux fois**, est une approche alternative pour effectuer un zoom numérique sur une image. Contrairement à la réplication de pixels, cette technique vise à produire des images zoomées de meilleure qualité en insérant de nouveaux pixels calculés à partir des pixels voisins existants. Son fonctionnement consiste à parcourir l'image ligne par ligne, puis colonne par colonne. Pour chaque paire de pixels adjacents, on calcule leur moyenne et on insère cette valeur moyenne comme nouveau pixel entre les deux pixels d'origine. Cette opération est répétée sur toutes les lignes, puis sur toutes les colonnes. Par exemple, pour zoomer une image 2×2 d'un facteur 2, la méthode insèrera d'abord un pixel entre chaque paire de pixels sur les lignes, résultant en une image 2×3 . Puis, elle insèrera un pixel entre chaque paire sur les colonnes, donnant une image finale de dimensions 3×3 .

La méthode de « **Zoom K-fois** » est considérée comme l'une des plus performantes pour effectuer un zoom numérique sur une image. Son principal avantage réside dans sa flexibilité à zoomer avec n'importe quel facteur entier K , contrairement aux méthodes plus limitées comme la réplication de pixels ou le Zero Order Hold. Son fonctionnement repose sur l'insertion de nouveaux pixels calculés à partir des différences entre pixels

voisins. Pour chaque paire de pixels adjacents, on calcule leur différence, que l'on divise ensuite par le facteur de zoom K pour obtenir une valeur intermédiaire. On insère alors $K-1$ nouveaux pixels entre les deux pixels d'origine, en ajoutant successivement la valeur intermédiaire au pixel le plus petit. Par exemple, pour zoomer une image 2×3 d'un facteur 3, on commencerait par calculer la différence entre les deux premiers pixels ($30-15=15$). On diviserait alors 15 par 3 pour obtenir 5 comme valeur intermédiaire. On insérerait ensuite deux nouveaux pixels (20 et 25) entre 15 et 30. Ce processus serait répété pour toutes les paires de pixels, ligne par ligne puis colonne par colonne.

La méthode de zoom utilisée pour notre implémentation est le **Zoom K-fois**, offrant une flexibilité sur le facteur de zoom tout en produisant des images de haute qualité.

Maintenant que nous avons compris les principes de base de la méthode de Zoom K-fois, penchons-nous plus en détail sur l'algorithme spécifique que nous avons utilisé pour notre implémentation.

1. Calcul de la région d'intérêt : Les dimensions de la région à zoomer autour du curseur sont calculées en fonction du facteur de zoom K . Cela garantit que la région zoomée reste centrée sur le curseur.
2. Extraction de la région : La région d'intérêt est extraite de l'image originale à l'aide d'un simple rognage (slicing) basé sur les coordonnées calculées précédemment.
3. Traitement ligne par ligne : Pour chaque ligne de la région extraite, le code itère sur les paires de pixels adjacents. Pour chaque paire :
 - La différence entre les deux pixels est calculée.
 - Cette différence est divisée par le facteur de zoom K pour obtenir une valeur intermédiaire.
 - $(K-1)$ nouveaux pixels sont insérés entre les deux pixels d'origine, en ajoutant successivement la valeur intermédiaire au pixel le plus petit.
4. Traitement colonne par colonne : Le même processus est répété pour chaque colonne de la région zoomée, insérant ainsi de nouveaux pixels dans les deux dimensions.
5. Tri naturel des valeurs insérées : Grâce au rognage ligne par ligne et colonne par colonne, les valeurs insérées se retrouvent naturellement triées par ordre croissant, assurant une transition en douceur entre les pixels.

La figure B-27 présente une comparaison entre le Zoom K-fois sur une image originale avec des facteurs de zoom 4x et 8x.

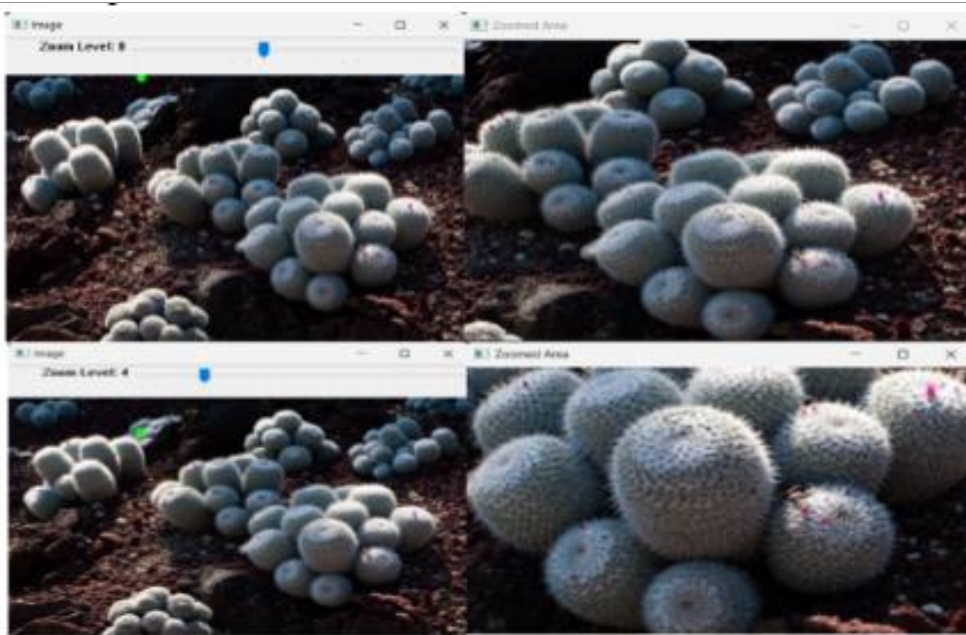


Figure B-27 : Comparaison visuelle de l'effet du Zoom K-fois

Le tableau B-4 présente une comparaison détaillée des méthodes de zoom numérique, à savoir la **réplication de pixels**, le **Zero Order Hold** et le **Zoom K-fois**, en mettant en évidence leurs avantages respectifs, leurs inconvénients et la complexité de leur implémentation.

Tableau B-4 : comparaison détaillée des méthodes de zoom numérique

Méthode	Avantages	Inconvénients
Réplication de pixels	<ul style="list-style-type: none"> - Facile à mettre en œuvre. - Rapide et efficace. - Aucun calcul complexe requis. 	<ul style="list-style-type: none"> - Qualité d'image médiocre, particulièrement avec des facteurs de zoom élevés. - Effets de crénelage et pixélisation visibles. - Perte de détails et de clarté dans l'image résultante.
Zero Order Hold (Zoom 2x)	<ul style="list-style-type: none"> - Amélioration de la qualité visuelle par rapport à la réplication de pixels. - Moins de flou et de pixélisation que la réplication de pixels. - Implémentation relativement simple. 	<ul style="list-style-type: none"> - Limitation au zoom par un facteur de puissance de 2 (2x, 4x, 8x ...). - Ne produit pas des images zoomées pour tous les facteurs de zoom désirés.

Zoom K-fois	<ul style="list-style-type: none"> - Qualité d'image améliorée grâce à l'insertion de nouveaux pixels calculés. - Flexibilité sur le facteur de zoom (n'importe quel entier K). - Moins de flou et de pixélisation par rapport aux méthodes plus simples - Convient pour des applications nécessitant un zoom de haute qualité. 	<ul style="list-style-type: none"> - Nécessite des calculs plus complexes et une gestion des valeurs insérées. - Ajout d'une étape de tri des valeurs insérées pour assurer la transition en douceur entre les pixels. - Plus exigeant en termes de ressources de calcul. - Complexité accrue de l'algorithme rendant l'implémentation plus difficile.
--------------------	---	--

4.3 Retournement (Flipping)

Le retournement, également appelé miroir, est une transformation spatiale qui inverse la disposition spatiale d'une image le long d'un ou de plusieurs axes. Il s'agit de réfléchir l'image sur un axe spécifique pour créer une image miroir. Le retournement peut être effectué horizontalement, verticalement ou les deux.

4.3.1 Paramètres

Code de retournement : détermine l'axe ou les axes sur lesquels l'image est retournée. Il peut prendre l'une des valeurs suivantes : (0 : retournement horizontal (autour de l'axe vertical) ,1 : retournement vertical (autour de l'axe horizontal). -1 : retournement horizontal et vertical (autour du point central).

4.3.1 Applications

- Création de compositions symétriques dans la conception artistique et graphique.
- Inversion de textes ou de graphiques à des fins d'affichage ou d'impression.
- Générer des données d'entraînement supplémentaires pour les modèles d'apprentissage automatique.

La figure B-28 présente une image originale, une image retournée verticalement et une image retournée horizontalement.



Figure B-28 : Retournements vertical et horizontal d'une image

4.4 Rognage (Cropping) :

Après avoir exploré les différentes techniques de zoom permettant d'agrandir et de se rapprocher des détails d'une image, passons maintenant à une autre technique qui va nous permettre de nous focaliser seulement sur une région précise de l'image sans prendre en considération la partie « out of Region Of Interest » (**ROI**). C'est le **cropping** ou **bien le rognage des images**.

Le rognage des images : Le rognage est une opération de traitement d'image fondamentale qui consiste à sélectionner une portion d'une image tout en supprimant le reste. Cette opération est couramment utilisée pour supprimer les parties indésirables d'une image, se concentrer sur des régions d'intérêt spécifiques ou ajuster le rapport d'aspect. Voici comment fonctionne généralement le rognage :

- **Sélection** : L'utilisateur spécifie la région d'intérêt en définissant un cadre englobant ou une forme polygonale autour de la zone souhaitée. Cette sélection peut être effectuée de manière interactive à l'aide d'une interface graphique ou par programmation en fournissant des coordonnées. Dans notre approche, nous préférons dessiner un polygone entourant la région d'intérêt. Cela peut être fait à l'aide d'un outil d'interface graphique ou en fournissant par programmation les coordonnées des sommets du polygone. L'avantage d'utiliser un polygone est qu'il permet une sélection plus précise, en particulier lorsque la région d'intérêt à une forme irrégulière.
- **Extraction** : Une fois que la région d'intérêt est définie par le polygone, les pixels à l'intérieur de cette zone sont extraits de l'image d'origine. Ce processus implique la copie des valeurs de pixels de la région sélectionnée et la création d'une nouvelle image contenant uniquement ces pixels.
- **Affichage** : L'image rognée, contenant uniquement la région d'intérêt sélectionnée, est affichée. Cependant, au lieu d'afficher simplement la région rognée, ma méthode remplit la zone en dehors du polygone avec des pixels noirs.

Cela permet de se concentrer uniquement sur la région d'intérêt en la séparant visuellement du reste de l'image.

- **Post-traitement optionnel** : Selon l'application, des étapes de traitement supplémentaires telles que le redimensionnement, le filtrage ou l'amélioration peuvent être appliquées à l'image rognée pour obtenir le résultat souhaité.

En utilisant un polygone pour définir la région d'intérêt et en remplissant la zone extérieure avec des pixels noirs, cette méthode de rognage garantit que l'attention du détective est dirigée uniquement sur la partie pertinente de l'image. Cette approche est particulièrement utile lorsque l'on travaille avec des images complexes contenant plusieurs objets ou régions d'importance variable, car elle permet d'isoler et d'analyser précisément des zones spécifiques.

Après avoir défini le rognage, la figure B-29 illustre un exemple concret de son application sur une image originale, en montrant le résultat obtenu après ce processus.

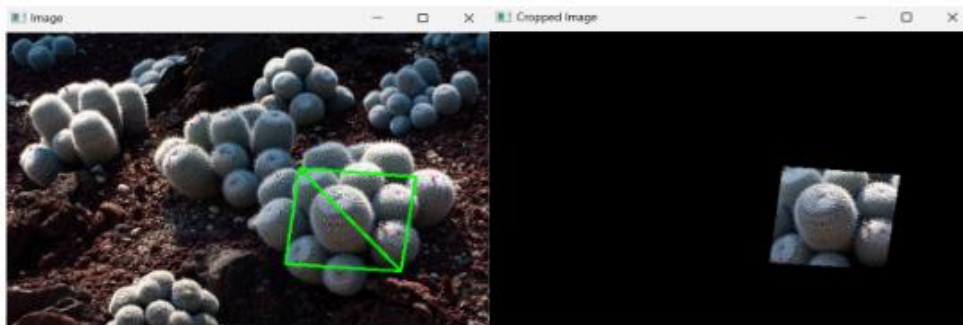


Figure B-29 : Exemple de l'application du rognage sur une image originale et résultat obtenu

Au-delà des données visuelles proprement dites, les images numériques comportent souvent une couche supplémentaire d'informations, appelées métadonnées. Ces métadonnées qui veulent dire les « données sur les données » constituent un ensemble de données structurées décrivant le contenu, le contexte ou les caractéristiques techniques d'une image et font référence aux détails sur l'image et son mode de production. Certaines de ces métadonnées sont générées automatiquement par le dispositif de capture.

5. Les métadonnées

L'analyse des métadonnées associées à une image peut apporter des informations précieuses, complémentaires à l'analyse du contenu visuel. En effet, les métadonnées peuvent révéler des éléments surs :

- L'historique de l'image (date, heure, appareil photo utilisé...)
- Son contexte de création (localisation, conditions de prise de vue...)

-
- Les modifications éventuellement apportées (éditions, retouches...)

La confrontation de ces informations de métadonnées avec le contenu de l'image peut donc permettre de détecter des incohérences ou des preuves de manipulations.

Pour procéder efficacement à l'analyse des métadonnées, il est essentiel de posséder certains prérequis fondamentaux. Tout d'abord, une compréhension approfondie des principaux standards de métadonnées tels que Exif, IPTC et XMP, ainsi que de leur structure, est indispensable. Ensuite, des capacités solides en matière d'extraction et de lecture des métadonnées à partir de fichiers d'image sont requises. De plus, une connaissance détaillée des informations typiquement contenues dans les métadonnées et de leur signification respective est cruciale.

Dans ce contexte, nous allons maintenant approfondir notre compréhension des métadonnées associées aux fichiers d'images numériques. Ces précieuses informations descriptives sont stockées dans différentes sections dédiées à cet effet. Parmi les principales sections utilisées pour le stockage des métadonnées, on retrouve les normes bien connues telles que Exif, IPTC et XMP.

La section Exif (**Exchangeable Image File Format**) rassemble principalement les métadonnées techniques liées à la prise de vue elle-même. On y trouve les paramètres de l'appareil photo et de l'objectif utilisés, ainsi que les réglages appliqués lors de la capture de l'image. Deux éléments particulièrement importants sont également encodés dans Exif : la date et l'heure précises auxquelles le cliché a été réalisé, ainsi que les coordonnées GPS du lieu de prise de vue, si disponibles. Il permet aussi de stocker des informations telles que les réglages de l'appareil photo (comme l'ouverture, la vitesse d'obturation, la sensibilité ISO) et le modèle de l'appareil photo, et d'autres détails liés à la prise de vue. Cependant, Exif souffre d'une limitation notable, celle d'être restreint à l'utilisation des caractères ASCII, ce qui limite grandement les possibilités de description textuelle.

La section IPTC (**International Press Telecommunications Council**) a été conçue dès l'origine dans un objectif d'archivage fin des images, en permettant l'ajout de métadonnées descriptives enrichies. On peut ainsi y renseigner le nom du photographe, une légende détaillée, des mots-clés, des informations sur le contenu éditorial : le titre, les crédits et les droits d'auteur. Initialement limitée aux caractères ASCII/ANSI, IPTC supporte désormais Unicode/UTF-8. Néanmoins, cette norme vieillissante comporte d'autres limites, comme des tailles de champs restreintes et un manque de champs pour certaines métadonnées devenues indispensables. Ses spécifications ne sont d'ailleurs plus activement maintenues, l'organisation IPTC ayant choisi de migrer progressivement vers XMP.

XMP (Extensible Metadata Platform) est un format de métadonnées flexible et extensible, sans limitation de taille ou de nombre de champs, développé par Adobe Systems. La section XMP peut contenir à la fois les métadonnées Exif, IPTC, mais également celles issues d'autres schémas comme celui d'Adobe. Avec XMP, plus de contraintes, ce qui est à la fois un avantage et un risque potentiel. En effet, son contenu pourrait devenir énorme et désordonné si son utilisation n'est pas bien encadrée. C'est pourquoi les logiciels récents adoptent XMP pour le stockage centralisé de la majorité des métadonnées, tout en conservant les sections Exif et IPTC pour certaines données clés.

5.1 ExifTool : Gestion Avancée des Métadonnées

ExifTool, développé par Phil Harvey, est un outil robuste et polyvalent permettant d'extraire, de visualiser et d'éditer efficacement les métadonnées d'une grande variété de fichiers, notamment les images, les vidéos et les fichiers audio. Construit comme une bibliothèque Perl avec une application en ligne de commande, ExifTool est multiplateforme et prend en charge de nombreux formats de métadonnées, tels que EXIF, IPTC, XMP, et d'autres normes courantes. Il excelle dans la lecture et l'écriture des notes de fabrication des appareils photo numériques, ce qui en fait un choix prisé des photographes et des professionnels de l'imagerie numérique.

Les avantages d'ExifTool sont nombreux. Il offre un support complet des différents formats de métadonnées, y compris l'EXIF, l'IPTC et le XMP. Sa compatibilité multiplateforme lui permet d'être utilisé sur divers systèmes d'exploitation. Les capacités étendues d'extraction et d'écriture offrent une grande flexibilité pour manipuler les métadonnées. De plus, ExifTool est reconnu pour sa personnalisation, offrant de nombreuses options de formatage de sortie et un support multilingue. Il bénéficie d'un développement actif soutenu par une communauté dynamique, assurant des mises à jour régulières et l'ajout de nouvelles fonctionnalités. ExifTool permet un traitement par lot efficace, ce qui est idéal pour gérer des collections d'images et de fichiers multimédias. Enfin, ExifTool est un logiciel gratuit et open-source, accessible à tous sans coût supplémentaire. Avec ses fonctionnalités avancées et ses nombreux avantages, ExifTool est devenu indispensable pour ceux qui travaillent avec la gestion et la manipulation des métadonnées.

Le tableau B-5 présente une comparaison des outils d'extraction, d'édition et de visualisation de métadonnées d'image, à savoir ExifTool, exiv2, exif, Windows Photo Metadata Toolkit et Adobe Bridge, en fonction de leurs caractéristiques, compatibilité, fonctionnalités et coût

Tableau B-5 : outils d'extraction, d'édition et de visualisation de métadonnées d'image

Caractéristique / Outil	ExifTool	exiv2	exif	Windows Photo Metadata Toolkit	Adobe Bridge
Formats de métadonnées pris en charge	EXIF, IPTC, XMP, JFIF, et autres	EXIF, IPTC, XMP, et autres	Principalement EXIF	Principalement les formats pris en charge par Windows	Formats pris en charge par Adobe Creative Suite
Compatibilité	Multiplateforme (Perl)	Principalement pour les systèmes Linux	Unix-like systems	Windows	Windows, macOS
Fonctionnalités d'extraction	Extraction, visualisation, édition	Extraction, édition, visualisation	Extraction basique	Extraction basique	Extraction, édition, visualisation
Personnalisation	Options avancées de formatage de sortie	Options de personnalisation	Personnalisation limitée	Personnalisation limitée	Personnalisation étendue
Support de la communauté	Actif et soutenu par la communauté Perl	Actif et soutenu par la communauté	Soutien communautaire limité	Fait partie des ressources Microsoft	Soutenu par la communauté Adobe
Mises à jour régulières	Oui	Oui	Moins fréquentes	Mises à jour régulières	Mises à jour régulières
Traitement par lot	Oui	Oui	Limité	Intégré à l'environnement Windows	Oui
Coût	Gratuit et open-source	Gratuit et open-source	Gratuit	Inclus dans le système Windows	Commercial (inclus dans Adobe Creative Cloud)

La figure B-30 permet de visualiser les métadonnées de l'image originale et de l'image modifiée (croppée). On peut y constater plusieurs modifications, telles que la date de dernière modification, qui a été mise à jour suite au processus de cropping.

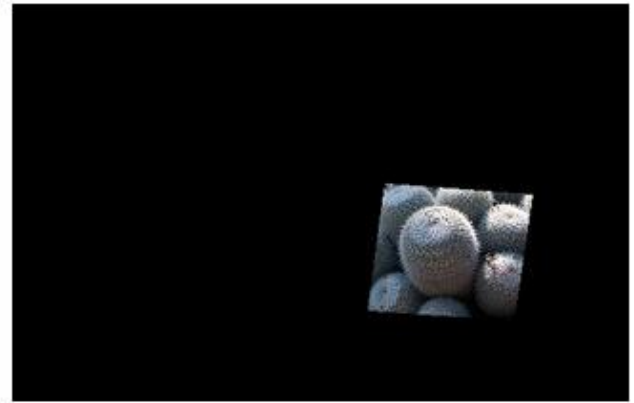


```

metadata:
{
  "sourceFile": "C:/Users/Computer House/Desktop/FFI/Codes/Methodes simples/sharpening_cacti.jpg",
  "exifToolVersion": 12.81,
  "fileName": "sharpening_cacti.jpg",
  "directory": "C:/Users/Computer House/Desktop/FFI/Codes/Methodes simples",
  "fileSize": "92 KB",
  "zoneIdentifier": "Exists",
  "fileModifyDate": "2024:03:09 13:29:19+01:00",
  "fileAccessDate": "2024:07:02 09:59:23+02:00",
  "fileCreateDate": "2024:03:09 13:29:19+01:00",
  "filePermissions": "-rw-rw-rw-",
  "fileType": "JPEG",
  "fileTypeExtension": ".jpg",
  "MIMEType": "image/jpeg",
  "JFIFVersion": 1.01,
  "ResolutionUnit": "None",
  "xResolution": 1,
  "yResolution": 1,
  "imageWidth": 908,
  "imageHeight": 321,
  "encodingProcess": "Baseline DCT, Huffman coding",
  "bitsPerSample": 8,
  "colorComponents": 3,
  "YCbCrSubsampling": "YCbCr4:2:0 (1 1)",
  "imageSize": "308x311",
  "megapixels": 0.101
}

```

(a)



```

metadata:
{
  "sourceFile": "C:/Users/Computer House/Desktop/FFI/Codes/Methodes simples/cropped_image.jpg",
  "exifToolVersion": 12.81,
  "fileName": "cropped_image.jpg",
  "directory": "C:/Users/Computer House/Desktop/FFI/Codes/Methodes simples",
  "fileSize": "14 KB",
  "fileModifyDate": "2024:04:24 13:23:16+01:00",
  "fileAccessDate": "2024:07:02 09:58:45+01:00",
  "fileCreateDate": "2024:03:31 10:40:40+01:00",
  "filePermissions": "-rw-rw-rw-",
  "fileType": "JPEG",
  "fileTypeExtension": ".jpg",
  "MIMEType": "image/jpeg",
  "JFIFVersion": 1.01,
  "ResolutionUnit": "None",
  "xResolution": 1,
  "yResolution": 1,
  "imageWidth": 508,
  "imageHeight": 321,
  "encodingProcess": "Baseline DCT, Huffman coding",
  "bitsPerSample": 8,
  "colorComponents": 3,
  "YCbCrSubsampling": "YCbCr4:2:0 (1 1)",
  "imageSize": "168x321",
  "megapixels": 0.051
}

```

(b)

Figure B-30 : a) l'image originale avec ses métadonnées associées et b) l'image modifiée (croppée) avec ses métadonnées mises à jour

Annexe C

Modeling

1. Introduction

Cette section de l'annexe fournit des détails sur la bibliothèque Albumentations et les diverses techniques d'augmentation de données utilisées. Elle explique également les modèles pré-entraînés employés avec le transfer learning et décrit les technologies utilisées pour déployer notre solution.

2. Bibliothèque de Albumentations

Cette bibliothèque offre une gamme étendue de fonctionnalités pour améliorer et diversifier les images utilisées dans les modèles de vision par ordinateur. Parmi ces fonctionnalités, on trouve :

- **PadIfNeeded** : Ajoute des bordures pour atteindre une taille minimale spécifiée.
- **HorizontalFlip et VerticalFlip** : Pour diversifier les orientations des images en les retournant horizontalement et verticalement.
- **Transpose** : Échange les axes des images, ce qui peut aider à diversifier davantage les orientations.
- **HueSaturationValue** : Modifie la teinte, la saturation et la valeur des images pour enrichir la variété des couleurs.
- **ElasticTransform** : Applique des déformations élastiques aux images pour simuler des variations réalistes.
- **GridDistortion** : Applique des distorsions en forme de grille pour augmenter la diversité géométrique des images.
- **OpticalDistortion** : Simule des distorsions optiques qui peuvent apparaître dans des conditions de prise de vue réelles.
- **RandomBrightnessContrast** : Modifie aléatoirement la luminosité et le contraste pour simuler des conditions d'éclairage variées.
- **RandomGamma** : Applique des ajustements de gamma pour varier la luminosité de manière non linéaire.

- **Resize** : Redimensionne les images à une taille spécifique pour s'assurer qu'elles sont compatibles avec les exigences du modèle.

L'avantage d'Albumentations réside dans sa rapidité d'exécution et la possibilité de combiner plusieurs transformations de manière efficace. Cela permet de générer un grand nombre de variations d'images, augmentant ainsi la quantité de données disponibles pour l'entraînement du modèle sans avoir besoin de collecter de nouvelles images.

3. Les modèles sélectionnés pour le transfer learning

- **EfficientNet (B0, B4, B7)** : EfficientNet (montrée dans la figure C-1) est une famille de modèles de réseaux neuronaux convolutifs qui ont été conçus pour optimiser l'équilibre entre profondeur, largeur et résolution du réseau. Les modèles EfficientNet utilisent une technique d'échelle composée pour ajuster la taille du réseau en fonction des ressources disponibles et des exigences de performance. EfficientNetB0 est le modèle de base de la première génération d'EfficientNet. Il offre un bon compromis entre performance et efficacité computationnelle. EfficientNetB4 propose un niveau de complexité intermédiaire, avec une profondeur et une largeur accrue par rapport à EfficientNetB0. EfficientNetB7 est le plus grand et potentiellement le plus performant de cette génération, avec une profondeur et une largeur encore plus grande.

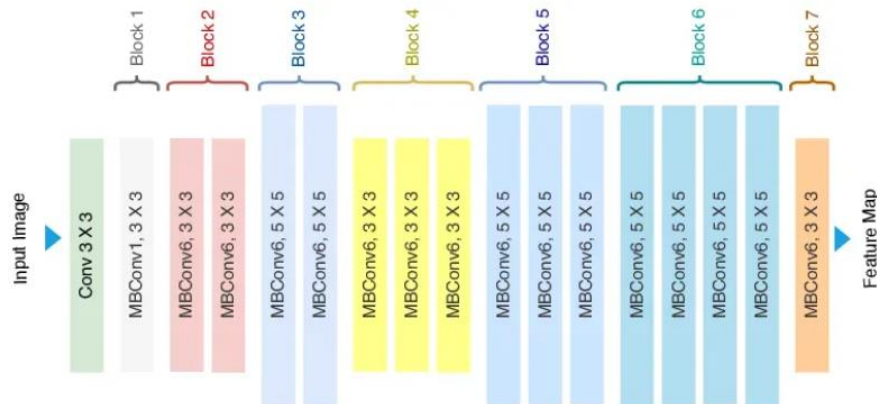


Figure C-1 : Architecture de EfficientNetB7, composée de différents blocks de MBconv « Mobile Inverted Bottleneck Convolution »

- **EfficientNetV2 (B3, M, L)** : EfficientNetV2 (montrée dans la figure C-2) est la deuxième génération d'EfficientNet, qui apporte des améliorations en termes de vitesse d'entraînement et d'efficacité. Les modèles EfficientNetV2 utilisent une technique d'échelle composée améliorée, ainsi que des optimisations de l'architecture du réseau pour une meilleure précision et une meilleure efficacité. EfficientNetV2B3 offre une performance améliorée par rapport à son équivalent

de première génération, EfficientNetB3. EfficientNetV2M (Medium) fournit un équilibre entre performance et coût computationnel, avec une profondeur et une largeur intermédiaire. EfficientNetV2L (Large) est le plus complexe et potentiellement le plus précis de cette sélection, avec une profondeur et une largeur encore plus grande.

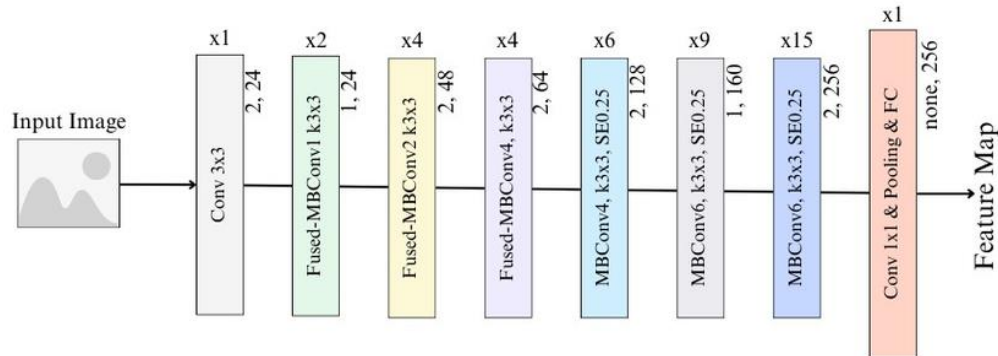


Figure C-2 : Architecture de EfficientNetV2, composée de différents blocks de Fused-MBconv « Mobile Inverted Bottleneck Convolution »

- **InceptionV3** : C'est un modèle de réseau neuronal convolutif qui utilise des modules de convolution parallèles pour capturer des caractéristiques à différentes échelles. Cette architecture (figure C-3) permet à InceptionV3 de détecter des motifs à diverses résolutions dans une même image. InceptionV3 utilise également des techniques d'optimisation telles que la factorisation de convolution et la régularisation auxiliaire pour améliorer l'efficacité et la précision.

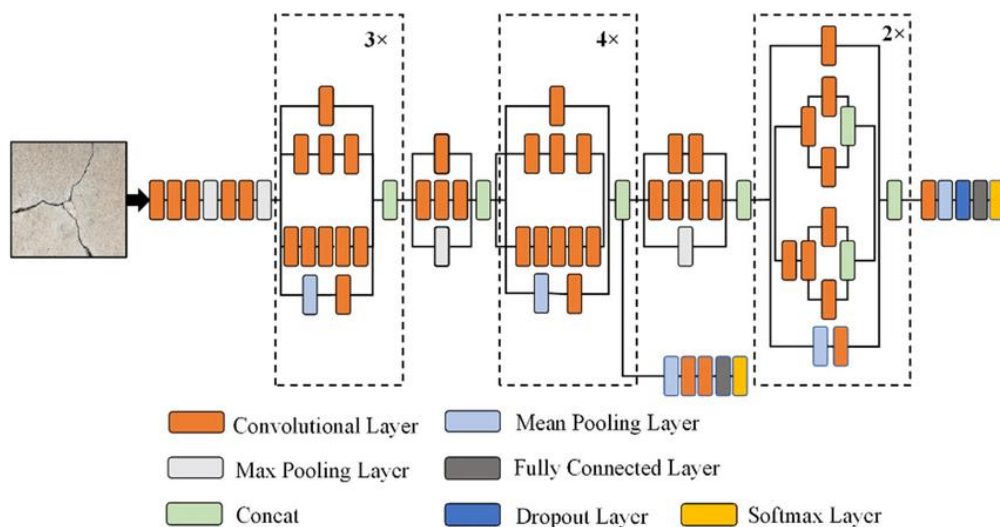


Figure C-3 : Architecture de InceptionV3

4. Technologies utilisées dans le déploiement

Pour réaliser cette application, nous avons choisi d'utiliser les technologies suivantes :

4.1 Electron

C'est un framework open-source permettant de créer des applications de bureau multiplateformes en utilisant des technologies web telles que HTML, CSS et JavaScript. Nous avons choisi Electron pour les raisons suivantes :

- **Multiplateforme** : Electron permet de développer une application unique qui fonctionne sur Windows, macOS et Linux.
- **Chromium** : Electron utilise Chromium pour le rendu des interfaces utilisateur, garantissant ainsi une performance rapide et une compatibilité avec les standards web modernes.
- **Node.js** : Electron intègre Node.js, ce qui permet d'utiliser des modules et des bibliothèques JavaScript côté serveur dans une application de bureau.
- **Écosystème riche** : Electron bénéficie d'un large écosystème de modules et de bibliothèques open-source, facilitant l'ajout de fonctionnalités supplémentaires et l'intégration de services externes.
- **Développement simplifié** : L'utilisation de technologies web familières (HTML, CSS, JavaScript) permet aux développeurs web de transitionner facilement vers le développement d'applications de bureau, réduisant ainsi la courbe d'apprentissage et augmentant la productivité.

4.2 API RESTful

4.2.1 Fonctionnalités

Pour assurer une communication efficace et standardisée entre le frontend (interface utilisateur) et le backend (logique de traitement et modèle de prédiction), nous avons implémenté une API RESTful. Une API (Application Programming Interface) RESTful est un ensemble de règles et de conventions permettant à différentes applications de communiquer entre elles de manière standardisée via des appels HTTP. REST (Representational State Transfer) est une architecture de services web qui repose sur des principes et des méthodes HTTP standard (GET, POST, PUT, DELETE) pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur les ressources.

Voici quelques avantages de cette architecture :

- **Séparation des responsabilités** : L'API RESTful permet une séparation claire entre le frontend et le backend. Le frontend se concentre sur l'interface utilisateur et l'expérience utilisateur, tandis que le backend gère la logique de traitement, l'accès aux données, et l'exécution des modèles de prédiction.
- **Standardisation** : En utilisant des méthodes HTTP standard (GET, POST, PUT, DELETE), l'API RESTful assure une communication cohérente et prévisible entre le frontend et le backend.
- **Flexibilité et scalabilité** : L'API RESTful permet d'ajouter facilement de nouvelles fonctionnalités et de modifier les existantes sans affecter le frontend. Cette flexibilité facilite l'évolution de l'application pour répondre à de nouveaux besoins ou intégrer de nouvelles technologies.
- **Interopérabilité** : L'API RESTful peut être utilisée par différentes applications et services, facilitant ainsi l'intégration avec d'autres systèmes et la collaboration avec des partenaires externes.
- **Sécurité** : En utilisant des pratiques de sécurité standard telles que l'authentification et l'autorisation via des tokens, l'API RESTful peut protéger l'accès aux données sensibles et aux fonctionnalités critiques de l'application.

4.2.2 Fonctionnement d'une API RESTful

Une API RESTful fonctionne en exposant des endpoints, c'est-à-dire des URL spécifiques, auxquelles les clients peuvent envoyer des requêtes HTTP. Chaque endpoint correspond à une ressource spécifique ou à une collection de ressources. Voici un aperçu du fonctionnement de base :

- **Requêtes HTTP** : les utilisateurs de l'application envoient des requêtes HTTP aux endpoints de l'API pour interagir avec les ressources : (**GET** : Récupérer des données (lecture), **POST** : Créer de nouvelles données (écriture), **PUT** : Mettre à jour des données existantes (mise à jour), **DELETE** : Supprimer des données (suppression))
- **Endpoints** : Chaque endpoint correspond à une ressource ou à une collection de ressources. Par exemple :
 - GET /images : Récupère une liste d'images
 - POST /images : Ajoute une nouvelle image
 - GET /images/{id} : Récupère une image spécifique par son identifiant

-
- PUT /images/{id} : Met à jour une image spécifique par son identifiant
 - DELETE /images/{id} : Supprime une image spécifique par son identifiant
- **Représentation des Ressources** : Les ressources sont généralement représentées en JSON (JavaScript Object Notation), un format léger et facile à lire/écrire pour les échanges de données.
 - **Statut des Réponses** : L'API RESTful renvoie des codes de statut HTTP pour indiquer le résultat de la requête : (**200 OK** : La requête a réussi, **201 Created** : Une nouvelle ressource a été créée, **204 No Content** : La requête a réussi mais ne retourne pas de contenu, **400 Bad Request** : La requête est mal formée, **404 Not Found** : La ressource demandée n'a pas été trouvée, **500 Internal Server Error** : Une erreur côté serveur s'est produite)

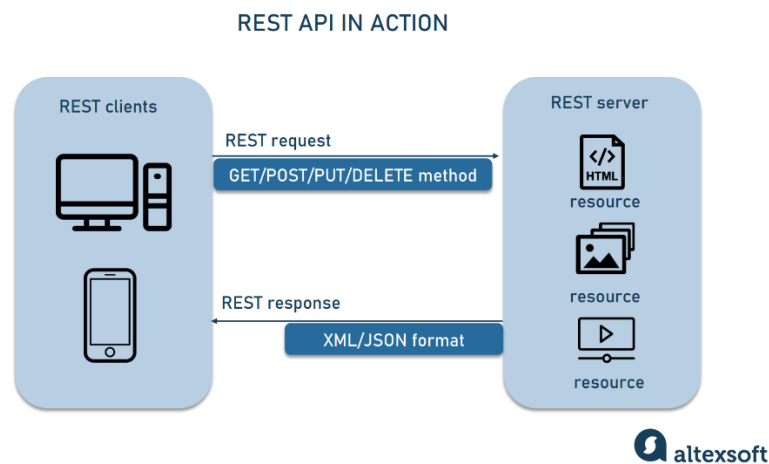


Figure C-4 : Fonctionnement du RESTful API

En combinant Electron pour le développement de l'interface utilisateur et une API RESTful pour la communication avec le backend, nous avons pu créer une application de bureau puissante, flexible et évolutive, capable de répondre aux besoins complexes de notre tâche.

Pour concrétiser cette vision, nous avons exploité les technologies web standard : HTML5 pour la structure, CSS3 pour le style et la mise en page, et JavaScript pour la logique et l'interactivité côté client.



Figure C-5 : Technologies Web utilisées

L'utilisation de ces technologies web au sein du framework Electron nous a permis de bénéficier à la fois de la familiarité et de la puissance des outils de développement web modernes, et des avantages d'une application de bureau native. HTML5 nous a fourni une base solide pour organiser le contenu de manière sémantique et accessible. CSS3, avec ses fonctionnalités avancées comme Flexbox et Grid, nous a permis de créer une interface réactive et esthétiquement plaisante. JavaScript, quant à lui, a joué un rôle crucial dans l'implémentation des interactions dynamiques et la gestion des communications avec notre API RESTful.