N° d'ordre : ....

UNIVERSITÉ MOHAMMED SEDDIK BENYAHIA
JIJEL
FACULTÉ DE SCIENCES EXACTES ET D'INFORMATIQUE

MEMOIRE DE MASTER

Présenté pour l'obtention du diplôme de :

MASTER

En **INFORMATIQUE**

Option : INFORMATIQUE LEGALE ET MULTIMEDIA

Thème

Sécurisation de la base des templates du réseau veineux

Présentée par :
BOUROMAH Chiraz
BOUAZI Sana

Encadrée par :
M. **BIROUK Wafa**

Année Universitaire : 2021/2022

Remerciements

Tout d'abord nous remercions de plus profond de nos cœurs ***ALLAH*** *le tout puissant, de nous avoir éclairées vers le bon chemin.*

Nous remercions en particulier notre encadrant Mademoiselle ***Wafa Birouk*** *pour sa grande disponibilité et ses précieux conseils, idée et ses encouragement tout au lang de rédaction de ce mémoire.*

Nos remerciements s'adressent aussi aux membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger et d'examiner notre travail, ainsi que tous les enseignants et tous les enseignantes de l'université de Jijel.

Sans oublier de remercier nos amis, nos proches et toutes les personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Et pour finir, nous présentons nos remerciements les plus sincères à nos chères familles, particulièrement à nos parents pour leur soutien inconditionnel et pour leur encouragement durant toutes nos années d'étude.

Dédicace

Je remercie et je loue tout d'abord Allah de m'avoir aidé à défier tous les obstacles, afin de compléter ce modeste travail. C'est ainsi que je dédie ce travail à :

A ma chère mère ***Hassina****, A mon cher père* ***Mouhemed***

Qui non jamais cessé, de formuler des prières à mon égard de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes chers frères ***Aymene****,* ***Hichem*** *et* ***Djamel Eddine***
mes chères sœurs ***Imen*** *et* ***Sara***

Pour son soutien moral et leurs conseils tout au long de mes études.

A mon cher binôme, ***Chiraz*** *pour son entente et sa sympathie.*

À toute la promotion Master2 informatique légale et multimédia 2021/2022.

À tous ce que j'aime et qui m'aiment.
À tous mes amis.

Sana

Dédicace

Je remercie et je loue tout d'abord Allah de m'avoir aidé à défier tous les obstacles, afin de compléter ce modeste travail

C'est ainsi que je dédie ce travail à :

Ma Tendre Mère

La lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ma maman chérie que j'adore.

Mon très cher Papa

Le guide de mes désirs, le donneur avec plaisir, à toi ma fierté et mon pouvoir, merci.

La mémoire de « mon grand-père », que dieu le garde dans son vaste paradis

***Mes très chères sœur et à mes chers frères**.*

Pour son soutien moral et leurs conseils tout au long de mes études.

*À Mon binôme **Sana** avec qui j'ai partagé la fatigue et les bons moments durant cette période de travail.*

À toute la promotion Master2 informatique légale et multimédia 2021/2022.

À tous ce que j'aime et qui m'aiment.
À tous mes amis.

Chiraz

Résumé

Les fonctions de hachage perceptuel sont fortement inspirées des fonctions de hachage cryptographique, elles se basent sur l'aspect visuel des données à hacher permettant d'établir une correspondance perceptuelle entre l'image originale et l'image d'authentifier. Les manipulations acceptables (compression, PEG, filtrage, rotation...) préservent l'aspect visuel de l'image authentifié par contre, les manipulations malveillantes (l'ajout de nouveaux objets, la suppression ou la modification majeure d'objets existants par exemple) changent le contenu sémantique de l'image. Ces dernières années ont vu beaucoup de chercheurs se pencher sur cette nouvelle approche de sécurité des données multimédia et la donnée biométrique comme les réseaux veineux de main. Par conséquent, la sécurisation de ce Template est nécessaire

Dans ce mémoire, nous essayons d'implémenter une technique de hachage perceptuel hautement sécurisée basée sur les caractéristiques fractales structurelles du codage d'image et sur la partition en anneau, afin de construire un Template hautement sécurisé pour un utilisateur

Mots-clés *: Réseau veineux de main, Hachage perceptuel, Codage fractale, manipulations acceptables, manipulations malveillantes*

Abstract

Perceptual hash functions are strongly inspired by cryptographic hash functions, they are based on the visual aspect of the data to be hashed, making it possible to establish a perceptual correspondence between the original image and the image to authenticate. Acceptable manipulations (compression, PEG, filtering, rotation, etc.) preserve the visual aspect of the authenticated image, on the other hand, malicious manipulations (the addition of new objects, the deletion or major modification of existing objects by example) change the semantic content of the image. In recent years, many researchers have looked into this new approach to multimedia data security and biometric data such as hand vein networks. Therefore, securing this Template is necessary

In this thesis, we try to implement a highly secure perceptual hashing technique based on the structural fractal characteristics of image coding and on the ring partition, in order to build a highly secure Template for a user

Keywords : *Hand vein network, Perceptual hashing, Fractal coding, acceptable manipulations, malicious manipulations.*

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTES DES ACRONYMES

ADN	**A**cide **D**esoxyribo **N**ucleique
TFA	**T**aux **F**aux **A**cceptation
TFR	**T**aux **F**aux **R**ejet
TEE	**T**aux **E**rreur **E**gal
IR	**I**dentification **R**ate
FNIR	**F**alse-**N**egative **I**dentification-**E**rreur **R**ate
FPIR	**F**alse-**P**ositive **I**dentification-**E**rreur **R**ate
GAR	**G**enuine **A**ccept **R**ate
CN	**C**rossing **N**umber
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
DCT	**D**iscrete **C**osine **T**ransform
DWT	**D**iscrete **W**avelet **T**ransform
SHA	**S**ecure **H**ash **A**lgorithm
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
DH	**D**istance de **H**amming
SVD	**S**ingular **V**alue **D**ecomposition
RGB	**R**ed **G**reen and **B**lue
SRM	**S**pectral **R**esidual **M**odel
LRR	**L**ow-**R**ank **R**epresentation
RP	**R**ing **P**artition
QGT	**Q**uaternion **G**yrator **T**ransform

INTRODUCTION GÉNÉRALE

Aujourd'hui, les nouveaux développements technologiques ont grandement valorisé nos modes de communication et d'échange d'informations. Cette évaluation explosive des nouvelles technologies de l'information nous a permis d'échanger facilement et rapidement des informations sous toutes ces formes : texte, audio et vidéo, sur des réseaux publics de plus en plus larges. Par exemple, les échanges en ligne font désormais partie de notre quotidien, nous permettant de réaliser diverses opérations comme acheter sur une boutique en ligne, passer des commandes bancaires ou encore tout simplement échanger des données multimédias avec des contenus personnels. Par conséquent, le grand nombre de ces applications rend urgent le besoin de communications plus sécurisées, car il motive les fraudeurs et l'aide à vaincre les systèmes de sécurité actuels.

Malheureusement, il n'existe pas d'outils puissants qui s'adapteront aux nouvelles menaces en temps opportun. La vérification de l'intégrité et de l'authenticité des images numériques est une étape essentielle en raison de la haute qualité des informations visuelles qu'elles transmettent. En fait, ces données sont faciles à pirater, modifier et redistribuer sans perte notable de qualité. La protection est une nécessité incontournable si l'on veut assurer la qualité des services rendus. Dans le domaine de la sécurité multimédia, un type d'approche très populaire a été proposé ces dernières années pour répondre à ces exigences, il s'agit ici du hachage perceptuel.

Dans ce mémoire nous intéressons aux fonctions de hachage perceptuel pour la sécurisation et le contrôle d'intégrité des images numériques des réseaux veineux de main. Les fonctions de hachage perceptuel sont inspirées des fonctions de hachage cryptographie pour authentifier les données multimédia. Traditionnellement, la vérification d'intégrité des données est traitée par des fonctions de hachage cryptographique, qui sont très sensibles à chaque bit du message d'entrée. Par conséquent, les images peuvent subir des manipulations acceptables tel que (Filtrage, rotation...). L'intégrité du message n'est validée que lorsque chaque bit du message est inchangé. Cela présente un inconvénient majeur des techniques cryptographiques pour authentifier les images. En d'autres termes, l'authentification des images devrait se baser sur leurs contenus visuels et non pas sur leurs contenus binaires, c'est le rôle de fonction de hachage perceptuel. Quand nous voulons vérifier l'authenticité d'un objet multimédia' les signatures de hachage de ce dernier et de l'objet original sont comparés en utilisant des fonctions prédéfinies. Ces fonctions renvoient une distance ou score de similarité entre deux signatures de hachage perceptuel où

la dernière décision est basée sur un seuil choisi.

Les fonctions de hachage perceptuel sont des solutions potentielles, dans ces cas-là, elles permettent d'établir une "correspondance perceptuelle" entre l'image originale et l'image à authentifier.

Dans ce contexte, ce manuscrit se compose de quatre chapitres principaux délimités par une introduction générale, une conclusion et des perspectives. Le premier chapitre est consacrée à la présentation générale de la biométrie, ainsi vu détaillée sur le système de reconnaisse de réseau veineux de main. Dans le deuxième chapitre nous présentons les différents techniques de protection d'images. Puis nous décrivons quelques techniques de hachage perceptuel des images qui s'existent dans la littérature, avec une description détaillée de la méthode à implémenté dans notre travail dans le troisième chapitre. Ce mémoire composée des chapitres suivants :

Chapitre 1 : Ce chapitre expose des généralités sur la biométrie. Il présente la définition liée à la biométrie, leur caractéristique et les principes modalités morphologiques et comportementales et biologiques, et leur Architecture et modes de fonctionnements, et les mesure de performance. Enfin, les applications de la biométrie dans les domaines commerciales et gouvernementales et légale et détaillé sur le système de reconnaissance de réseau veineux de main .

Chapitre 2 : Dans ce chapitre, nous présentons les différentes techniques permettant d'assurer la protection des données comme (la cryptographie, le filigrane), un service de contrôle d'intégrité et de sécurisation des images naturel par le hachage perceptuel .

Chapitre 3 : Dans ce chapitre, nous présentons une vue globale sur les différentes méthodes de hachage perceptuel qui se divisent en méthode de décomposition en bloc, et en méthode globale. Puis nous détaillons la méthode de hachage perceptuel d'image à base des caractéristiques fractales structurelles qui sera appliquée dans notre travail et qui doit être robuste et sure.

Chapitre 4 : Ce chapitre est consacré essentiellement à la présentation de différentes interfaces de l'application réalisée,ainsi que les résultats des testes effectués sur cette application.

Enfin nous clôturons ce mémoire par conclusion générale et perspective.

CHAPITRE 1

SYSTÈME RECONNAISSANCE BIOMÉTRIQUE DE RÉSEAU VEINEUX DE MAIN

1.1 Introduction

A notre époque, la sécurité des personnes est devenue une préoccupation majeure. Le besoin en augmente de jour en jour. Les méthodes de sécurité des systèmes d'information classiques ne sont plus efficaces aujourd'hui. Le scientifique a donc eu recours à la biométrie, qui est une méthode émergente qui permet de vérifier l'identité d'un individu à travers l'utilisation de ses caractéristiques personnelles. Aujourd'hui, cette technique est de plus en plus utilisée pour prouver la reconnaissance de personnes dans un grand nombre de différents classements.

Dans ce chapitre, nous allons commencer par une présentation de quelques généralités sur la biométrie, telles que (leur définition, leurs caractéristiques, et leur principales modalités et Architecture des systèmes biométriques et modes de fonctionnements.) puis nous détaillons les étapes de la reconnaissance par la technique de réseau veineux de la main .

1.2 Système reconnaissance biométrique

1.2.1 Définition de la biométrie

Le terme de biométrie est originaire d'une contraction des deux anciens termes grecs : « bios »qui signifié : la vie « métrique » qui se traduit par : mesure. C'est-adire « mesure du vivant »

La biométrie est une mesure des caractéristiques biologiques pour l'identification ou l'authentification d'un individu à partir de certaines de ses caractéristiques : comportementales (exemple de la dynamique de frappe au clavier), physiques ou physiologiques (exemple de l'ADN) .Cette technique est utilisée de plus en plus aujourd'hui pour établir la reconnaissance des personnes dans un grand nombre d'applications diverses. [1]

1.2.2 Les Caractéristiques biométriques

Ces éléments sont considérés comme biométriques car ils constituent un ensemble de données qui comprend toutes les informations de base permettant de reconnaître et de distinguer les personnes [2] [3] :

- **Universelle** : existe chez tous les individus.
- **Unique** : différente pour chaque individu.
- **Permanente** : stable dans le temps.
- **Mesurables** : non coûteuse et non intrusives.
- **Utilisable** : acceptation par l'utilisateur.
- **Non imitable** : difficilement copiable.

.

1.2.3 principales modalités biométriques

Il 'y un très grand nombre de modalités biométriques nous pouvons distinguer trois grandes catégories. Le tableau 1.1 représente classification de la biométrie :

- **La biométrie physiologique ou morphologique :**

Cette catégorie est basée sur l'utilisation de parties du corps humain telles que les empreintes digitales, les veines et la reconnaissance de l'iris. Ces facteurs ont l'avantage d'être stables dans la vie d'une personne. [4]

- **La biométrie comportementale :**

Elle repose sur une analyse de certaines caractéristiques personnelles du comportement d'une personne. Il s'agit d'une étude comportementale répétitive et générale menée par des personnes. Méthode de saisie au clavier, résumé de signature, méthode de marche, etc. [5] [6]

- **L'analyse des traces biologiques :**

L'authentification est basée sur les facteurs biologiques d'un individu (ADN, salive, odeur, etc.). Ces éléments sont très complexes à mettre en œuvre et sont réservés exclusivement à des applications médico-légales ...

Biométrie physiologique	**Biométrie comportementale**	**L'analyse des traces biologiques**
-Les empreintes digitales -La géométrie de la main (hand-scan) -L'iris -La rétine -La reconnaissance faciale -L'oreille -Veines	-Reconnaissance vocale -Reconnaissance de la dynamique de la frappe au clavier -Reconnaissance de la dynamique de signature -La démarche	-Reconnaissance de l'ADN -Reconnaissance de l'odeur -Salive -Cheveux

TABLE 1.1 – Classification de biométrie

1.2.3.1 la biométrie physiologique ou morphologique

• **Les empreintes digitales** : Est une série de lignes, de fourches et de points qui forment des motifs différents pour chaque individu. Cette dernière est analysée en fonction de caractéristiques qui sont des points spécifiques de l'empreinte digitale. La chose la plus importante pour la reconnaissance d'empreintes digitales est les détails de l'empreinte digitale.

Les empreintes digitales se répartissent en trois catégories principales : les arcs, les tourbillons et les boucles (Figure 1.1). Chacune de ces catégories comporte de nombreux éléments qui nous distinguent. En plus des cicatrices, il existe des fourches, des îles et des espaces qui confèrent aux impressions un caractère unique.

FIGURE 1.1 – L'empreinte digitale.

• **La géométrie de la main** : la technologie biométrique utilisant la mesure manuelle est encore l'une des technologies les plus populaires aujourd'hui. Une "mesure" de la main se compose de plusieurs mesures, telles que les dimensions des doigts, les caractéristiques des articulations, les paumes et leur forme. .. La technologie associée à cela est principalement l'imagerie infrarouge. [9]

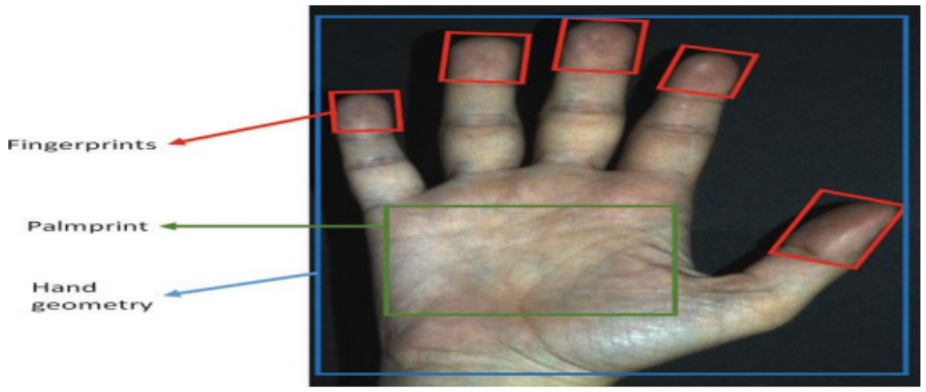


FIGURE 1.2 – Aperçu schématique de l'empreinte palmaire ainsi que des autres traits biométriques de la main, en fonction de la taille de la zone analysée.

• **L'iris** : C'est une zone unique en forme d'anneau entre la pupille et le blanc de l'œil. L'iris a une structure extraordinaire et possède de nombreuses textures propres à chaque personne. La biométrie avec cette fonctionnalité est la plus récente et la plus efficace. [10]

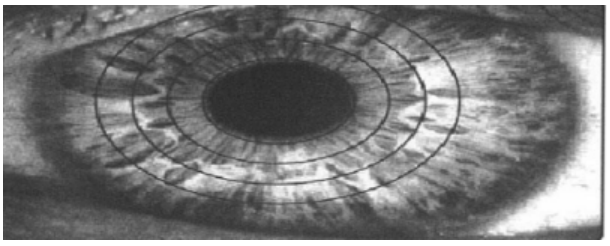


FIGURE 1.3 – Texture de l'iris.

• **La rétine** : Les mesures devant être effectuées à très courte distance du capteur, l'utilisation est très faible et mal acceptée par le grand public (Figure 1.4). Le motif formé par les veines sous la surface de la rétine est unique et stable dans le temps. Ne pouve affecté que par certaines maladies. Pour ces raisons, la reconnaissance rétinienne est actuellement l'une des méthodes biométriques les plus sûres [12]

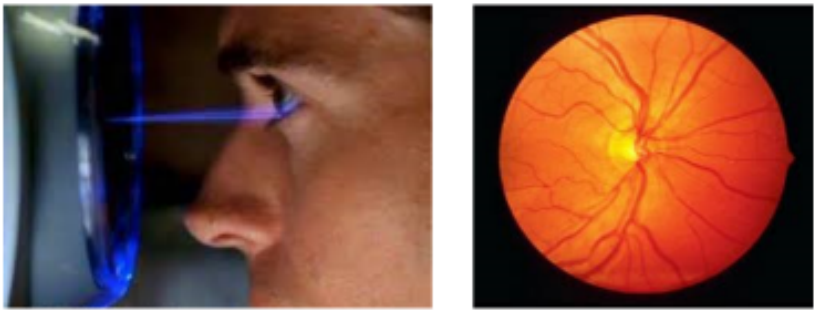


FIGURE 1.4 – Reconnaissance de la rétine.

• **La reconnaissance faciale** : Rien n'est plus naturel que d'utiliser votre visage pour identifier une personne. C'est la biométrie la plus courante et la plus populaire. Il couvre une variété de mesures des traits du visage et dépend moins de facteurs de nature

variable, tels que l'utilisation de coupes de cheveux et de produits cosmétologiques. Néanmoins, les visages humains peuvent changer avec le temps, et cette réalité continue d'être un défi pour les systèmes de reconnaissance faciale tels que le changement d'expression, la maladie, la vieillesse et d'autres facteurs normaux. en autre, les facteurs humains et environnement joueront un très grand rôle dans l'efficacité d'un système de reconnaissance faciale. [12]

• **Veines :** Le motif des veines sur les doigts ou les paumes sert de critère pour authentifier les personnes. Grâce à la caméra grand angle intégrée au scanner infrarouge, le système enregistre la structure veineuse et donc l'identité unique de la personne en quelques millisecondes seulement. Plus d'informations sur cette technologie biométrique dans le prochain chapitre. La figure 1.5 représente ce système biométrique

FIGURE 1.5 – Système biométrique basé sur les veines de la main.

1.2.3.2 La biométrie comportementale

• **Reconnaissance vocale** : Cette technique peut être forgée très facilement en utilisant l'enregistrement. Les mesures biométriques de la voix traitent les données de facteurs physiologiques qui dépendent de facteurs comportementaux tels que l'âge, le sexe, le ton de la voix, l'accent, la vitesse et le rythme. Ces facteurs ont l'avantage d'être stables dans la vie d'un individu [4].

• **Reconnaissance de la dynamique de la frappe au clavier** : La dynamique de frappe varie d'une personne à l'autre. C'est une sorte de graphologie moderne car on écrit plus souvent avec le clavier qu'avec le stylo. Les éléments analysés sont : Vitesse de saisie, séquence de caractères, temps de saisie, pause... Le but est de développer une solution biométrique économique pour renforcer la sécurité des mots de passe qui ne cessent de croître au quotidien

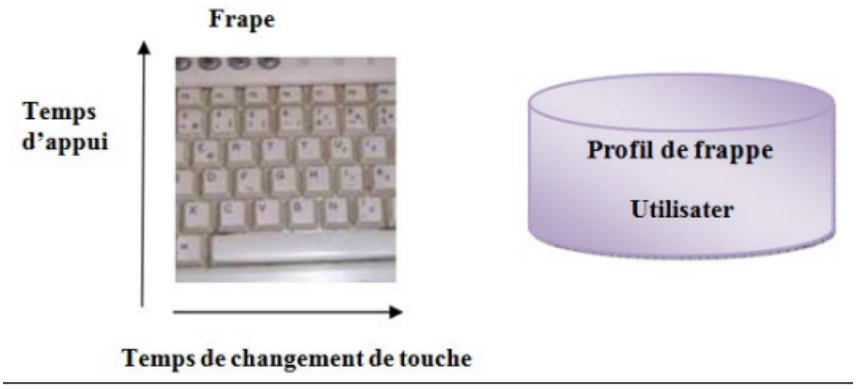


FIGURE 1.6 – Reconnaissance de la dynamique de la frappe au clavier.

• **Reconnaissance de la dynamique de signature** : Ce système d'identification impose à l'utilisateur de se connecter à la tablette graphique à l'aide d'un stylo électronique comme le montre la figure 1.7. Le système analyse ensuite les changements de vitesse, d'accélération et de pression du stylet. Il justifie son intégrité ou permet de les confondre devant un document signé

FIGURE 1.7 – Reconnaissance de la dynamique de signature.

• **La démarche** : Chacun a une méthode de marche spécifique. Vous pouvez identifier le type de mouvement des jambes, des bras ou des articulations, ou un mouvement spécifique reçu d'une caméra vidéo et l'envoyer à un ordinateur pour analyse afin de déterminer la vitesse et l'accélération de chaque individu...

1.2.3.3 L'analyse des traces biologiques

• **Reconnaissance de l'ADN** : Il se produit dans les cellules du corps et est individuellement spécifique et peut être identifié de manière fiable à l'aide de simples fragments de peau, de traces de sang ou de gouttelettes de salive. Actuellement, le temps et le coût requis pour l'analyse limitent son utilisation en dehors de l'identification médico-légale.

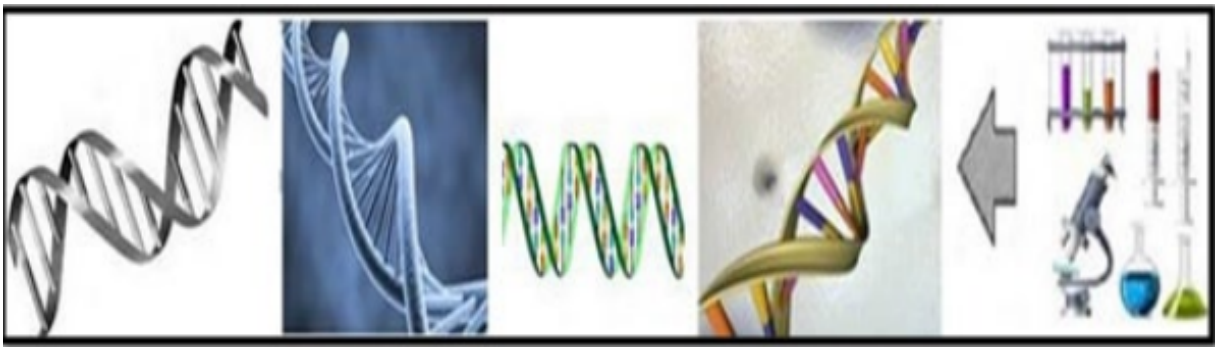


FIGURE 1.8 – système reconnaissance de L'A.D.N

• **Reconnaissance de l'odeur** : Chacun dégage une odeur spécifique définie par la composition chimique. Ce système biométrique basé sur les modalités analyse ces composants pour extraire des données comparatives.

1.2.4 Architecture des systèmes biométriques et modes de fonctionnements

Un système biométrique est essentiellement un système de reconnaissance de formes. Le système enregistre les fonctions biométriques, construit des modèles, compare ces modèles aux fonctions précédemment stockées dans la base de données, et enfin prend des mesures ou décide en fonction des résultats de cette comparaison. Il fonctionne en vous permettant de baisser.

1.2.4.1 Architecture des systèmes biométriques

L'architecture d'un système biométrique contient cinq modules comme le montre la figure 1.9 [12] :

1. **Module d'acquisition ou capture** : Le module d'acquisition peut mesurer la fonction biométrique d'origine à l'aide d'une caméra, d'un lecteur d'empreintes digitales, d'une caméra de sécurité, etc.
2. **Module de prétraitement** : Il consiste en un prétraitement et une atténuation du bruit, et en l'application d'une série d'opérations continues (filtrage, normalisation, etc.) pour révéler des paramètres pertinents et utiles.
3. **Module d'extraction des caractéristiques** : Il est utilisé pour représenter les données biométriques prétraitées à l'étape précédente à l'aide d'une nouvelle représentation ou dite modèle. Ces modèles sont obtenus en extrayant les caractéristiques les plus pertinentes. Idéalement, ces modèles devraient être uniques à chacun et relativement constants aux changements dans la classe.
4. **Module du stockage** : Contient tous les modèles biométriques des utilisateurs enregistrés du système. Fondamentalement, les informations stockées ne sont pas l'image d'origine, mais un modèle mathématique des éléments qui distinguent un échantillon biométrique d'un autre.
5. **Module de Matching et de décision** : Il s'agit de l'étape finale au cours de laquelle vous pouvez calculer la similitude entre le modèle et la ligne de base, puis prendre les décisions appropriées en fonction des besoins de votre application.

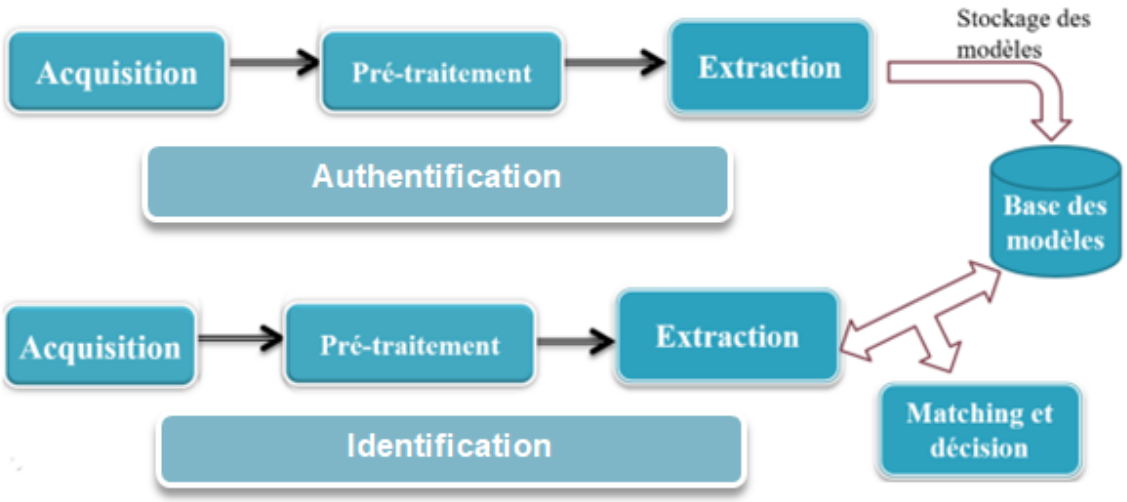


FIGURE 1.9 – Architecture d'un système biométrique.

1.2.4.2 Modes de fonctionnement

Les systèmes biométriques peuvent fonctionner en deux modes principaux : l'authentification (vérification) et l'identification. Il existe une étape avant les deux modes précédents qui s'appelle "l'enrôlement". [9]

1. **Enrôlement** : Est une phase d'apprentissage visant à collecter des informations biométriques sur la personne identifiée. Il s'agit de la première étape au cours de laquelle un utilisateur est enregistré dans le système et une ou plusieurs modalités biométriques sont enregistrées et stockées dans la base de données (voir Figure 1.10), qui enregistre l'ajout d'informations biométriques à la base de données.
2. **Authentification** : Active la preuve d'identité demandée par l'utilisateur (voir Figure 1.11). Le système doit répondre à des questions telles que «Suis-je bien la personne que je prétends être ? » Techniquement, l'appareil est un code (identifiant) qui est soit tapé au clavier, soit lu par l'échantillon biométrique fourni en passant un identifiant (carte à puce, carte magnétique, proximité, etc.) correspondant au modèle spécifié.
3. **Identification** : Vous pouvez vérifier si l'identité de la personne qui vous présente existe dans la base de données de référence (voir Figure 1.12). Le système doit deviner l'identité de l'individu. Répondez au type de question « Qui suis-je ? » L'appareil recherche dans la base de données le modèle correspondant à partir de l'échantillon biométrique fourni.

Par conséquent, l'identification et l'authentification sont deux sujets différents. L'identification peut être une tâche ardue si la base de données contient des millions d'identités, surtout si le système est contraint en temps réel. Ces problèmes sont similaires à ceux que les systèmes d'indexation de documents multimédias tendent à résoudre. [14]

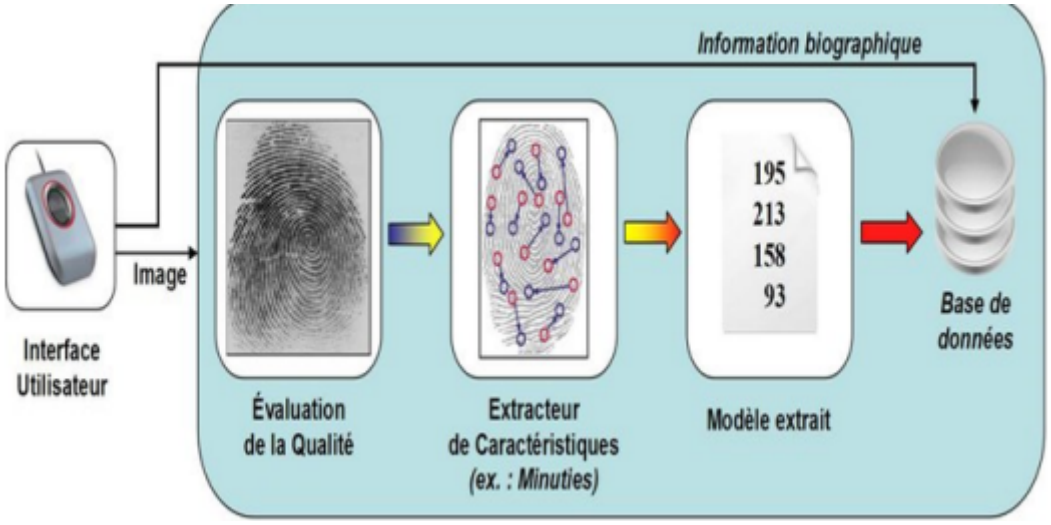


FIGURE 1.10 – Enrôlement d'une personne dans un système biométrique.

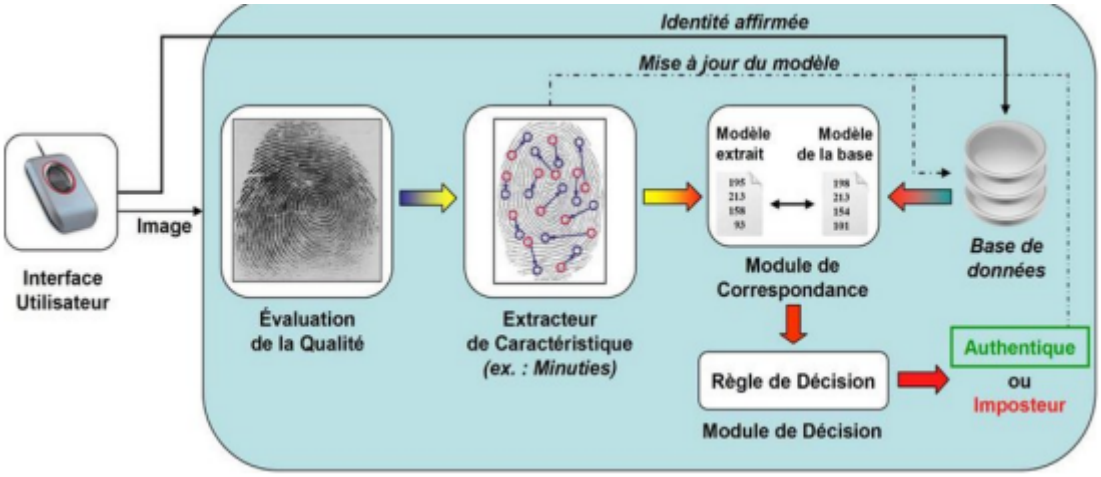


FIGURE 1.11 – Authentification d'un individu dans un système biométrique.

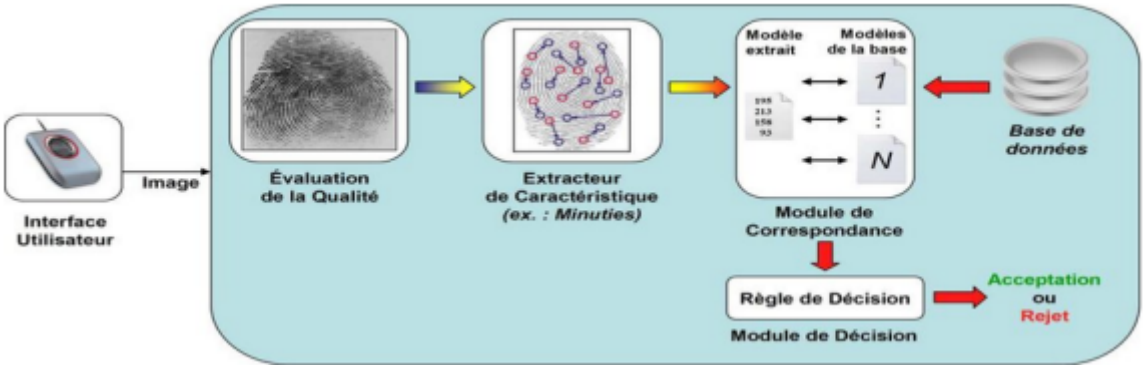


FIGURE 1.12 – Identification d'un individu dans un système biométrique.

1.2.5 Mesure de performance d'un système biométrique

1.2.5.1 Taux d'erreur de système d'authentification

Les systèmes d'authentification sont généralement évalués par le taux de faux rejets et le taux de fausses acceptations. Le système d'identification peut être évalué sur la base du taux d'identification, du taux de faux négatifs d'identification, du taux de faux positifs d'identification et de l'erreur d'algorithme de présélection.

Dans taux d'erreur du système d'authentification Plusieurs métriques peuvent être trouvées pour mesurer les performances d'un système biométrique particulier lors de l'authentification. Les plus importants sont : Taux de fausse acceptation (TFA), taux de faux rejet (TFR) et taux d'erreur égal (TEE) comme le montre dans la figure 1.13 [16] :

1. **TFA (FAR)** : taux d'acceptation inexact (« taux d'acceptation inexact » ou FAR). Ce taux représente le pourcentage de personnes qui ne devraient pas être reconnues, mais qui sont quand même acceptés par le système.

$$TFA = \frac{\text{Nombre imposteurs acceptes (FA)}}{\text{Nombre total d'axes imposteur}}$$

2. **TFR (FRR)** : Taux de Faux Rejets, ("False Reject Rate" ou FRR). Ce taux représente le pourcentage de personnes qui devraient être approuvées mais rejetées par le système :

$$TFR = \frac{\text{Nombre de client rejetées}}{\text{Nombre total d'acces clients}}$$

3. **TEE (EER)** : Taux d'Erreur Egale, ("Equal Error Rate" ou EER). Donne un point sur lequel : TFA = TFR.

$$TEE = \frac{\text{Nombre de fausses acceptations (FA)} + \text{Nombre de faux rejets (FR)}}{\text{Nombre total d'acces}}$$

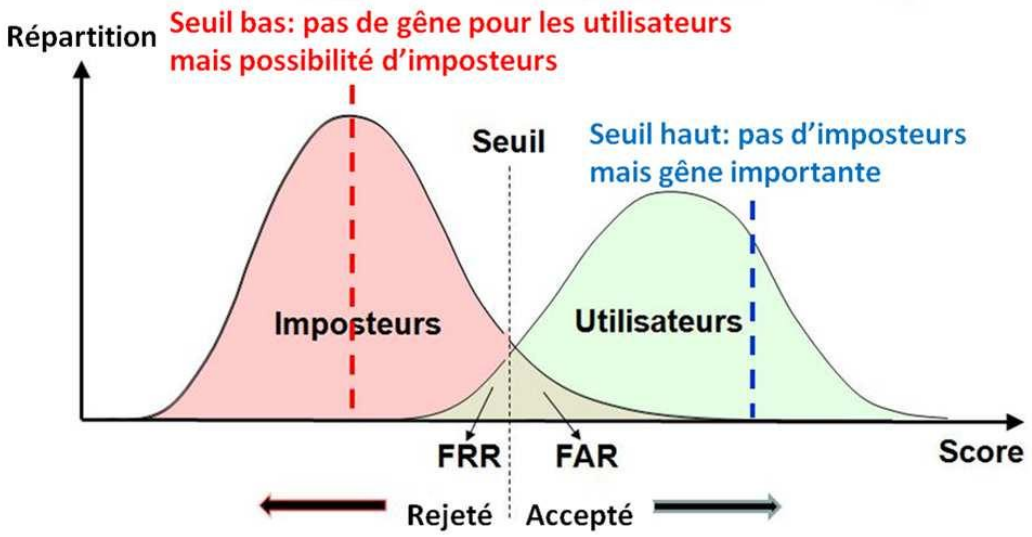


FIGURE 1.13 – Taux de vraisemblances des utilisateurs légitimes et des imposteurs d'un système biométrique (dont la comparaison est basée sur le calcul d'une similarité).

1.2.5.2 Taux d'erreur de systèmes d'identification

Dans le cas des systèmes d'identification, on peut trouver les taux suivants [6] [10] :

1. **Taux d'identification (identification rate, IR)** : Appelé aussi « taux de reconnaissance ". Il s'agit du taux d'identification de rang-1 (Rank-1). Il représente le pourcentage de véritables tentatives d'identification pour lesquelles l'enregistrement correct est spécifié dans la liste des identifiants.

$$Rang - 1 = \frac{N_i}{N}.100\%$$

   Où N_i représente le nombre d'images attribuées avec succès à l'identité correcte (bien classées) et N représente le nombre total d'images essayant d'assigner une identité .

2. **Taux de faux-négatif d'identification (false-négative identification-erreur rate, FNIR)** : Proportion de transactions d'identification, par des utilisateurs enrôlés dans le système, pour lesquels l'identifiant de l'utilisateur ne figure pas dans la liste des identifiants retournée.
3. **Taux de faux-positif d'identification (false-positive identification-erreur rate, FPIR)** : Proportion de transactions d'identification, par des utilisateurs non enrôlés dans le système, pour lesquels la liste des identifiants retournée est non vide.
4. **Erreur de l'algorithme de présélection (pre-selection error)** : L'algorithme de présélection réduit le nombre de modèles biométriques qui sont comparés aux images prises lors de la phase d'identification. Des erreurs d'algorithme de présélection se produisent lorsque le modèle correspondant aux données biométriques capturées ne figure pas dans la liste de modèles renvoyée.

1.2.5.3 GAR (Genuine Accept Rate)

C'est le taux des véritables clients acceptés par le Système biométrique. GAR est calculé par l'équation [16] :

$$TFA = GAR(T) = 1 - FRR(T)$$

1.2.6 Type d'application

La technologie de la biométrie est utilisée dans plusieurs domaines, et son champ d'application peut inclure tous les domaines de sécurité où les personnes. Peut être divisée en trois groupes principaux.

- **Applications commerciales**

Ouverture du réseau informatique, sécurité des données électroniques, commerce électronique, accès Internet, cartes de crédit, contrôle d'accès physique, téléphones mobiles, gestion des dossiers médicaux, formation à distance, etc.

- **Applications gouvernementales**

Carte d'identité, permis de conduire, sécurité sociale, contrôle aux frontières, gestion des passeports, etc.

- **Applications légales**

Nous citons : Identification personnelle, enquête criminelle, identification terroriste, l'identification de cadavre, l'analyse de l'ADN.

FIGURE 1.14 – Les applications des systèmes biométriques.

1.3 Système de reconnaissance de réseau veineux de main

1.3.1 Anatomie de réseau veineux de la main

Le réseau veineux dorsal de la main est un réseau de veines du fascia superficiel du dos de la main formé par les veines métacarpiennes dorsales. Il est situé sur le dos de la main et se connecte à des veines telles que la veine céphalique et la veine basilique.

1.3.2 Les étapes de reconnaissance de système de réseau veineux de la main

Nous utilisons dans cette partie la référence [17]

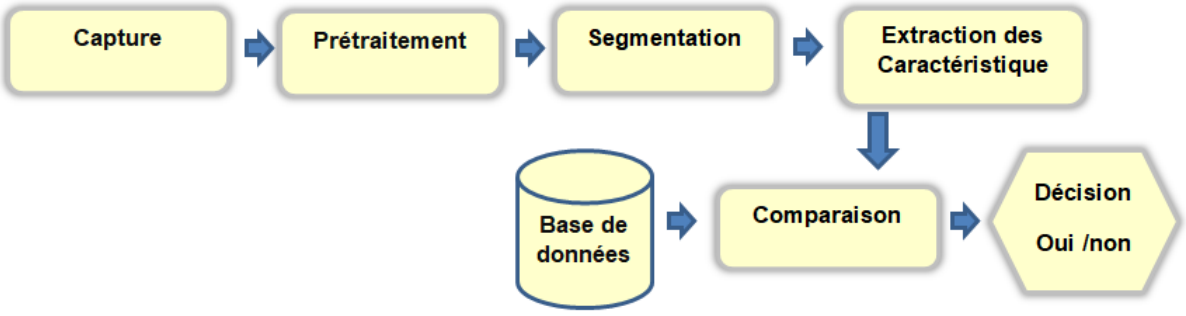


FIGURE 1.15 – Le processus de reconnaissance des veines.

1.3.2.1 L'acquisition d'image

Système de détection via le réseau veineux de la main. La première étape consiste à prendre une photo. Le but de cette étape est de prendre une photo de main.

Le réseau veineux de la main est sous la peau et est invisible à l'œil humain. Par conséquent, il n'est pas possible de filmer avec de la lumière visible dans la bande de longueur d'onde de 400 à 700 nm. Le schéma vasculaire de la main peut être détecté par la lumière proche infrarouge (NIR), qui occupe une bande de longueur d'onde d'environ 800 à 1000 nm, et peut pénétrer la peau à cette longueur d'onde.

L'hémoglobine oxydée dans les vaisseaux sanguins absorbe les rayons infrarouges plus fortement que la peau [18], et les veines apparaissent sous forme de lignes noires dans les images enregistrées avec une caméra CCD (dispositif à couplage de charge) photosensible.

A la sortie de la caméra CCD, l'image est disponible en niveaux de gris avec une résolution de 8x8 bits par pixel. La figure 1.16 montre un exemple d'image de main prise avec la lumière visible et infrarouge. [19]

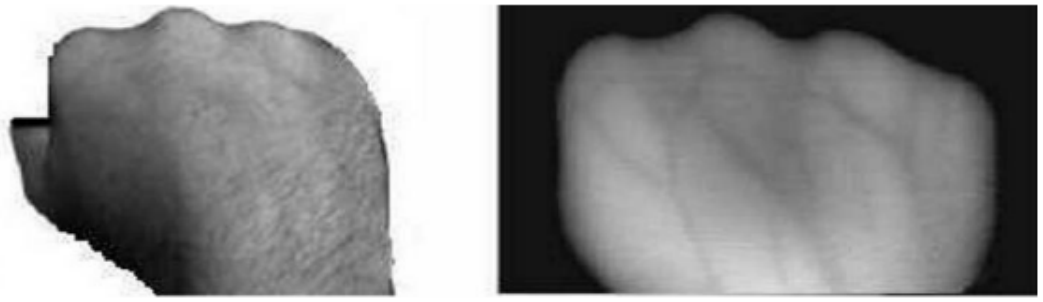


FIGURE 1.16 – L'image de la main obtenue par la lumière visible (à gauche) et la lumière infrarouge (à droite).

Pour capturer une image du réseau veineux sous lumière infrarouge NIR, le scanner utilise une série de LED pour émettre de la lumière et éclairer la main. Prenez une image à l'aide d'une caméra CCD photosensible infrarouge NIR. Un filtre infrarouge NIR placé devant la caméra CCD est utilisé pour bloquer la lumière visible indésirable provenant de sources lumineuses externes.

Il existe deux méthodes d'appareils d'imagerie pour capturer des images de réseaux veineux :

1. **Méthode de transmission** : la matrice LED est placée sur la main et la caméra CCD sur le côté opposé de la matrice LED de la main pour capturer la lumière que la caméra CCD traverse la main. La figure 1.17 montre la configuration du réseau de LED illuminées et de la caméra CCD.

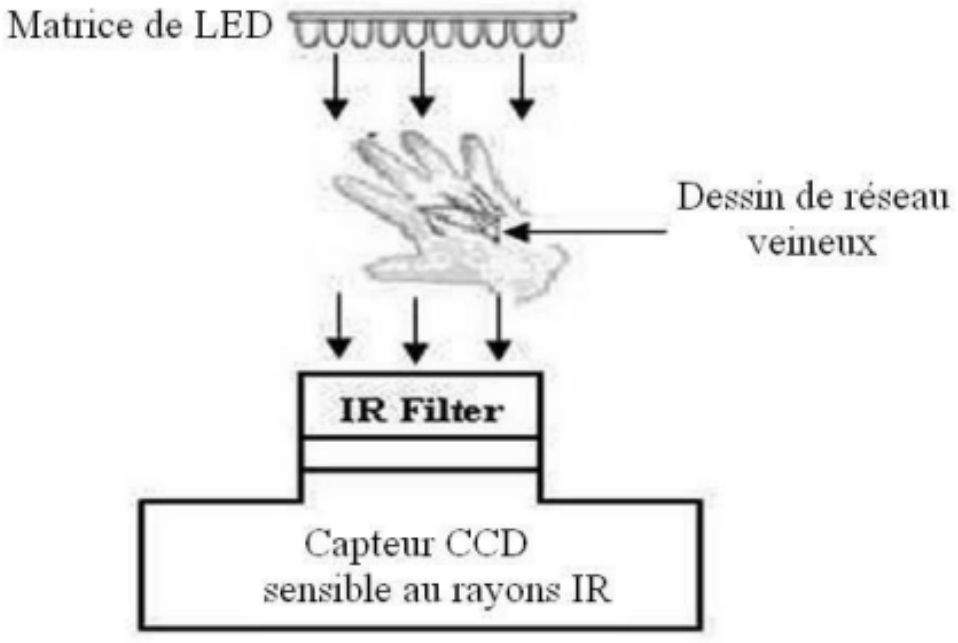


FIGURE 1.17 – Configuration de la méthode d'acquisition basée sur la transmission.

2. **Méthode de réflexion** : Ici la matrice de LED et la caméra CCD sont positionnés au même endroit de la main, la caméra CCD capture la lumière qui est réfléchie par la main. La Figure 1.18 montre la configuration de la matrice de LED d'éclairage et la caméra CCD.

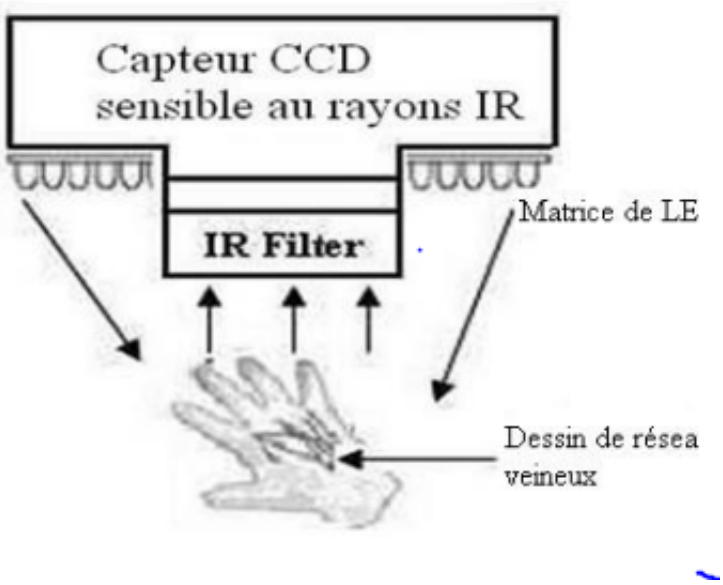


FIGURE 1.18 – Configuration de la méthode d'acquisition basée sur la réflexion .

La méthode de réflexion est préférable car la méthode de transmission est souvent sensible aux fluctuations de la transmission lumineuse à la main, qui sont facilement affectées par la température. Une autre raison pour laquelle la méthode de réflexion est préférée est sa construction simple puisque l'éclairage LED et la caméra CCD au même endroit facilitent l'intégration de le système dans un petit appareil

1.3.2.2 Prétraitement

La deuxième étape consiste à prétraiter cette image capturée. Au cours de cette étape, vous utiliserez diverses techniques pour extraire de la main un échantillon approprié du réseau veineux.

Le prétraitement des images est généralement la première étape importante dans de nombreux systèmes de reconnaissance de formes. La section suivante présente les bases du traitement d'images et les techniques utilisées dans cette étape.

C'est parce que le système d'enregistrement produit beaucoup de bruit dans l'image enregistrée. L'amélioration de l'image a deux objectifs principaux. Le premier est le lissage et la réduction du bruit. La seconde est l'amélioration du contraste. Ceci est nécessaire car le motif des veines peut être faible.

1.3.2.2.1 Lissage et suppression du bruit : Il existe plusieurs façons de traiter le bruit d'image. Certaines méthodes tirent parti du fait que le bruit est une variable aléatoire avec une moyenne de 0 et est ajouté à l'image. Une autre méthode annule les effets du bruit en faisant la moyenne des images. Malheureusement, cela produit une image avec des tâches où les petites veines peuvent être perdues dans le processus .

Le filtre gaussien : Un filtre gaussien est un filtre de lissage basé sur une distribution gaussienne. Il agit comme un filtre passe-bas qui atténue le bruit à haute fréquence sans modifier la basse fréquence, ce qui aide à éliminer le bruit de l'image. Il est défini dans une dimension comme :

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{1.1}$$

Ou σ est l'écart type.
Et en deux dimensions :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.2}$$

Les figures 1.19 et 1.20 montrent le graphe des deux fonctions. Les propriétés du filtre passe-bas gaussien peut être vu à partir de sa transformée de Fourier qui est-elle même une fonction gaussienne :

$$G(\omega) = e^{-\frac{\omega^2\sigma^2}{2}} \tag{1.3}$$

Cela signifie que le filtre atténuera les changements rapides dans l'image et la lissera efficacement. La qualité du lissage dépend de l'écart type σ choisi. Écart-type σ élevé , image de résultat plus lisse.

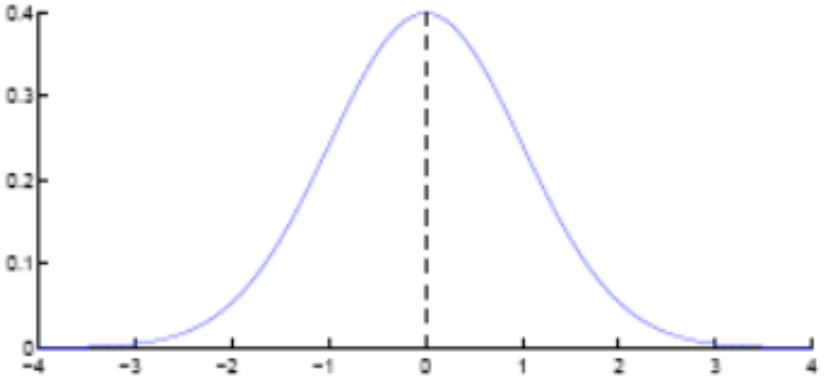


FIGURE 1.19 – Une courbe montrant la distribution gaussienne avec $\sigma = 1$ dans une dimension.

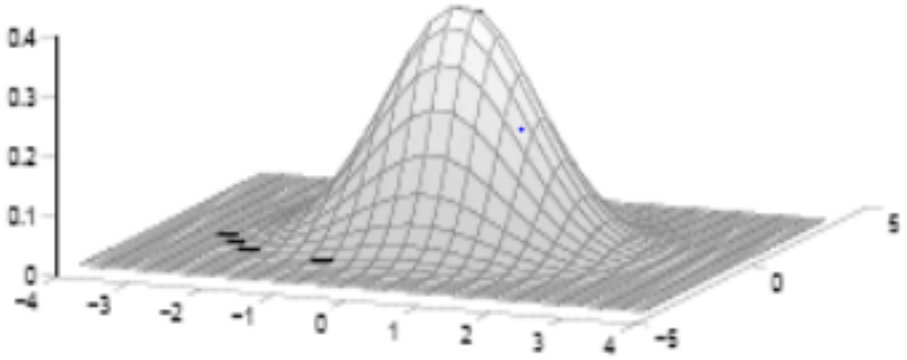


FIGURE 1.20 – Une courbe montrant la distribution gaussienne avec $\sigma = 1$ dans deux dimension.

Le filtre médian : Les cheveux sont une autre source de bruit dans l'image, apparaissant sous forme de lignes noires très fines. Une façon de les supprimer est d'utiliser un filtre médian. Le filtre médian fonctionne en remplaçant la valeur du pixel par la médiane de cette plage. Cela se fait en itérant sur tous les pixels de l'image pour voir les pixels adjacents à une certaine distance. Ces valeurs de pixels sont ensuite combinées et triées.

La valeur au centre de l'ensemble résultant est sélectionnée comme nouvelle valeur de pixel moyenne. À titre d'exemple, considérons la fenêtre de pixels dans le tableau suivant.

124	124	124
126	203	203
128	130	124

TABLE 1.2 – Fenêtre de pixels de taille 3×3.

L'ensemble de valeurs de pixels est :

$$P = 124, 124, 124, 126, 203, 203, 128, 130, 124$$

La nouvelle valeur du pixel du centre est 126, le résultat devient :

X	X	X
X	126	X
X	X	X

TABLE 1.3 – Fenêtre de pixels de taille 3×3 après l'application du filtre médian.

1.3.2.2.2 Amélioration du contraste avec étirement d'histogramme : Un graphique qui peut être utilisé pour montrer la distribution des variables. Pour les images, l'histogramme vous donne une idée du nombre de pixels pour un niveau de gris particulier. Nous pouvons utiliser cet outil pour obtenir un certain nombre d'informations sur une image. Nous pouvons ensuite faire certaines choses à partir de l'histogramme, telles que l'étirement.

L'utilisation de l'infrarouge pour capturer une image de la main rend les veines plus claires, mais nécessite souvent une amélioration supplémentaire du contraste avant la segmentation de l'image. Un moyen simple mais très efficace de le faire est d'utiliser l'étirement de l'histogramme. Cette méthode exploite parti du fait que les valeurs de pixels dans les images en niveaux de gris ne dépassent pas la plage 0-255. Dans l'image d'entrée, les valeurs de pixels ont tendance à être étroitement réparties vers le centre de l'histogramme. Comme le montre la figure 1.21.

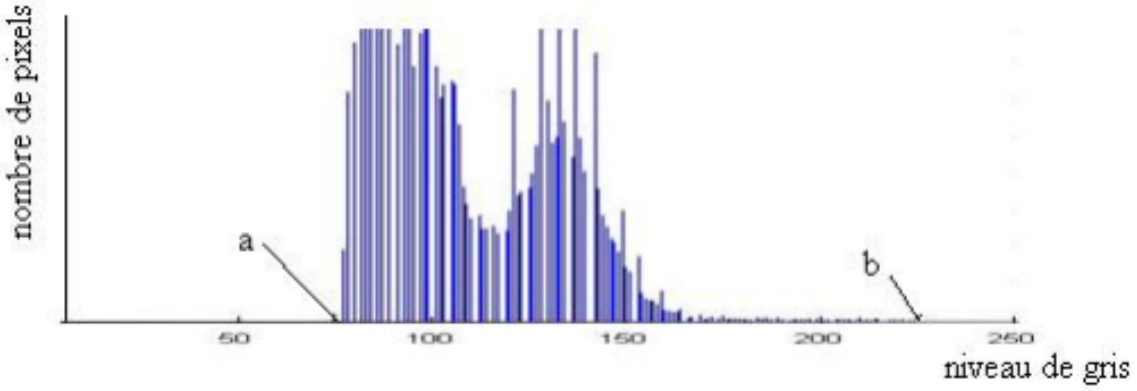


FIGURE 1.21 – Histogramme d'une image.

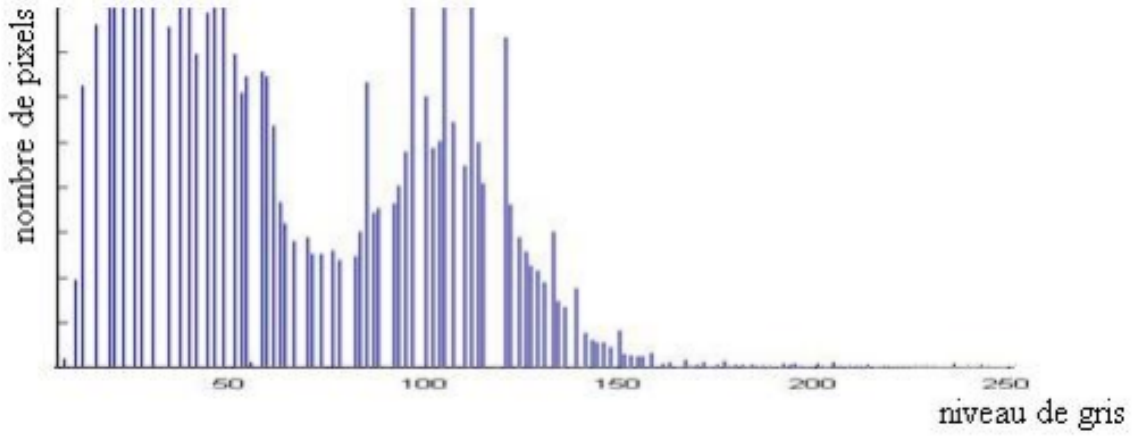


FIGURE 1.22 – Histogramme étiré de l'image .

Sous une forme simple, l'algorithme d'étirement de l'histogramme utilise la limite inférieure a et la limite supérieure b pour convertir la couleur de l'image. Toutes les valeurs de couleur entre a et b sont converties pour couvrir l'ensemble de 0 à 255. Les couleurs inférieures à sont définies sur 0 et les couleurs supérieures à b sont définies sur 255. La première étape consiste à trouver c qui est la moyenne de a et b

$$c = a + \frac{(b-a)}{2} \tag{1.4}$$

Chaque pixel de l'image est transformé comme suit :

$$T(x) = \begin{cases} 128 + \frac{x-c}{b-c}.127 & b \geq x \geq c \\ \frac{x-c}{b-c}.127 & c \geq x \geq a \\ 0 & a > x \\ 255 & x > b \end{cases} \tag{1.5}$$

- x : est un pixel de l'image.
- $T(x)$: est la transformée de pixel x par l'étirement d'histogramme.

De cette façon, l'espace colorimétrique est uniformément étiré autour de la moyenne des deux limites. Après cette étape, nous disposons d'une image filtrée rectangulaire de résolution 8×8 bits par pixel en niveau de gris, l'image est prête à la phase de segmentation.

1.3.2.3 Segmentation

La segmentation est utilisée pour extraire le motif des veines dans une image. Ce processus de segmentation est essentiel aux performances du système, c'est pourquoi un grand soin est apporté à l'évaluation des méthodes disponibles. Nous présentons trois méthodes de segmentation.

1.3.2.3.1 Laplacien de gaussienne : La segmentation utilisant le laplacien de gaussienne est une méthode couramment utilisée pour la détection des contours et la réduction du bruit. Le filtre de Laplace-Gaussien est un filtre combiné dérivé de deux méthodes courantes de traitement d'images : la détection des contours à l'aide du filtre de Laplace et la suppression du bruit à l'aide du filtre gaussien. L'opérateur de Laplace est une mesure 2D de la dérivée spatiale quadratique d'une image utilisée pour segmenter les bords le long d'un objet. Ce processus est très sensible. C'est-à-dire que la plupart des bords peuvent être segmentés, mais ils sont plus bruyants. Pour pallier cette faiblesse, la détection des contours de Laplace a été associée à un filtrage du bruit gaussien. Cela supprime les bords bruyants et préserve les bords de l'objet réel.

Description

Fondamentalement, la méthode gaussienne de Laplace est une application de la méthode gaussienne à un filtre gaussien, et cette méthode est mathématiquement décrite comme suit.

$$LOG = \Delta G_\sigma(x, y) \tag{1.6}$$

Où ;

- LOG : est le laplacien de l'opérateur de Gauss.
- Δ : est l'opérateur de Laplace.
- $G_\sigma(x, y)$: est le filtre de gauss en 2D.
- σ : est l'écart type de l'opérateur gaussien

L'opérateur de Gauss est exprimé comme suit :

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.7}$$

Où : x et y sont les coordonnées de pixel
L'opérateur laplacien est exprimé comme :

$$\Delta = \frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} \tag{1.8}$$

En utilisant cette information, l'opérateur LOG peut être dérivé :

$$\begin{aligned} LOG &= \Delta G_\sigma(x,y) \\ &= \left(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2}\right)\left(\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) \end{aligned}$$

$$= \frac{1}{2\pi\sigma^2}\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] \tag{1.9}$$

Cela utilisera un filtre discret dans le traitement de l'image.

1.3.2.3.2 Seuillage Global : La caractéristique la plus simple que les pixels d'une zone peuvent avoir en commun est l'intensité. Par conséquent, ces zones peuvent être segmentées par seuil. Cela se fait en séparant les zones blanches et noires.

Le seuil crée une image binaire avec de niveaux de gris. Tous les pixels en dessous d'une certaine valeur fixe, appelée seuil, à zéro et tous les pixels au-dessus du seuil sont mis à 1. Si $g(x,y)$ est l'image de seuil de l'entrée $f(x,y)$ du seuil T, le processus de traitement de seuil peut être décrit comme suit :

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) \geq T \\ 0 & \text{ailleurs} \end{cases} \tag{1.10}$$

A partir de l'équation (1.10), nous pouvons voir que l'image entière est binarisée par un seul seuil T. De ce fait, le traitement global du seuil ne donne pas des résultats satisfaisants pour les images avec de beaucoup variations d'intensité [20].

1.3.2.3.3 Seuillage local adaptatif : Le seuillage local adaptatif sélectionne un seuil individuel $t(x,y)$ pour chaque pixel en fonction des voisins locaux. L'algorithme choisit un seuil différent pour chaque pixel, sur la base d'une analyse des pixels adjacents environnants. Un pixel est défini comme étant proche du pixel s'il se trouve dans la fenêtre carrée $w \times w$ centrée sur le pixel en question. Le seuil est calculé en calculant la moyenne des valeurs des pixels adjacents [21].

En utilisant $t(x,y)$ comme seuil pour chaque pixel de l'image d'entrée $f(x,y)$, la version de seuil $g(x,y)$ en comparant chaque pixel à $t(x,y)$ est obtenue.

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) \geq t(x,y) \\ 0 & \text{ailleurs} \end{cases} \tag{1.11}$$

Le seuil $t(x,y)$ est la somme des valeurs des pixels entourant le pixel central divisée par le nombre de ces pixels adjacents. Pour chaque image en niveaux de gris, les seuils pour toutes les fenêtres de taille w × w peuvent être calculés à l'aide de l'équation (1.12)

$$t(x,y) = \frac{1}{w^2} \sum_{i=x-\frac{w}{2}}^{x+\frac{w}{2}} \sum_{j=y-\frac{w}{2}}^{y+\frac{w}{2}} f(i,j) \tag{1.12}$$

Ou ;

- $f(i, j)$: la valeur du pixel dans i, j
- w : la taille du fenêtre

1.3.2.4 Extraction des caractéristiques

Cette étape est destinée à extraire les paramètres de la structure veineuse de l'image et est stockée dans une base de données pour comparaison.

Les méthodes d'extraction des paramètres :

Cette section décrit deux manières d'extraire des caractéristiques d'un dessin de veine. La première est l'extraction des minuties (terminaisons et bifurcations) du dessin des veines de la main. La seconde consiste à extraire la matrice des moments de Zernike. Les deux méthodes produisent les caractéristiques qui sont analysés à la phase de reconnaissance.

a) L'extraction des minuties :

Deux phases de préparation de l'extraction (segmentation et squelettisation) rendent cette phase beaucoup plus facile. En fait, nous disposons maintenant une image binaire squelette. Les pixels noirs prennent la valeur 1, les pixels blancs prennent la valeur 0 et la largeur de la bande est égale à 1 pixel. Pour chaque point du squelette, nous calculons le nombre de transitions divisé par 2 entre les pixels blancs et noirs pour obtenir le nombre de CN (Crossing Number) dans les stries. À partir de ce point, nous pouvons simplement déterminer le type des Pixels (voir Figure 1.23). Nous mettons une fenêtre de 3×3 pixels, comme indiqué dans le tableau suivant.

P_1	P_2	P_3
P_8	P	P_4
P_7	P_6	P_5

TABLE 1.4 – Fenêtre de taille 3×3 pixels.

Le nombre CN pour le pixel P s'écrit comme suit :

$$CN(P) = \frac{1}{2} \sum_{i=1}^{8} \left| P_i - P_{i-1} \right| \text{ avec } P_8 = P_0 \text{ et } P_i = (0 \text{ ou } 1) \tag{1.13}$$

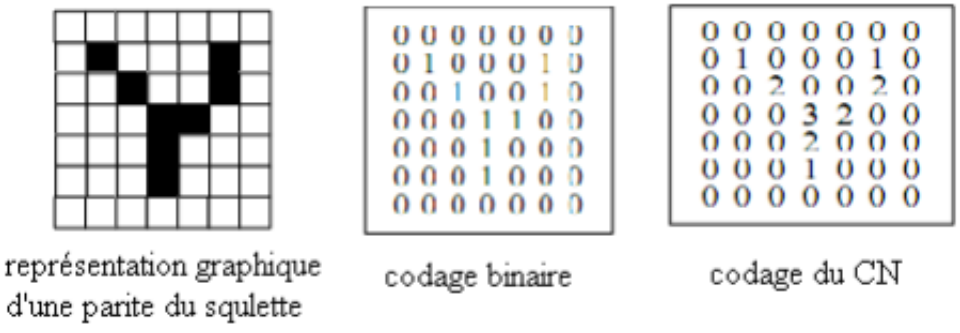


FIGURE 1.23 – Les différentes représentations du squelette d'une image quelconque.

Pour chaque pixel P qui appartient à la bande (c'est-à-dire chaque pixel avec une valeur de 1), on peut déterminer le type de pixel selon le calcul CN .

- $CN(P) = 0$: dans ce cas il s'agit d'un pixel isolé et nous n'en tenons pas compte car même si ce type de minutie existe il est très rare et à ce stade du traitement de l'image il est probablement dû à un résidu de bruit.
- $CN(P) = 1$: dans ce cas nous avons à faire à une minutie de type terminaison.
- $CN(P) = 2$: c'est le cas le plus courant, les pixels sont sur la bande et il n'y a pas de détails.
- $CN(P) \geq 3$: nous sommes en présence d'une bifurcation.

b) Les moments de Zernike :

Le Zernike Moment a été introduit par F. Zernike en 1934. [22] Dans le domaine du traitement de l'information. Les moments de Zernike sont largement utilisés pour leur orthogonalité, qui permet la génération de descripteurs non redondants, et pour leurs propriétés d'invariance de translation, d'échelle et de rotation. Les moments de Zernike se retrouvent dans de nombreux travaux sur la reconnaissance d'images, l'indexation d'images dans des bases de données, l'analyse et la description de la forme d'objets 2D ou 3D, etc

$$Z_{nm} = \frac{n+1}{n} \int_x \int_y f(x,y).[V_{nm}(x,y)]^* dxdy \tag{1.14}$$

Ou : $x^2 + y^2 \leq 1$

- $[\ldots]^*$: est utilisé pour indiquer la valeur complexe conjuguée.
- n : représente ici l'ordre de décomposition ($n = 0, 1, 2, \ldots, \infty$), dit aussi ordre radial, et m le nombre de répétitions de la décomposition pour un ordre n donné. L'ordre et la répétition sont liés par les deux conditions suivantes :
  $n - |m|$ toujours pair et $|m| \leq n$
- V_{nm} représente les polynômes de Zernike constituant la base orthogonale de projection. Ils s'écrivent en général en représentation polaire sous la forme suivante :

$$V_{nm}(r, \Theta) = R_{nm}(r).e^{im\Theta} \tag{1.15}$$

Ou $R_{nm}(r)$ sont des polynomes radiaux de la forme :

$$R_{nm}(r) = \sum_{k=|m|}^{n} \frac{(-1)^{\frac{(n-k)}{2}}.(n+k)!}{\frac{(n-k)}{2}!\frac{(k+m)}{2}!\frac{(k-m)}{2}!}.r^k \tag{1.16}$$

L'application des moments de Zernike sur une fonction discrète $h(x, y)$ (image par exemple) nécessite la réécriture de l'équation (1.14) comme suit :

$$Z_{nm} = \frac{n+1}{n} \sum_x \sum_y h(x,y).[V_{nm}(x,y)]^* \tag{1.17}$$

Approximations

Pour maintenir l'orthogonalité de base au niveau du moment calculé, la fonction $h(x, y)$ doit être convertie en une représentation polaire (r, Θ) et recalculée dans le cercle unité.

Tel que le centre d'image soit le centre du cercle unité. Ensuite, la relation de conversion est décrite.

$$x = r.cos\theta \qquad \text{et} \qquad y = r.cos\Theta$$

Avec : $r = \sqrt{x^3 + y^2}$ et $\Theta = \tan^{-1}(\frac{y}{x})$

La forme de l'image $h(x, y)$ est rectangulaire ou carrée et est incompatible avec la forme du cercle unité. Cela vous donne le choix de supprimer certains points (notamment des coins) de l'image ou d'introduire des points différents de la fonction d'origine.

Les nouvelles coordonnées sont écriées comme suit :

$$\begin{cases} x_j = c + \frac{j(d-c)}{N-1} \\ y_i = d - \frac{i(d-c)}{M-1} \end{cases} \quad \text{et} \quad \begin{cases} r_{ij} = \sqrt{x_j^2 + y_i^2} \\ \Theta_{ij} = \tan^{-1}(\frac{y_i}{x_j}) \end{cases} \tag{1.18}$$

Où ;

- i et j sont les coordonnées du point de l'image originale.
- x_j et y_i sont les nouvelles coordonnées de ce point dans le nouveau système de coordonnées (cercle unitaire).
- M et N sont respectivement les dimensions horizontale et verticale de cette image.
- c et d sont des paramètres qui permettent de recalculer tout ou partie ($c = 1$ et $d = 1$) de la fonction $h(x, y)$ ($c = \frac{-1}{\sqrt{2}}$ et $d = \frac{1}{\sqrt{2}}$) dans le cercle unité, comme le montre la figure 1.24.

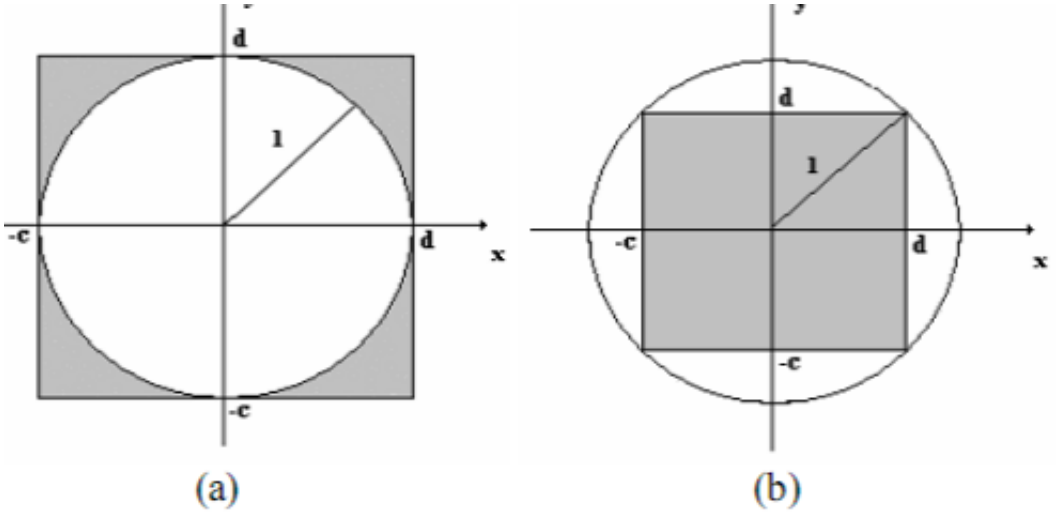


FIGURE 1.24 – Façon de recalculer de la fonction h(x, y) (rectangle gris) dans le cercle unité (a) cas de $c = -1$ et $d = 1$ (b) cas de $c = \frac{-1}{\sqrt{2}}$ et $d = \frac{1}{\sqrt{2}}$.

Après normalisation des images, chaque image est complètement recalculée dans le cercle unitaire, comme le montre la figure 1.24 (b). En effet, les approximations qui consistent à calculer partiellement l'image à l'intérieur du cercle unité réduiront considérablement les performances, surtout après avoir exclu les coins de l'image.

1.3.2.5 Comparaison

Une fois les paramètres extraits, ils sont comparés avec ceux de la base de données pour prendre une décision, l'image est identifiée si les caractéristiques d'entrée sont semblables à celles enregistrées dans la base de données, sinon elle est rejetée.

1.4 Conclusion

Dans ce chapitre nous avons expliqué quelques généralités sur la biométrie, nous expliquons la notion de la biométrie, les caractéristiques biométriques, les modalités biométriques, l'architecture des systèmes biométriques et modes de fonctionnements, mesure de performance d'un système biométrique, et les applications de systèmes biométriques et enfin nous avons expliqué en détail les étapes de la reconnaissance par la technique de réseau veineux de la main.

Dans le chapitre qui suit, nous allons entamer quelques outils de sécurité des images en général, afin de détailler ensuite la méthode de hachage perceptuel et qui est la méthode qui sera appliquer dans notre travail.

CHAPITRE 2

SÉCURISATION DES IMAGES VIA HACHAGE PERCEPTUEL

2.1 Introduction

Nous vivons dans le monde ou presque tout peut être digitalisé et envoyé d'une partie du monde, l'autre en seulement quelques secondes. L'opération de traitement de contenu est devenue facile, et de nombreux utilisateurs de ces outils modifier le contenu multimédia presque chaque jour.

Dans le souci d'aboutir à une vérification efficiente de l'intégrité du contenu et la prévention efficace des falsifications, plusieurs techniques d'authentification ont vu aujourd'hui .

Dans ce chapitre nous abordons différentes techniques de sécurisation d'image ou nous intéressons plus particulièrement sur le hachage perceptuel, ses fonctionnalités ses étapes, ainsi que ses propriétés.

2.2 Les outils de sécurité des images numérique

2.2.1 La sténographie et la cryptographie

La sténographie vise à dissimuler des messages secrets dans des médias sans méfiance, tels que des images, des sons et des vidéos, appelés médias de couverture. Cette dissimulation doit se faire en changeant de support de recouvrement, avec la contrainte que les changements apportés au support soient minimisés et masqués. Plus le message secret est petit et plus le média de couverture est grand, plus ce changement sera probablement ignoré. Par conséquent, utilisez des fichiers image, audio ou vidéo au lieu de petits fichiers texte.

La sténographie repose sur l'idée de sécurité par l'obscurité [23] : Si personne ne sait qu'il y a un message caché, personne ne cherchera à le regarder ou à, le récupérer.

Le cryptage crypte les messages de Fagon de manière incompréhensible pour garantir la confidentialité. Ce processus garantit que seul l'utilisateur cible du message peut accéder

au message. Il existe deux types de chiffrement : le chiffrement symétrique et le chiffrement asymétrique.

2.2.2 Le tatouage numérique

Le tatouage numérique ou watermarking est une technique permettant d'ajouter des informations de droit d'auteur ou d'autres messages de vérification à de l'audio, de la vidéo, des images ou d'autres documents ou fichiers numériques en insérant des marques invisibles sur des supports numériques. [24]

Le message contenu dans un signal d'urgence est communément appelé un marqueur ou simplement un message et est un ensemble de bits dont le contenu varie d'une application à l'autre [25]. La marque peut prendre la forme d'un nom ou d'un identifiant du créateur, du propriétaire, de l'acheteur ou d'une signature décrivant un signal hâte.

Le tatouage numérique permet d'insérer des informations (une signature) dans un document informatique. L'ajout de cette signature doit être imperceptible et indécelable par tout système ignorant son mode d'insertion. En particulier, il faut qu'il soit totalement invisible pour L'œil humain.

2.2.3 Hachage des images

2.2.3.1 Hachage cryptographique

La fonction de hachage cryptographique H est un algorithme qui permet de générer un hachage de taille fixe $H(M)$ de n bits i à partir de données M de taille quelconque.

$$\begin{aligned} H \ : \ (0,1)^* &\rightarrow (0,1)^n \\ M &\rightarrow H(M) \end{aligned}$$

Ou la notation $(0,1)^*$ Désigne l'ensemble des chaines de bits de longueur arbitraire finie et la notation $(0,1)^n$ désigne l'ensemble de chaines de bits de longueur exactement n. En pratique, n est de l'ordre de plusieurs centaines de bits.

La valeur de hachage du hachage chiffré est également aléatoire. Le message utilisé pour générer le hachage agit comme un graine aléatoire, de sorte que les mêmes données produisent exactement le même résultat, mais différents messages produisent des hachages complètement différents. Les fonctions de hachage cryptographique comprennent deux choses principales : Si les hachages sont différents, les données sont différentes ou vice versa [26].

Par conséquent, l'intégrité du message ne peut être vérifiée que si tous les bits du message n'ont pas changé. Ce comportement est fondamentalement caractéristique de ce type de hachage.

2.2.3.2 Hachage perceptuel

Cependant, il existe plusieurs raisons pour lesquelles la cryptographie ne peut pas être utilisée directement pour résoudre les problèmes de sécurité multimédia [27]. Contrairement aux données textuelles transmises sur un support sans perte, les données multimédia telles que les images (ou audio, vidéo, etc.) peuvent être transmises et stockées à l'aide

d'un support avec perte, ce qui permet d'économiser de la bande passante et de l'espace de stockage.

Par conséquent, lors de la vérification de l'intégrité ou de l'authentification du contenu multimédia à l'aide des fonctions de hachage cryptographiques traditionnelles, le problème est qu'un seul changement de bit dans le contenu peut modifier considérablement la valeur de hachage. D'autres algorithmes de hachage conviennent à l'analyse du contenu multimédia.

De plus, les gens peuvent facilement faire la distinction entre plusieurs images et déterminer s'il s'agit de même. Cependant, comme les ordinateurs voient tout d'un point de vue complètement différent, les tâches simples pour les humains sont très compliquées pour les ordinateurs..

Plusieurs images peuvent avoir des représentations numériques différentes, mais dans la perception humaine, elles sont toutes identiques. Cela soulève le problème que le contenu multimédia peut être illégalement divulgué aux algorithmes de recherche si seul le hachage cryptographique est pris en compte. Les données telles que les images numériques peuvent subir diverses opérations telles que la compression et la rotation. Avec les hachages cryptographiques traditionnels, il n'est pas possible de reconnaître une image modifiée et d'en déduire sa source.

2.3 Hachage perceptuel des images

2.3.1 Définition de hachage perceptuel

La fonction de hachage perceptuel d'image extrait des caractéristiques d'une image et calcule une valeur de hachage basée sur ces caractéristiques. Ces caractéristiques ont été proposées pour établir une égalité de perception dans le contenu de l'image. L'authentification de l'image est effectuée pour comparer les valeurs de hachage de l'image d'origine et de l'image de l'authentificateur. Ces dernières années, des recherches sur le hachage perceptuel d'images ont été menées et reçoivent une attention croissante dans la littérature

2.3.2 Les fonctions de hachage perceptuel

Les fonctions de hachage perceptives sont fortement influencées par les fonctions de hachage cryptographiques. Ce dernier étant très sensible au contenu binaire des données hachées, des fonctions de hachage perceptuel ont été proposées comme solution alternative à l'application principale pour les données multimédias, notamment les images. La fonction de hachage perceptuel est basée sur l'aspect visuel des données hachées. [28]
La figure (2.1) montre un exemple des exigences attendues d'une fonction de hachage perceptive. La figure (2.1.a) montre la version compressée JPEG (taux de compression visuellement acceptable) de l'image originale illustrée à la figure (2.1.b) [29]. L'image originale et sa version compressée JPEG doivent avoir la même signature.

Par contre, quand l'image originale subie des opérations malicieuses en changeant son contenu comme illustré figure (2.1.c), la signature associée de l'image malicieusement modifiée doit être radicalement différente d celle de l'image originale.

(a) Image compressée JPEG (b) Image originale (c) Image Modifiée

FIGURE 2.1 – Exemple qui illustre les exigences d'un hachage perceptuel dans le scénario d'authentification de contenu. Les signatures perceptuelles des images (b) et (a) doivent être égales et différentes de celle de l'image (c).

2.3.3 Manipulations acceptables vs manipulations malveillantes

Une image numérique peut subir différentes formes de transformations ou de manipulations qui peuvent affecter son contenu binaire et/ou visuel. Certaines applications ont avoir besoin d'appliquer certaines manipulations acceptables afin d'améliorer la qualité de l'image originale tels que le filtrage, la compression, ou même d'effectuer d'autres opérations permettant l'amélioration de l'image en question. Certaines applications peuvent également nécessiter une compression avec pertes pour satisfaire les contraintes de ressources sur la bande passante ou d'espace de stockage. [28]

Ces manipulations acceptables modifient uniquement les valeurs de pixels, qui se traduisent par différents niveaux de distorsion visuelle de l'image, mais le contenu de l'image, qui porte la même information visuelle vers le récepteur, est encore conservé, D'autre part, les manipulations malveillantes changent le contenu de l'image originale afin de porter une information visuelle différente pour le récepteur. [28]

Un exemple typique de modification malveillante est de remplacer certaines parties de l'image avec des contenus différents pour une utilisation malveillante. Une classification, non exhaustive, des manipulations acceptables préservant le contenu et les manipulations malveillantes changeant le contenu, cela est présentée dans le tableau (2.1). [28]

Les fonctions de hachage perceptuel doivent être capables de survivre i, des manipulations acceptables qui préservent le contenu et de rejeter les manipulations malveillantes [29]

Manipulations acceptables	Manipulations Malveillantes
• Ajout de bruit. • Erreur de transmission. • Mise à l'échelle. • Compression et Quantification. • Conversion de colleurs. • Filtrage. • Réduction de résolution. • Rotation. • Réglage de contraste. • Changements de luminosité, teinte et de saturation • Croping • Shearing etc.	• Suppression des objets sur l'image • Déplacement des éléments de l'image pour changer leurs positions • Ajout de nouveaux objets • Changements des caractéristiques de l'image : couleur, textures, structure, impression, etc. • Modifications du contexte de l'image : de jour ou d'remplacement • Les changements de conditions d'éclairage : manipulations d'ombre • Dégradation de la qualité etc. ....

TABLE 2.1 – Manipulations acceptables et manipulations malveillantes.

2.3.4 Hachage perceptuel vs Hachage cryptographique

Les fonctions de hachage cryptographique et les fonctions de hachage perceptuel ont les mêmes objectifs [30]. Les deux types de fonctions de hachage vérifient l'authenticité et contrôlent l'intégrité des données à hacher. Quand une authentification d'un fichier exécutable est exigée, il est très important que toutes les valeurs des bits correspondent exactement aux valeurs originales. Dans ce cas, les fonctions de hachage cryptographique sont les plus adéquates à utiliser.

Pour authentifier une donnée multimédia, il est nécessaire de vérifier son contenu visuel sans tenir compte de sa représentation numérique [26]. Dans ce cas, les fonctions de hachage cryptographique ne présentent pas une bonne solution. Pour cela, les fonctions de hachage perceptuel sont proposées pour satisfaire les besoins particuliers de sécurité des images numériques. Par analogie aux fonctions de hachage cryptographique, les fonctions de hachage perceptuel doivent générer une signature :

- Courte : la signature doit être courte de l'ordre de quelques centaines de bits.
- Robuste : avoir la même signature pour des données multimédia de même contenus visuels.
- Sécurisée : impossible de générer les données originales à partir de leurs signatures et en même temps avoir des signatures totalement différentes pour des données multimédia n'ayant pas le même contenu visuel.

2.3.5 Schéma général d'un système de hachage perceptuel

Un système de hachage perceptuel, comme illustré figure (2.2), se compose généralement de quatre étapes : l'étape de transformation, l'étape d'extraction des caractéristiques, l'étape de crypto-compression.

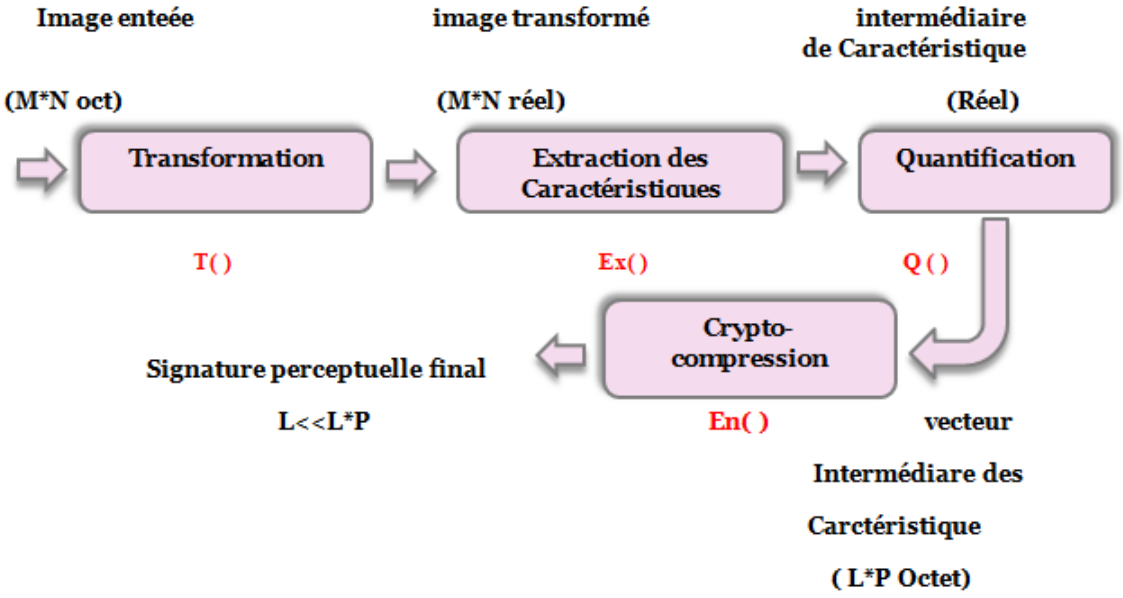


FIGURE 2.2 – Présentation des quatre étapes d'un système de hachage perceptuel.

Dans l'étape de transformation, l'image d'entrée subit une transformation spatiale et/ou fréquentielle. Ces transformations transfèrent toutes les caractéristiques extraites, dans l'étape suivante, et les rendent dépendantes des valeurs de pixel ou des coefficients fréquentiels de l'image d'entrée [27]. Dans l'étape d'extraction des caractéristiques, le système de hachage perceptuel extrait les caractéristiques de l'image h partir de l'image transformée pour générer un vecteur intermédiaire des caractéristiques contenant des valeurs réelles. Ensuite, le vecteur intermédiaire des caractéristiques est quantifié pour former le vecteur intermédiaire des caractéristiques discrètes durant l'étape de quantification. Enfin, le vecteur intermédiaire des caractéristiques discrètes est compressé et crypté dans une courte signature perceptuelle durant l'étape de crypto-compression

2.3.5.1 Étape de transformation

Durant la transformation, l'image d'entrée de taille $M \times N$ octets subit des transformations spatiales telles que la transformation de couleur, le lissage, ou des transformations fréquentielles comme la transformée en cosinus discrète (DCT) ou la transformée de en ondelettes (DWT) [27]. Quand une transformée en ondelettes discrètes est appliquée, la plupart des systèmes de hachage perceptuel ne prennent que la sous-bande LL en compte. En effet, la sous-bande LL est une version grossière de l'image originale et contient toutes les informations perceptuel de l'image. L'objectif principal de ces transformations est de rendre toutes les caractéristiques extraites dépendantes des valeurs de pixel de l'image (dans le cas de la transformation spatiale) ou des coefficients fréquentiels (en cas de transformation fréquentielle).

2.3.5.2 Étape d'extraction des caractéristiques

Dans l'étape d'extraction des caractéristiques, le système de hachage perceptuel extrait les caractéristiques de l'image à partir de l'image transformée pour générer le vecteur intermédiaire des caractéristiques réelles de L éléments, ou, $L < M \times N$. A noter que chaque caractéristique peut contenir P éléments de type réel, ce qui signifie que le vecteur des caractéristiques est composé de $L \times P$ réels A, cette étape [27]. Une autre sélection des caractéristiques peut être ajoutée à cette étape, comme l'illustre la figure (2.3), ou les caractéristiques les plus pertinentes sont sélectionnées. Elles sont statistiquement plus résistantes contre certaines manipulations spécifiques tolérées, comme l'ajout de bruit, la compression JPEG et le filtrage. Les caractéristiques sélectionnées peuvent être présentées comme un vecteur intermédiaire des caractéristiques de $K \times P$ réels, ou $K < L$. Notez que les caractéristiques visuelles sélectionnées sont généralement connues du public et peuvent donc être modifiées. Cela pourrait menacer la sécurité, du fait que la valeur de la signature perceptuel pourrait être ajustée malicieusement pour correspondre d'une autre image.

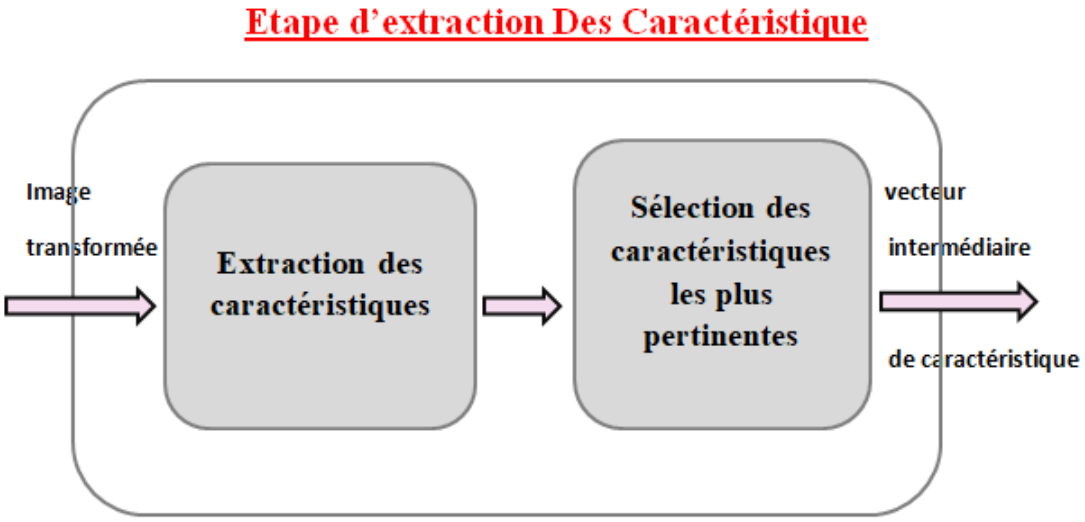


FIGURE 2.3 – Sélection des caractéristiques les plus pertinentes.

2.3.5.3 Étape de quantification

Dans l'étape de quantification, le vecteur intermédiaire des caractéristiques réelles est quantifié et peut être codé sur $K \times P$ octets. Le nouveau vecteur intermédiaire de caractéristiques contient des valeurs discrètes. La quantification uniforme peut être appliquée pour quantifier chaque composante du vecteur des caractéristiques réelles. La quantification adaptative est également un schéma de quantification largement utilisé dans les techniques de hachage perceptuel [27]

2.3.5.4 Étape de compression et cryptage

L'étape de crypto-compression est la dernière étape d'un système de hachage perceptuel qui garantit à la fois la sécurité du système et la longueur fixe de la signature

perceptuel finale. Le vecteur intermédiaire des caractéristiques discrètes est compressé et crypté dans une courte signature perceptuel de taille fixe de 1 octet, ou $I < K \times P$ [27], qui présente la signature perceptuelle permettant la vérification et l'authentification d'image au niveau du récepteur. Cette étape peut être assurée par les fonctions de hachage cryptographique comme, par exemple, la fonction de hachage cryptographique SHA-1, qui généré une signature de taille 160 bits.

2.3.6 Propriétés de hachage perceptuel d'image

Les fonctions de hachage perceptuel doivent conserver quatre propriétés principales afin de garantir leur compatibilité, efficace et sécurisé à la fois :

- La robustesse
- Discriminabilité
- Imprévisibilité
- La compacité

Pour faciliter l'idée, supposons que (O) est l'image originale et $(O^{''})$ est une image modifiée, mais conserve les propriétés perceptuel de l'original (O). En d'autres mots, les (O) et $(O^{''})$ sont les mêmes images du point de vue humain, mais différentes en maniéré cryptographique [31]. Le représente d'une image complètement différente de celle d'origine est (0). Soit $\Theta, \Theta^{'}$ (deux valeurs positives satisfaisant $0 < \Theta, \Theta^{'} < 1$, La fonction de hachage perceptuel H produit une empreinte de hachage dont la longueur dépend d'une clé secrété.

2.3.6.1 La robustesse

La robustesse signifie que si la même clé est utilisée, des images perceptuelle ment similaires génèrent un hachage similaire [30]. Cette propriété peut être représentée par l'équation :

$$P(H(O, K) = H(O^{\circ}, K)) \approx 1 \tag{2.1}$$

Dans l'équation (2.1), P désigne la probabilité. H est la fonction de hachage perceptuel qui produit le hachage final basé sur l'image d'entrée O et K la clé secrète. Le O est la version de I'image modifié avec les propriétés de perception préservées de l'image d'origine (O). Les valeurs de hachage perceptuel des images similaires doivent être identiques ou avec une très petite distances de similarité, même si l'image $(O^{''})$ été modifiée. Dans ce cas, la valeur de probabilité devrait être égale ou très proche de 1.

La robustesse garantit que deux images perceptuelle ment identiques doivent avoir des hachages similaires. L'objectif principal des images similaires est d'être suffisamment robuste pour la différente manipulation acceptable telle que la transformation avec perte JPEG, la rotation, le bruit, le flou, etc. Perceptuelle ment, ces images dans le système visuel humain (SVH) sont identiques, même si certains des bits ont été changés.

2.3.6.2 Discriminabilité

La discriminabilité signifie que la même clé est utilisée. Des images perceptuelle ment différentes génèrent les différents hachages. Cette propriété peut être représentée par l'équation :

$$P(H(O,K) \neq H(M,K)) \approx 1 \tag{2.2}$$

Dans l'équation (2.2), P désigne la probabilité. Et H la fonction de hachage perceptuel qui produit le hachage final basé sur l'image d'entrée O et la clé secrète K. Le (M) est complétement une image différente. Les valeurs de hachage des différentes images ne devraient pas être les mêmes ou avec une distance similaire. Dans ce cas, la valeur de la probabilité doit être égale ou très proche de 0.

La discriminabilité garantit que le hachage perceptuel de deux images est distinct [32]. En d'autres termes, les hachages de deux images complétement différentes ne doivent pas être égaux, Il devrait y avoir une très faible probabilité proche de 0.

2.3.6.3 Imprévisibilité

L'imprévisibilité garantit que l'attaquant n'aura pas le même hash pour l'objet multimédia original en manipulant certains des bits de données d'objet multimédia modifiés

$$H(O,K); fn(1) \approx fn(0) \approx 0.5 \tag{2.3}$$

Dans l'équation (2.3), H est la fonction de hachage perceptuel qui produit le hachage final basé sur l'image d'entrée O et la clé secrète K. Ou fn est la fonction de masse de probabilité pour hash h. Avec cette propriété, les valeurs de hachage doivent être équitablement réparties La sécurité est une préoccupation importante pour le hachage. Cela rendra la procédure de hachage suffisamment en sécurité [31], pour diminuer la probabilité pour l'attaquant de deviner la clé secrète et estimer la valeur de hachage correcte.

2.3.6.4 La compacité

La compacité est une autre propriété importante presque dans tous les schémas de hachage. Cette propriété peut être représentée par l'équation

$$Size(H(O,K)) \ll Size(O) \tag{2.4}$$

Dans l'équation (2.4), $Size$ représente la quantité de bits totale. H est la fonction de hachage perceptuelle qui produit le hachage final basé sur l'image d'entrée O et Ia clé secrète K. Si nous comparons la taille de hachage avec l'image originale devrait être sensiblement plus petite. La propriété de compacité résoud le problème des grandes bases de donnée, et simplifie le processus de recherche [32]. En outre, il n'est pas nécessaire de restaurer l'image originale de hachage, mais en tenant compte uniquement des « caractéristiques perceptuel».

2.3.7 Distance / Fonctions de similarité pour les hachages perceptuels

Une fonction de hachage perceptuel calcule des valeurs de hachage perceptuel similaires pour des objets multimédias identiques. Pour comparer deux valeurs de hachage perceptuel, des mesures appropriées doivent être utilisées. Les plus souvent utilisées sont : Le taux d'erreur binaire (TEB), la distance de Hamming, la distance euclidienne et le coefficient de corrélation linéaire.

• **Le taux d'erreur binaire (TEB) :**

Le TEB définit p comme le nombre i d'erreurs des bits du hachage perceptuel normalisé par la longueur k du hachage perceptuel [32] :

$$P := \frac{i}{k}$$

tandis que i ∈ { 0,1,....,K } et $0 \leq p \geq 1$

Le nombre d'erreurs des bits i est égal i la distance de Hamming des valeurs de hachage perceptuel. Lors de la comparaison d'images perceptuellement différentes, le TEB devrait être environ 0,5. C'est le TEB auquel nous pouvons s'attendre lors de la comparaison de deux valeurs de hachage perceptuel tirées d'une distribution aléatoire uniforme de $(0,1)^n$ Perceptuellement des images égales devraient donner un TEB proche à 0 .

• **Distance de Hamming :** Est une mesure de la différence de deux chaines, ces chaines peuvent être par exemple les nombres codés en binaires, mais ils pourraient aussi bien être constitués d'éléments d'autres systèmes de numération ou d'alphabets (voir le tableau suivant pour quelques exemples).

String 1	String 2	String 3
00101	10101	1
12345	13344	2
well	well	0

TABLE 2.2 – Exemples de calcul de la distance de Hamming. Les chaines sont issues de trois alphabets différents (système binaire, système à décennie et alphabet latin).

Soit A un alphabet de longueur finie. $X = (x_1, .., x_n)$ dénote une chaine de longueur égale, alors que, $x \in A$. La même chose vaut pour $y : (y_1, .., y_n)$.Ensuite, la distance de Hamming Δ entre x et y est définie comme :

$$\Delta(X, Y) := \sum_{xi \neq yi} 1i = 1...n$$

• **Distance Euclidienne :**

La distance euclidienne est adaptée pour les vecteurs nonbinaires comme les nombres entiers. Elle est parfaite pour mesurer la similitude et la capacité de discrimination entre deux haschs lorsqu'il n'est pas représenté dans la forme binaire. Le tableau 2.3 représente les différentes exemples.

La distance euclidienne est représentée par l'équation suivante :

$$DR(H_1, H_2) := \sqrt{sum_{i=1}^{n}(h_1(i) - h_2(i))^2}$$

Ou DE représente la distance euclidienne. H_1 et H_2 sont les haschs décimaux avec la même longueur L, h_1 est le premier bit de hachage décimal, h_2 est le second. Pour la distance euclidienne, la longueur des deux haschs examinés doit être égale. Les deux valeurs de hachage sont plus proches

A	**B**	**Distance euclidienne**
0100100	1100110	1.414
4391256	5341255	5.196

TABLE 2.3 – Exemples de calcul de la distance euclidienne.

• **Le coefficient de corrélation :** Est utilisé pour distinguer le degré linéaire entre les séquences de hachage obtenues par f image.

L'intervalle de corrélation est [-1,1], ou la plus grande valeur représente la plus grande similitude.

2.4 Conclusion

Dans ce chapitre nous avons présenté d'une manière générale les techniques de sécurisation des données, tel que la sténographie et le tatouage numérique, ainsi que le hachage cryptographie qui représentent une forte source d'inspiration des techniques de hachage perceptuel des images.

Nous avons abordé aussi les fonctions de hachage perceptuel, en commençant par la présentation des objectifs attendus de telles fonctions. Ensuite, nous avons détaillé toutes les étapes dans un système de hachage perceptuel. Puis une présentation de ses propriétés fondamentales qu'un système de hachage perceptuel doive les vérifier.

Dans le chapitre suivant nous allons montrer les différentes méthodes de ce hachage perceptuel, en détaillant la méthode à exploiter dans ce mémoire.

CHAPITRE 3

HACHAGE D'IMAGES ROBUSTE À BASE CODAGE FRACTALE

3.1 Introduction

Les images numériques subissent souvent un traitement normal telles que la compression JPEG, transformation géométrique et la conversion de format Après ces opérations, les représentations numériques des images sont modifiées, mais leurs apparences visuelles sont encore préservées. Ainsi, leurs hach d'image devaient être identiques ou très similaires. En général, une fonction de hachage d'image doit avoir deux propriétés de base la robustesse perceptuel et la capacité discriminante. [33]

Dans ce chapitre nous mentionnons quelques méthodes de hachage qui se décompose en méthode de décomposition en bloc, et les méthodes globales, ainsi nous détaillons le hachage d'image robuste à base codage fractale cette méthode est décomposé en trois étapes. Son idée principale est de diviser l'image de réseau de veineux de main en plusieurs anneau pour former un image secondaire puis appliquer le codage fractale sur l'image secondaire pour extraite le hache de l'image et crypter en fin pour améliorer la sécurité.

3.2 Méthodes de hachage perceptuel

3.2.1 Méthodes de hachage perceptuel par bloc

3.2.1.1 Méthode par transformation des caractéristiques d'échelle-invariante (SIFT) et la décomposition de valeurs singulières (SVD)

Dans cette méthode L'auteur propose un algorithme de hachage basé sur la transformation de caractéristiques invariantes à l'échelle (SIFT) et la décomposition en valeurs singulières (SVD) de cette manière. L'image d'entrée est prétraitée puis algorithme SIFT est utilisé pour extraire le point clé h. Les valeurs de hachage sont générées par le traitement des blocs et les opérations SVD Comme la montre la figure 3.1 suivante : [34]

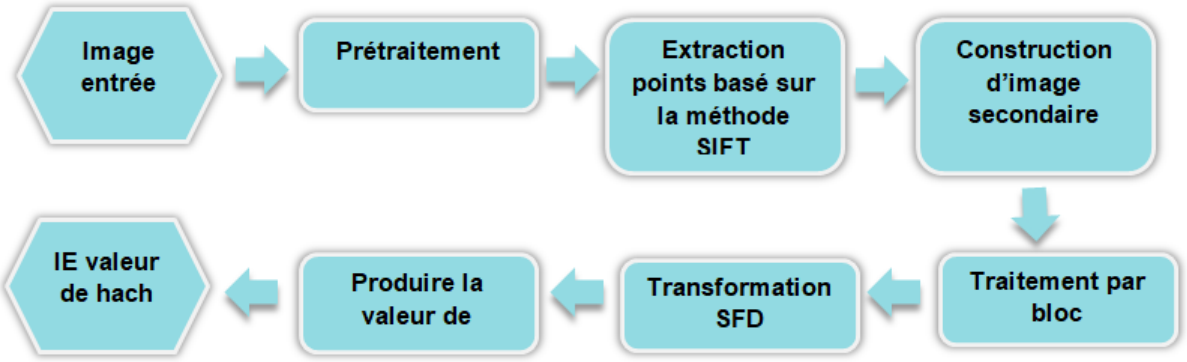


FIGURE 3.1 – Méthode transformation des caractéristiques d'échelle-invariante (SIFT) et la décomposition de valeurs singulières (SVD).

• **Prétraitement** : Dans cette étape, l'image d'entrée I est mise à l'échelle à la taille standard $(q \times q)$ à l'aide d'une interpolation bilinéaire afin que la valeur de hachage générée ait une longueur fixe. [34]

• **Extraction des points clés** : Après le prétraitement, la méthode SIFT est utilisée pour extraire les points clés de l'image en niveaux de gris mis à l'échelle, mais seuls les points clés créés après le filtrage des points négatifs sont utilisés. [34]

• **Construction d'image secondaire** : Dans cette étape, l'auteur convertit l'image J obtenue à l'étape précédente en une image noir et blanc I et restitue le pixel J (i, j) représentant le point clé avec la valeur 1 fixée à 0. [34]

• **Décomposition en bloc et application SVD** : Une fois l'image secondaire crée, l'image I en noir et blanc est divisée en blocs non superposés, de taille 46×46 après nous avons appliqués sur chaque bloc l'application SVD pour obtenir IA valeur singulière maximale pour chaque bloc utilisant, si la moyenne de toute les blocs Supérieur à la valeur singulière maximale pour chaque bloc on a 0 sinon 1, dans la fin, ou construisons le hach binaire. [34]

• **Mesure de similitude** : La similitude entre deux hach est mesurée à l'aide de la distance de Hemming (DH), la DH normalisé entre deux hach d'image est directement Proportionnel d la similitude des images. Un seuil prédéfini est utilisé pour classer des similaires entre différentes images. [34]

3.2.1.2 Hachage d'image robuste avec représentation de rang inférieur et Cloison de sonnerie

Dans cette méthode l'auteur propose un hachage d'image peut être divisé en quatre étapes : prétraitement, calcul de la représentation visuelle, représentation de bas rang et partition en anneau, Figure 3.2 est le diagramme de notre algorithme. Le prétraitement consiste à produire une image normalisée, et le calcul de la représentation visuelle consiste

à générer une représentation d'image qui peut indiquer les régions saillantes de l'image. L'utilisation de la représentation de bas rang consiste à extraire les principales caractéristiques de l'image pour effectuer le hachage discriminant, et l'utilisation de la partition en anneau peut rendre le code d'entité extrait invariant à la rotation de l'image. Ces étapes sont décrites en détail dans les sections suivantes :

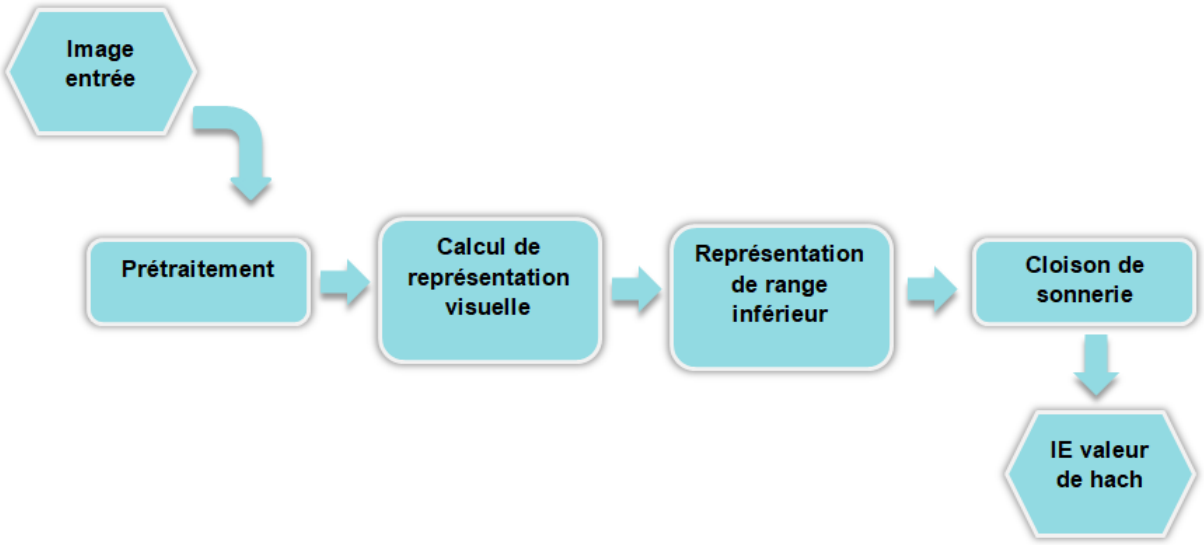


FIGURE 3.2 – Schéma de l'algorithme de hachage proposé.

• **Prétraitement** : Cette étape comprend trois opérations l'interpolation bilinéaire, le filtrage passe-bas gaussien et la conversion de l'espace colorimétrique. L'interpolation bilinéaire consiste à redimensionner l'image d'entrée à une taille $n \times n$ standard. Cette opération peut rendre notre algorithme résilient à la mise à l'échelle de l'image. Les 3 x 3 filtrages passe-bas gaussiens est ensuite appliqués à l'image redimensionnée, Enfin l'image filtrée dans l'espace colorimétrique RVB est convertie en espace colorimétrique HSV. [36]

• **Calcul de la représentation visuelle** : est exploité pour trouver la carte de saillance de l'image. Ensuite, la représentation visuelle de l'image peut être déterminée en combinant la carte de saillance et la composante de luminosité de l'image prétraitée. Ici, nous sélectionnons le SRM comme méthode de détection de la saillance [37]

• **Représentation de rang inférieur** : La représentation de rang inférieur (LRR) est une technique utile pour capturer la structure globale des données, Le LRR est robuste au bruit et peut extraire la représentation de rang le plus bas de toutes les données. [38]

• **Partition en anneau (RP)** : prend le centre de l'image comme centre du cercle et divise le cercle inscrit de l'image en un ensemble d'anneaux. Figure 3.3 présente un exemple de RP avec 4 anneaux d'image. [39]

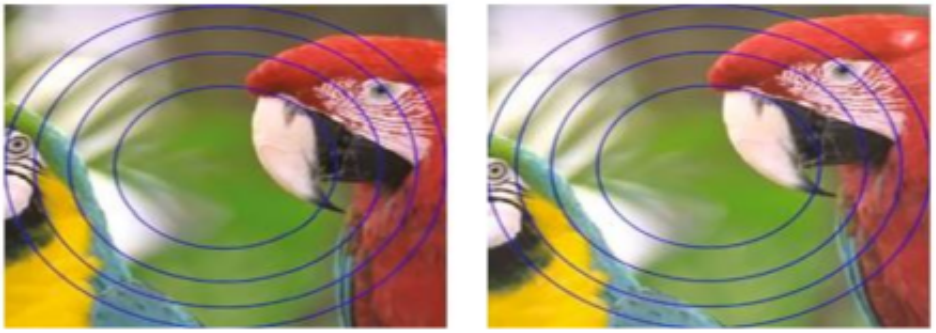


FIGURE 3.3 – Exemple de partition en anneau avec 4 anneaux .

3.2.1.3 Hachage robuste basé sur la transformation du gyrateur de quaternion pour l'authentification d'image

Dans cette méthode l'auteur propose un algorithme de hachage d'image en trois étapes comme le montre la Figure 3.4. **Dans la première** étape, l'image d'entrée en une taille uniforme par interpolation bicubique, et diviser l'image normalisée en blocs d'images qui ne se chevauchent pas. Les uns les autres. **Dans la deuxième** étape, la carte des caractéristiques locaux blocs est extraite par la transformée quaternion gyrator. **Dans la troisième** partie, en compressant la carte de caractéristiques en un vecteur et en effectuant un produit interne avec le vecteur de caractéristiques en un vecteur et un produit interne avec le vecteur aléatoire, elle est converti en hachage final par la fonction de binarisation. La différence entre les hachages sera calculée par la distance euclidienne [40]

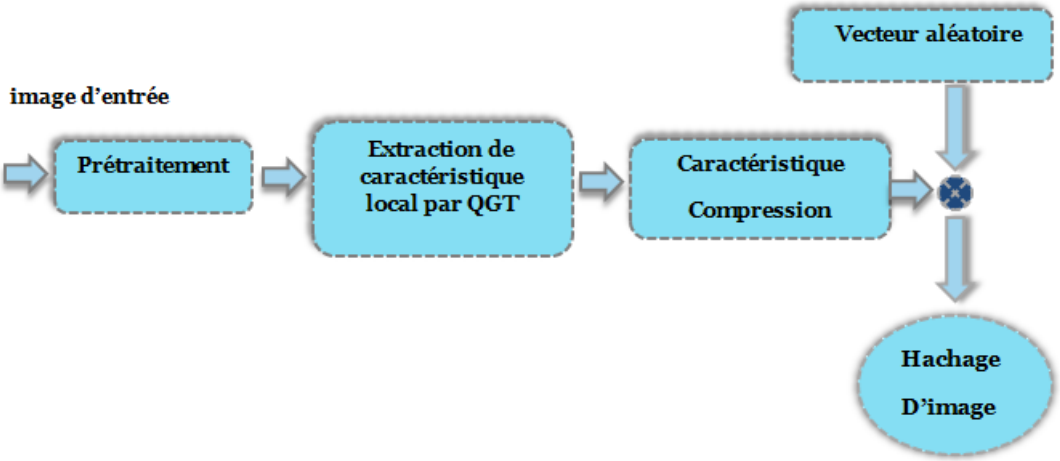


FIGURE 3.4 – Schéma de l'algorithme de hachage proposé.

• **Prétraitement** : Puisque les images réelles auront des tailles différentes d'unifier la taille de l'image d'entrée à l'aide de méthode d'interpolation bicubique, pour s'assurer

que le hachage généré plus tard est une taille fixe. Toutes les images d'entrée sont uniformément mises à l'échelle à la taille de $M \times M$, puis l'image d'entrée est divisée en $N \times N$ sous-blocs par des blocs non chevauchants. [40]

• **Extraction de caractéristiques** : la partie extraction de caractéristiques utilise la transformation QGT avec un angle α. La transformation gyrator quaternion peut être réalisée en divisant l'opération du gyrateur à un seul canal en trois fois [41], ou utilisé une autre méthode pour réaliser la transformation gyrator quaternion par la transformée de Fourier quaternion.

• **Génération de hachage** : pour chaque vecteur de caractéristiques normalisé $\hat{C}_i$ nous utilisons la formule suivante pour obtenir la valeur de hachage approximative de la première étape : [40]

$$H_B(i) = \left[\hat{C}_i * L + 0.5\right]$$

Ou L est un vecteur aléatoire généré par une clé à nombre entier positif
Ensuite, nous prenons l'opération de binarisation suivante pour chaque $H_B(i)$

$$\hat{H}_B(i) = \begin{cases} 1, & \text{if } h_j > h_{j+1} \\ 0, & \text{if } h_j \leq h_{j+1} \quad h_j \in H_B(i) \end{cases}$$

Où $j = 1, 2, \ldots k$

• **Evaluation de similarité** : La similarité entre les images est calculée par la distance euclidienne de leurs hachages correspondants. [40]

3.2.2 Méthodes de hachage perceptuel globales

3.2.2.1 Méthode utilisée filtre de Gabor et la probabilité d'absorption de Markov

Dans cette méthode l'auteur proposer le hachage robuste d'image perceptuel et il a utilisé Le filtre de Gabor et la probabilité d'absorption Markov. Les caractéristiques globales et locales sont extraites pour la formation du hachage. Le filtre de Gabor est utiliser pour extraire les caractéristiques globales.

Le filtre de Gabor conventionnel est modifié pour avoir une bonne propriété invariante contre la rotation et la rotation-invariant, le filtre est aléatoirement utilisé pour faciliter l'extraction et la sécurité des caractéristiques. La probabilité d'absorption de Markov est appliquée pour la détection des régions saillies [42] , puis la position et la texture des vecteurs sont calculées pour extraire les caractéristiques locales. Le cadre de cette méthode est représenté par la figure suivante :

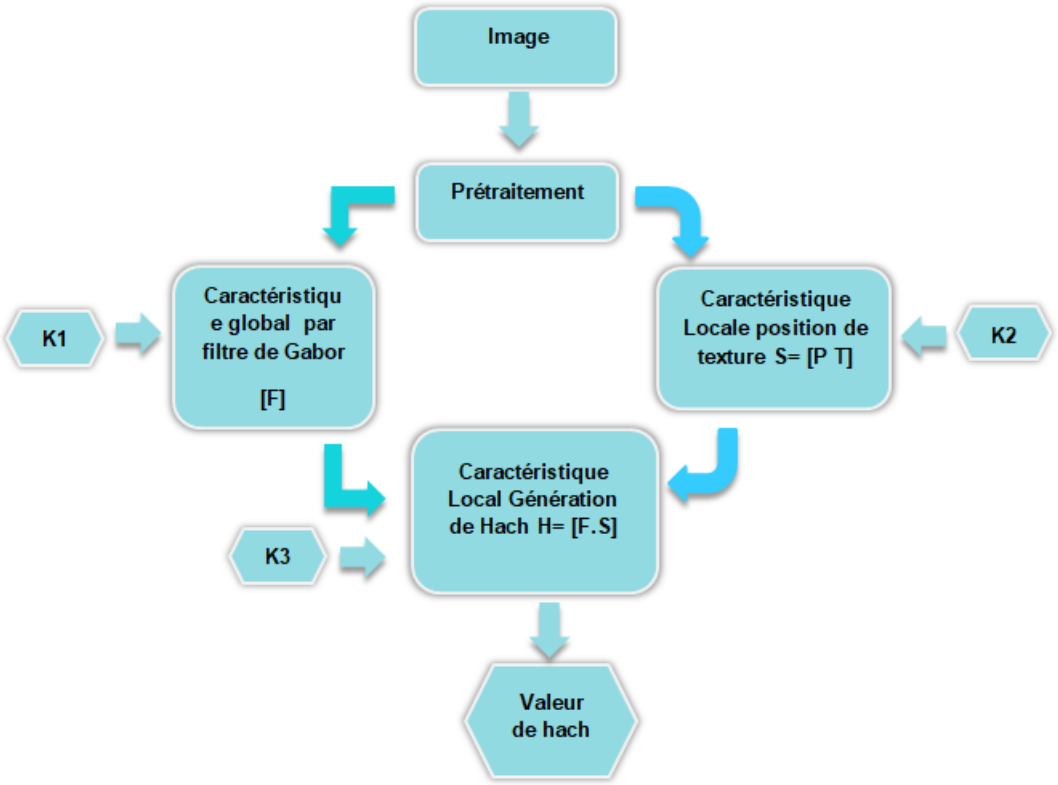


FIGURE 3.5 – Méthode utilisé le filtre de Gabor et la probabilité d'absorption de Markov.

Les étapes de cette méthode sont expliquées dans les paragraphes suivantes : [42]

• **Étape d'extraction des caractéristiques globales** : Premièrement, normaliser l'image d'entrée avec un filtre gaussien 5×5 ayant un écart standard de 3 puis redimensionner l'image dans une taille fixe de 256×256 , Après un ensemble des rayons est généré pour l'extraction des caractéristiques il l'aide de la clé sécurisée K_r. En suit nous calculons la fonctionnalité des rotation-invariante $F(i)$ pour chaque anneau.

• **Étape d'extraction des caractéristiques locales** : Diviser l'image en super pixels, âpres nous avons construire un graphique il deux anneaux pour refléter l'information de voisinage. Converti le modèle RVB au modèle CIE Lab. En utilisant les nœuds absorbants pour la limite supérieure et la limite gauche, nous obtenons les cartes de saillante, la carte de saillante est redéfinie pour obtenir la version finale des régions importantes. Après nous avons formez un vecteur de position d'élément K en utilisant les coordonnées du coin supérieur gauche et la largeur / hauteur de chaque rectangle autour de la région saillante. Les caractéristiques de texture sont extraites pour construire un vecteur K-élément. Le vecteur de position est concaténé avec le vecteur de texture. Une clé K_2 est utilisée pour le cryptage après de passer à la deuxième fonction de hachage intermédiaire en utilisant des fonctionnalités locales.

• **Étape génération de hache** : Enfin, les deux vecteurs de hachage intermédiaires sont concaténé à l'aide d'une clé sécurisée $K(3)$ brouillée pour obtenue le hachage d'image finale.

3.2.2.2 Méthode de hachage moyen

L'algorithme de hachage moyen est une version très simple d'un hachage perceptive. Cette méthode est très utilisable si nous voulons trouver des images similaires. Les principaux avantages de cet algorithme est sa rapidité et sa simplicité Cette méthode est décrite par le DrNeal Krawetz sur le blog Hacker Factor. L'idée de base était de filtrer les hautes fréquences dans une image et garder les basses fréquences. Avec les hautes fréquences, nous donnons des images en détail, tandis que les basses fréquences montrent la structure de cette dernière. Une image très petite manque de détails, il est donc toutes basses fréquences. l'algorithme Dr Neal décrit [48]

• **Étape 1** : Redimensionner a une taille de 8×8 communs. Le moyen le plus rapide pour supprimer les fréquences élevées pour rétrécir l'image.

• **Étape 2** : Niveaux de gris. Cela modifie le hachage de 64 pixels (64 rouges, 64 verts, 64 bleus) il 64 couineurs au total.

• **Étape 3** : Calculer la valeur moyenne des 64 couleurs. C'est la moyenne de la valeur de hach.

• **Étape 4** : Convertir les 64 couleurs i, 64 bits. Chaque bit est simplement régler en fonction de la moyenne (la valeur de couleur est au-dessus d'la moyenne).

• **Étape 5** : Construire le hachage.

• **Étape 6** : Pour comparer deux images, calculer la distance de Hemming entre deux haches moyenne. Une distance de zéro indique qu'il s'agit probablement d'une image très proche (ou une variante de la même image). Une distance de 5 signifie quelque choses peuvent être différentes, mais ils sont probablement encore assez proche pour être similaire. Mais une distance de 10 ou plus, c'est probablement une image très différente.

Selon le Dr Neal [48], le hach ne changera pas si l'image est redimensionnée. Augmenter ou diminuer la luminosité ou la contraste, ou même modifier les couleurs ne changera pas considérablement la valeur de hach.

3.3 Hachage d'image perceptuel basé sur les caractéristiques fractales structurelles du codage d'image et de la partition en anneau

Dans notre méthode implémenté le système de hachage d'images implémenté comprend trois étapes principales, comme l'illustre la figure (3.6) Dans la première étape, l'image d'entrée est prétraitée afin de construire l'image normalisée.

Ensuite, l'anneau d'image est extrait de l'image normalisée afin de construire la version secondaire de l'image partitionnée. Et dans la dernière étape, le codage fractal est appliqué à l'image secondaire pour pouvoir extraire le contenu visuel de chaque anneau et finalement le crypter pour améliorer la sécurité.

Les sous-sections suivantes donnent La description détaillée de chaque étape (nous

utilisons la article [43] :

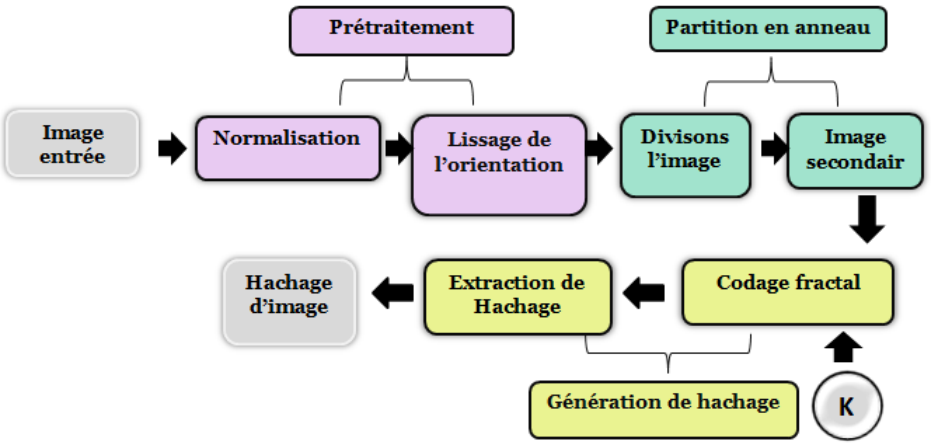


FIGURE 3.6 – Schéma de l'algorithme de hachage implémenté.

• **Prétraitement** :

Dans la prenant de la photo du réseau veineux il y a deux problèmes [43] :

1. d'abord c'est que les images capturé n'est pas toujours de la même taille. Alors les images de différentes résolutions ne pas conservent la même valeur de hachage pour cela nous avons utilisé. L'interpolation bilinéaire redimensionne l'image d'entrée I_0 avec une taille arbitraire de I avec une 0 taille de $M \times M$.
2. Il existe aussi de nombreuses distorsions dans la main (brulures et, densité de cheveux, etc.) ou du dispositif d'acquisition (bruit dans l'image enregistrée), la qualité de l'image entrée peut nettement décroitre. Une mauvaise qualité d'image influera grandement le processus d'extraction.

Pour remédier à cela, nous utilisons une technique d'amélioration qui repose sur le filtre de Sobel.

Ce dernier est un opérateur utilisé en traitement d'image pour la détection de contours. Consiste à appliquer des opérateurs afin de faire des transformations sur toute l'image ou à une partie d'elle pour améliorer la qualité visuelle de cette dernière.

Les étapes pour appliquer Sobel sont les suivantes :

1. L'image donnée doit être divisée en plusieurs blocs d'images en blocs non chevauchants de taille 16×16 pixels et l'extraction de caractéristiques est appliquée à chaque bloc.

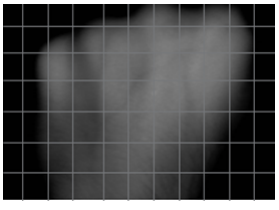


FIGURE 3.7 – Division de l'image en plusieurs blocs.

2. $d_x(i,j)$ et $d_y(i,j)$ les gradients au pixel (i,j) de chaque blocs pour l'extraction de caractéristiques sont obtenus.
3. Lisser les gradients obtenus ci-dessus, l'aide d'un filtre approprié.

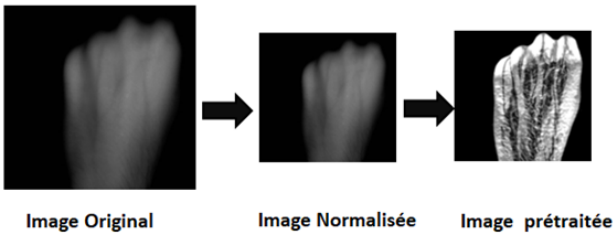


FIGURE 3.8 – Le résultat obtenu après le prétraitement.

• **Partition en Anneau** : A partir d'image de veineux préalablement traitée nous divisons cette image à des anneaux et les utilisons pour former une image secondaire.

Les étapes mises en œuvre :

1. Soit n le nombre total d'anneaux (nous chaisiers $n = 5$) et r_n le rayon des anneaux. Nous calculons la surface du cycle d'image inscrit comme $A = \pi r_n^2$, et chaque anneau concentrique est donné par $\alpha_A = |\frac{A}{n}|$.
   Par conséquent, le rayon du premier anneau peut être calculé comme suit :

$$r_1 = \frac{\sqrt{\alpha_A}}{\pi} \tag{3.1}$$

   De même, les rayons restants du cycle concentrique qui détermine le rayon de le $i^{ième}$ anneau sont donnés par l'équation suivante :

$$r_i = \frac{\sqrt{\alpha_A + \pi r_{i-1}^2}}{\pi} \tag{3.2}$$

2. Soit (x_c, y_c) les coordonnées du centre de l'image normalisée avec la distance entre $p(x,y)$. La valeur du pixel dans la $Y^{ième}$ rangée et la $X^{ième}$ colonne est désignée par :

$$d_{x,y} = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{3.3}$$

3. À partir des distances entre les pixels et des rayons des cycles concentriques obtenus, les valeurs des pixels peuvent être classées en n ensembles. R_i représente un ensemble de ces valeurs de pixel dans le $i^{ème}$ anneau $(i = 1, 2..., n)$. Il est donné comme suit :

$$R = \begin{cases} R_1 = \{p(x,y) & | \, d_{x,y} \geq r_1\} \\ R_2 = \{p(x,y) & | \, d_{x,y} \geq r_2\} \\ R_2 = \{p(x,y) & | \, d_{x,y} \geq r_3\} \end{cases} \tag{3.4}$$

$$R_i = \{p(x,y) \mid r_{i-1} \geq d_{x,y} \geq r_i\} \;\; (i = 3, 4, \ldots, n) \tag{3.5}$$

4. Les éléments de R_i $(i = 1, 2, \ldots n)$ sont ensuite réorganisés dans un ordre croissant pour obtenir le vecteur trié V_i qui n'est pas corrélé à l'opération de rotation. Cependant, ce vecteur trié V_i est également sujet à des déformations naturelles qui modifient les valeurs des pixels des anneaux. Pour surmonter cette limitation, V_i est ensuite mis en correspondance avec un nouveau vecteur W_i afin de s'assurer que les pixels de chaque anneau forment une colonne dans l'image secondaire S et restent inchangés après la rotation L'illustration schématique de cette approche est présentée à la figure 3.9. Par conséquent, puisque W_i n'est pas lié à la rotation, l'image secondaire S devient également invariante par rapport à la rotation. Ainsi, l'image secondaire S est obtenue à partir de l'équation suivante :

$$S = [w_1, w_2, w_3, \ldots w_n] \tag{3.6}$$

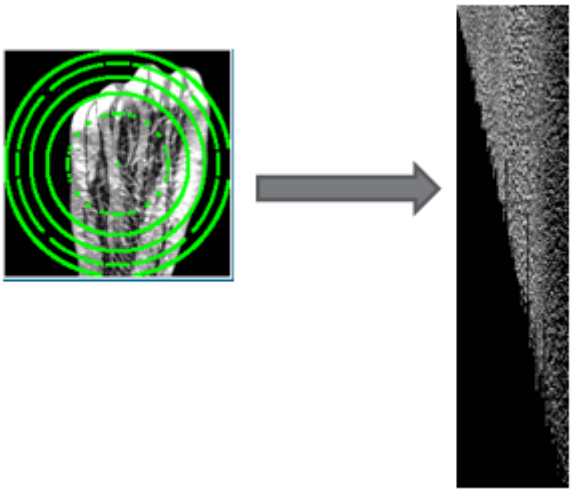


FIGURE 3.9 – Partition en anneau d'une image et image secondaire .

• **Génération de hachages** :

Après l'étape de partition en anneau, nous appliquons le codage d'image à l'image secondaire S obtenue afin de pouvoir générer les caractéristiques du contenu visuel de chaque anneau.

La procédure de notre génération de hachage est présentée comme suit :

1. L'image secondaire S est d'abord représentée comme un bloc de domaine fractal étendu. Bloc $\tilde{S}_i = (\tilde{S}_1, \tilde{S}_2, \ldots \tilde{S}_n)$.
2. Pour trouver l'élément correspondant pour chaque anneau dans le bloc de l'intervalle $\hat{S}_i = (\hat{S}_1, \hat{S}_2, \ldots \hat{S}_n)$, une transformée contractive affine M_i est définie comme suit :

$$M = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & \hat{c}_i \end{bmatrix} \begin{bmatrix} v_i \\ \vdots \\ v_n \end{bmatrix} + \begin{bmatrix} \tilde{S}_i \\ \hat{S}_i \\ \hat{b}_i \end{bmatrix} \tag{3.7}$$

Où ;

- x et y sont les coordonnées spatiales et z est la valeur du pixel.

- $\begin{bmatrix} v_i \\ \vdots \\ v_n \end{bmatrix}$ est le vecteur trié pour les éléments de chaque anneau.
- $\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ désigne l'une des huit transformations symétriques sur chaque anneau.
- $(S_i, \hat{S}_i)$ représente les coordonnées de l'image secondaire par rapport au bloc du domaine fractal.
- $\hat{b}_i$ et $\hat{c}_i$ sont respectivement les commandes de luminosité et de contraste.

3. Afin d'obtenir la meilleure correspondance entre le bloc de domaine fractal $\tilde{S}_i$ et celui de l'étendue $\hat{S}_i$ dans M, nous effectuons la minimisation comme dans l'équation (3.8) ci-dessous.

$$M(\tilde{S}_i, \hat{S}_i)^2 = \min_n \|p(\tilde{S}_i - j_m I) - (\hat{S}_i - k_m I)\|^2 \tag{3.8}$$

où p est le paramètre d'échelle optimal, j_m représente la moyenne de $\tilde{S}_i$, k_m désigne la moyenne de $\hat{S}_i$, I est la taille de $\hat{S}_i$ et $\|.\|$ est le pool de domaine à deux normes toutes égales à 1.

4. Le paramètre d'échelle optimal est obtenu en calculant la méthode des moindres carrés en utilisant l'équation (3.9)

$$p = \frac{\left\langle (\tilde{S}_i - j_m I) - (\hat{S}_i - k_m I) \right\rangle}{\|\tilde{S}_i - j_m I\|^2} \tag{3.9}$$

5. Afin d'assurer la compacité de notre hachage et que le résultat de la transformation de la caractéristique ainsi bloc de gamme sont aussi minimes que possible, nous définissons le paramètre d'échelle optimal p_n en déduisant la moyenne de chaque bloc comme ci-dessous. En déduisant la moyenne de chaque bloc comme dans le schéma ci-dessous.

$$p_n \approx p = \frac{\tilde{S}_i(\tilde{s}_1, \tilde{s}_2, \ldots \tilde{s}_n) - j_m}{\hat{S}_i(\hat{s}_1, \hat{s}_2, \ldots \hat{s}_n) - k_m} \tag{3.10}$$

6. Après la compression ci-dessus, nous concaténons les trois séquences de caractéristiques perceptuelles obtenues, c'est-à-dire. $p_n.I.M$. Puis nous générons une séquence binaire $H = [p_n.I.M]$. Cependant, afin d'assurer la sécurité de notre schéma de hachage, nous chiffrons le hachage H généré en utilisant une clé secrète K pour brouiller les bits de la séquence $H = [p_n.I.M]$.

La séquence de brouillage finale $H^{'}$ est le hachage final de l'image originale Nm. La figure 3.10 représente l'organigramme du schéma de génération de hachage. Et l'algorithme 1 montre le pseudo-code de la génération de hachage.

• **Évaluation de la similarité des hachages** : Pour mesurer la similarité entre deux hachages d'images, nous avons adopté la distance de Hamming normalisée comme

métrique donnée dans l'équation (3.11).

$$Dis(H_1^{'}, H_2^{'}) = \frac{1}{l}\sum_{i=1}^{l} \|H_1^{'}(i) - H_2^{'}(i)\| \tag{3.11}$$

Où $H_1^{'}$ et $H_2^{'}$ sont deux séquences de hachage de deux images distinctes, l'est la longueur de hachage et $H_1^{'}(i)$ et $H_2^{'}(i)$ désignent les i-èmes bits de $H_1^{'}$ et $H_2^{'}$ respectivement.

En général, plus la distance de Hamming normalisée, plus les images des hachages correspondants sont similaires.

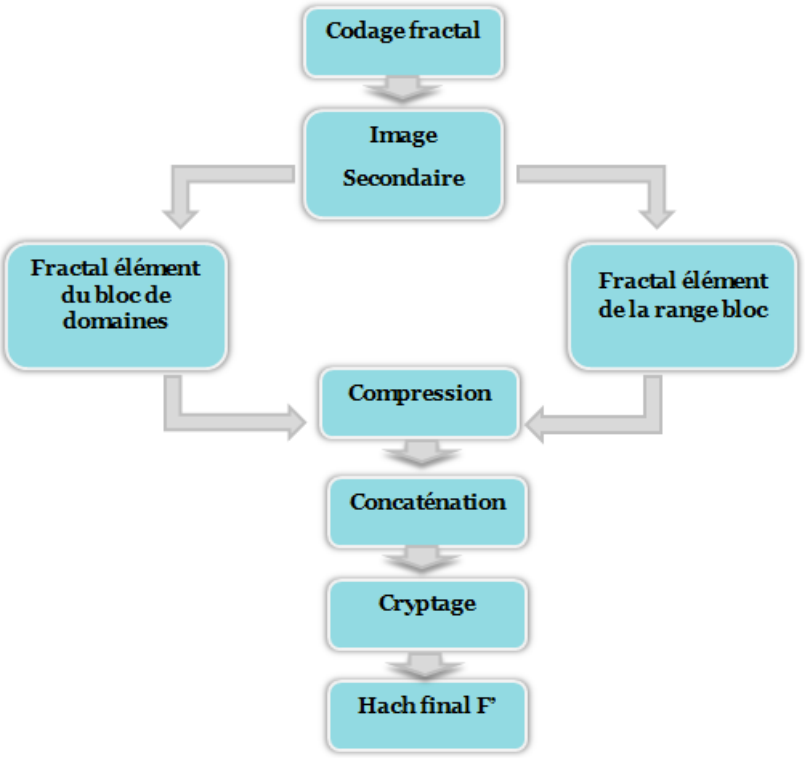


FIGURE 3.10 – Organigramme de la génération de hachage.

Si $Dis(H_1^{'}, H_2^{'})$ est inférieur au seuil prédéfini T, alors les images des hachages d'entrée sont considérées comme virtuellement identiques. Dans le cas contraire, elles sont considérées comme différentes

3.4 Méthode fractale

La compression fractale d'images a été proposée pour la première fois par Barnsley [44]. Cette méthode, basée sur le théorème du collage [45], montre qu'il est possible décoder des images fractales à l'aide de quelques transformations contractantes définissant un système de fonctions itérées (SFI). Barnsley proposa un algorithme pour construire, à partir d'une image donnée, un ensemble de transformations contractantes la représentant. Les signaux naturels ne possédant pas forcément la propriété d'auto-transformabilité globale, Jacquin

[46] proposa de rechercher des auto-transformabilités Locales ou partielles ce qui a conduit au premier algorithme de compression par SFI et à la notion de système de fonctions itérées locales (SFIL)

3.4.1 Contractions

Soit (E, d) un espace métrique complet et $f : E \to E$ une correspondance de E à E. Nous dirons que f est une contraction s'il existe $0 < s < 1$ tel que :

$$\forall x, y \in E,\ d(f(x), f(y)) \leq s\, d(x, y)$$

À partir de maintenant, f désignera une contraction avec un facteur de contractivités. Il existe deux théorèmes importants sur les contractions : le théorème de la cartographie des contractions et le théorème du collage. [47]

- **Théorème (Théorème de la cartographie par contraction)** : f à un unique point fixe x_0. A partir de maintenant, x_0 désignera le point fixe de f.
- **Théorème (Théorème de Collage)** : Si $d(x, f(x)) < \varepsilon$ alors $d(x, x_0) < \varepsilon 1 - s$.

Le deuxième théorème nous dit que si nous trouvons une contraction f telle que $f(x)$ est proche de x alors nous sommes sûrs que le point fixe de f est aussi proche de x.

Ce résultat sera fondamental dans la suite. En effet, au lieu de sauvegarder une image, nous ne sauvegarderons qu'une contraction dont le point fixe est proche de l'image.

3.4.2 Contractions pour les images

Dans cette partie, nous allons montrer comment construire une contraction telle que son point fixe soit proche d'une image donnée. [47]

Tout d'abord, définissons l'ensemble des images et une distance. Nous choisissons $E = [0, 1]^{h \times w}$. E est l'ensemble des matrices avec h lignes, w colonnes et avec des coefficients dans $[0, 1]$. On prend alors $d(x, y) = (\sum_{i=1}^{h} \sum_{j=1}^{w} (x_{ij} - y_{ij})^2)^{0,5}$. d est la distance obtenue à partir de la norme de Frobenius.

Soit maintenant $x \in E$ l'image que nous voulons compresser. Nous allons segmenter deux fois l'image en blocs :
Tout d'abord, nous partitionnons l'image en blocs de destination ou de plage $R_1, \ldots, R_L$. Ces blocs sont disjoints et ils couvrent toute l'image.
Ensuite, nous segmentons l'image en blocs de source ou de domaine $D_1, \ldots, D_K$. Ces blocs ne sont pas nécessairement disjoints et ne couvrent pas nécessairement l'image. Par exemple, nous pouvons segmenter l'image comme suit :

Ensuite, pour chaque bloc de plage R_l, nous choisirons un bloc de domaine D_{k_l} et une cartographie

$$f_l : [0, 1]^{D_{k_l}} \to [0, 1]R_l.$$

Enfin, nous pouvons définir notre fonction f comme :

$$f(x)_{ij} = f_l(xD_{k_l})_{ij} \quad \text{si } (i, j) \in R_l$$

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}
R_{17}	R_{18}	R_{19}	R_{20}	R_{21}	R_{22}	R_{23}	R_{24}
R_{25}	R_{26}	R_{27}	R_{28}	R_{29}	R_{30}	R_{31}	R_{32}
R_{33}	R_{34}	R_{35}	R_{36}	R_{37}	R_{38}	R_{39}	R_{40}
R_{41}	R_{42}	R_{43}	R_{44}	R_{45}	R_{46}	R_{47}	R_{48}
R_{49}	R_{50}	R_{51}	R_{52}	R_{53}	R_{54}	R_{55}	R_{56}
R_{57}	R_{58}	R_{59}	R_{60}	R_{61}	R_{62}	R_{63}	R_{64}

3.4.3 Segmentations

Nous garde les choses très simples. Les blocs source et les blocs destination segmentent l'image sous forme de grille, comme sur l'image ci-dessus. [47]

La taille des blocs est une puissance de deux, ce qui facilite les choses. Les blocs source sont de 8 par 8 tandis que les blocs de destination sont de 4 par 4.

3.4.4 Transformations

Dans cette section, nous allons montrer comment construire les contractions de D_k à R_l.

Nous voulons générer une correspondance f_l telle que $f(xD_k)$ est proche de xR_l. Ainsi, plus nous générons de mappings, plus nous avons de chances d'en trouver un bon. [47]

Cependant, la qualité de la compression dépend du nombre de bits nécessaires pour sauvegarder f_l. Ainsi, si nous avons un ensemble de fonctions trop grand, la compression sera mauvaise. Il y a donc un compromis à trouver.
J'ai choisi que f_l aura la forme suivante :

$$f_l(xD_k) = s \times \ rotate_\theta(flip_d\,(reduce(xD_k))) + b$$

Où reduce est une fonction permettant de passer de blocs 8 par 8 à des blocs 4 par 4, flip et rotate sont des transformations affines, s modifie le contraste et b la luminosité.

- Fonction **reduce** réduit la taille d'une image en calculant la moyenne des voisinages.
- Fonction **rotate** fait simplement pivoter l'image de l'angle donné. Afin de préserver la forme de l'image, l'angle θ sera en $\{0°, 90°, 180°, 270°\}$
- Fonction **flip** retourne l'image si direction est égal à -1 et ne le fait pas si elle est égale à 1
- Toute la transformation est effectuée par la fonction **apply_transformation** :

3.4.5 Compression

L'algorithme de compression est simple. Nous générons d'abord toutes les transformations affines possibles de tous les blocs sources en utilisant la fonction **generate_all_transfomred_blocks** [47]

Puis pour chaque bloc destination, on essaie tous les blocs sources transformés précédemment générés. Pour chacun nous optimisons le contraste et la luminosité à l'aide de la méthode **find_contrast_and_brightness2** et si la transformation testée est la

meilleure que nous ayons vue jusqu'à présent, nous l'enregistrons .
Dans la compression nous obtenons 6 paramètre k, l, direction, angle, contrast, brightness.

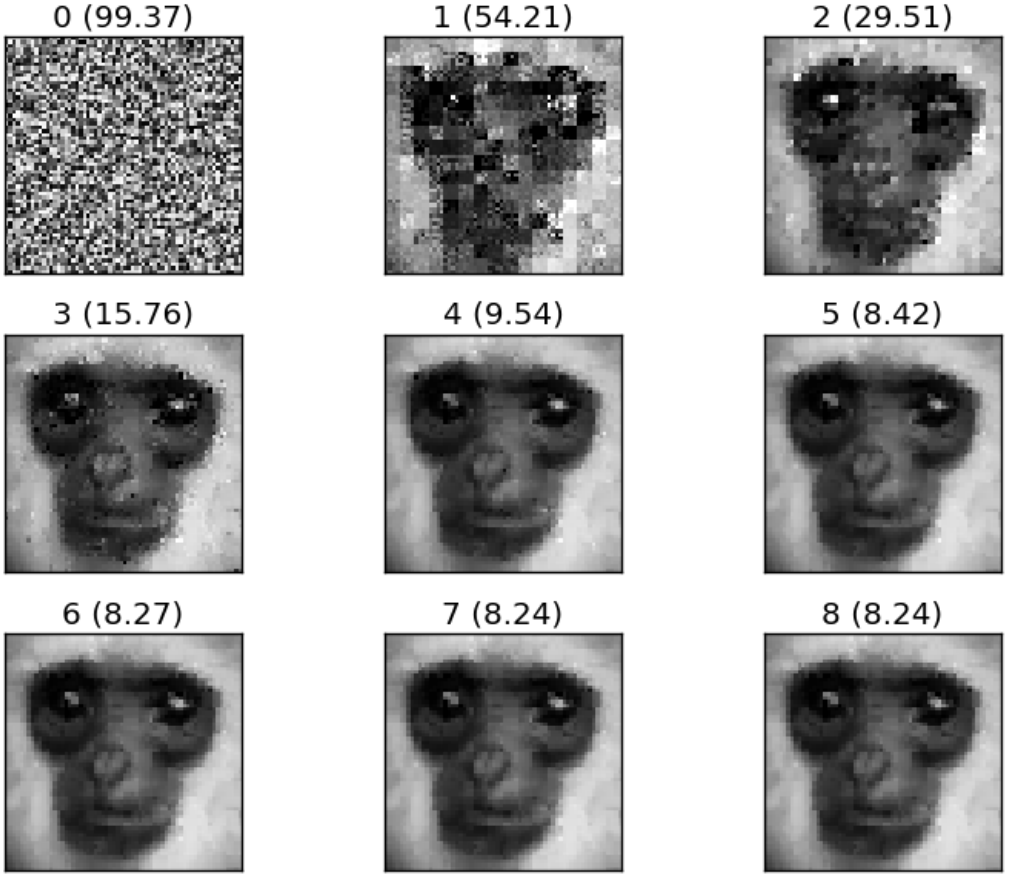
Pour trouver le meilleur contraste et la meilleure luminosité, la méthode **find_contrast_and_brightness2** résout simplement un problème des moindres carrés

3.4.6 Décompression

L'algorithme de décompression est encore plus simple. On part juste d'une image complètement aléatoire puis on applique la contraction F plusieurs fois [47]

Cela fonctionne parce que la contraction a un point fixe unique et quelle que soit l'image initiale que nous choisissons, nous y tendrons.

La fonction **test_greyscale** charge l'image, la compresse, la décompresse et affiche chaque itération de la décompression : [47]



3.5 conclusion

Dans ce chapitre nous avons mentionné les différentes méthodes utilisées dans le hachage perceptuel des images, qui se divisent en méthodes de décomposition en bloc et les méthodes globales ensuite nous avons détaillé les différents principes et étapes de la méthode implémenté, qui repose sur la construction d'un hach perceptuel qui doit être

robuste et sure en utilisant la méthode basé sur le codage fractale et de la partition en anneau. La performance de la méthode implémenté est évaluée en l'appliquant sur une data base, of les résultats obtenus seront analyser sous plusieurs critères, chose qui sera abordé dans le chapitre suivant.

CHAPITRE 4

TEST ET RÉSULTATS EXPÉRIENTAUX

4.1 Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes de la conception de la méthode proposé "Hachage image robuste à base le codage fractale" nous présentons dans ce chapitre un aperçu général sur la phase pratique de notre travail. Le but de ce projet est de construire un hach robuste et sure pour la sécurisation de notre base de données de réseau veineux de main en appliquant certaine manipulation acceptable comme la rotation, le scaling, la compression, nous mettons en évidence les raisons de nos choix technique, les tests sur l'application et les résultats obtenue.
Nous résumons cette mise en œuvre en trois parties :

- Environnement de développement
- Présentation l'application
- Tests et évaluation

4.2 Environnement de développement

4.2.1 Matériel

Nous avons utilisés deux machines avec les caractéristiques suivantes :

Caractéristiques	**Machine 1**	**Machine 2**
Processeur	2.16 GHz Intel(R) Pentium(R) CPU N3540	2.0 GHz Intel (R) Pentium(R) CPU B960.
RAM	4 Go	4 Go
Carte graphique	Intel(R) HD Graphics	Intel(R) HD Graphics 4400
Système d'exploitation	Windows 10 64 bits	Windows 10 64 bits

TABLE 4.1 – Caractéristiques de machines utilisées

4.2.2 Software

L'environnement logiciel utilisé pour la réalisation de notre application est : **Python**

FIGURE 4.1 – Logo Python

Nous avons utilisés la version **Python 3.7** comme langage de programmation, parmi les raisons de cette utilisation :

- Python fonctionne sur différentes plateformes (Windows, Mac, Linux).
- Il a une syntaxe simple claire, respecte les standards du domaine. Similaire à la langue anglaise.
- Langage peut être traite de manière procédurale, de manière orientée objet ou de manière fonctionnelle.

4.2.3 Bibliothèque

L'une des grandes forces du langage python est le grand nombre de bibliothèques logicielles externes disponibles. Une bibliothèque est un ensemble de fonctions. Elles sont regroupées et mises à disposition afin de pouvoir les utiliser sans avoir à les réécrire.

Ces bibliothèques peuvent être utilisées pour : le calcul numérique, les graphiques, la programmation internet ou réseau, la mise en forme de texte, la gestion de documents....

- Module PIL : La bibliothèque PIL (Python Imaging Librairie) permet manipulation de tout type d'images et fournit quelques fonctions de traitement d'images de base
- Numpy : Numpy est une bibliothèque numérique apportant le support efficace de larges tableaux multidimensionnels, et de routines mathématiques de haut niveau.
- MatPlotlib : Matplotlib est une bibliothèque destinée à tracer et visualiser des données sous formes de graphiques.

- Open CV : cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données.

- Tkinter : Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl.

- SciPy : La librairie SciPy contient de nombreuses boites à outils consacrées aux méthodes de calcul scientifique. Ses différents sous-modules correspondent à différentes applications scientifiques, comme les méthodes d'interpolation, d'intégration, d'optimisation, de traitement d'image, de statistiques, de fonctions mathématiques spéciales, etc

- Os : fonctions permettant d'interagir avec le système d'exploitation.

- Math toutes les fonctions utiles pour les opérations mathématiques (cosinus, sinus, exp, ... etc).

- Random : Des fonctions permettant de travailler avec des valeurs aléatoires

4.3 Présentation de l'application

Le but de notre travail est de tester la performance de notre méthode de hachage (coté robustesse et discrimination) contre quelques attaques acceptables, sur une base d'image de réseau veineux de main, nous pouvons mesurer la performance de cet algorithme

4.3.1 Base de données utilisée

Pour évaluer la méthode proposée dans ce mémoire nous l'avons appliqué sur certain images (48 images) de la base de données des veines dorsales de la main SUAS, la résolution de ces images est (640×480) ainsi que son format est JPEG.

Il existe dans cette base de données trois échantillons par réseau veineux de main. La figure 4.2 suivante montre cette base de données :

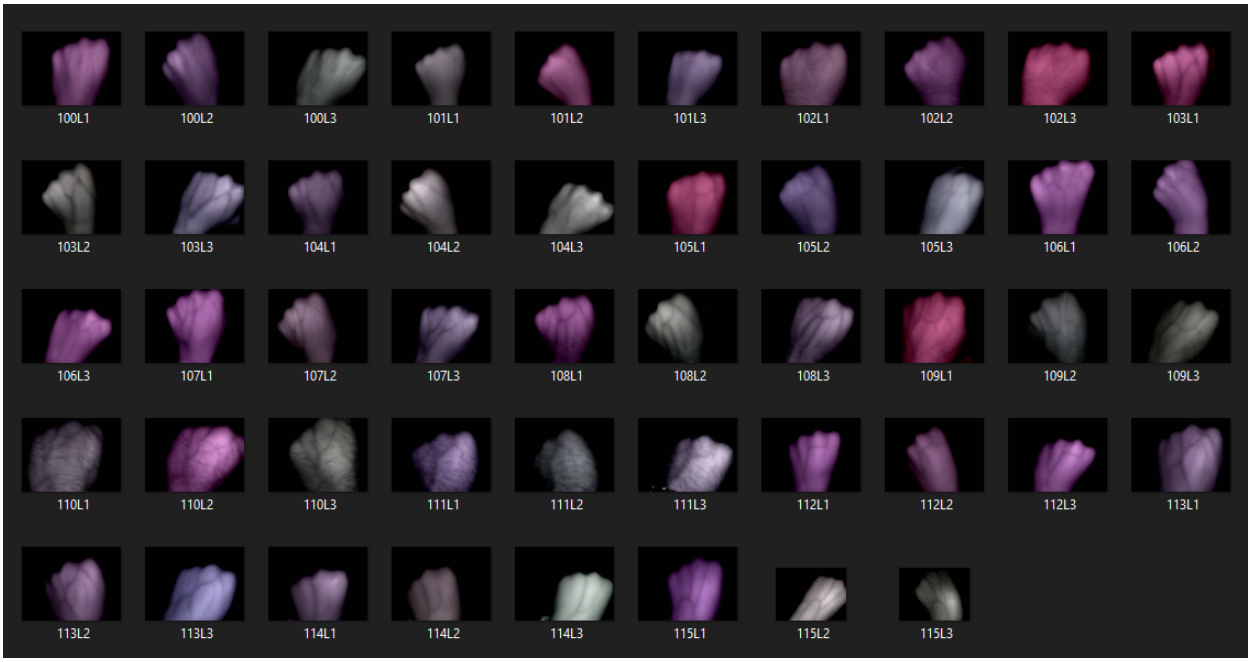


FIGURE 4.2 – La base de données utilisée

4.3.2 Attaques acceptables utilisées

Nous appliquons quelques manipulations acceptables tel que la rotation, la compression, bruits gaussien, correction Gamma et scaling. Le tableau 4.2 montre les paramètres de chaque attaque

Attaque	**Paramètre**
Rotation	+-10,+-15,+-30,+-45,+-90
Compression JPG	10, 20, 30, 40, 50, 60, 70, 80, 90
Correction Gamma	0.75, 0.9 ,1.1, 1.25, 1.3
Scaling	0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 2.0
Bruit Gaussien	0.001, 0.005, 0.010, 0.015, 0.020, 0.025, 0.030, 0.035
Bruit sel et poivre	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08

TABLE 4.2 – Paramètre utilisé pour chaque manipulation

4.3.3 Interface graphique

Dans cette partie, nous allons présenter les différentes phases de la réalisation de notre application. Après le lancement de l'application la fenêtre d'accueil s'affiche :

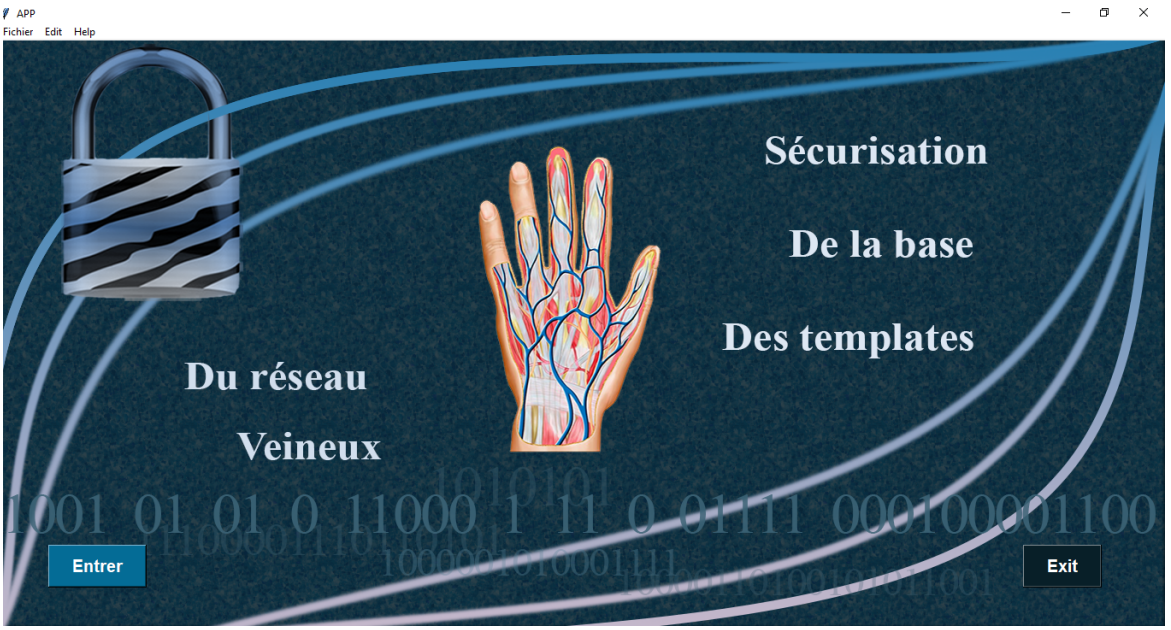


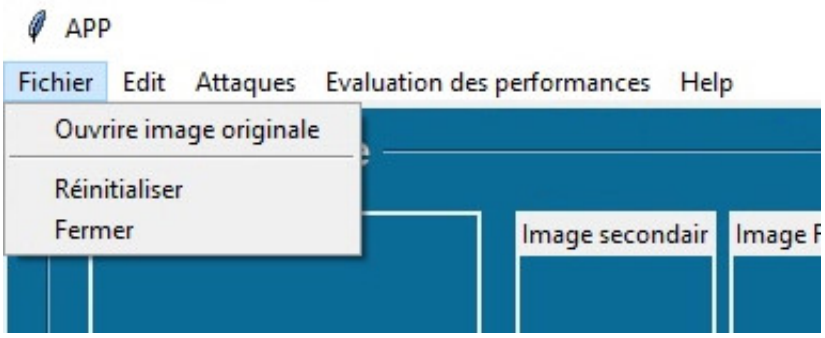
FIGURE 4.3 – La représentation de la page d'accueil

L'interface graphique donnée par la figure 4.3 est composé de deux boutons :

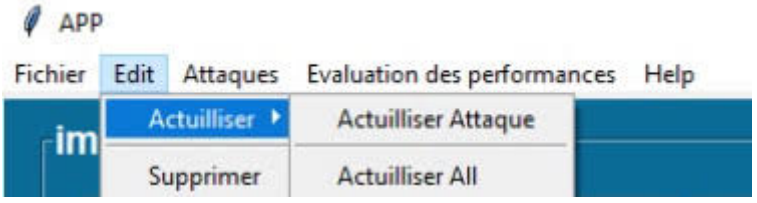
1. Le bouton **Exit** : pour fermer l'application.
2. Le bouton **Entrer** : pour accéder à la page principale, le but de cet onglet est d'accéder à la page qui fait le travail demandé, elle constitué de différentes étapes, la figure suivante montre cette page.

L'interface graphique donnée par la figure 4.4 est composé d'une barre des menus qui contient :

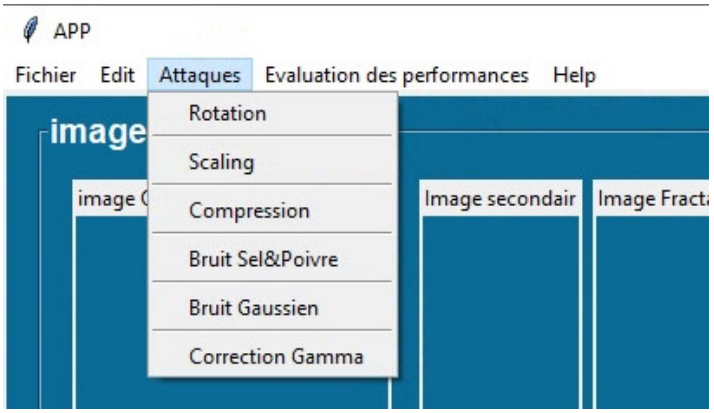
FIGURE 4.4 – La représentation de la page principale



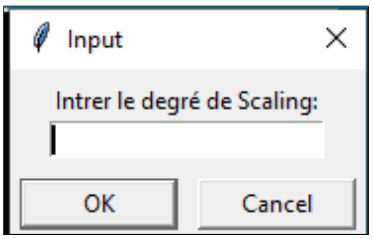
1. **Fichier** : contient les opérateurs suivants :
   - Ouvrir image originale : Permet d'ouvrir une image pour la traité, et à partir de cette action le bouton smothing de la partie d'image originale sera activé.
   - Réinitialiser : permet de Réinitialiser toutes les composants de l'interface graphique.
   - Fermer : Permet de quitter l'interface



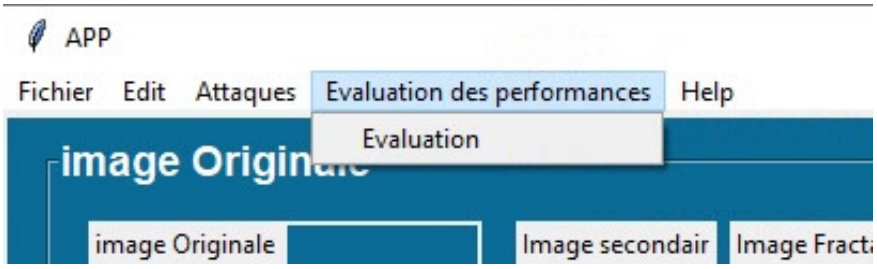
2. **Edit** : contient les opérateurs suivants :
   - Actualiser Originale : permet de Réinitialiser tous les composants de la partie graphique d'image attaquée.
   - Supprimer : permet de supprimer tous les dossiers qui sauvegarde les informations des images traitée.



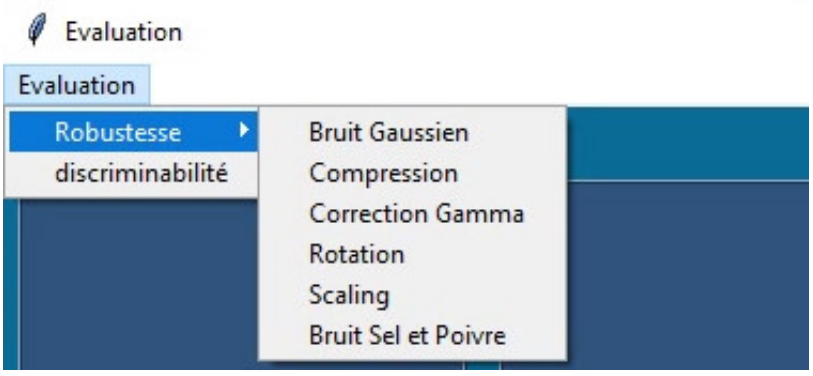
3. **Attaques** : Après avoir ouvrir l'image originale, le menu "Attaques" permet de choisir une attaque parmi les six attaques déclarées qui sont : rotation, scaling, compression, correction gamma, bruit gaussien, bruit sel et poivre.



Fenêtre



4. **Evaluation des performances** : permet d'afficher la fenêtre pour le calcul des résultats.



- **Evaluation** : Elle est décomposée sur deux pages, une pour la "Robustesse" et la deuxième pour la "Discrimination".

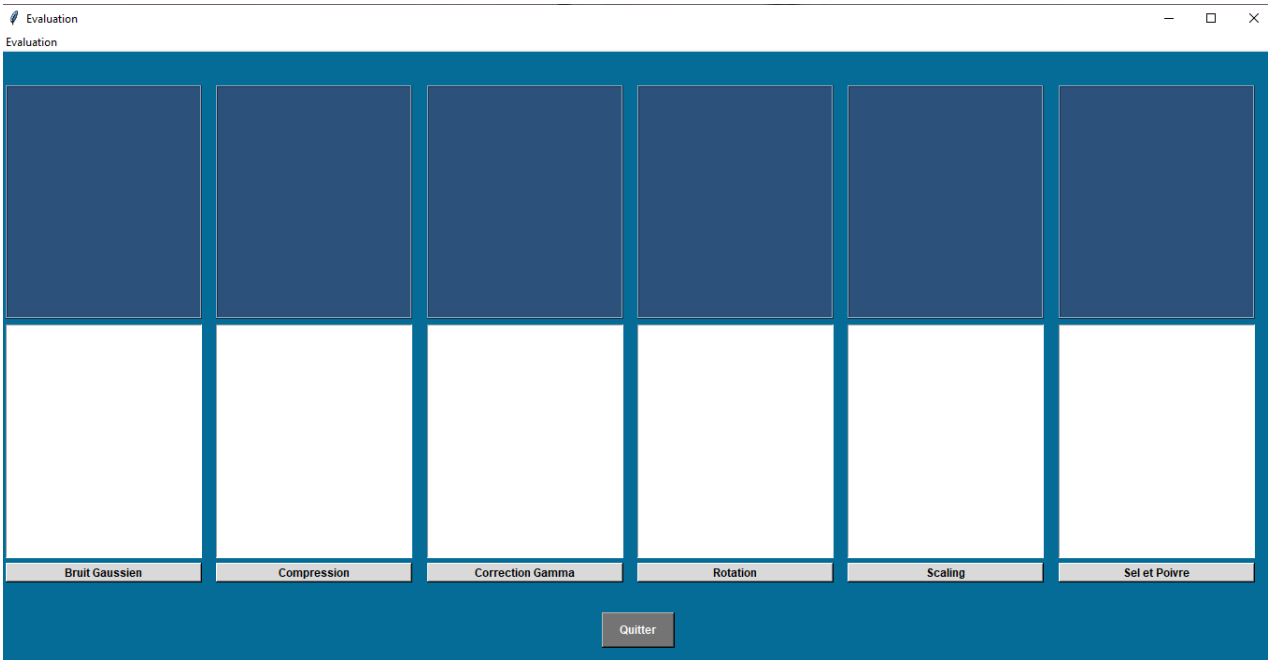


FIGURE 4.5 – Fenêtre pour L'évaluation des performances (Robustesse) .

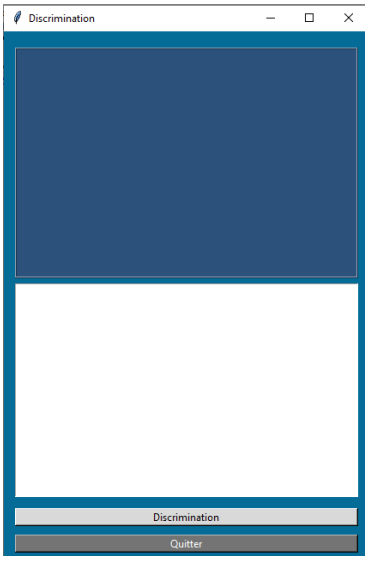
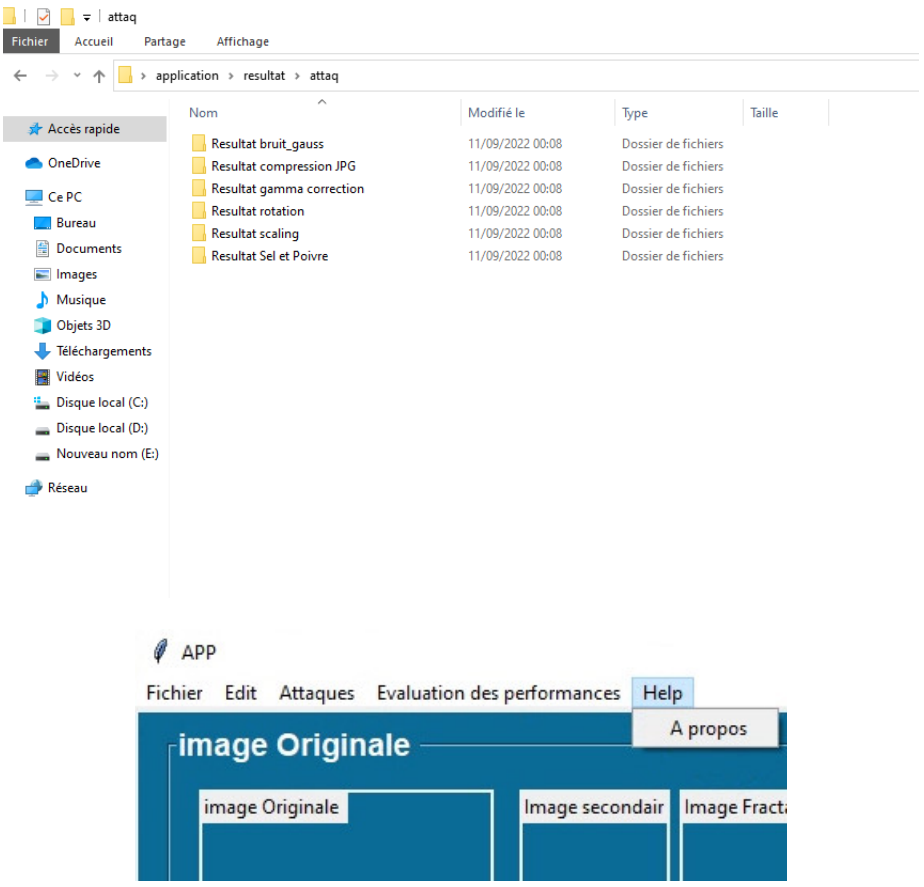


FIGURE 4.6 – Fenêtre pour la discrimination

Pour tous les boutons de la figure 4.5 tel que (Sel et poivre, rotation, correction gamma, scaling, bruit gaussien et compression) et les boutons de la deuxième figure 4.6 tel que discrimination, nous obtenons cette figure pour sélectionner les fichiers des résultats que nous voulons voir et elle changera d'un bottons à un autre.



5. **Help** : pour donner une idée générale sur le but d'application et tous les paramètres utilisés sur les attaques. La figure 4.7 montre cette fenêtre de "Help".

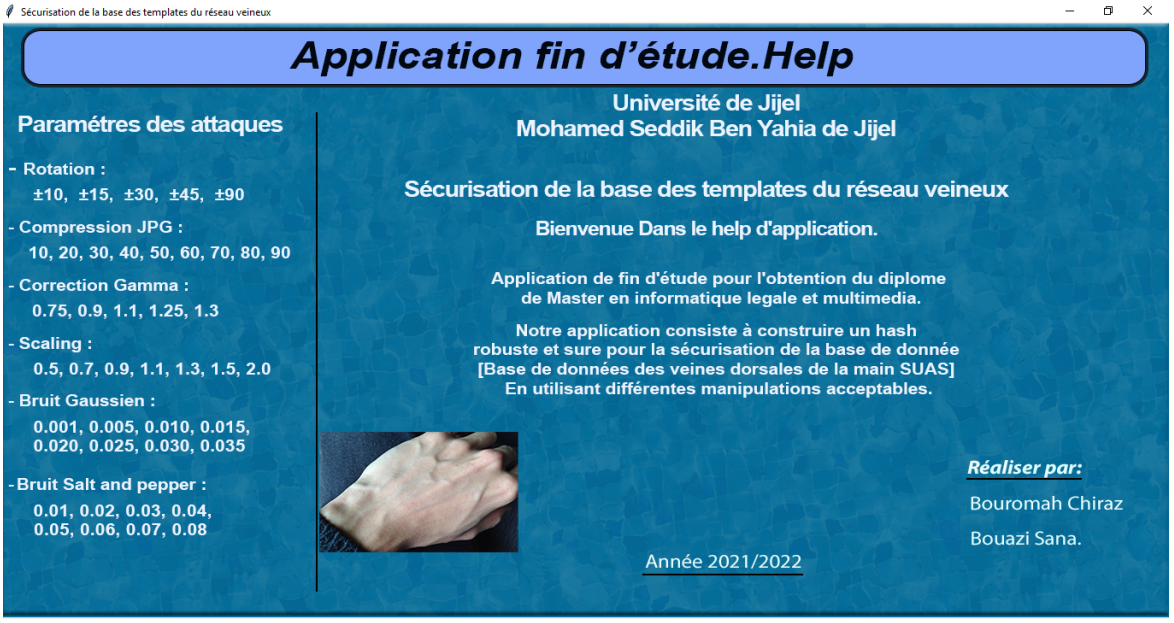


FIGURE 4.7 – help de l'application

L'interface graphique de la figure 4.8 est composée de trois parties (image originale, image attaquée, et mesure de similarité). Pour les deux premières étapes nous avons les

boutons suivants :
Bouton lissage : il sera actif que lorsque l'image originale est affichée pour la partie d'image originale, et lorsque l'attaque est choisi pour la partie d'image attaquée, il suffit nous cliquons sur ce bouton pour obtenir l'*image prétraité*
Bouton partition en anneau : il sera actif que lorsque l'image prétraitée est affichée, il suffit nous cliquons sur ce bouton pour obtenir d'*image partition en anneau.*
Bouton image secondaire : il sera actif que lorsque l'image partition en anneau est affichée, il suffit nous cliquons sur ce bouton pour obtenir *image secondaire.*
Bouton fractale : il sera actif que lorsque l'image secondaire est affichée, il suffit nous cliquons sur ce bouton pour obtenir *image de codage fractale.*
Bouton hash finale : il sera actif que lorsque le codage fractale est faite, il suffit nous cliquons sur ce bouton pour obtenir le hach final.
La figure suivante montre les résultats obtenus de chaque bouton.

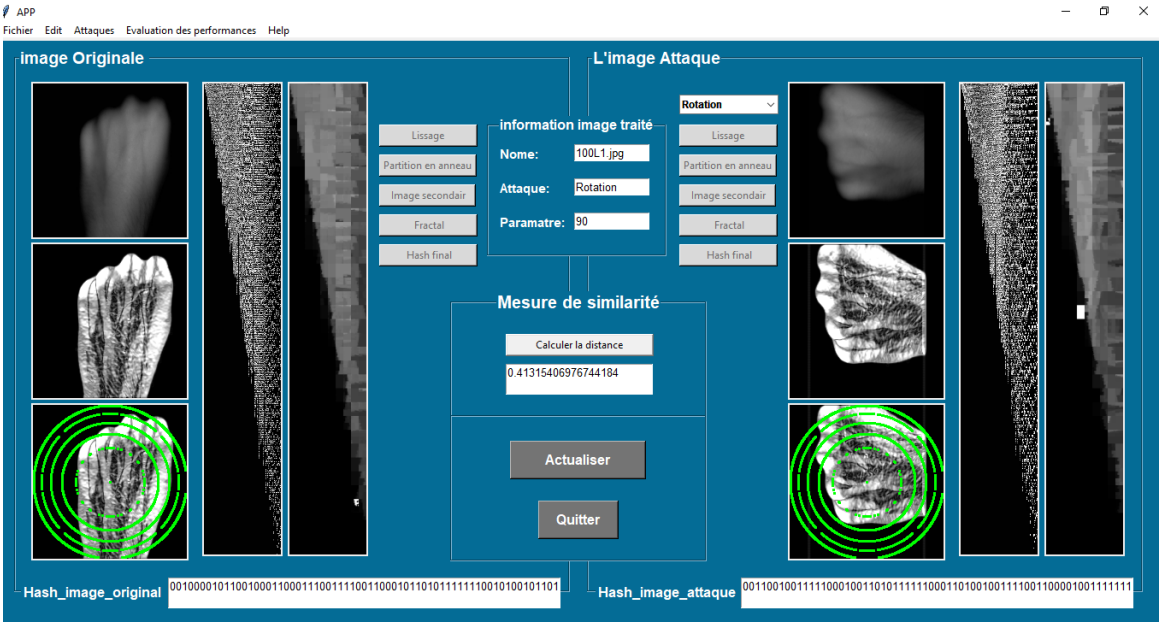


FIGURE 4.8 – Les résultats obtenus de la partie d'image originale et la partie d'image attaquée.

Pour chaque image originale et pour chaque image attaquée, les résultats obtenus doit être dans un dossier chaque image s'appelle par le nom d'image et l'attaque choisi pour l'image attaquée.

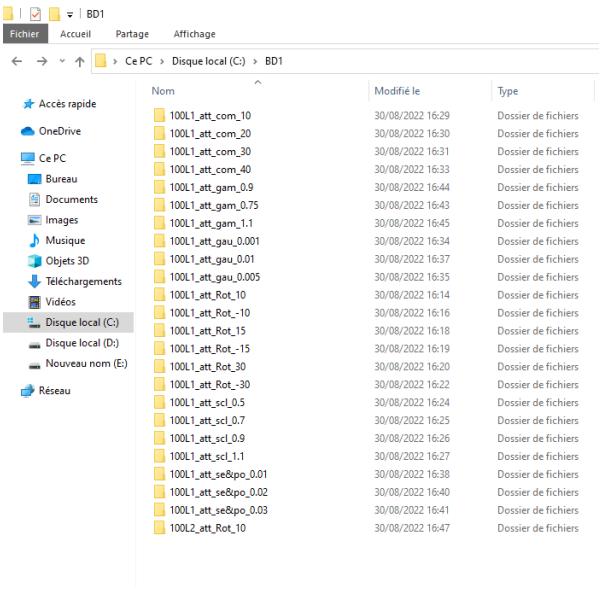
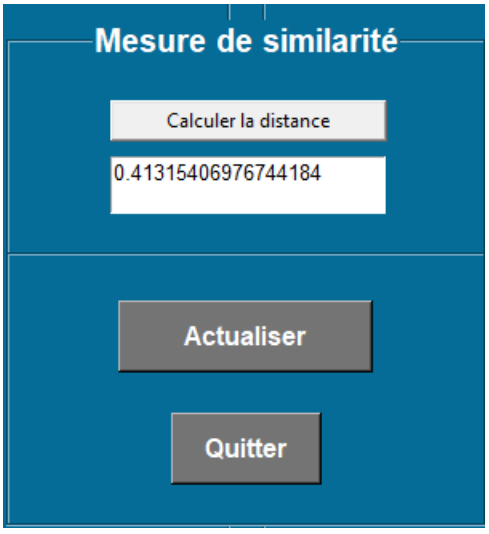


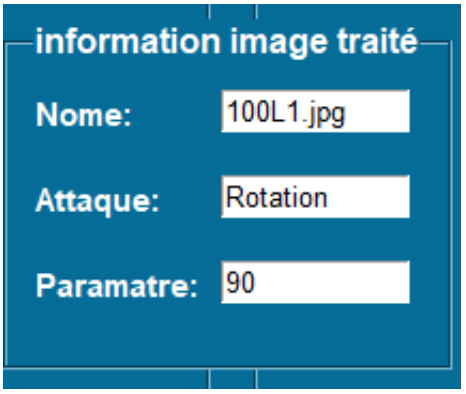
FIGURE 4.9 – Les résultats sauvegardés pendant le traitement

La mesure de similarité pour calculer la distance(en utilisant distance de hamming) entre la signature d'image originale et la signature attaque. Pour la partie de la mesure de similarité dans l'interface nous avons :

- Le bouton calculer : il sera actif que lorsque les deux haches sont affichés, il suffit de cliquer sur ce bouton pour obtenir la mesure de similarité a entre les deux images.



- Dans cette partie, nous affichons les informations sur l'image qui sera traité



4.4 Analyse et interprétation des résultats

4.4.1 Robustesse perceptuelle

Pour évaluer la robustesse de notre méthode contre les différentes manipulations acceptables et qui sont : la rotation, compression, scaling, correction gamma, bruit gaussien et bruit sel et poivre, nous calculons la mesure de similarité entre le hach de chaque image d'empreinte originale et son hach attaqué, pour chaque paramètre d'attaque, ensuite nous calculons la moyenne de cette mesure. Notre teste est appliqué sur 48 images choisies de la base de données des veines dorsales de la main SUAS.

Les résultats obtenus pour chaque attaque sont illustré dans les tableaux suivants :

Rotation :

Attaque	Paramètre	Mesure de similarité
Rotation	-10	0.418
	+10	0.421
	-15	0.41
	+15	0.415
	-30	0.413
	+30	0.407
	-45	0.412
	+45	0.425
	-90	0.418
	+90	0.416

TABLE 4.3 – La moyenne de similarité pour chaque paramètre de la rotation

Compression JPEG :

Attaque	Paramètre	Mesure de similarité
Compression JPEG	10	0.405
	20	0.389
	30	0.399
	40	0.398
	50	0.404
	60	0.386
	70	0.384
	80	0.394
	90	0.375

TABLE 4.4 – La moyenne de similarité pour chaque paramètre de la compression

Correction Gamma :

Attaque	Paramètre	Mesure de similarité
Correction gamma	0.75	0.381
	0.9	0.363
	1.1	0.365
	1.25	0.379
	1.3	0.387

TABLE 4.5 – La moyenne de similarité pour chaque paramètre de la correction gamma

Scaling :

Attaque	Paramètre	Mesure de similarité
Scaling	0.5	0.499
	0.7	0.505
	0.9	0.494
	1.1	0.498
	1.3	0.497
	1.5	0.496
	2.0	0.501

TABLE 4.6 – La moyenne de similarité pour chaque paramètre de scaling

Bruit Quassien :

Attaque	Paramètre	Mesure de similarité
Bruit Qaussien	0.001	0.374
	0.005	0.38
	0.01	0.387
	0.015	0.38
	0.02	0.378
	0.025	0.37
	0.03	0.379
	0.035	0.38

TABLE 4.7 – La moyenne de similarité pour chaque paramètre de Bruit Qaussien

Bruit Sel et poivre :

Attaque	Paramètre	Mesure de similarité
Bruit sel et poivre	0.01	0.427
	0.02	0.447
	0.03	0.441
	0.05	0.434
	0.06	0.446
	0.07	0.444
	0.08	0.437

TABLE 4.8 – La moyenne de similarité pour chaque paramètre de Bruit Sel et poivre

La moyenne de mesure de similarité générale pour toutes les attaques sont illustré dans le tableau suivant :

Attaque	Moyenne globale de similarité
rotation	0.415
Compression JPEG	0.393
Correction gamma	0.375
scaling	0.5
Bruit Gaussien	0.379
Bruit sel et poivre	0.44

TABLE 4.9 – La moyenne générale de la mesure de similarité pour chaque attaque

Nous avons utilisé la courbe ROC (receiver operating characteristics) pour représenter la robustesse de notre application contre les attaques cités auparavant. Les résultats sont présentés dans la figure 4.10

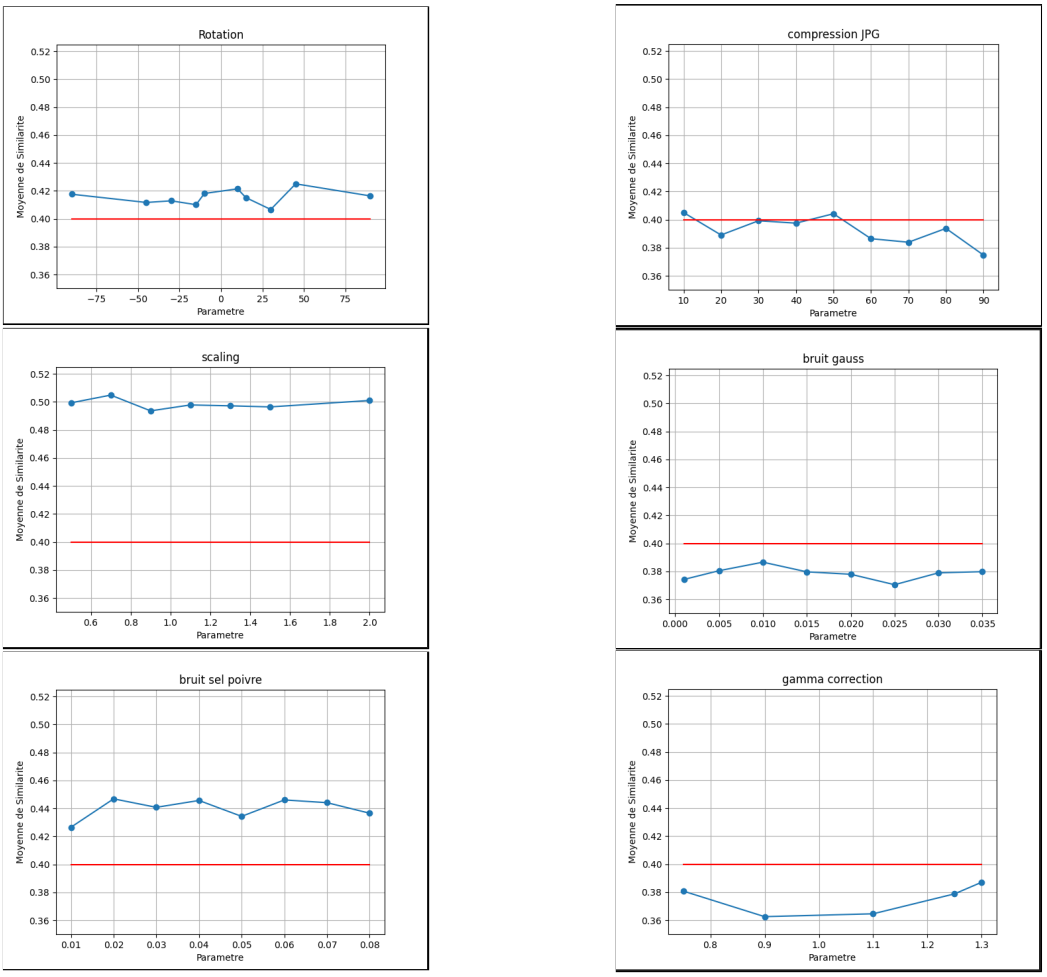


FIGURE 4.10 – Evaluation de la robustesse par les différentes manipulations acceptables

D'après les courbes ROC de la figure 4.10 et le tableau 4.9, nous remarquons que les valeurs de moyenne de distance entre le hache original et celui attaqué sont inférieures au seuil 0.4 pour la majorité des attaques sauf pour les attaques de rotation, scaling et bruit sel et poivre car ces dernières induisent des changements visuels notables sur les images des veines et, par conséquent, des variabilités importantes dans les codes de hachage. Ce qui implique que notre méthode donne une bonne robustesse contre les distorsions acceptables : compression JPEG, correction gamma et bruit Gaussien. .

4.4.2 Capacité de discrimination

Pour tester la capacité de discrimination de notre méthode, nous calculons la mesure de similarité entre chaque image originale de réseau veineux de main et les autres images originaux qui n'appartient pas aux réseaux veineux de la même personne. Les résultats sont représentés dans l'histogramme 4.11 ou l'axe des X sont les valeurs de la distance de hamming et l'axe des Y la fréquence d'apparition de chaque valeur .

La méthode proposée est appliquée à 48 images différentes des réseaux veineux de main de la même base de données, et la distance entre chaque paire de hach est calculée pour obtenir 2160 résultats, comme indiqué dans la Figure 4.11.

Les résultats obtenus sont illustré dans l'histogramme suivant :

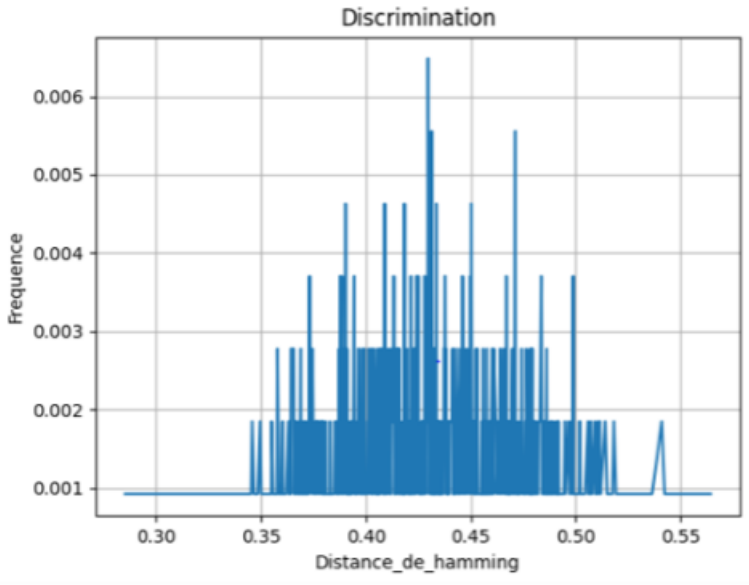


FIGURE 4.11 – Évaluation de discrimination

Si nous choisissons le seuil 0.4, le taux d'erreur calculé pour les images différentes et qui sont fausses identifiées comme des images similaire égale à 0.27. Si nous voulons maintenir une bonne discrimnation de notre méthode de hachage perceptuel, ce taux d'erreur est acceptable. Cela implique que dans notre méthode de hachage il ya un compris entre la robustesse et la discrimantion.

4.5 Conclusion

D'après les différents tests effectués sur notre système pour évaluer ses performances à travers les différentes attaques appliquées sur les images des réseaux veineux de main nous avons arrivé à conclure que notre méthode est robuste contre quelques attaques acceptables et elle est discriminante contre les différentes qui n'appartiennent pas à la même personne. Egalement nous concluons que notre méthode donne des bons résultats pour le seuil 0.4, dans le cas de correction Gamma et bruit Gaussien et dans le cas de la compression JPEG20, 60, 70, 80, 90. Ce qui implique que la robustesse de cet algorithme est bonne envers ces trois attaques. Les résultats expérimentaux ont montré que notre hachage est robuste contre les manipulations qui préservent le contenu normal.

CONCLUSION GÉNÉRALE ET PERSPECTIVE

Les fonctions de hachage perceptuel sont fortement inspirées des fonctions de hachage cryptographique, comme ces dernières sont très sensibles au contenu binaire des données à hacher, les fonctions de hachage perceptuel sont proposées comme une solution alternative pour une principale application aux données multimédias et spécialement aux images. Une image numérique tel que l'image de réseau veineux de main peut subir différentes formes de transformations ou de manipulations qui peuvent affecter son l'intégrité. Certaines applications peuvent avoir besoin d'appliquer certaines manipulations acceptables afin d'améliorer la qualité de l'image originale telles que la compression, le filtrage ou même d'effectuer d'autres opérations permettant l'amélioration de l'image en question. Certaines applications peuvent également nécessiter une compression avec pertes pour satisfaire les contraintes de ressources sur la bande passante ou d'espace de stockage.

Par conséquent, pour authentifier une image de réseau veineux de main, il faut tolérer des manipulations acceptables que pourrait subir une image. Les fonctions de hachage perceptuel sont des solutions potentielles dans ces cas-là permettant d'établir une « correspondance perceptuelle » entre l'image originale et l'image d'authentifier.

Le projet réalisé dans ce mémoire a été de proposer une méthode de hachage perceptuel basée à fractale et partitions en anneau sur les images des réseaux veineux de main pour garantir leur sécurité contre les différentes modifications acceptable comme correction gamma, compression et bruit gaussien

Nous avons vu les étapes utilisées dans le système de reconnaissance biométrique de réseaux veineux de main pour l'identification et l'authentification.

Ensuite, notre travail a essentiellement consisté par la description de différentes techniques de protection des images parmi eux la technique de hachage perceptuel.

Après ça, nous avons donné une description détaillée de notre méthode de hachage perceptuel de réseau veineux de main d'image, basé sur le codage fractal d'image et de la partition en anneau, nous avons expliqué ses différentes phases.

Enfin, nous avons essayé d'illustrer l'importance de cette technique en montrant les résultats de de hachage afin de construire un hach qui permet d'assuré la robustesse et la discrimination de ces images

Nous conclut que notre méthode appliqué dans ce mémoire est robuste et sur contre les attaques acceptables la compression, bruit gaussien et correction gamma A l'issue de ce travail, les perspectives suivantes peuvent être proposées pour poursuivre les recherches dans ce domaine :

- Mettez d'autres types d'attaques pour analyser les signatures perceptuelles.
- Étude de la robustesse à d'autres types de caractéristiques extraites..

Annexe

Algorithme 1

Algorithm 1: Pseudocode for generating perceptual hash using fractal coding

Input: Secondary image in extended fractal domain block $\bar{S}_i$,
Output: Generated hash H'.

1. Initialize: n =5

Stage 1: Search for matching elements in $\bar{S}_i$.

// n = number of rings

2. **for** $i = 1$ *to* n **do**
3. **if** $\bar{S}_i = (\hat{s}_1, \hat{s}_2, \ldots \hat{s}_n)$, **then search**
4. $M_i = \Omega(\bar{S}_i)$ //find the matching elements in every ring in $\bar{S}_i$
5. **break loop**
6. **end if**
7. **end for**

Stage 2: Get the best match between $\bar{S}_i$ and $\bar{S}_i$ in M.

8. Initialize: $\frac{\bar{S}_i}{M} = j_m$ and $\frac{\bar{S}_i}{M} = k_m$;
9. **for** $i = 1$ *to* n **do**
10. **if** $M(\bar{S}_i, \bar{S}_i) \neq 0$ **then**
$M(\bar{S}_i, \bar{S}_i)^2 = \min_p \|p(\bar{S}_i - j_m I) - (\bar{S}_i - k_m I)\|^2$;
11. **end if**
12. **end for**

Stage 3: Achieve compactness

13. // compute the least square of the optimal square parameter p.
14. // reduce feature transformation and range block to the minimal.
15. **for** n ≤ 1 *to* I **do**
16. **for** $p_n \leq I$ **do**
17. $p_n \approx p = \frac{\bar{S}_i(\bar{s}_1, \bar{s}_2, \ldots \bar{s}_n) - j_m}{\bar{S}_i(\bar{s}_1, \bar{s}_2, \ldots \bar{s}_n) - k_m}$;
18. **end for**
19. **end for**
20. // concatenate the perceptual features p_n, I *and* M to generate a binary sequence $H = [p_n. I. M]$.
21. // scramble H using k for enhanced security and obtain the final hash H'

Output H'

Fonction reduce

```
def reduce(img, factor):
    result = np.zeros((img.shape[0] // factor, img.shape[1] // factor))
    for i in range(result.shape[0]):
        for j in range(result.shape[1]):
            result[i,j] = np.mean(img[i*factor:(i+1)*factor,j*factor:(j+1)*fact
or])
    return result
```

Fonction rotate

```
def rotate(img, angle):
    return ndimage.rotate(img, angle, reshape=False)
```

Fonction flip

```
def flip(img, direction):
    return img[::direction,:]
```

Fonction apply_transformation

```
def apply_transformation(img, direction, angle, contrast=1.0, brightness=0.0):
    return contrast*rotate(flip(img, direction), angle) + brightness
```

Fonction generate_all_transformed_blocks

```
def generate_all_transformed_blocks(img, source_size, destination_size, step):
    factor = source_size // destination_size
    transformed_blocks = []
    for k in range((img.shape[0] - source_size) // step + 1):
        for l in range((img.shape[1] - source_size) // step + 1):
            # Extract the source block and reduce it to the shape of a destinat
ion block
            S = reduce(img[k*step:k*step+source_size,l*step:l*step+source_siz
e], factor)
            # Generate all possible transformed blocks
            for direction, angle in candidates:
                transformed_blocks.append((k, l, direction, angle, apply_transf
orm(S, direction, angle)))
    return transformed_blocks
```

Fonction compress

```
def compress(img, source_size, destination_size, step):
    transformations = []
    transformed_blocks = generate_all_transformed_blocks(img, source_size, dest
ination_size, step)
    for i in range(img.shape[0] // destination_size):
        transformations.append([])
        for j in range(img.shape[1] // destination_size):
            print(i, j)
            transformations[i].append(None)
            min_d = float('inf')
            # Extract the destination block
            D = img[i*destination_size:(i+1)*destination_size,j*destination_siz
e:(j+1)*destination_size]
            # Test all possible transformations and take the best one
            for k, l, direction, angle, S in transformed_blocks:
                contrast, brightness = find_contrast_and_brightness2(D, S)
                S = contrast*S + brightness
                d = np.sum(np.square(D - S))
                if d < min_d:
                    min_d = d
                    transformations[i][j] = (k, l, direction, angle, contrast,
brightness)
    return transformations
```

Fonction find_contrast_and_brightness2

```
def find_contrast_and_brightness2(D, S):
    # Fit the contrast and the brightness
    A = np.concatenate((np.ones((S.size, 1)), np.reshape(S, (S.size, 1))), axis
=1)
    b = np.reshape(D, (D.size,))
    x, _, _, _ = np.linalg.lstsq(A, b)
    return x[1], x[0]
```

Fonction decompress

```
def decompress(transformations, source_size, destination_size, step, nb_iter=
8):
    factor = source_size // destination_size
    height = len(transformations) * destination_size
    width = len(transformations[0]) * destination_size
    iterations = [np.random.randint(0, 256, (height, width))]
    cur_img = np.zeros((height, width))
    for i_iter in range(nb_iter):
        print(i_iter)
        for i in range(len(transformations)):
            for j in range(len(transformations[i])):
                # Apply transform
                k, l, flip, angle, contrast, brightness = transformations[i][j]
                S = reduce(iterations[-1][k*step:k*step+source_size,l*step:l*st
ep+source_size], factor)
                D = apply_transformation(S, flip, angle, contrast, brightness)
                cur_img[i*destination_size:(i+1)*destination_size,j*destination
_size:(j+1)*destination_size] = D
        iterations.append(cur_img)
        cur_img = np.zeros((height, width))
    return iterations
```

BIBLIOGRAPHIE

[1] **Benchennane Ibtissam**, *"Etude et mise au point d'un procédé biométrique multimodale pour la reconnaissance des individus"*, Thèse de Doctorat, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, 2015.

[2] **B.Fatima**, *"Caractéristiques Biométrique pour l'identification"*, Mémoire de Magister, Université Ahmed ben Bella, Oran, 2015

[3] **L.ALLANO**, *"La Biométrie multimodale : stratégies de fusion de scores et mesures de dépendance appliquées aux bases de personnes virtuelles"*, Thèse de doctorat, Université D'Every Val D'Essonne, 2009

[4] **KHELLAT-KIHEL Badra**, *"Sécurisation des réseaux WIFI par authentification biométrique par empreinte digitale"*, Mémoire d'ingéniorat en informatique, Juin 2009.

[5] **KHELLAT-KIHEL Souad**, *"Reconnaissance des individus par leurs réseaux veineux"*, Mémoire de master en informatique, Juin 2012.

[6] **Souilla Benkhaira**, *"Systèmes multimodaux pour l'identification et l'authentification biométrique"*, Mémoire de Magister en Informatique, Université 20 Août 1955-Skikda, 2014

[7] **C. L. Tisse**, **L. Torres**, **L. Martin** et **M. Robert**, *"Systèmes biométriques pour la vérification d'individu, Un exemple : l'iris"*, Center for Autonomous System, University of Sydney, The Rose Street Building J04, NSW 2006, Australia

[8] **Berredjem Achref**, *"La reconnaissance des individus par leur empreinte des articulations des doigts"*, Mémoire de Magister, Universite 8 Mai 1945, Guelma, 2019

[9] **Benabdi Mouad**, *"Identification des personnes par les empreintes d'articulation des doigts et le deep learning"*, Thèse de doctorat, Universite mohamed boudiaf, M'sila, 2019

[10] **EL-ABED Mohamad**, *"Evaluation de système biométrique Cryptographie et sécurité"*, Thèse de doctorat Université de Caen, 2011. Français

[11] **F. Perronnin** et **J.-L. Dugelay**, *"Introduction à la biométrie authentification des individus par traitement audio-vidéo"*, Traitement du signal, vol.19, no. 4, 2002.

[12] **Khellat-Kihel Souad**, *"Identification biométrique par fusion multimodale de l'empreinte d'articulation, l'empreinte digitale et l'empreinte veineuse du doigt"*, Thèse de doctorat, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, 2017.

[13] **Abdelhafid Oualid** et **Senouci Abdelkrim**, *"L'identification Biométrique Par Les Veines Des Doigts"*, Mémoire de Magister, Université Mohamed Boudiaf, M'sila, 2020

[14] **Belghechi Rima**, *"Contribution à la reconnaissance d'empreintes digitales par une approche hybride"*, Mémoire de master, Institut National de formation en Informatique (I.N.I), 2006

[15] **Nicolas Morizet**, *"Reconnaissance biométrique par fusion multimodale du visage et de l'iris"*, Thèse de doctorat, Télécom Paris Tech, 3 2009.

[16] **M. Mohamed El Abed**, *"Evaluation de systèmes biométrique"*, Thèse de Doctorat, Université de Caen Basse-Normandie France, 2006

[17] **DIB Fouad**, *"Identification des personnes par le réseau veineux de la main"*, Université Ferhat Abbas, Sétif, 2013

[18] **L. Wang** and **G. Leedham**, *"Near- and Far- Infrared Imaging for Vein Pattern Biometrics"*, Proceedings of the IEEE International Conference on Video and Signal, pp. 52-59, 2006

[19] **A. M. Badawi**, *"Hand Vein Biometric Verification Prototype : A Testing Performance and Patterns Similarity"*, IPCV, pp. 3-9, 2006

[20] **L. Wang**, **G. Leedham**, and **D. S.-Y. Cho**, *"Minutiae feature analysis for infrared hand vein pattern biometrics"*, Pattern Recognition, vol. 41, pp. 920-929, 2008.

[21] **L. Wang** and **G. Leedham**, *"A thermal hand vein pattern verification system"*, Lecture Notes in Computer Science, pp. 58-65, 2005.

[22] **F. Zernike**, *"Diffraction theory of the cutprocedure and its improved form, the phase contrast method"*, , Physical,pp.689-704, 1934.

[23] **S. Katzenbeisser** and **F. A. Petitcolas**, *"Hiding Techniques for Steganography and Digital Watermarking"*, , ArtechHouse, Inc, Norwood, MA, USA, lst édition, IS8Ni580530354, 2000.

[24] **Vincent Martin**, *"Contribution Des FiltresLptu Et Des Techniques D'interpolation Au Tatouage Numérique"*, 2006.

[25] **Yoo C.D**, **Lee S** and **T.Kalker**, *"Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform"*, IEEE transaction on Information Forensics and Security, Vol.2, No.3.pp.321-330, Septembre 2007.

[26] **C. Paul van Oorschot**, **Vanstone.A. Scott Menezes J. Alfred**, *"Handbook of applied cryptography"*, Boca Raton. CRC Press, 1998.

[27] **Brahim Ait Es Said Azhar Hadmi**, **Witliam Puech** and **Abdellah Ait Ouahman**. *"Perceptual Image Hashing, Watermarking"*, Volume 2. Dr, Mithun Das Gupta (Ed.), ISBN : 97 & 953-51-0619-7, InTech, DOI : 10.5772137435, 2012.

[28] **Azhar HADMIA**, *"protection des données visuelles Analyse des fonctions de hachage perceptuel"*, Laboratoire d'informatique de robotique et de Microélectronique de Montpellier, 26 octobre 2012.

[29] **V. Monga** , *"Perceptually Based Methods for Robust Image Hashing"*, , Phd dissertation. University of Texas at Austin, 2005.

[30] **Yong Wang**, *"New Way to Construct Cryptographic Hash Function"*, 16 February 2014.

[31] **Yinian Mao Swaminathan**, **Ashwin** and **Min Wu**, *"Robust and Secure Image Hashing"*, IEEE Transactions on Information Forensics and Security, June 2006.

[32] **Viktor Popkov**, *"Robust Image Hashing Using Image Normalization and SVD Decomposition"*, Department of Computer Engineering, IAG70LT, 132458IAPM, 2015.

[33] **AIBECHE NARIMANE** and **AMIAR YOUCEF**, *"Hachage robuste à base d'entropie d'images d'empreintes digitales"*, Mémoire de fin d'études pour l'obtention du diplôme master , université de Jijel, 2019.

[34] **Arambam Neelima** and **Kh Manglem Singh**, *" Perceptual Hash Function based on Scale-Invariant Feature Transform and Singular Value Decomposition"*, Département of Computer Science, Engineering, NIT Manipur, Imphal, India ,2016.

[35] **Zhenjun Tang**, *"Robust Image Hashing with Low-Rank Representation and Ring Partition"*, Wireless Communications and Mobile Computing, 2020.

[36] **Z. Tang**, **X. Li**, **J. Song**, **M. Wei**, and **X. Q. Zhang**, *"Color space selection in image hashing : an experimental study"*, IETE Technical Review, vol. 34, no. 4, pp. 440–447, 2016.

[37] **X. Hou** and **L. Zhang**, *"Saliency detection : a spectral residual approach"*, in 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, Minneapolis, MN, USA, 2007.

[38] **Q. Wang**, **X. He**, and **X. Li**, *"Locality and structure regularized low rank representation for hyperspectral image classification"*, IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 2, pp. 911–923, 2019.

[39] **Z. Tang**, **X. Q. Zhang**, **L. Huang**, and **Y. Dai**, *"Robust image hashing using ring-based entropies"*, Signal Processing, vol. 93, no. 7, pp. 2061–2069, 2013.

[40] **JUNLIN OUYANG**, **XIAO ZHANG 1** and **XINGZI WEN**, *"Robust Hashing Based on Quaternion Gyrator Transform for Image Authentication"*, , Digital Object Identifier 10.1109/ACCESS.2020.3043111, 2020.

[41] **S. Zhu-Hong**, *"A study on quaternion transforms for color image processing"*, Ph.D. dissertation, School Computer. Sci. Eng., Southeast Univ., Dhaka, Bangladesh, 2015.

[42] **R. H.**, **RICHHARIYA Bhanu Bhai Ram Kumar**, **LASKAR**, *"Robust and secure hashing using Gabor filter and markov absorption probability in : Communication and signal processing (iccsp)"*, 2016.

[43] **Fares Khelaifi** and **HongJie He**, *"Perceptual image hashing based on structural fractal features of image coding and ring partition, Multimedia Tools and Applications"*, 2020.

[44] **Barnsley M. F** and **Sloan A. D**, *"A better way to compress images"*, BYTE magazine, 1988, pp. 215-223.

[45] **Barnsley M. F**, *"Fractal every where"*, New-york : Academic Press, California, 1988.

[46] **Jacquin A. E**, *"A fractal theory of iterated Markov operators on spaces of measures with applications to digital image coding"*, PhD Thesis, Georgia Institute of Technology, 1989.

[47] **Pierre Vigier**, *"fractal image compression"*, 14 may 2018, https ://pvigier.github.io/2018/05/14/fractal-image-compression.html consulté le 10 aout 2022

[48] **Yachun Feng**, **Hong Zhang Hao**, **Chen Ding Yuan** and **Helong Wang**, *"Visual tracking via multi-experts combined with average hash model"*, 2015