République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Seddik Ben Yahia de Jijel
Faculté des Sciences Exactes et informatique
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme Master de Recherche en Informatique

Option : ***Informatique légale et multimédia***

Thème

Hachage robuste à base d'entropie d'images d'empreintes digitales

Présenté par :
Aibeche Narimane.
Amiar Youcef.

Encadré par :
M^{elle}.Birouk Wafa

Promotion : 2019.

⁂ Remerciements ⁂

⁂ Nous tenons à remercier en premier dieu qui nous a donné la force, la volonté, et le courage pour réaliser ce modeste travail.

⁂ Nous remercions en particulier notre encadreur M^{elle}. Birouk Wafa pour la confiance qu'il nous a accordé en acceptant de diriger notre travail de mastère. C'est grâce à son aide inestimable, aux conseils précieux qu'il n'a jamais cessé de nous donner que ce manuscrit a pu voir le jour. Nous vous serons toujours reconnaissants.

⁂ Nos remerciements s'adressent aussi aux membres du jury pour l'honneur qu'ils nous a fait en acceptant de juger et d'examiner notre travail, et à l'ensemble des enseignants et enseignantes de l'université de Jijel.

⁂ Et en fin, un très grand merci à tous ceux qui, de près ou de loin ont contribué par leurs conseils, leurs encouragements et leur assistance jusqu'à l'aboutissement de ce travail.

⁂ Dédicaces ⁂

♡♡♡ *Je dédie ce modeste travail* ♡♡♡

⁂ *A l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon cher Papa.*

♡♡♡♡

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

♡♡♡♡

⁂ *A mes adorable sœures Imene et Sara, mon cher frère Imad.*

♡♡♡♡

⁂ *A mes petits neveux Abderrahmen et Haythéme.*

♡♡♡♡

⁂ *A ma tante Fatiha.O et toute la famille Aibeche et Oubadi.*

♡♡♡♡

⁂ *A tous mes chère amies Amina.A Nihed.A et Yousra.A*

♡♡♡♡

⁂ *A mes belles Mimi.B et Sara.E*

♡♡♡♡

⁂ *A mon binôme Yousef.A*

♡♡♡♡

⁂ *A tous mes collègues de promotion 2019*

♡♡♡♡

⁂ *A tous ceux que j'aime je dédie ce travail.*

Narimane

⁕ Dédicaces ⁕

♡♡♡ *Je dédie ce modeste travail* ♡♡♡

⁕ *A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.*

♡♡♡♡

⁕ *A mes chères sœur Adila et Manel pour leurs encouragements permanents, et leur soutien moral.*

♡♡♡♡

⁕ *A mes chers frères Lamin, Messaoud, Hicham et Islam pour leur appui et leur encouragement.*

♡♡♡♡

⁕ *A toute ma famille pour leur soutien tout au long de mon parcours universitaire.*

♡♡♡♡

⁕ *A tous mes chère amies Salim.A, Salah.F, Mounir.B, Jilali.B, Said.H, Hossem.k et Salim.k*

♡♡♡♡

⁕ *A mon binôme Narimane.A*

♡♡♡♡

⁕ *A tous mes collègues de promotion 2019.*

♡♡♡♡

⁕ *A tous ceux que j'aime je dédie ce travail.*

Yousef

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

PIN Personal Identification Number

ADN Acide désoxyribo Nucléique

TFR Taux Faux Rejet

FRR False Reject Rate

FR False Reject

TFA Taux Fausse Acceptatio

FAR False Accepte Rate

FA False Accepte

TEE Taux Erreur Egal

EER Equal Error Rate

ROC Receveur Operating Caractéristique

TPR True Positive Rate

FPR False Positive Rate

CN Crossing Number

JPEG Joint Photographic Experts Group

DCT Discrete Cosine Transform

DWT Discrete Wavelet Transform

SHA Secure Hash Algorithm

HVS Human Visual System

BER Binary Error Rate

DE Distance Euclidien

DH Distance de Hamming

SVD Singular Value Decomposition

SIFT Scale-Invariant Feature Transform

FMT Fourier-Mellin Transform

Introduction générale

De nos jours, les nouvelles avancées technologiques ont fait fortement évoluer nos méthodes de communication et d'échange d'information. Cette évolution permis d'échanger facilement et rapidement l'information sous toutes ses formes : textuelle, sonore et visuelle sur des réseaux publics de plus en plus larges. L'importance des enjeux, motive les fraudeurs à mettre en échec les systèmes de sécurité existants.

Il serait particulièrement dramatique de ne pas disposer en moment voulu d'outils performants adaptés aux nouvelles menaces. La vérification de l'intégrité et l'authenticité des images numériques s'impose comme étape incontournable vu la qualité importante d'information visuelles qu'elle véhicule. En effet, ces données sont faciles à pirater, à modifier et à rediffuser sans aucune perte de qualité perceptible. La protection est une nécessité incontournable si nous souhaitons assurer la qualité des services offerts. Dans le domaine de sécurité multimédia, un type d'approche très populaire a été proposée pour répondre à ces exigences ces dernières années, il s'agit ici du hachage perceptuel.

Dans ce mémoire nous nous intéressons aux fonctions de hachage perceptuel pour la sécurité et le contrôle d'intégrité des images numériques d'empreinte digitale. 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 on veut 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. la dernière décision est basée sur un seuil choisi. Par conséquent, pour authentifier une image, il faut tolérer des manipulations acceptables que pourrait subir une image telles que la compression JPEG. En effet, ces manipulations préservent l'aspect visuel de l'image. En même temps, un système de hachage perceptuel

doit être suffisamment sensible pour détecter les manipulations malveillantes qui modifient l'interprétation du contenu sémantique de l'image. 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 est composé de quatre parties principales borduré par une introduction générale et une conclusion et perspectives. La première partie présente un tour d'horizon sur la biométrie, ainsi une vue détaillée sur le système de reconnaissance d'empreint digitale. Puis nous présentons les différentes techniques de protection des images qui sont abordées en deuxième partie de ce manuscrit. Nous présentons aussi l'étude des caractéristiques extraites d'une empreinte digitale dans un système de hachage perceptuel. Et nous terminons par la présentation de l'application réalisée et les résultats effectués. Plus précisément, ce mémoire est composé des chapitres suivants :

✓ **Le premier chapitre :** en introduisant la notion de la biométrie et les applications qui en découlent. Nous insistons plus particulièrement sur l'utilisation des empreintes digitales dans les systèmes d'identification des personnes.

✓ **Deuxième chapitre :** nous dressons 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.

✓ **Le troisième 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 percetuel d'image à base d'entropie qui sera appliquée dans notre travail et qui doit être robuste et sure.

✓ **Le quatrième chapitre :** est consacré essentiellement à la présentation de différentes interfaces de l'application réalisée, ainsi que les résultats des tests effectués sur cette application.

Enfin, nous clôturons ce mémoire par une conclusion générale pour faire le point sur l'ensemble des travaux effectués. Nous y présentons également différentes perspectives d'études et d'amélioration de notre approche.

Chapitre 1

Système de reconnaissance biométrique d'empreinte digitale

1.1 Introduction

La sécurité n'est pas une découverte récente. L'être humain a apprécié la nécessité en a développé des théories structurées et cela dés le début de son existence. Quelque soit le type d'organisation, le genre d'activité, le lieu géographique, l'état de développement du pays, les protocoles de la sécurité d'information sont devenues une nécessité incontournable.
Avec le développement rapide de la technologie une nouvelle technique de contrôle a vu le jour, c'est le système biométrique.

Dans ce chapitre nous commençons par présenter la biométrie de manière générale ainsi que les diverses applications qui en découlent, en insistant plus particulièrement sur l'utilisation des empreintes digitales. Puis nous détaillons les différentes étapes composant un système complet de reconnaissance d'empreinte.

1.2 Système de reconnaissance biométrique

1.2.1 Définition de la biométrie

La biométrie est un ensemble des technologies (appelées les technologies biométriques)qui exploitent des caractéristiques humaines physiques ou comportementales telles que l'empreinte digitale, la signature, l'iris, la voix, le visage, la démarche et la géométrie de la main pour différencier des personnes. Ces caractéristiques sont traitées par des ordres de processus automatisés [1],[2] à l'aide de dispositifs comme les modules de balayage ou les appareils-photo. À la différence des mots de passe, des PINs (Numéros d'Identification Personnelle), qui sont facilement oubliés ou exposés à l'utilisation frauduleuse, des clefs ou des cartes magnétiques qui doivent être portées par l'individu et sont faciles à être volées, copiées ou perdues, les caractéristiques biométriques sont uniques à chaque individu et il y a peu de possibilité que d'autres individus peuvent avoir ces caractéristiques[3]. Donc les technologies biométriques sont considérées comme les plus puissantes en termes de sécurité.

1.2.2 Système biométrique

Un système biométrique est essentiellement un système de reconnaissance des formes qui permet d'obtenir ces données biométriques d'un individu, d'extraire un ensemble de caractéristiques des données acquises et de comparer ces propriétés à la signature de la base de données. Il est utilisé pour vérifier l'identité d'une personne en utilisant une ou plusieurs méthodes spécifiées (voix, iris, empreinte digitale, visage) etc..

On peut dire que le système de contrôle biométrique est un système de mesure automatique [4] basé sur la reconnaissance des caractéristiques individuelles.

1.2.3 Les différentes techniques biométriques

Parmi les différentes techniques biométriques disponibles figurent trois catégories :

1.2.3.1 Analyse morphologique

empreintes digitales, iris, forme de la main, les traits du visage, et réseau veineux de la rétine.

1.2.3.2 Analyse des effets biologiques

ADN, sang, salive, urine, odeur, thermique.

1.2.3.3 Analyse comportementale

Reconnaissance de la parole, dynamique de frappe au clavier, la dynamique de signature, la manière de marche.

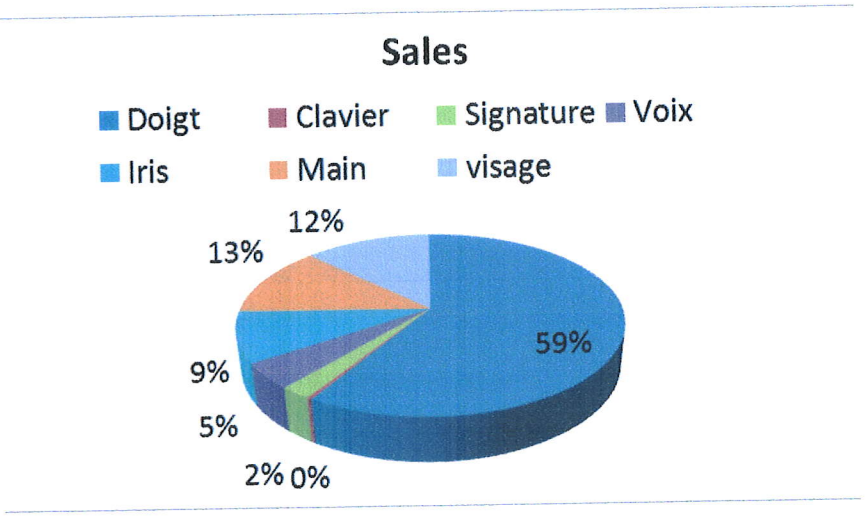


FIGURE 1.1 – Part de marché comparatif par technologie biométrie[3]

Entre les nombreuses méthodes d'identification biométrique utilisées, nous trouvons les empreintes digitales. La méthode la plus efficace avec une part de marché de 59%, nous y reviendrons plus en détail dans la section suivante. Cependant, d'autres moyens ont commencé à trouver leur place sur le marché biométrie (voir Figure 1-1 et Figure 1-2) :

la géométrie de la main : arrive en deuxième position avec 13% des part de marché, Jusqu'à 90 propriétés de la main sont mesurées (forme de la main et des articulations, longueur et largeur des doigts, longueurs communes inter articulations)[6]. Les taux erreurs de reconnaissance sont très élevés, en particulier pour les personnes appartenant à la même famille en raison de la forte similitude. En outre, la forme de la main change beaucoup avec l'âge.

la forme du visage : Plusieurs parties du visage (joues, yeux, nez, bouche) sont extraites d'une photo ou d'une vidéo et analysées géométriquement [7] (distances entre différents points, positions, formes). Le problème de cette méthode vient des possibles perturbations pouvant transformer le visage [8] (maquillage, Basse lumière, barbe ou lunettes, expressions faciales inhabituelles, changement avec l'âge.)

L'iris : est une technique extrêmement fiable car l'iris contient une infinité des points caractéristiques (ensemble fractionnaire), mais la fraude est possible avec des lentilles. L'acquisition de l'iris est effectuée par une caméra pour compenser les mouvements inéluctable de la part de pupille.

C'est très sensible (précision et réflexion) et relativement désagréable pour l'utilisateur [9], car l'œil doit rester grand ouvert et il est éclairé par une source lumineuse pour assurer un contraste correct.

Reconnaissance vocale : les caractéristiques du timbre de la voix et de la prononciation sont analysées [10]. La qualité de l'enregistrement peut poser problème et il est possible de frauder avec un échantillon vocal pré enregistré.

La dynamique du tracé de la signature : Il s'agit d'une analyse comportementale où différents éléments (mesure de la vitesse, ordre d'écriture, pression exercée, accélérations)[11] sont mesurés lors de la signature. La falsification est possible en passant par une phase d'apprentissage, et elle peut varier selon le stress de l'utilisateur.

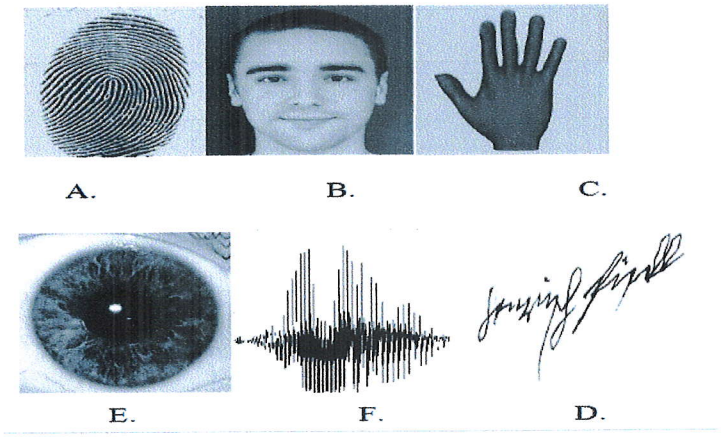


FIGURE 1.2 – Exemples de différentes caractéristiques biométriques : empreinte digitale (A), visage (B), main (C), iris (E), empreinte vocale (F), signature (D).

1.2.4 Modes de fonctionnement d'un Système biométrique

Tout système biométrique peut fonctionner en mode d'enrôlement ou en mode de vérification ou bien en mode d'identification.

1.2.4.1 Le mode d'enrôlement

C'est une phase d'apprentissage qui a pour but de recueillir des informations biométriques sur les personnes à identifier. Plusieurs campagnes d'acquisitions de données peuvent être réalisées afin d'assurer une certaine robustesse au système de reconnaissance aux variations temporelles des données [12]. Dans cette phase, les caractéristiques biométriques des individus sont saisies par un capteur biométrique, puis représentées sous forme numérique (signatures), et enfin stockées dans la base de données.

1.2.4.2 Le mode de vérification

Il s'agit d'une comparaison entre les données biométriques capturées (modèle de test) et les données stockées dans sa propre base de données (modèles d'apprentissage). Dans un tel système, un individu qui désire être identifié réclame une identité [12], habituellement par l'intermédiaire d'un PIN (numéro d'identification personnelle), d'un nom d'utilisateur, d'une carte d'identité, etc. Le système doit alors répondre à la question suivante "Suis-je réellement la personne que suis-je entrain de proclamer ?"

1.2.4.3 Le mode d'identification

Le système identifie l'individu en recherchant des signatures (Template) pour tous les utilisateurs de la base de données [13]. Par conséquent, le système conduit plusieurs comparaisons 1-à-N pour établir l'identité d'un individu. En résumé, un système biométrique opérant en mode identification répond à la question "Suis-je bien connu du système ?".

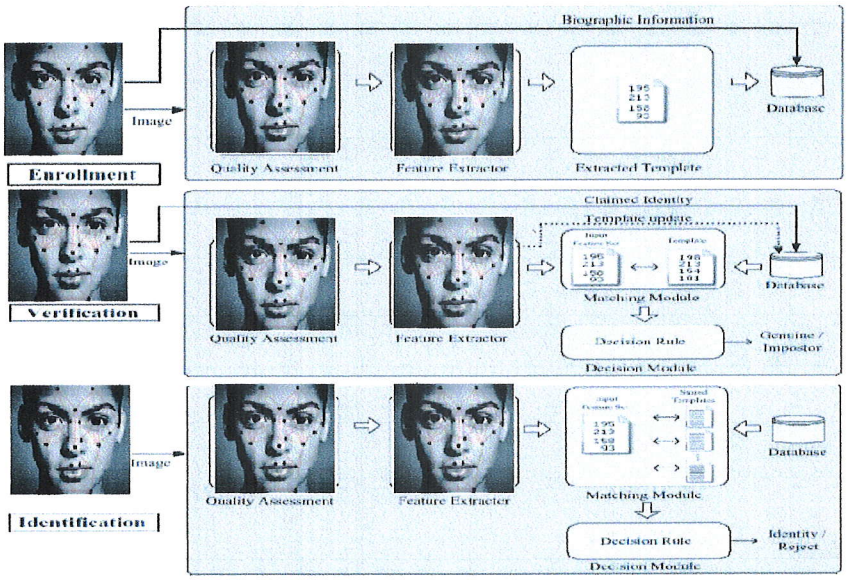


FIGURE 1.3 – Mode de fonctionnement d'un système biométrique.

1.2.5 Architecture d'un système biométrique

Un système biométrique typique peut être représenté par cinq modules comme le montre-la Figure 1-4 :

1.2.5.1 Le module de capture

Consiste à acquérir les données biométriques afin d'extraire une représentation numérique. Cette représentation est ensuite utilisée pour l'enrôlement, la vérification ou l'identification. Il s'agit d'un capteur biométrique qui peut être de type sans ou avec connexion.

1.2.5.2 Le module d'extraction des caractéristiques

Réduit la représentation numérique extraite afin d'optimiser la quantité de données à stocker lors de la phase d'enrôlement, ou pour faciliter le temps de traitement pendant la phase de vérification et l'identification. Ce module peut avoir un test de qualité pour contrôler les données biométriques acquises.

1.2.5.3 Le module du stockage

Contient des modèles biométriques pour les utilisateurs enrôlés du système.

1.2.5.4 Le module de similarité

Compare les données biométriques extraites par le module d'extraction des caractéristiques a un ou plusieurs modèles préalablement enregistré. Ce module détermine ainsi le degré de similarité(ou de divergence)entre deux vecteurs biométriques.

1.2.5.5 Module de décision

Vérifie l'identité affirmée par un utilisateur ou détermine l'identité d'une personne [14] basée sur le degré de similitude entre les caractéristiques extraites et le(s) modèle(s) stocké(s).

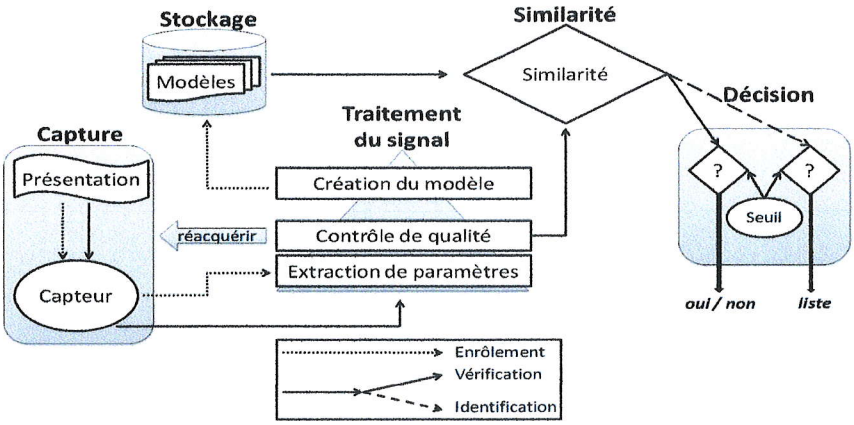


FIGURE 1.4 – L'architecture d'un système biométrique.

1.2.6 L'évaluation des performances d'un système biométrique

Lors de l'utilisation de système biométrique, il est difficile d'obtenir des résultats à 100% sans erreur. La raison est peut-être à chercher dans les différents environnements lors de l'acquisition des données (éclairage, température..etc) et dans les différents équipements utilisés (Caméras, Scanners.. etc). Avant tout, pour comprendre comment déterminer la performance d'un système biométrique, il nous faut définir clairement trois critères principaux :

1.2.6.1 Le taux de faux rejets (TFR)

("False Reject Rate" ou FRR), c'est la possibilité que le système produise un faux rejet. Ce taux représente le pourcentage des personnes censées être reconnues mais qui sont rejetées par le système. Il est également connu sous le nom de "taux de faux négatifs".

$$FRR = \frac{nombre\ de\ client\ rejeté(FR)}{nombre\ total\ d'accès\ clients} \tag{1.1}$$

1.2.6.2 Le taux de fausse acceptation (TFA)

("False Accepte Rate" ou FAR) est la probabilité qu'un système biométrique identifie de manière incorrecte une personne ou ne réussisse pas à rejeter un imposteur. Il mesure le pourcentage d'intrants non valides qui sont acceptés à tort. Il est également connu sous le nom de "taux de faux Positif".

$$FAR = \frac{Nombre\ imposteurs\ acceptés\ (FA)}{Nombre\ total\ d'accés\ imposteur} \tag{1.2}$$

1.2.6.3 Le taux d'erreur (TEE)

("Equal Error Rate" ou EER), ce taux est calculé à partir des deux premiers critères et constitue un point de mesure performance courant. Ce point correspond au lieu où FRR = FAR, c'est-à-dire le meilleur compromis entre le faux rejet et la fausse acceptation.

$$EER = \frac{nombre\ de\ fausses\ acceptations\ (FA) + \ nombre\ de\ faux\ rejets}{nombre\ totale\ d'accés} \tag{1.3}$$

Comme la Figure 1.5 illustre, nous avons toujours une zone de recouvrement. nous aimons évidemment avoir les deux distributions parfaitement disjointes, ce qui permettrait idéalement de séparer les authentiques des imposteurs, mais ce n'est jamais le cas dans la réalité.

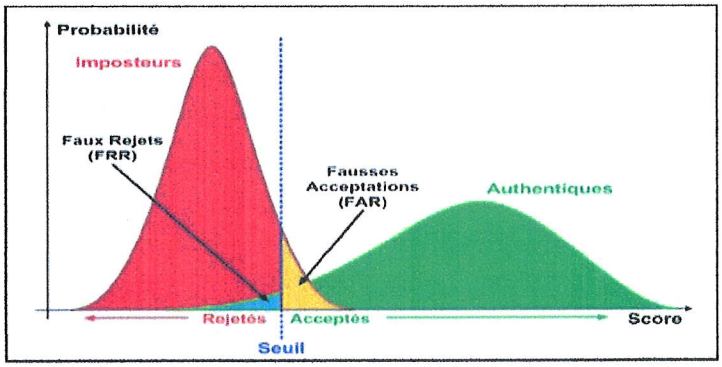


FIGURE 1.5 – Illustration du FRR et du FAR.[15]

Lorsque le système est en mode d'authentification, nous utilisons ce que l'on appelle une courbe ROC (Receveur Operating Caractéristique) [15]. La courbe ROC (Figure I.6) trace le taux de faux rejet en fonction du taux de fausse acceptation. Plus cette courbe tend à épouser la forme du repère, plus le système est performant, c'est-à-dire possédant un taux de reconnaissance global élevé.

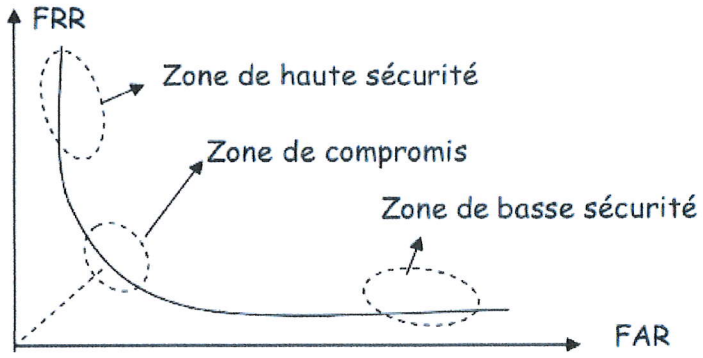


FIGURE 1.6 – Courbe ROC.[15]

1.2.6.4 Comparaison des performances

Le vrai taux positif (TPR) P_{TPR} et le taux de faux positifs (FPR) P_{FPR} ont été calculés comme suit [18] :

$$P_{TPR} = \frac{n\ similaire}{N\ identique}$$

$$P_{FPR} = \frac{n\ distinct}{N\ different}$$

Où $n\ similaire$ est le nombre des paires d'images visuellement identiques correctement identifiées comme des images similaires, $N\ identique$ est le nombre total de paires d'images visuellement identiques, $n\ distinct$ est le nombre des paires d'images distinctes considérées par erreur comme des images similaires, et $N different$ est le nombre total de paires d'images différentes. Il est clair que le TPR et le FPR sont respectivement des indicateurs de robustesse et de discrimination.

1.2.7 Applications des systèmes biométriques

Les premières applications dans uns cadre officiel du concept de la reconnaissance biométrique fut dans le domaine médico-légale tel que l'identification physique, l'identification d'un criminel ...etc. La biométrie toucha par la suite le domaine gouvernemental tel que (la carte d'identité nationale, le permis de conduire, la sécurité des frontières, le contrôle des passeports ...etc.) pour atteindre de nos jours le domaine commercial par exemple (l'ouverture de réseau informatique, la sécurité de données électroniques, le commerce, l'accès internet, la carte de crédit, le contrôle d'accès physique, le téléphone cellulaire, la gestion des registres médicaux ...etc.).

FIGURE 1.7 – Différentes applications de la biométrie dans notre vie.[19]

1.3 Système de reconnaissance d'empreinte digitale

1.3.1 Les caractéristiques des empreintes

Une empreinte digitale est constituée d'un ensemble de lignes localement parallèles formant un motif unique pour chaque individu (Figure 1-8), on distingue :
Les stries (ou crêtes) : Les lignes en contact avec une surface au toucher.
Les sillons : les creux qui se trouvent entre les stries.
Les pores : des trous qui sont régulièrement espacés dans les stries.

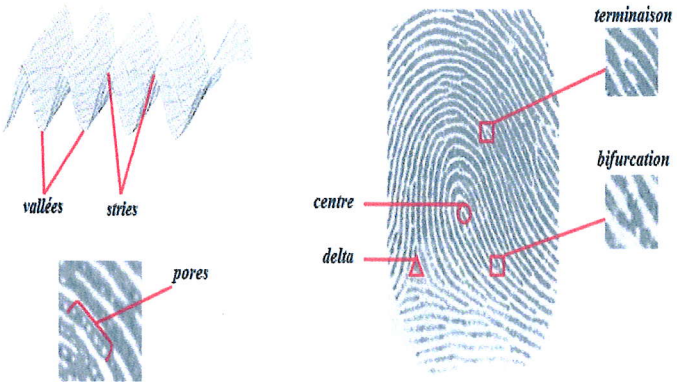


FIGURE 1.8 – Caractéristiques d'une empreinte digitale.

L'empreinte digitale est caractérisée par ses propriétés globales et locales. Typiquement, les représentations globales sont utilisées pour la classification d'empreinte digitale, alors que les représentations locales sont destinées à la comparaison de cette modalité.

1.3.1.1 La représentation globale

Chaque empreinte digitale a un ensemble des points singuliers globaux qui sont les centres et les deltas, symbolisés respectivement par θ et Δ.
Le centre est le lieu de convergence des stries (il est aussi appelé le core), alors que le delta correspond au lieu de divergence [17]. La position et le nombre de ces points permettent la classification des empreintes digitales, c'est ainsi que Francis Galton les a subdivisées en trois grandes familles :

Les Spires (Whorls) : représentent 30% des empreintes des doigts humains.
Les Arches (Archs) : ne représentent que 5% des empreintes des doigts humains.
Les Boucles (Loops) : représentent 65% des empreintes des doigts humains.

Edward Henry les a classées En six sous-classes principales [3] : arche, boucle à gauche (leftloop), boucle à droite (Right loop), arche penchée (tentedarch), spires et spires im-

briquées ou boucles jumelles (voir la Figure 1-9).

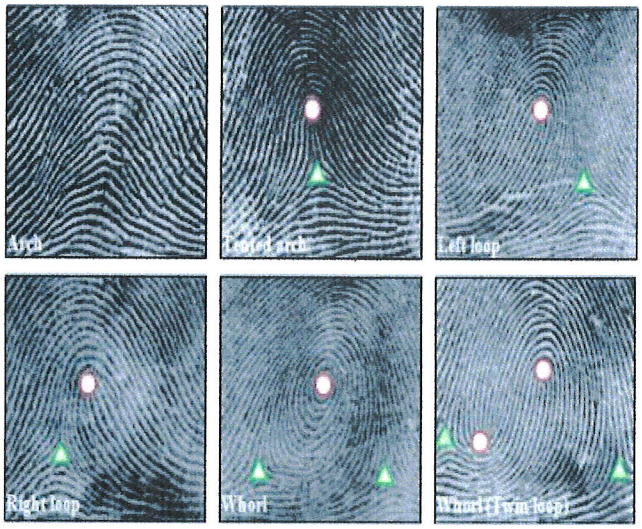


FIGURE 1.9 – Les principales classes d'empreintes digitales selon la classification de Galton-Henry.[19]

1.3.1.2 La représentation locale

Il s'agit des caractéristiques les plus utilisées "Les minuties" (littéralement : petits détails), qui sont en fait les points d'irrégularités qui se trouvent sur les lignes capillaires. Nous pouvons distinguer jusqu'à seize types de minuties différentes, mais dans les algorithmes on s'intéresse qu'aux deux types suivants parce qu'ils sont facilement détectables :

La bifurcation : c'est le point où la strie se divise en deux.
La terminaison : c'est le point où la strie s'arrête.

En fait les autres types de minuties ne sont que les résultats des combinaisons des minuties de terminaison et de bifurcation. Par exemple, les boucles peuvent être visualisées en tant que deux bifurcations.
Chaque minutie est représentée par les cordonnées (x, y) de sa position dans l'image, et l'angle (θ)qui est la direction associé à la strie (voir la figure 1-10).
Donc chaque minutie de l'empreinte est repérée par un vecteur de la forme : (Type de minutie, x, y,θ).

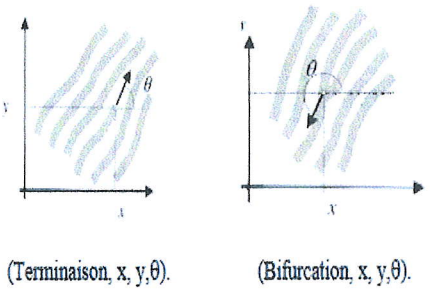


FIGURE 1.10 – Représentation des vecteurs de terminaison et de bifurcation.[19]

1.3.2 Structure d'un système de reconnaissance d'empreinte digitale

1.3.2.1 Principe générale

Un système automatique complet de reconnaissance d'empreintes digitales est une chaîne de processus qui à partir du doigt d'un utilisateur en entrée renvoie un résultat en sortie [20], permettant ainsi à l'utilisateur d'accéder ou non à des éléments nécessitant une protection. La réalisation d'un tel système a fait l'objet de très nombreuses recherches et des méthodes très différentes de traitement ont été proposées. Néanmoins ces systèmes répondent toujours à la même structure (Figure 1-11).

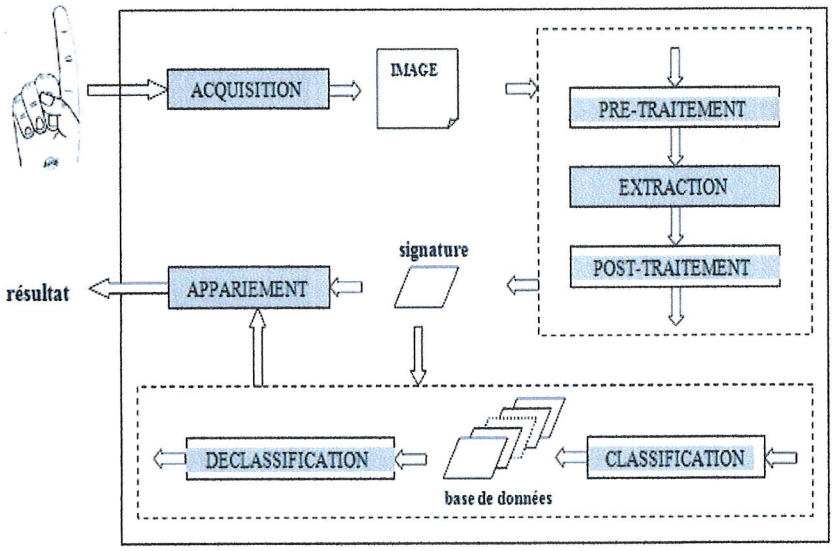


FIGURE 1.11 – Architecture générale d'un système complet de reconnaissance d'empreintes.

1.3.2.2 Acquisition

Il existe 2 méthodes pour l'acquisition d'une image d'empreinte digitale :

L'acquisition indirecte : où il existe 2 méthodes :

L'empreinte acquise par encre.
Les empreintes latentes.

L'acquisition directe.

1.3.2.3 Pré-traitement

Pour permettre une reconnaissance fiable un pré-traitement est alors nécessaire pour améliorer la qualité de l'image obtenue et éviter les erreurs [23]. L'image est donc filtrée pour augmenter l'efficacité du traitement, les caractéristiques locales des stries (direction et fréquence) sont généralement utilisées.

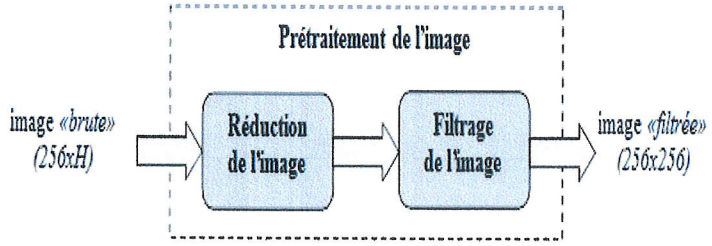


FIGURE 1.12 – Le principe de prétraitement de l'image.

1.3.2.4 Extraction des caractéristiques

L'ensemble des minuties d'empreinte (signature) est extrait à partir de l'image filtrée d'empreinte digitale. Pour cela nous étudierons une méthodes qui s'appelle "La méthode classique", et qui consiste à extraire l'information sur un squelette binaire (noir et blanc) de l'image filtrée.

La méthode classique :

Dans cette approche, l'image en niveau de gris est convertie en une image binaire. Puis elle sera amincie (squelettisée) afin de diminuer l'épaisseur des stries à ce stade [25], les minuties seront bien visibles et faciles à détecter, comme montre la figure suivante :

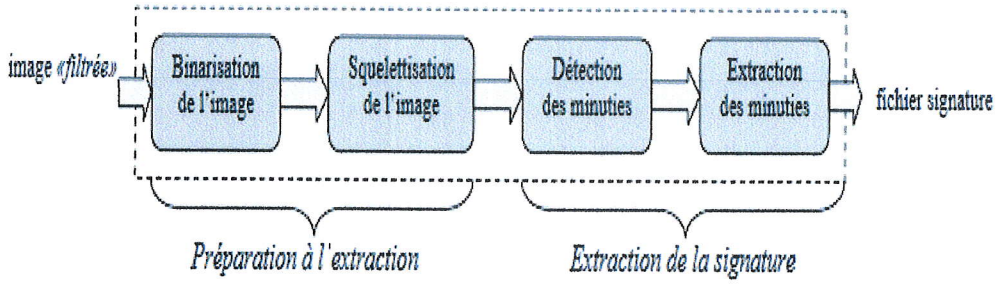


FIGURE 1.13 – La phase dextraction de la signature

La Binarisation de l'image :

Pour augmenter la visibilité des minuties, l'image doit d'abord être binarisée, il s'agit de donner une intensité aux stries et une autre intensité différente aux vallées [26]. La technique utilisée consiste à comparer chaque pixel à un seuil S, en lui assignant la valeur un (noir) si son intensité est supérieure à S sinon il prend la valeur zéro (blanc). Le seuil S peut être global fixé dès le départ ou local adaptatif obtenu en calculant la moyenne des pixels de chaque bloc de l'image de l'empreinte digitale.

$$I_T(x,y) = \begin{cases} 1 \text{ si } I(x,y) \geq T \\ 0 \text{ si } I(x,y) \leq T \end{cases} \quad (1.4)$$

FIGURE 1.14 – Résultats de l'étape de binarisation

La squelettisation :

Pour pouvoir détecter rapidement les minuties l'image doit être squelettisée : c'est une sorte d'amincissement basée sur des opérations morphologiques itératives, dans le but d'obtenir une image plus schématique de l'empreinte [27] dans laquelle toutes les stries ont un pixel de largeur tout en conservant la connexité (c.à.d. respecter la continuité des stries).

L'algorithme consiste a [35] :

Pour chaque pixel P_0 nous considérons son voisinage immédiat $\{P_i i \in [1...8]\}$ de 8 pixels. Pour l'algorithme de squelettisation nous considérons les définitions suivantes :

1. P_0 est un point frontière Nord si $P_2 = 0$.

2. P_0 est un point frontière Est si $P_4 = 0$.

3. P_0 est un point frontière Sud si $P_6 = 0$.

4. P_0 est un point frontière Ouest si $P_8 = 0$.

5. P_0 est un point 8-terminal si un seul de ses voisins est noir ($\exists i \in [1...8] P_i = 1$), il s'agit en fait d'une minutie de type terminaison.

6. P_0 est un point 8-isolé si aucun de ses voisins n'est noir ($\forall i \in [1...8] P_i = 0$)

7. P_0 est un point 8-simple si la connexité de ses 8 voisins n'est pas altérée quand on le transforme en pixel blanc.

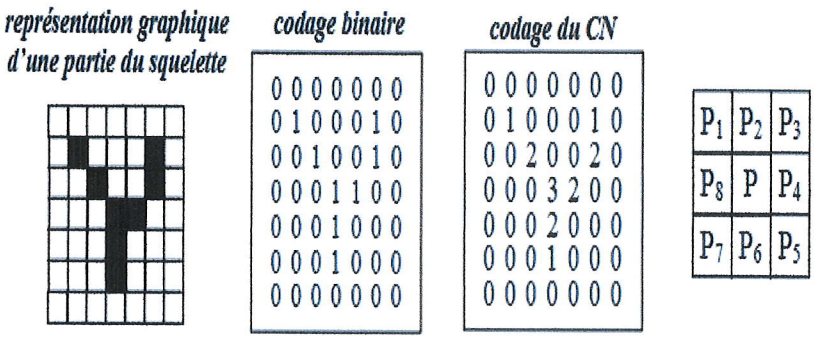


FIGURE 1.15 – Les différentes représentations de squelette.

La squelettisation consiste à répéter les opérations d'érosion suivantes jusqu'à ce que plus aucun pixel ne soit changé :

Étape 1 : tous les pixels noirs vérifiant (1) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Nord).

Étape 2 : tous les pixels noirs vérifiant (2) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Est).

Étape 3 : tous les pixels noirs vérifiant (3) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Sud).

Étape 4 : tous les pixels noirs vérifiant (4) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Ouest).

• La propriété (7) peut être ignorée dans les étapes d'érosion car bien qu'un point unique corresponde à une minutie sa présence à ce stade du traitement est très probablement due à un résidu de bruit, il vaut donc mieux l'effacer. Il est à noter que plus l'épaisseur des stries sera importante et plus le processus sera long. La Figure 1.16 montre le résultat obtenu à partir d'une image binaire filtrée.

FIGURE 1.16 – Squelette de limage binaire de lempreinte

Une fois l'image squelettisée, nous passons à l'extraction de signature. Cette dernière doit être la plus représentative de l'empreinte digitale c.à.d. le nombre de minuties extraites doit être suffisant, 12 au minimum (pour pouvoir effectuer une comparaison fiable entre les empreintes), et fiable (c.à.d. les minuties retenues ne font pas partie des zones bruitées ni des zones altérées temporairement de l'empreinte digitale comme les blessures par exemple).

La détection des minuties :

Les deux étapes de préparation à l'extraction (binarisation et squelettisation) ont grandement facilité cette phase. Dans cette étape la méthode utilisée est celle initiée par Arcelli.[28]

En effet nous disposons maintenant d'une image binaire squelettisée : un pixel noir prend la valeur 1, un pixel blanc prend la valeur 0 et la largeur des stries est égale à 1 pixel. Si nous calculons le nombre de transitions divisé par 2 entre un pixel blanc et un pixel noir pour chaque point du squelette, nous obtenons le nombre CN de stries partant de ce point (*CrossingNumber*) et nous pouvons donc déterminer simplement le type d'un pixel (voir Figure 1-17).

$$CN(P) = \frac{1}{2}\sum_{i=1}^{8} |P_i - P_{i-1}| avec P_8 = P_0 et P_i \in \{0,1\} \tag{1.5}$$

Ainsi pour chaque pixel P appartenant à une strie (c'est-à-dire pour chaque pixel ayant une valeur)

Le calcul de CN peut prendre cinq valeurs :
CN (P)= 0 : Dans ce cas il s'agit d'un pixel t, 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, le pixel se situe sur une strie, il n y a pas de minutie.

CN (P)= 3 : nous sommes en présence d'une bifurcation triple.

CN (P)= 4 : nous sommes en présence d'une bifurcation quadruple.Ce type de minutie étant assez rare il est probablement à du bruit et nous l'ignorons.

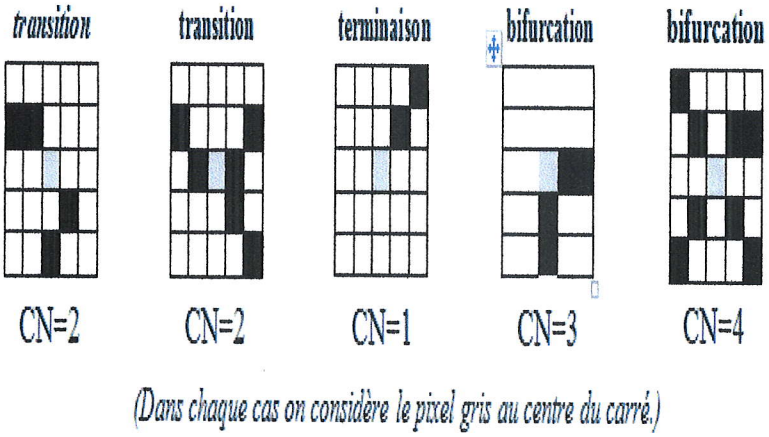


FIGURE 1.17 – Exemples de détermination du type de minutie en fonction du calcul de CN.

À la fin de ce processus, nous acquérons un grand nombre des minuties, 100 minuties en moyenne. Parmi lesquelles environ 60% sont des minuties erronées, y compris celles introduites lors des étapes de binarisation et de squelettisation. Donc un post-traitement est nécessaire pour éliminer les fausses minuties avant de passer à l'étape suivante.

1.3.2.5 Post-traitement

Dans la section précédente nous avons vu qu'un traitement supplémentaire est nécessaire pour éliminer la multitude de fausses minuties produites au cours des étapes de binarisation et de squelettisation. Comme le montre la Figure 1-18, ces fausses minuties sont diversts et variées. Pour cela nous utilisons des considérations heuristiques[30], [31] basées sur le fait que la distance entre deux minuties voisines ne doit pas dépasser un certain seuil. Pratiquement, si nous trouvons plusieurs minuties dans une petite région cela indique la présence de bruit, deux terminaisons plus proches indiquent une coupure dans la strie [24], ainsi que beaucoup de fausses minuties se situent généralement au bord de l'image.

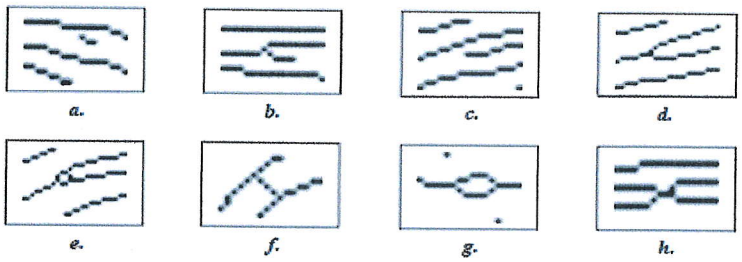


FIGURE 1.18 – Exemples de minuties détectées, segment trop court (a), branche parasite (b), vraie terminaison (c), vraie bifurcation (d), triangle (e), pont (f), îlot (g), segment trop court (h).

Le traitement des terminaisons détectées :

Pour éliminer les fausses terminaisons, nous suivons les règles suivantes :

✓ S'il existe un bloc adjacent au bloc contenant la terminaison $T(xT, yT)$ et appartenant au bord de l'image, alors T est une fausse minutie et nous élimine.

✓ Pour les terminaisons restantes T, on parcourt la strie qui lui est associée sur une distance maximum K_1, jusqu'à atteindre le point A.

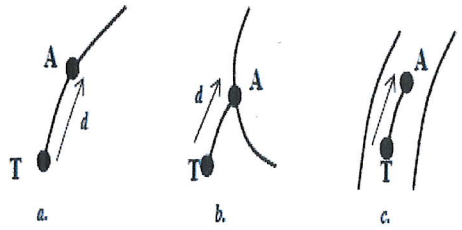


FIGURE 1.19 – Validation des terminaisons détectées, cas d'une vraie terminaison (a), branche parasite (b), segment trop court (c).

Le traitement des bifurcations détectées :

Lorsque nous détectons un point B candidat pour le titre de bifurcation ($CN(B) = 3$), nous parcourons les trois stries qui lui sont associées sur une distance maximum de K_1 jusqu'à atteindre trois points A_1, A_2 et A_3.

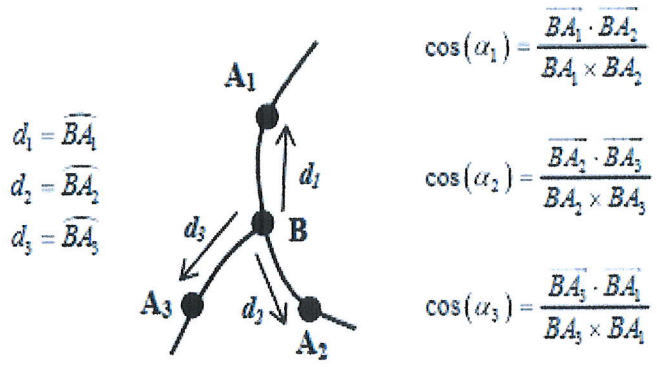


FIGURE 1.20 – Définitions associées à une bifurcation lors de la phase de validation

✓ Si $d_1 < k_1$, $d_2 < k_1$ et $d_3 < k_1$: la zone circulaire de centre B et de rayon k_1 contient au moins 4 minuties. nous considérons alors que nous sommes dans une zone très bruitée et que B est une fausse bifurcation.

✓ Si $CN(A_1) = 1$ ou $CN(A_2) = 1$ ou $CN(A_3) = 1$: au moins une des stries mène à une terminaison. Nous sommes dans le cas d'une branche parasite, le point B et les terminaisons atteintes ne sont pas validés.

✓ Si $A_1 = A_2$ ou $A_2 = A_3$ ou $A_3 = A_1$: deux des stries mènent au même point. Nous sommes dans le cas d'un lac, le point B et la bifurcation atteinte ne sont pas validés.

✓ Nous avons deux des stries qui mènent à deux bifurcations A_1 et A_2 ($CN(A_1) = 3$) et $CN(A_2) = 3$). Dans ce cas nous calculons la différence angulaire α_1 ainsi que la distance $\|\overline{A_1.A_2}\|$ entre les deux bifurcations rencontrées. Si les conditions $|cos(\alpha_1)| > cos\frac{\pi}{4}$ et $\|\overline{A_1.A_2}\| < \lambda$ sont réunies alors on est dans le cas d'un triangle et nous considérons que B est une vraie bifurcation tandis que A_1 et A_2 sont des fausses.

✓ Une seule des stries mène à une bifurcation A_1 ($CN(A_1) = 3$) nous calculons les différences angulaires α_2 et α_3 ainsi que la distance entre A_1 et B. Si $|cos(\alpha_2)| < cos\frac{\pi}{4}$, $|cos(\alpha_1)| < cos\frac{\pi}{4}$ et $\left|\widehat{BA_1}\right| \leq \lambda$ alors nous somme dans le cas d'un pont et A_1 et B sont considérées comme des fausses minuties.
Dans tous les autres cas le point B est validé en tant que vraie bifurcation.

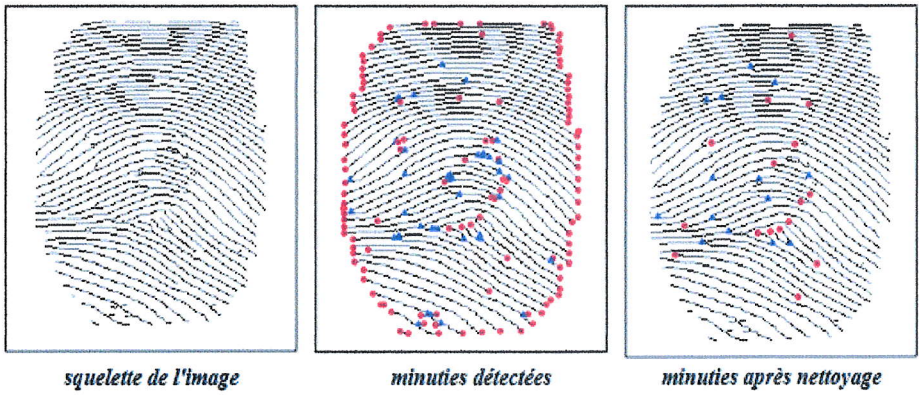


FIGURE 1.21 – Résultat de la phase d'élimination des fausses minuties.

1.3.3 La comparaison des empreintes digitales

La comparaison des empreintes digitales consiste à réaliser un accord entre la signature à identifier et les signatures stockées dans la base de données. Cependant cette tâche n'est pas facile, notamment à cause de la variabilité dans les différentes impressions d'une même empreinte (variation intra-classe).

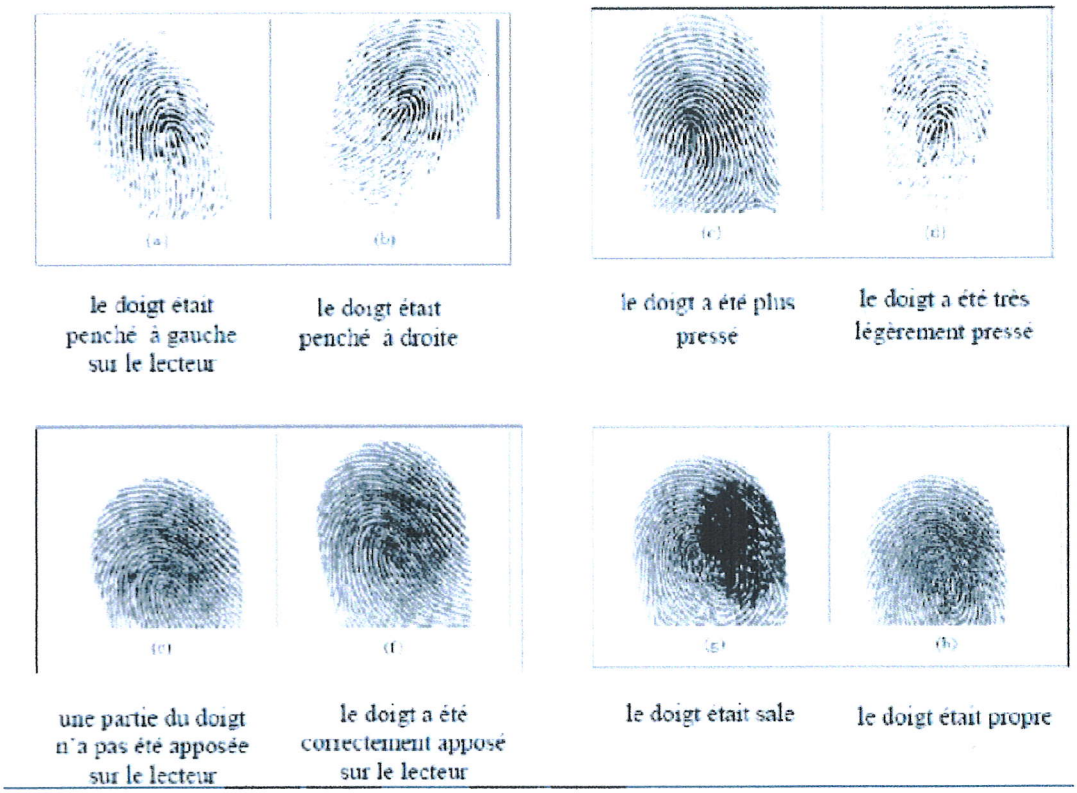


FIGURE 1.22 – Les différentes positions de même doigt.[19]

Les différentes approches de comparaison des empreintes digitales sont classées en trois

grandes familles :

1.3.3.1 Les approches basées sur la corrélation

Dans ces approches, deux images d'empreinte digitale sont superposées et la corrélation entre les pixels correspondants sera calculée pour différents alignements (ex : Rotation, déplacement).

1.3.3.2 Les approches basées sur les minuties

Ce sont les approches les plus utilisées. Les minuties sont extraites de deux empreintes digitales et représentées sous forme d'un ensemble de points dans un plan à deux dimensions selon le modèle des coordonnées.

1.3.3.3 Les approches basées sur les rides

lorsque la qualité de l'image de l'empreinte digitale est mauvaise l'extraction des minuties est très difficile voire même impossible.

1.4 Conclusion

Dans ce chapitre nous avons vu l'émergence des méthodes biométriques de reconnaissance et des problèmes qui en découlent. Les caractéristiques des empreintes digitales ainsi que la structure globale d'un système de reconnaissance d'empreintes ont également été décrit, il s'agit à l'heure actuelle de la technique biométrique la plus fiable.

Dans le prochain chapitre, nous allons présenté les différentes classes d'outils de sécurité des images qui s'existent dans la littérature en détaillant la technique de hachage perceptuel, son schéma général ainsi que ses propriétés.

Chapitre 2

Sécurité des images via le Hachage perceptuel

2.1 Introduction

Nous vivons dans le monde ou presque tout peut être digitalisé et envoyé d'une partie du monde à l'autre en seulement quelques secondes. L'opération de traitement de contenu est devenue trivial, 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 falsification, plusieurs techniques d'authentification ont vu le jour.

Dans ce chapitre nous abordons les différentes techniques de sécurisation d'image où 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 stéganographie et la cryptographie

La stéganographie a pour but de dissimuler un message secret dans un médium insoupçonné appelé médium de couverture, qui peut être par exemple une image, un son, une vidéo. Cette dissimulation doit se faire en modifiant le médium de couverture, mais avec la contrainte de minimiser les changements effectués sur le médium pour le rendre imperceptible. Plus le message secret est réduit et le médium de couverture volumineux, plus cette altération a des chances de passer inaperçue. D'où l'utilisation de fichiers image, son ou vidéo plutôt que de fichiers texte, de taille plus réduite.

La stéganographie repose sur l'idée de sécurité par l'obscurité [37] : si personne ne sait qu'il y a un message caché, personne ne cherchera à le regarder ou à le récupérer.

La cryptographie consiste à chiffrer un message de façon à le rendre inintelligible et incompréhensible, afin de garantir sa confidentialité. Cette opération permet de s'assurer que seules les personnes auxquelles les messages sont destinés pourront y accéder. Il existe deux type de chiffrement : chiffrement symétrique est chiffrement asymétrique.

2.2.2 Le tatouage numérique

Le tatouage numérique ou watermarking est une technique permettant d'ajouter des informations de copyright ou d'autres messages de vérification à un fichier ou signal audio, vidéo, une image ou un autre document numérique [38], qui consiste à insérer une marque invisible dans un support numérique.
Le message inclus dans le signal hôte, généralement appelé marque ou bien simplement message, est un ensemble de bits, dont le contenu dépend de l'application [39]. La marque peut être un nom ou un identifiant du créateur, du propriétaire, de l'acheteur ou encore une forme de signature décrivant le signal hôte.
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

Une fonction de hachage cryptographique H est un algorithme permettant de générer une empreinte $H(M)$ de taille fixe de n bits à partir d'une donnée M de taille quelconque.

$$H : (0,1)^* \longrightarrow (0,1)^n$$
$$M \longrightarrow H(M)$$

Où la notation $(0,1)^*$ désigne l'ensemble des chaînes de bits de longueur arbitraire finie et la notation $(0,1)^n$ désigne l'ensemble de chaînes de bits de longueur exactement n . En pratique, n est de l'ordre de plusieurs centaines de bits.
De plus, avec hash cryptographique, les valeurs de hachage sont aléatoires. Le message qui est utilisé pour générer les actes de hachage comme une graine aléatoire de sorte que les mêmes données provoquent exactement le même résultat, mais le message différent produira un hachage tout à fait différent. Les fonctions de hachage cryptographique impliquent deux choses principales :
Si les hashs sont différents, les données sont différentes ou vice-versa [40]. En conséquence, l'intégrité du message ne peut être validé que lorsque chaque bit du message est inchangé. Un tel comportement caractérise essentielle de ce type de hash.

2.2.3.2 Hachage perceptuel

Il y a, cependant, plusieurs raisons qui empêchent l'utilisation directe des techniques cryptographiques pour résoudre les problèmes de sécurité multimédia [41]. Contrairement aux données textuelles qui sont transmises à travers un milieu sans perte, les données multimédias comme des images (ou audio, vidéo, etc.) peuvent être transmises et stockées à l'aide d'un support avec perte pour économiser la bande passante et l'espace de stockage.

Par conséquent, l'utilisation des fonctions de hachage cryptographique traditionnel pour la vérification de l'intégrité ou l'authentification du contenu multimédia a un problème qu'un seul changement de bit dans le contenu changera de façon significative la valeur de hachage. Pour analyser le contenu multimédia, d'autres algorithmes de hachage sont préférables.

De plus, les humains peuvent facilement distinguer plusieurs images et dire si oui ou non ce sont les mêmes. Un ordinateur, voit cependant tout dans une perspective très différente, donc cette tâche facile pour un être humain est assez compliqué pour un ordinateur. Plusieurs images peuvent avoir différentes représentations numériques, mais pour la perception humaine, ils sont tous pareils. Cela nous amène au problème que le contenu multimédia peut être distribué illégalement imperceptiblement pour les algorithmes de recherche si on considère uniquement les techniques de hachage cryptographique. Les données telles que des images numériques peuvent subir diverses manipulations telles que la compression, la rotation...etc. Il est impossible au crypto-hachage traditionnel de repérer les images modifiées et déduire l'origine.

2.3 Hachage perceptuel des images

2.3.1 Définition de hachage perceptuel

C'est un algorithme qui produit une empreinte numérique particulièrement adaptée aux fichiers multimédias. L'intérêt est que le résultat est proche si les fichiers le sont, contrairement aux algorithmes précédents, il est utilisé pour détecter les violations des droits d'auteur sur des images, des articles, des fichiers audio, ou en criminalistique numérique.

Les fonctions de hachage perceptuel d'image extraient certaines caractéristiques de l'image et calculent une valeur de hachage en fonction de ces caractéristiques. Ces fonctions ont été proposées pour établir l'égalité de perception du 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 à authentifier. Ces dernières années, des recherches ont été menées sur le hachage perceptuel d'image qui retient de plus en plus l'attention dans la littérature.

2.3.2 Les fonctions de hachage perceptuel

La plupart des fonctions de hachage produisent des empreintes radicalement différentes si l'entrée est légèrement modifiée [43]. Ce phénomène est particulièrement visible avec les fonctions de hachage cryptographiques qui se comportent de manière imprévisible grâce à l'effet avalanche.

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. Les fonctions de hachage perceptuel se basent sur l'aspect visuel des données à hacher. [42] La figure (2.1) illustre un exemple des exigences attendues des fonctions de hachage perceptuel. La figure (2.1.a) montre une version compressée JPEG (avec un facteur de compression acceptable visuellement) de l'image originale présentée la figure (2.1.b) [44]. L'image originale et sa version compressée JPEG doivent alors 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 à l'image malicieusement modifiée doit être radicalement différente à 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) [44].

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.

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.

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).
Les fonctions de hachage perceptuel doivent être capables de survivre à des manipulations acceptables qui préservent le contenu et de rejeter les manipulations malveillantes [44].

Manipulations acceptables	**Manipulations malveillantes**
-Ajout de bruit.	-Ajout de nouveaux objets.
-Erreurs de transmission.	-Déplacement des éléments de l'image pour changer leurs positions.
-Mise à l'échelle.	-Suppression des objets sur l'image.
-Compression et Quantification.	-Changements des caractéristiques de l'image : couleur, textures, structure, impression, etc ...
-Conversion de couleurs.	-Modifications du contexte de l'image : de jour ou d'emplacement.
-Filtrage.	-Les changements de conditions d'éclairage : manipulations d'ombre.
-Réduction de résolution.	/
-Rotation.	etc ...
-Réglage de contraste.	/
-Changements de luminosité, teinte et de saturation.	/
etc ...	/

TABLE 2.1 – Manipulations acceptables et manipulations malveillantes .[45]

2.3.4 Hachage perceptuel vs Hachage cryptographique

Les fonctions de hachage cryptographique et les fonctions de hachage perceptuel ont les mêmes objectifs [46]. 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 [46]. 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 quantification et l'étape de crypto-compression.

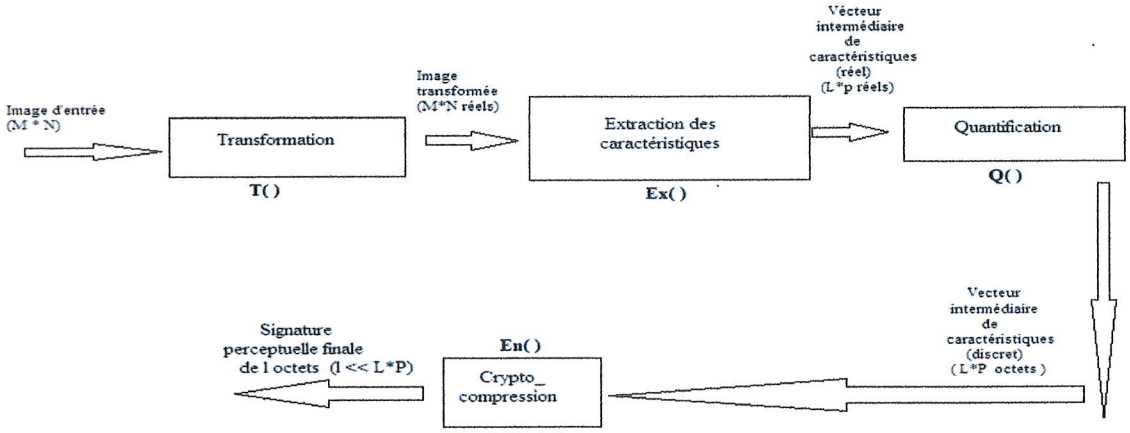


FIGURE 2.2 – Présentation des quatre étapes d'un système de hachage perceptuel [45].

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 [45]. 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 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 perceptuel durant l'étape de crypto-compression.

2.3.5.1 Étape de transformation

Durant la transformation, l'image d'entrée de taille $M * N$ 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 en ondelettes (DWT) [45]. Quand une transformée en ondelettes discrète 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, où $L \leq M * N$. À noter que chaque caractéristique peut contenir p éléments de type réel, ce qui signifie que le vecteur des caractéristiques est composé de $L * p$ réels à cette étape [45]. Une autre sélection des caractéristiques peut être ajoutée à cette étape, comme l'illustre la figure (2.3), où 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 $K * p$ réels, où $K < L$. 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 à d'une autre image.

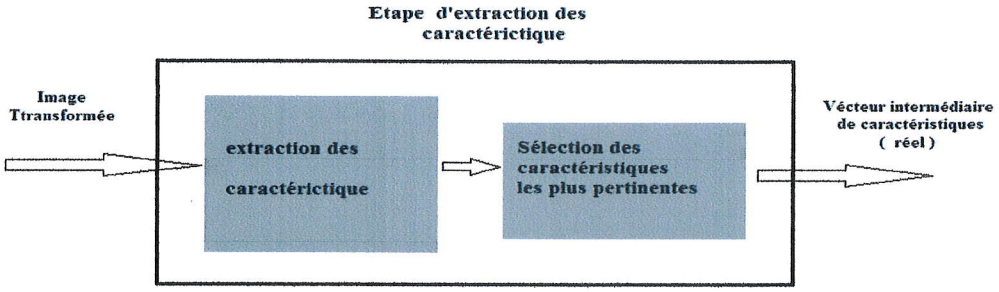


FIGURE 2.3 – Sélection des caractéristiques les plus pertinentes.[45]

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 $K * p$ 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 [45].

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 l octet, où $l \leq K * p$ [45], 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 manière cryptographique [47]. Le représente d'une image complètement différente de celle d'origine est (0). Soit θ',θ (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 secrète.

La robustesse : La robustesse signifie que si la même clé est utilisée, des images perceptuellement similaires génèrent un hachage similaire [46]. 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, 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 l'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°) a été modifiée. Dans ce cas, la valeur de probabilité devrait être égale ou très proche de 1.

La robustesse garantit que deux images perceptuellement identiques doivent avoir des hachages similaires. L'objectif principal des images similaires est d'être suffisamment robuste pour la différents manipulations acceptables tels que la transformation avec perte JPEG, la rotation, le bruit, le flou, etc. Perceptuellement, ces images dans le système visuel humain (SVH) sont identiques, même si certains des bits ont été changés.

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

$$P(H(O, K) \neq H(M, K)) \approx 0 \tag{2.2}$$

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 complètement 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 [49]. En d'autres termes, les hachages de deux images complètement différentes ne doivent pas être égaux. Il devrait y avoir une très faible probabilité, proche de 0.

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. Où 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 [47], pour diminuer la probabilité pour l'attaquant de deviner la clé secrete et estimer la valeur de hachage correcte.

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

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

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 la 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 [49]. 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 [49] :

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

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

Le nombre d'erreurs des bits i est égal à la distance de Hamming des valeurs de hachage perceptuel. Lors de la comparaison d'images perceptuellement différentes, le TEB devrait être environ 0,5 [51]. 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 chaînes, ces chaînes 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 chaînes sont issues de trois alphabets différents (système binaire, système à décennie et alphabet latin). [49]

Soit A un alphabet de longueur finie. $x = (x_1, .., x_n)$ dénote une chaîne 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 :[51] :

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

✓ **Distance Euclidienne :** La distance euclidienne est adapté pour les vecteurs non-binaires comme les nombres entiers. Elle est parfaite pour mesurer la similitude et la capacité de discrimination entre deux hashs 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 [50] :

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

Où DE représente la distance euclidienne. H_1 et H_2 sont les hachs décimaux avec la même longueur$L$, h_1 est le premier bit de hachage décimal, h_2 est le second [49]. Pour la distance euclidienne, la longueur des deux hashs examinés doit être égale. Les deux valeurs de hachage sont plus proches.

A	B	Distance Euclidienne
0100100	1100110	$1 \cdot 414$
4391256	5341255	$5 \cdot 196$

TABLE 2.3 – Exemples de calcul de la distance euclidienne .[49]

✓ **Le coefficient de corrélation :** est utilisé pour distinguer le degré linéaire entre les séquences de hachage obtenues par l'image.
L'intervalle de corrélation est [-1,1] [49], où 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 stéganographie 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 d'entropie

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 devraient 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.

Dans ce chapitre nous mentionnons les différents 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 d'entropie et la décomposition elliptique, cette méthode est décomposé en quatre étapes. Son idée principale est de diviser l'image d'empreinte digitale en plusieurs ellipses, chaque ellipse se caractérise par un ensemble des minuties située dans des différents pixels.

3.2 Les Méthodes de hachage perceptuel

3.2.1 Les Méthodes de hachage perceptuel par bloc

3.2.1.1 Hachage d'image robuste à l'aide de la normalisation d'image et de la décomposition SVD

Dans cette méthode l'auteur utilise la normalisation des images et la fonction de décomposition SVD pour générer un hach robuste d'image numérique. La normalisation des images est une technique qui s'est révélée robuste par rapport à différents types d'attaques géométriques telles que la rotation mise à l'échelle et le renversement. Le schéma de la méthode est illustré par la figure suivante :

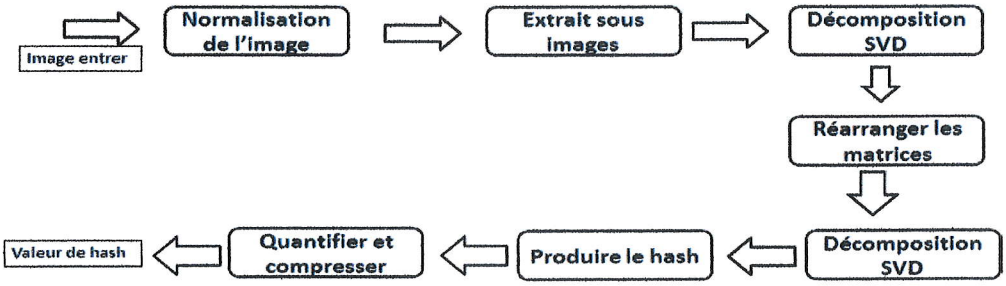


FIGURE 3.1 – Méthode de la normalisation de l'image et la décomposition SVD.

Les étapes d'algorithme de cette méthode résumé comme suit [48] :

Étape 1 : Nous appliquons la normalisation sur l'image entrer I pour générer l'image normalisée G.

Étape 2 : Après l'étape de normalisation nous extrayons des sous images A_i à partir de l'image normalisée G, chaque sous image a de taille $(m \times m)$.

Étape 3 : Nous appliquons à chaque sous image A_i la décomposition SVD pour obtenir les matrices U et V.

Étape 4 : A partir des matrices U et V, nous composons une nouvelle matrice B.

Étape 5 : Nous appliquons la décomposition SVD sur la matrice B pour obtenir les matrices U et V.

Étape 6 : Produire une séquence de hach à partir de la première colonne de U et la première ligne de V.

Étape 7 : Enfin, nous quantifions le résultat de vecteur de hach statistique, et transféré ce vecteur à une séquence de hach binaire, après nous compressons ce hach binaire avec le décodeur Reed-Muller pour obtenir le hach h.

3.2.1.2 Méthode par transformation des caractéristiques d'échelle-invariante (SIFT) et la décomposition des valeurs singuliers (SVD)

L'auteur dans cette méthode propose un algorithme de hachage basé sur la transformation des caractéristiques d'échelle-invariante (SIFT) et la décomposition des valeurs singulières (SVD), l'image d'entrée subira une étape de prétraitement suivie de l'extraction des points clés à l'aide de l'algorithme SIFT. La valeur de hach est ensuite générée par un traitement de bloc et l'opération SVD. Comme dans la figure suivante :

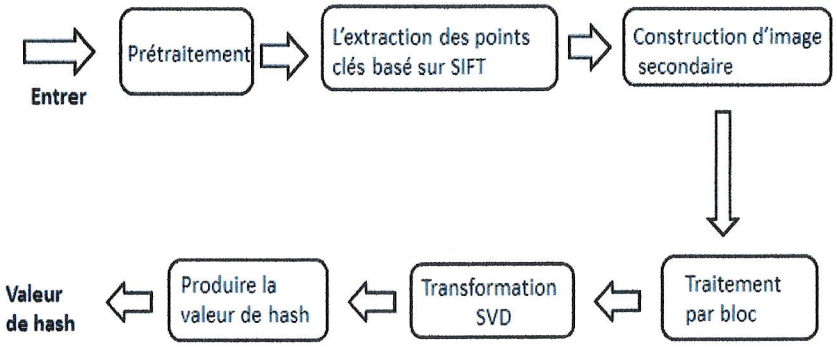


FIGURE 3.2 – Méthode transformation des caractéristiques d'échelle-invariante (SIFT) et la décomposition des valeurs singuliers(SVD).

Les étapes de cette méthode sont expliquées dans les paragraphes suivants [55] :

✓ **Prétraitement :** Dans cette étape l'image d'entrée I est redimensionnée à une taille standard de $(q \times q)$ en utilisant une interpolation bilinéaire pour s'assurer que la valeur de hachage générée a une longueur fixe.

✓ **Extraction des points clé :** Après le prétraitement les points-clés sont extraits de l'image en niveau de gris dimensionnée basé sur la méthode SIFT. Cependant, nous n'utilisons que les points principaux créés après le filtrage des points négatifs.[55]

✓ **Construction d'image secondaire :** pour cette étapes l'auteur a converti l'image J résultante de l'étape précédente a une image noir et blanc I, tel qu'il rend le pixel J(i, j) qui représente les points clés à la valeur 1 le reste à 0.

✓ **Décomposition en bloc et application SVD :** Une fois l'image secondaire créée, 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 la valeur singulière maximale pour chaque bloc utilisant, si la moyenne de toute les blocs supérieur a la valeur singulière maximale pour chaque bloc on a 0 sinon 1, dans la fin nous construisons le hach binaire.

✓ **Mesure de similitude :** La similitude entre deux hachs est mesurée à l'aide du distance de Hamming (DH), La DH normalisé entre deux hachs d'image est directement proportionnel à la similitude des images. Un seuil T prédéfini est utilisé pour classer des similaires entre différentes images.

3.2.1.3 Transformation de Fourier-Mellin pour un hachage d'image robuste

L'algorithme de hachage d'image dans cette méthode applique d'abord un prétraitement à l'image d'entrée afin de réduire l'effet des opérations communes de traitement des signaux. L'image pré-traitée est divisée en blocs imbriqués. Ces blocs sont alors commandés via une clé secrète K_1. Le bloc sera transformé par FMT. Les coefficients des caractéristiques sont sélectionnés au hasard via une autre clé K_2 de la basse fréquence. Le choix de coefficients est concaténé pour former le hachage final.
Le schéma de la méthode est illustré par la figure :

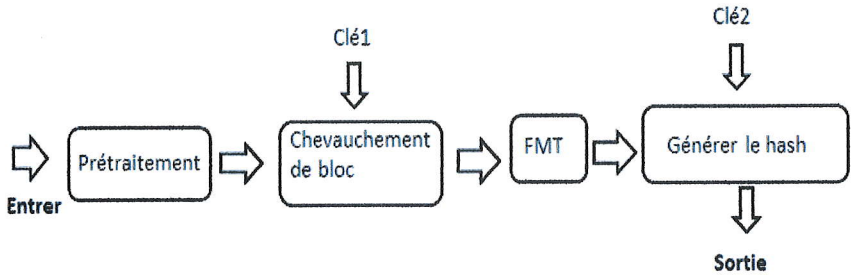


FIGURE 3.3 – Transformation de Fourier-Mellin pour un hachage d'image robuste

L'algorithme est résumé comme suit [57] :
Étape 1 : Entrer l'image I avec la taille originale.
Étape 2 : Appliquer le filtrage et l'histogramme à faible passe d'égalisation sur l'image originale. L'objectif est de réduire l'effet du traitement commun des signaux opérations.
Étape 3 : Diviser l'image en blocs imbriqués avec une sélection aléatoire en utilisant la clés K_1. Les blocs sont de taille $m * m$ et dénotés par Iblock1, Iblock2, Iblock3,..., IblockN, dans lequel N est le numéro du bloc.
Étape 4 : FMT est appliqué sur chaque bloc d'image pour extraire les caractéristiques.
Étape 5 : Les coefficients de basse fréquence sont aléatoirement sélectionnés à partir de chaque bloc dans une zone de taille $n * n$ en utilisant d'autres clés secrètes K_2.
Étape 6 : Le vecteur de hachage final VF est formé pour concaténer les coefficients extraits.

Après cette étape en utilisant la distance euclidienne pour mesurer la similitude entre deux images.

3.2.2 Les Méthodes de hachage perceptuel globales

3.2.2.1 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 [56] :
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) à 64 couleurs 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 à 64 bits. Chaque bit est simplement réglé en fonction de la moyenne (la valeur de couleur est au-dessus de la moyenne).
Étape 5 : Construire le hachage.
Étape 6 : Pour comparer deux images, calculer la distance de Hamming entre deux hachs 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 [56], 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.

3.2.2.2 Méthode utilisé le filtre de Gabor et la probabilité d'absorption de Markov

L'auteur dans cette méthode a proposer un 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 [58], 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 :

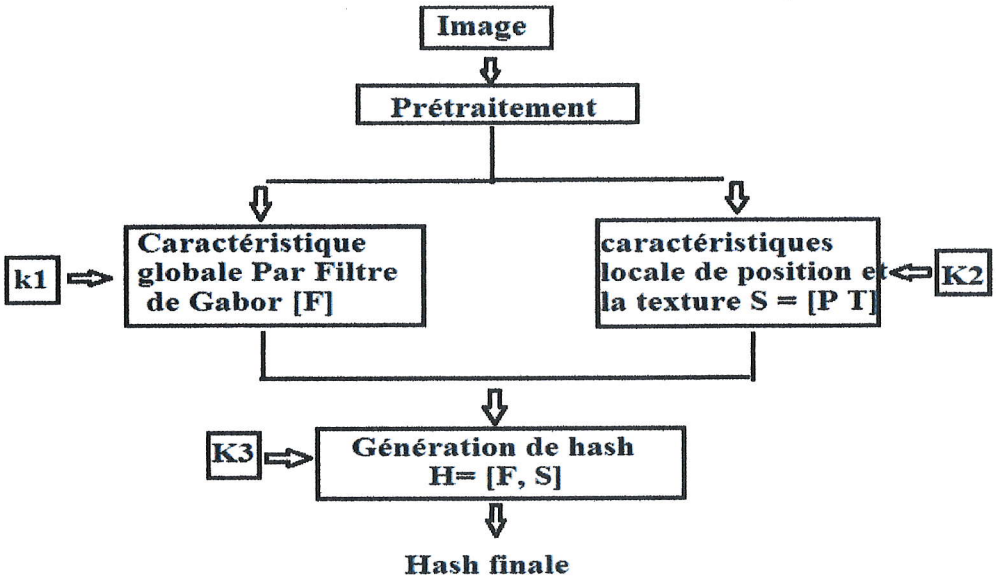


FIGURE 3.4 – Méthode utiliser le filtre de Gabor et le probabilité d'absorption de Markov

Les étapes de cette méthode sont expliquées dans les paragraphes suivantes : [58]

✓ **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 à l'aide de la clé sécurisée K_1. 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 à deux anneaux pour refléter l'information de voisinage. Converti le modèle RVB au modèle CIE Lab. En utilisant les noeuds absorbants pour la limite supérieure et la limite gauche, obtenez 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.

3.2.2.3 Hachage perceptuel des images via les points caractéristiques

C'est un algorithme de hachage visuel basé sur les points caractéristiques très importants. Les points caractéristiques devraient être largement invariants sous distorsions perceptive insignifiants. Pour cela Monga a proposé un détecteur de fonction itérative pour extraire la géométrie significative pour préserver les points caractéristiques. Il a appliqué la quantification probabiliste sur les caractéristiques dérivées pour introduire l'aléatoire, et à son tour réduit la vulnérabilité aux attaques adverses [59]. L'algorithme de hachage proposé résiste aux attaques standards y compris la compression, distorsions géométriques de mise à l'échelle, la rotation aux petits angles, et des opérations de traitement de signaux communs. Le cadre général de cet algorithme est représenté par la figure 3-5 :

FIGURE 3.5 – Diagramme du bloc de la fonction de hachage

❒ **Ondelettes End-stopped**

Des études identifiée psycho-visuellement la présence de certaines cellules appelées hypercomplexe ou End-Stopped cellules, dans le cortex visuel primaire ces cellules réagissent fortement à l'image extrêmement robuste tels que le coin comme stimulus et points de haute courbure. Le terme End-Stopped vient de la forte sensibilité de ces cellules au point finale de structures linéaire. Bhattacherjee et al, construisent les ondelettes "End-Stopped" pour capturer ce comportement. Les structures linéaires ayant une certaine orientation sont sélectionnées. Ces Structures linéaires sont ensuite traitées pour détecter les files d'extrémités (coins) et de forte courbure. Les ondelettes de Morlet peuvent être utilisées pour détecter des structures linéaires ayant une orientation spécifique [59].

Les ondelettes End-Stopped combinent les 2 étapes : La première pour détecter les lignes qui ont une orientation spécifique(**ondelette de Morlet**).
La deuxième c'est pour détecter à chaque ligne, les points à l'extrémité (**première dérivation de Gaussian FDoG**).

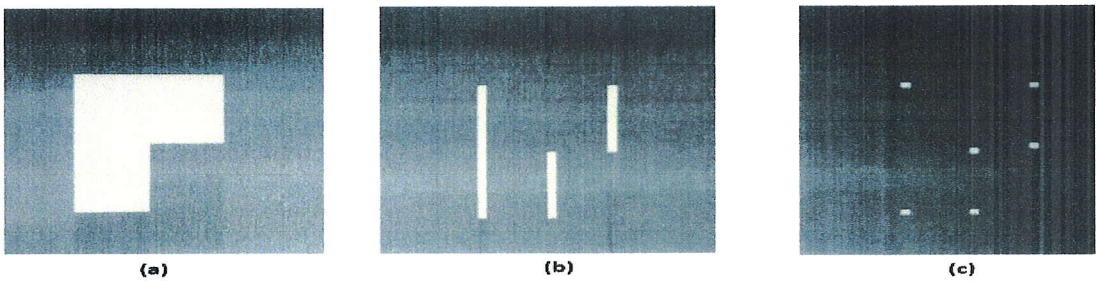


FIGURE 3.6 – Comportement ondelettes End-Stopped sur une image synthétique : notez la réponse forte aux points de haute courbure et les coins. (a) image synthétique dans la forme de L. (b) réponse d'une ondelettes End-Stopped.

❒ **Méthode de détection des caractéristiques proposée par Monga :**
Cette approche de détection des caractéristiques calcule une transformée en ondelettes repose sur les ondelettes End-Stopped obtenu en appliquant l'opérateur FDoG de l'ondelette de Morlet.

➤ Les étapes pour l'extraction des caractéristique sont les suivantes :
❶ Calculer les importants caractéristiques.
❷ Identifier les caractéristique importantes en cherchant des maxima locaux de l'amplitude des coefficients d'ondelette.
❸ Parmi les points sélectionnés en étapes **2**, sélectionner les points caractéristiques finals satisfaisant (élimine maxima locaux parasite a l'aide d'un seuil).

➢ La méthode de détection des caractéristiques a deux paramètres libres : échelle **T** entier et seuil **i** réelle [59]. Le seuil est adapté pour sélectionner un nombre fixe (paramètre défini par l'utilisateur) des points caractéristiques de l'image. Un vecteur de caractéristique d'image est formé en collectant les amplitudes des coefficients d'ondelettes à la fonction sélectionnée.

3.3 Le hachage d'image robuste à base d'entropie et la décomposition elliptique

Comme représenté sur la Figure (3.7), notre algorithme est composé de quatre étapes. La première étape est de faire un pré-traitement pour améliorer la qualité d'image et pour produire une image normalisée pour l'extraction des caractéristiques robustes. Puis, nous faisons l'extraction des minuties et la détection de core à partir d'une image de l'empreinte préalablement traitée, nous extrayons grâce à algorithme de (Crossing Number) une structure de données (ou signature). Dans l'étape suivante, nous faisons la décomposition en ellipse de l'image normalisée. Enfin, nous extrayons l'entropie d'image de ces ellipses et les utiliser pour former le hach d'image. Les sous-sections suivantes donnent la description détaillée de chaque étape.

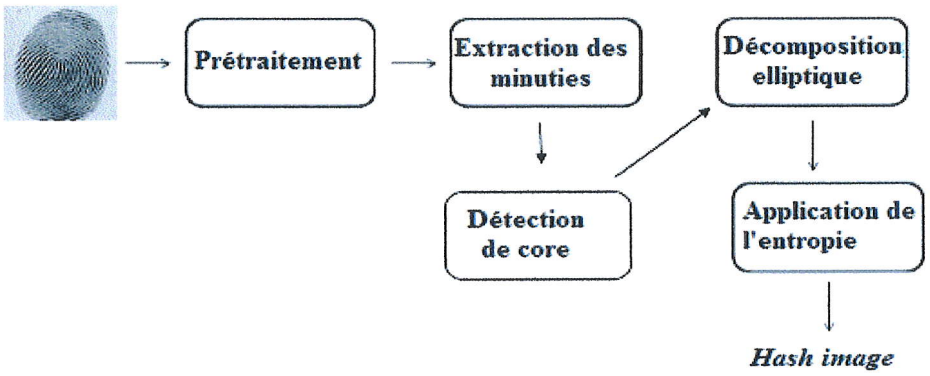


FIGURE 3.7 – Schéma représente le principe de hachage.

3.3.1 Pré-traitement

A cause de l'état de l'épiderme (sécheresse, humidité, etc.) ou du dispositif d'acquisition, la qualité de l'image en entrée peut nettement décroitre. Cela se traduit généralement par une cassure dans le flux global des crêtes et des vallées. 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 Gabor [61].
Ce dernier est une opération de traitement d'images numériques qui 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 filtres de Gabor font partie de la catégorie des filtres directionnels. Le filtrage directionnel a pour objectif d'extraire l'information utile dans la direction recherchée. Pour cela, les filtres de Gabor sont très souvent utilisés car ce sont des filtres passe bande simples à créer et ont une résolution conjointe spatiale/fréquentielle optimale [62] c.-à-d qu'ils offrent la meilleure localisation simultanée en temps et en fréquence d'après le principe d'incertitude d'Heisenberg.
Nous appelons fonction de Gabor l'association d'une courbe de Gauss et d'une sinusoïde orientée. En traitement d'images, nous travaillons dans le domaine spatial en dimension 2, ce qui nous permet d'écrire la fonction de Gabor de la manière suivante :

$$G(x, y, \theta, f) = \mathrm{e}^{\frac{-1}{2}(\frac{x_\theta^2}{\sigma_x{}^2}+\frac{y_\theta^2}{\sigma_y^2})} \cos(2\pi f x_\theta) \tag{3.1}$$

avec : $x_\theta = x\cos\theta + y\sin\theta$
et : $y_\theta = y\cos\theta + x\sin\theta$

Où θ est l'orientation de la sinusoïde, f sa fréquence et σ_x (respectivement σ_y) l'écart type de la gaussienne selon l'axe des abscisses.

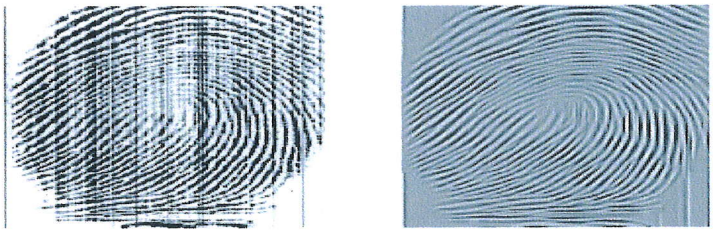


FIGURE 3.8 – Le résultat obtenu après le prétraitement.

3.3.2 Extraction des minuties

Comme nous avons vu dans le chapitre 1, l'extraction des minuties c'est le processus final qui complète l'obtention de la "signature" de l'empreinte.
A partir d'une image d'empreinte préalablement traitée, nous extrayons grâce à l'algorithme (Crossing Number) une structure de données (ou signature).

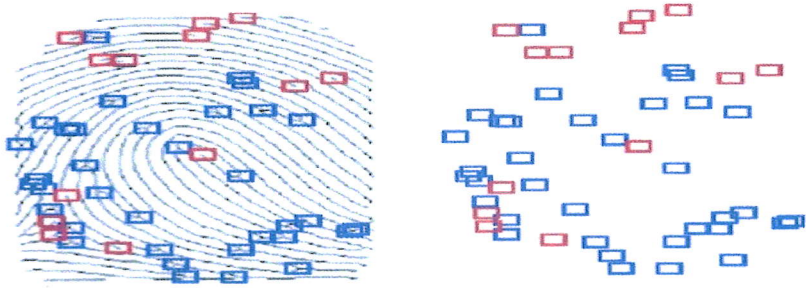


FIGURE 3.9 – Le résultat obtenu après l'extraction des minuties.

3.3.3 Détection de core

La plupart des approches proposées dans la littérature pour la détection de singularité fonctionnent sur l'image d'orientation de l'empreinte finale.
La localisation des points centraux est l'étape principale des différents systèmes de vérification et d'authentification des empreintes digitales. Il est facile d'identifier visuellement les points centraux des empreintes digitales, mais faire reconnaître les points par un ordinateur peut sembler une tâche ardue.
Une approche mathématique utilisant la méthode de Poincaré est très pratique à cet égard.

L'implémentation pas à pas de l'algorithme développé par Jin Bo Et Al[60], permet d'obtenir une estimation pratique du noyau . Avant de suivre leur approche, il est sage de

faire l'amincissement et la rotation à l'image afin d'obtenir des résultats uniformes et précis.

Les étapes mises en œuvre :

❶ L'image donnée doit être divisée en plusieurs blocs d'images plus petites. Cela ressemble à ceci :

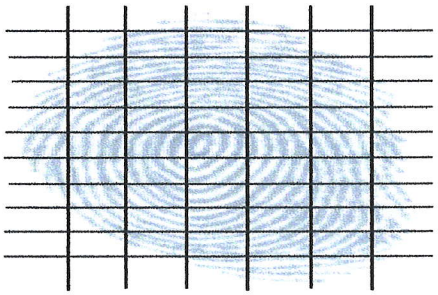


FIGURE 3.10 – Dividion de l'image en plusieur blocs.

❷ Les gradients de chaque bloc sont obtenus en appliquant un opérateur Kernel sur l'ensemble de l'image. Le résultat obtenu est G_x et G_y.
G_{xx}, G_{yy} et G_{xy} sont obtenus en utilisant ce qui suit :

$$\begin{cases} \mathrm{G}_{xx} = {G_x}^2 \\ \mathrm{G}_{xy} = G_x * G_y \\ \mathrm{G}_{yy} = {G_y}^2 \end{cases}$$

❸ Lisser les moments obtenus ci-dessus à l'aide d'un filtre approprié.

❹ Appliquez les opérations suivantes sur les moments obtenus ci-dessus :

$$sin\Theta = \frac{G_{xy}}{\sqrt{{G_{xy}}^2 + (G_{xx} - G_{yy})^2}}$$

$$cos\Theta = \frac{G_{xy} - Gyy}{\sqrt{{G_{xy}}^2 + (G_{xx} - G_{yy})^2}}$$

❺ Lissez les composants ci-dessus comme indiqué, en utilisant une taille de fenêtre appropriée.

❻ Calculez le champ d'orientation à l'aide de la formule suivante :

$$orientation = \frac{\pi}{2} + \frac{\tan^{-1}(sin\Theta, cos\Theta)}{2}$$

$\frac{\pi}{2}$ apparaît car la direction du flux est perpendiculaire à la dépendance directionnelle. Obtenu ainsi le champ d'orientation.

❼ Le calcul de l'indice de Poincaré au pixel (x,y)

$$Poincare(x, y) = \frac{1}{2\pi} \sum_{k=0}^{N-1} \Delta(k)$$

Where

$$\Delta(k) = \begin{cases} \delta(k) & |\delta(k)| < \frac{\pi}{2} \\ \delta(k) + \pi & \delta(k) \leq -\frac{\pi}{2} \\ \pi - \delta(k) & \delta(k) \geq \frac{\pi}{2} \end{cases}$$

$$\delta(k) = \theta(x_{(k+1) \bmod N}, y_{(k+1) \bmod N}) - \theta(x_k, y_k).$$

Où

θ, donne la direction de n'importe quel pixel de l'image, k est l pas.

❽ Détection de points singuliers :

Si Poincaré (i, j) = 0.5, le point central est présent à l'emplacement (i, j).

Si Poincaré (i, j) = - 0.5, le point delta est présent à l'emplacement (i, j).

Si Poincaré (i, j) = 1, un double point principal est présent.

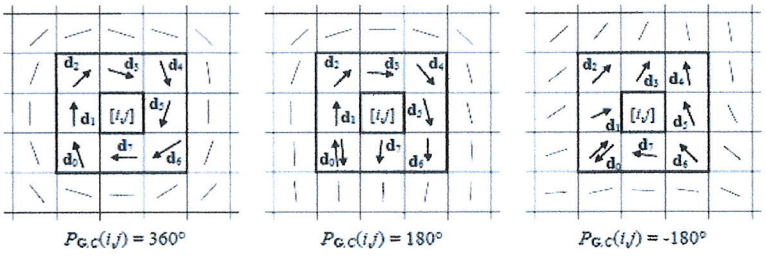


FIGURE 3.11 – Exemples de calcul de l'indice de Poincaré dans le voisinage de 8 points appartenant (de gauche à droite) à une singularité de verticille, boucle et delta, respectivement. Notez que, pour les exemples de boucle et de delta (centre à droite), la direction de d_0 est d'abord choisie vers le haut (pour calculer l'angle entre d_0 et d_1) puis successivement vers le bas (pour calculer l'angle entre d_7 et d_0).

3.3.4 Décomposition elliptique

Pour rendre le hachage résistant aux images d'empreinte digitale, nous divisons l'image normalisée dans cette étape en différents ellipses.
Nous avons vu que chaque empreinte digitale a un point singulier global qui est le centre (core). Nous concéderons que le core de l'empreinte est le centre des ellipses, et nous faisons la décomposition elliptique.
n le nombre d'ellipse et R_k c'est l'ensemble des pixels des minuties détecter du K-ième ellipses tel que $K = (1, 2, ...n)$.

Pour tracer un ellipse nous avons besoin de :

- De deux rayons ou le premier rayon r_1 doit être le double de deuxième rayon $r_2 = r_1 * 2$.
- Déterminer les coordonnées de point central c'est le core d'empreinte(c_x,c_y).
- A partir du core nous mesurons les deux axes (rayons).
- L'angle entre le grand axe et l'axe des x.

Les rayons des ellipses se calculent par la manière suivante :

- Nous avons n nombre d'ellipses $\kappa = (1, 2, ..., n)$.
- Nous détectons le contour d'empreinte, et à partir de ce contour nous obtenons la longueur et la largeur de cette dernière.
- Nous calculons le $rayon_n$ à partir de la largeur $(larg)$ d'empreint comme suit :

$$R = \frac{larg}{2}$$

R est le r_1 c'est le premier rayon d'ellipse n où $rayon_n = (r_1; (r_1 * 2))$.

Autrement, pour calculer le $rayon_k$ où $k \in \kappa$, nous calculons le PAS de chaque rayon :

$$PAS = \frac{R}{n}\ ;\ r_k = PAS * k$$

Donc : $rayon_k = ((r_k); (r_k * 2))$

L'appartenance des minuties :

Cette approche doit vérifier si un point de minutie se trouve dans une ellipse, en fonction du centre, de la largeur, la hauteur et de l'angle de l'ellipse. nous trouvons les coordonnées x et y du point de minutie par rapport au centre de l'ellipse, puis nous les transformons à l'aide de l'angle en coordonnées le long des axes majeur et mineur. Enfin, nous trouvons la distance normalisée du point à partir du centre de la cellule.
➢ Nous avons les cordonnés suivants :

- *angle* : est l'angle d'ellipse.
- (c_x, c_y) : le centre d'ellipse.
- (r_1, r_2) : sont les rayons d'ellipse i.

$$cos_{angle} = cos(radians(180 - angle))$$

$$sin_{angle} = sin(radians(180 - angle))$$

$\forall X \in \Bbbk$ où $\Bbbk$ est l'ensemble des minuties $\{X_1, X_2, ..., X_n\}$ et $X_i = (x_i, y_i)$ est la position des minuties numéro i.
Pour chaque minuties (x_i, y_i), nous calculons :

$$xc = x_i - c_x$$

$$yc = y_i - c_y$$

Autrement :

$$xcT = xc * cos_{angle} - yc * sin_{angle}$$

$$ycT = xc * sin_{angle} - yc * cos_{angle}$$

Donc :

$$S = \frac{(xcT)^2}{\frac{r_{i1}{}^2}{2}} + \frac{(ycT)^2}{\frac{r_{i2}{}^2}{2}}$$

$$\begin{cases} Si\ S = 1 \text{ Alors } X_i = (x_i, y_i) \text{ est sur l'ellipse.} \\ Si\ S \prec 1 \text{ Alors } X_i = (x_i, y_i) \text{ est à l'intérieur de l'ellipse.} \\ Si\ S \succ 1 \text{ Alors } X_i = (x_i, y_i) \text{ est à l'extérieur de l'ellipse.} \end{cases}$$

Pour calculer les minuties qui sont dans l'ellipse (i+1), il faut soustrais l'ellipse i et les minuties qui lui appartient.

3.3.5 Application de l'entropie

L'entropie est un concept de base de la théorie de l'information[64], c'est une mesure de son contenu moyen.

Nous avons i nombre d'ellipse, où chaque ellipse contient N nombres des minuties dans des pixels déférents. Laisser e_i être un événement, $p(e_i)$ soit "le nombre de minuties dans une couleur d'ellipse sur le nombre de minutie d'un ellipse", et elle nous donne sa probabilité d'occurrence, et E un ensemble se formant par $e_i (i = 1, 2, ..N)$. Ainsi, l'entropie de E peut être définie comme suit :

$$H(E) = -\sum_{i=1}^{N} p(e_i) \log_2 p(e_i) \tag{3.2}$$

Où $p(e_1) + p(e_2) + ... + p(e_N) = 1$. Pour les images numériques, l'entropie peut être utilisée pour caractériser la texture de l'image. Dans ce cas, E est l'image d'entrée, e_i est une valeur

de pixel de chaque minutie, et $p(e_i)$ est la probabilité de e_i se produire dans l'image d'entrée pour une image en niveau de gris, $e_i \in [0, 255]$ et le $N = 256$. Dans ce travail, nous prenons l'entropie de chaque ellipse comme une caractéristique et nous l'utilisons pour former un hach d'image. Ceci est basé sur le fait que les entropies en ellipse sont approximativement modifiées linéairement par des opérations préservant le contenu. Soit h_l le $l - ième$ élément du hach h. Ainsi, il peut être calculé par :

$$h_l = H(R_l)(l = 1, 2, 3, ..., n) \tag{3.3}$$

Clairement, la longueur de hach est égale au nombre d'ellipse. Plus le nombre d'ellipses est petit, plus la longueur de hach est courte. Cependant, peu d'ellipse signifient peu de caractéristiques, ce qui nuira inévitablement à la capacité de discrimination. Par conséquent, nous devons garder un équilibre entre la longueur de hach et la capacité de discrimination dans le choix du numéro d'ellipse.

3.3.6 Mesure de similarité

Étant donné que les valeurs de hachage sont modifiées de manière approximativement linéaire, nous exploitons le coefficient de corrélation pour mesurer la similarité de hachage. Tel que : $h^{(1)} = h_1^{(1)}, h_2^{(1)}, .., h_n^{(1)}$ et $h^{(2)} = h_1^{(2)}, h_2^{(2)}, .., h_n^{(2)}$ être deux hachs. Ainsi, le coefficient de corrélation est défini comme :

$$S = \frac{\delta_{h^{(1)},h^{(2)}}}{\delta_{h^{(1)}}\delta_{h^{(2)}} + \varepsilon} \tag{3.4}$$

Où ε est une petite constante pour éviter la singularité lorsque $\delta_{h^{(1)}}\delta_{h^{(2)}} = 0$, $\delta_{h^{(1)}}$ et $\delta_{h^{(2)}}$ sont l'écart type de $h^{(1)}$ et $h^{(2)}$, et $h^{(1)}$,$h^{(2)}$ est leur covariance calculée par l'équation suivante :

$$\delta_{h^{(1)},h^{(2)}} = \frac{1}{n-1}\sum_{i=1}^{n}[h_i^{(1)} - \mu_1][h_i^{(2)} - \mu_2] \tag{3.5}$$

Où μ_1 et μ_2 sont les moyens de $h^{(1)}$ et $h^{(2)}$, respectivement. L'intervalle de S est $[-1, 1]$. Plus les images d'entrée sont similaires, plus la valeur S est proche de 1 [63]. Si S est supérieur à un seuil prédéfini T_{seuil}, les images d'entrée sont considérées comme des images visuellement identiques. Sinon, elles sont visualisées sous forme d'images différentes.

3.4 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 proposée, qui repose sur la construction d'un hach perceptuel qui doit être robuste et sure en utilisant l'application de l'entropie et la décomposition elliptique. La performance de la méthode proposée est évaluée en l'appliquant sur une database, où les résultats obtenus seront analyser sous plusieurs critères, chose qui sera abordé dans le chapitre suivant.

Chapitre 4

Tests et résultats expérimentaux

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 d'entropie et décomposition elliptique", 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 d'empreinte digitale en appliquant certaine manipulation acceptable comme la rotation, le bruit, la compression, nous mettons en évidence les raisons de nos choix technique, les testes sur l'application et les résultats obtenue.

Nous résumons cette mise en œuvre en trois parties :

➢ Environnement et outils de développement.

➢ Présentation de l'application.

➢ Analyse et interprétation des résultats.

4.2 Environnement et outils de développement

4.2.1 Langage de programmation

Nous avons utilisés Python 2.7 comme langage de programmation, parmi les raisons de cette utilisation :

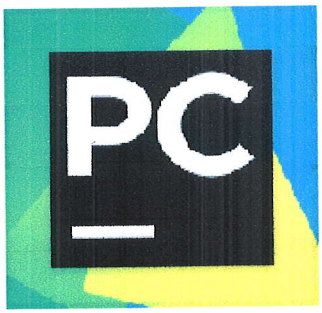
✓ Python fonctionne sur différentes plateformes (Windows, Mac, Linux).

✓ Il a une syntaxe simple claire, respecte les standards du domaine. similaire à la langue anglaise.

✓ Ce langage peut être traité de manière procédurale, de manière orientée objet ou de manière fonctionnelle.

4.2.2 Environnement de programmation

Nous allons utilisés la version gratuite de PyCharm Communauté pour le développement pur en Python, est un environnement de développement intégré (IDE) utilisé en programmation informatique, spécialement pour le langage Python. Il est développé par la société tchèque JetBrains. ces caractéristiques :



- ✓ PyCharm permet de compléter le code de manière intelligente, d'inspecter le code, ainsi que de refactoriser le code automatiquement et offre des fonctionnalités de navigation avancées.

- ✓ PyCharm comprend notre projet en profondeur, pas seulement les fichiers individuels.

- ✓ PyCharm s'intègre à IPython Notebook, dispose d'une console Python interactive et prend en charge des nombreux packages scientifiques, notamment matplotlib et NumPy.

4.2.3 Bibliothèque

Une des grandes forces du langage Python réside dans le nombre important de bibliothèques logicielles externes disponibles. Une bibliothèque est une ensemble de fonctions. Celles-ci sont regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire.

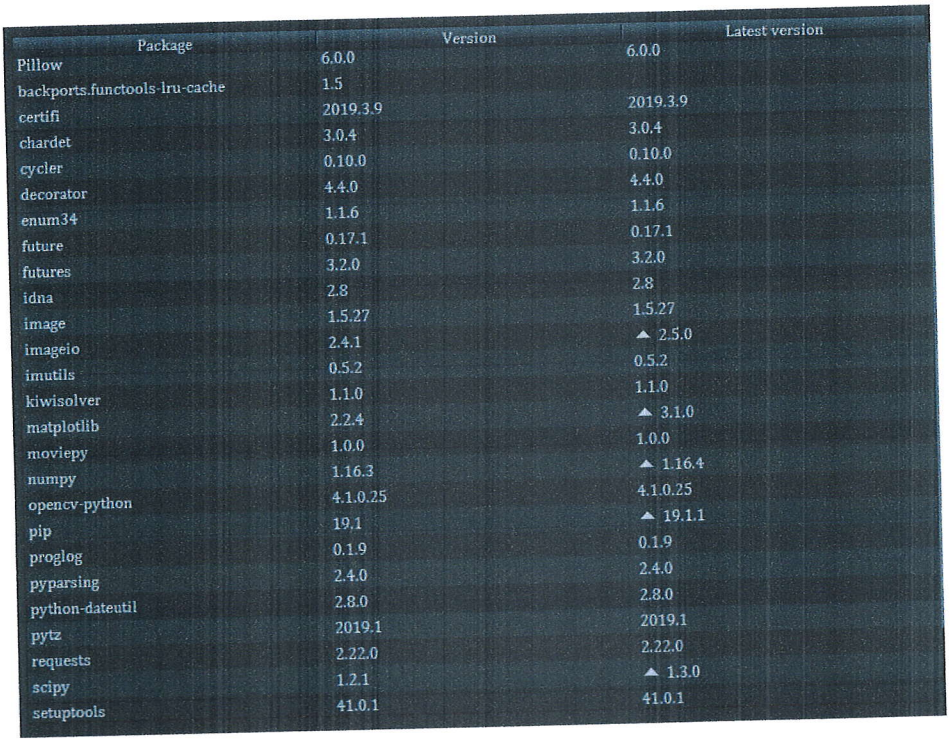
Celles-ci permettent de faire : du calcul numérique, du graphisme, de la programmation internet ou réseau, du formatage de texte, de la génération de documents...
La figure 4.1 montre les différentes bibliothèques utilisés dans notre application.

Module PIL :
La bibliothèque PIL (Python Imaging Librairie) permet la 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.

OpenCv :
Cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données.



Package	Version	Latest version
Pillow	6.0.0	6.0.0
backports.functools-lru-cache	1.5	
certifi	2019.3.9	2019.3.9
chardet	3.0.4	3.0.4
cycler	0.10.0	0.10.0
decorator	4.4.0	4.4.0
enum34	1.1.6	1.1.6
future	0.17.1	0.17.1
futures	3.2.0	3.2.0
idna	2.8	2.8
image	1.5.27	1.5.27
imageio	2.4.1	▲ 2.5.0
imutils	0.5.2	0.5.2
kiwisolver	1.1.0	1.1.0
matplotlib	2.2.4	▲ 3.1.0
moviepy	1.0.0	1.0.0
numpy	1.16.3	▲ 1.16.4
opencv-python	4.1.0.25	4.1.0.25
pip	19.1	▲ 19.1.1
proglog	0.1.9	0.1.9
pyparsing	2.4.0	2.4.0
python-dateutil	2.8.0	2.8.0
pytz	2019.1	2019.1
requests	2.22.0	2.22.0
scipy	1.2.1	▲ 1.3.0
setuptools	41.0.1	41.0.1

FIGURE 4.1 – Bibliothéques utilisés.

4.2.4 Caractéristique de la plateforme

Nous allons utilisés deux machines avec les caractéristiques suivantes :

Machine 1	**Machine 2**
Processeur 1.80 GHz Intel core i3-3217U **RAM** : 4.00 GO **Carte graphique** :Intel(R) HD Graphics 4000 **Système d'exploitation** : Windows 7 64 bits	**Processeur** : 2.50 GHz Intel Core i5-2450M **RAM** : 4.00 GO **Carte graphique** : Intel(R) HD Graphics 4000 **Système d'exploitation** : Windows 7 64 bits

TABLE 4.1 – Caractéristique des machines utilisés

4.3 Présentation de l'application

Le but de notre travail est de tester la performance de notre algorithme (hachage d'image robuste à base d'entropie et la décomposition elliptique) contre quelques attaques acceptables, sur une base d'image d'empreinte digitale. Grâce à la corrélation linéaire que nous avons choisi comme métrique de mesure, nous pouvons mesurer la performance de cet algorithme.

4.3.1 DataBase

Pour évaluer la méthode proposée dans ce mémoire nous l'avons appliqué sur certain images (104 images) de la base **FVC 2002 Set A**[65], cette base a été recueillies à l'aide d'appareils de balayage en direct utilisant trois scanners différents : un optique à basse résolution (256 dpi), un CMOS (zone de balayage plus petite de (0.6×0.8)) et un scanner optique complet (1×1) à échelle de gris, la résolution de ces images est (388 × 374) ainsi que son format est TIFF, en outre il existe dans cette base de données huit échantillons par doigt ce qui permet aux algorithmes de comprendre les variations entre les doigts correspondants. La figure suivante montre une partie de cette base de données :

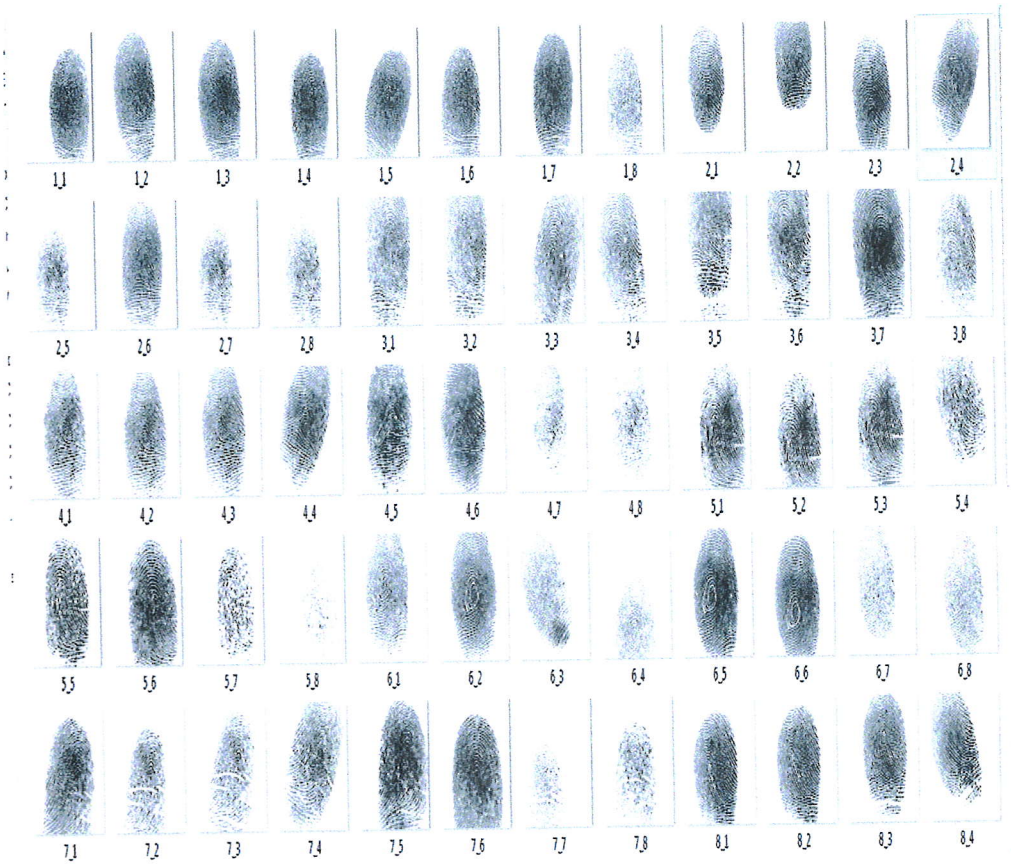


FIGURE 4.2 – Une partie de la base de données utilisée.

4.3.2 Implémentation et Interface graphique

Au lancement de l'exécution de l'application, la fenêtre suivante est apparue :

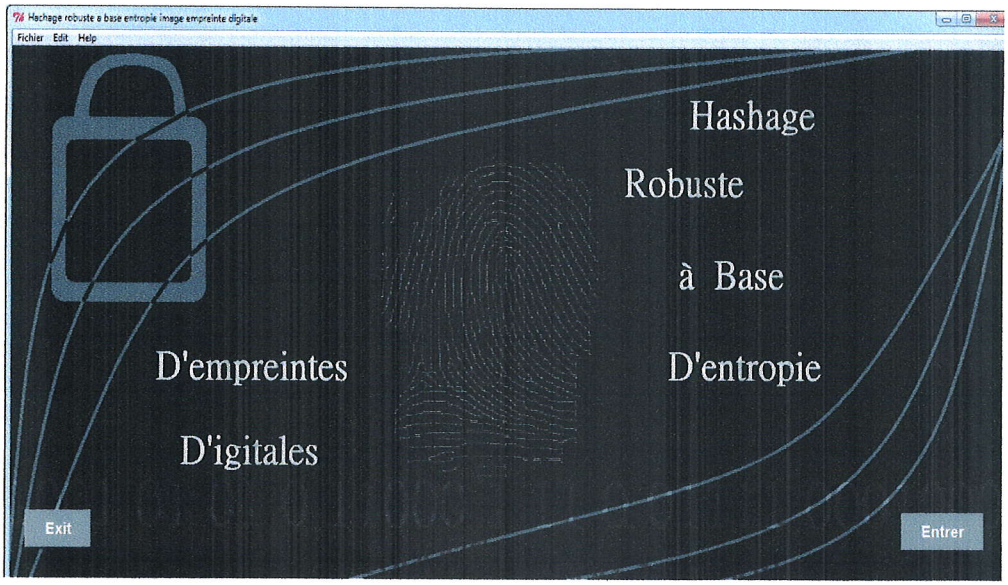


FIGURE 4.3 – La représentation de la page d'accueil.

❒ 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.

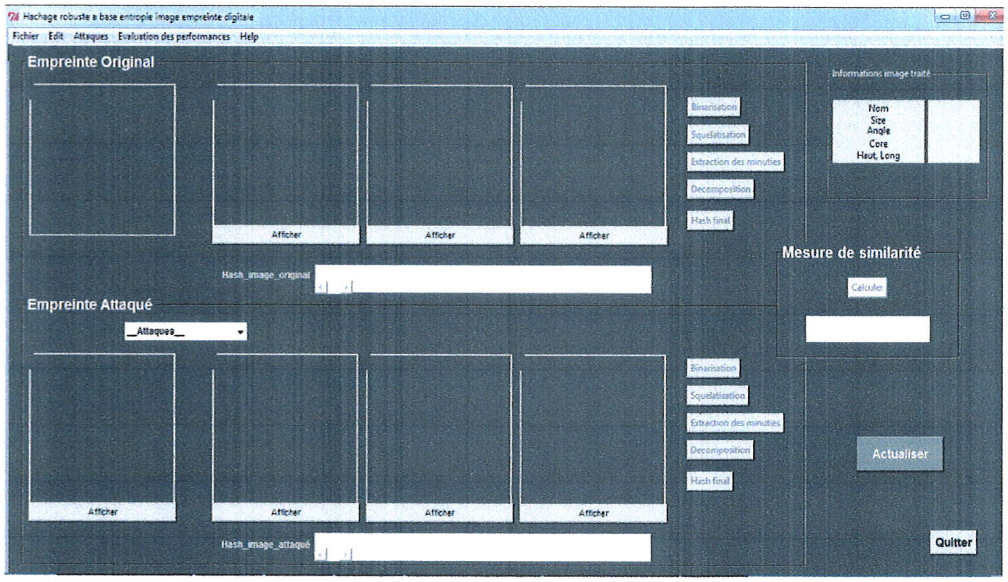
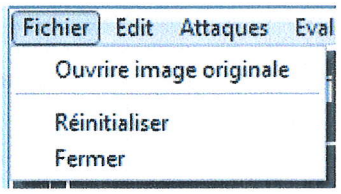


FIGURE 4.4 – La représentation de la page principale.

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

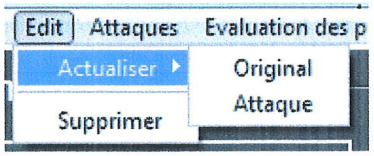
1) Fichier : contient les opérateurs suivants :
➢ Ouvrir image originale : Permet d'ouvrir une image pour la traité, et à partir de cette action le bouton binarisation de la partie d'image originale sera activé.



➢ Réinitialiser : permet de Réinitialiser touts les composants de l'interface graphique.

➢ Fermer : Permet de quitter l'interface.

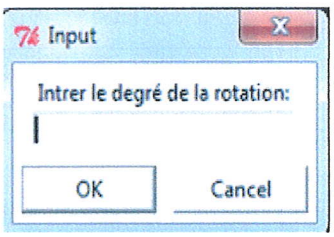
2) Edit : contient les opérateurs suivants :
➢ Actualiser Originale : permet de Réinitialiser tous les composants de la partie graphique d'image originale.



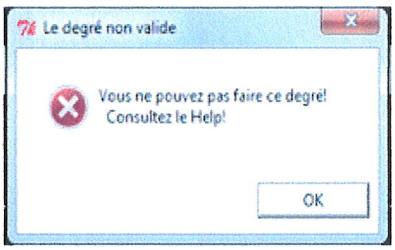
➢ Actualiser Attaque : permet de Réinitialiser tous les composants de la partie graphique d'image attaquée .

➢ Supprimer : permet de supprimer tous les dossiers qui sauvegarde les informations des images traitée.

3) Attaques : Après avoir ouvrir l'image originales, 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. Lorsque nous choisissons une attaque, la fenêtre 1 sera affichée pour saisir son paramètre, et si le paramètre est faux la fenêtre 2 sera affichée.



Fenêtre 1



Fenêtre 2

4) **Évaluation des performances :** permet d'afficher la fenêtre pour le calcule des résultats. Elle est décomposée sur deux pages, une pour la "Robustesse" et la deuxième pour le "Taux" et la "Discrimination".

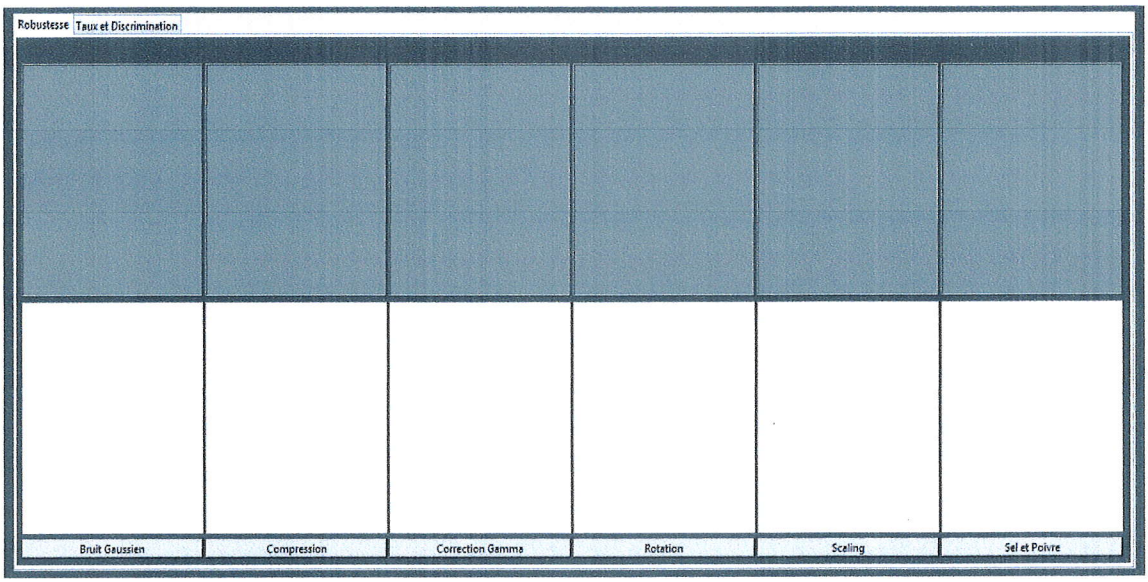


FIGURE 4.5 – Fenetre pour l'evaluation des performances (Robustesse) .

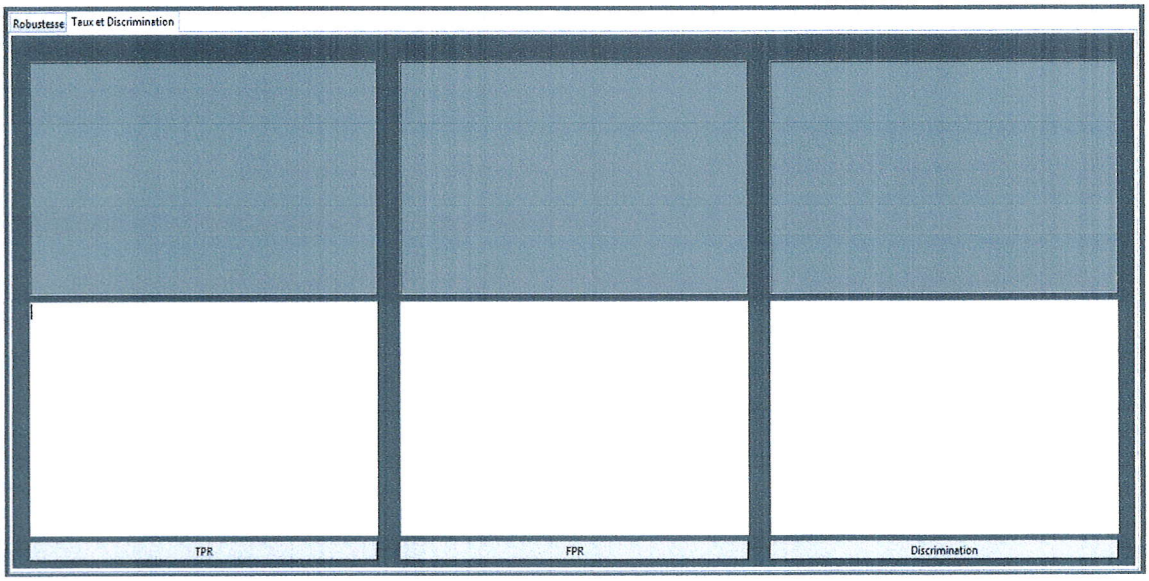
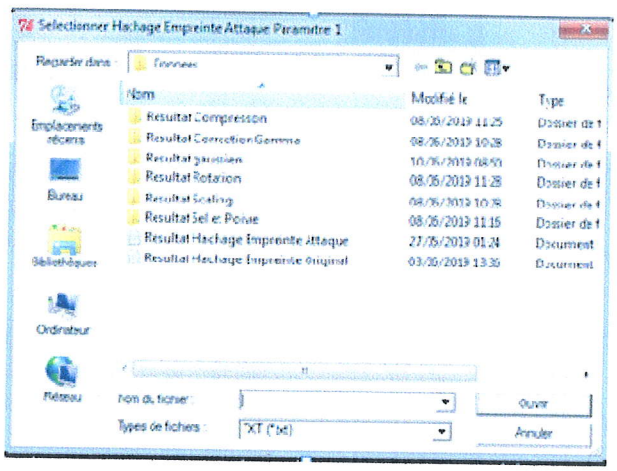


FIGURE 4.6 – Fenetre pour l'evaluation des performances (Taux et discrimination) .

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 (TPR, FPR et discrimination), nous obtenons cette figure pour sélectionner les fichier des résultats que nous voulons voir et elle changera d'un bottons à un autre.



La figure 4.7 montre le résultat obtenu de la robustesse :

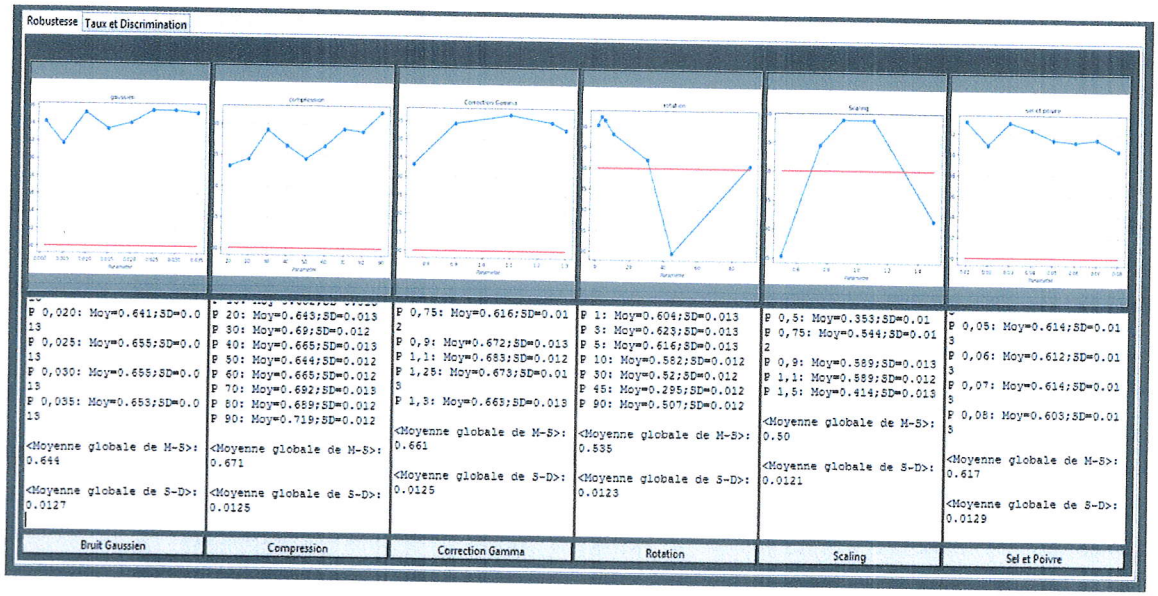


FIGURE 4.7 – Resultat de Fenetre Robustesse.

La figure 4.8 montre le résultat obtenu de TPR FPR et de la discrimination :

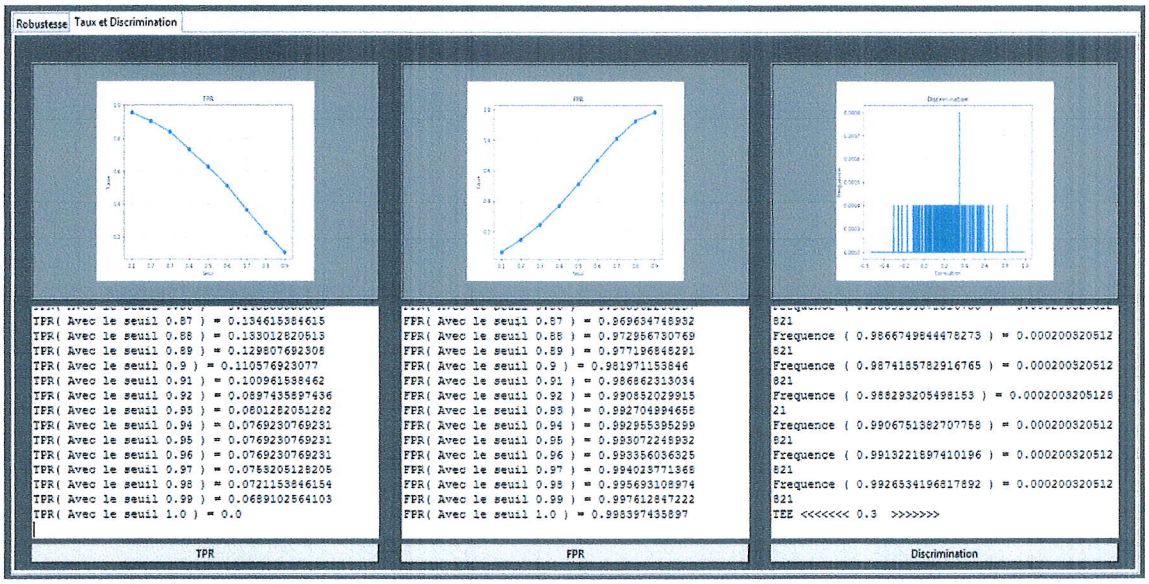


FIGURE 4.8 – Resultat de la Fenetre Taux et décrimination.

5) Help : pour donner une idée générale sur le but d'application et tous les paramètres utilisés sur les attaques. la figure 4.9 montre cette fenêtre de "Help".

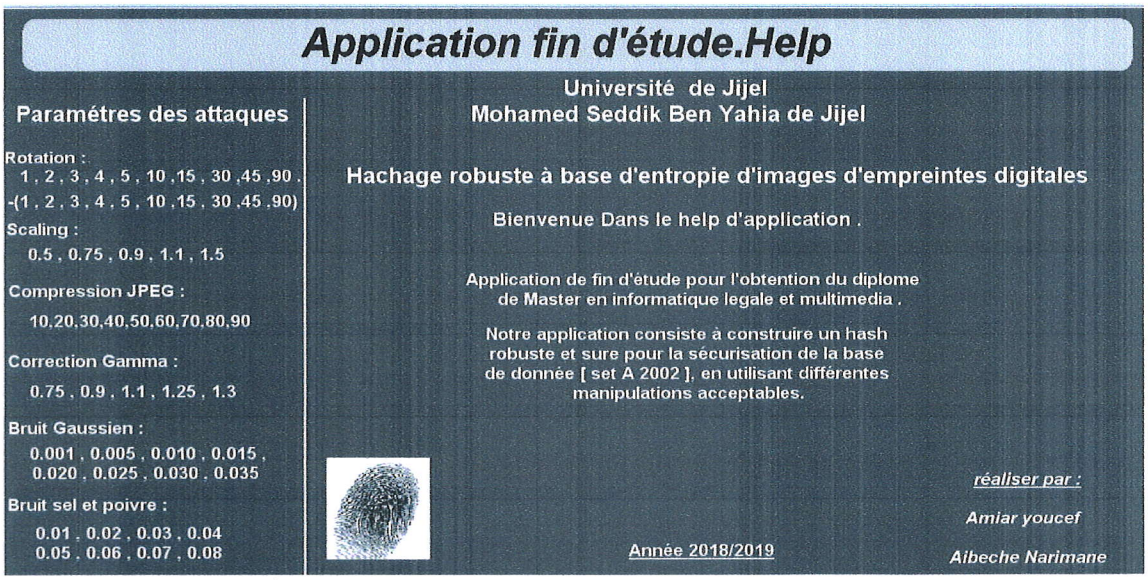


FIGURE 4.9 – Le Help de l'application.

❒ L'interface graphique de la figure 4.4 est composée de trois parties (image originale, image attaquée, et mesure de similarité). Pour les deux premières étapes nous avons les boutons suivants :

Bouton Binarisation : 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 de cliquer sur ce bouton pour obtenir l'image binarisée.

Bouton Squelettisation : il sera actif que lorsque l'image binarisée est affichée, il suffit de cliquer sur ce bouton pour obtenir l'image squelattisée.

Bouton Extraction des minuties : il sera actif que lorsque l'image squelettisée est affichée, il suffit de cliquer sur ce bouton pour obtenir l'ensemble des minuties extraits sur l'image squelettisée.

Bouton décomposition : il sera actif que lorsque les minuties sont extraits, il suffit de cliquer sur ce bouton pour obtenir la décomposition elliptique .

Bouton hach Final : il sera actif que lorsque la décomposition est faite, il suffit de cliquer sur ce bouton pour obtenir le hach final .

Bouton Afficher : permet d'afficher les images de chaque traitement.

La figure suivante montre les résultats obtenus de chaque bouton :

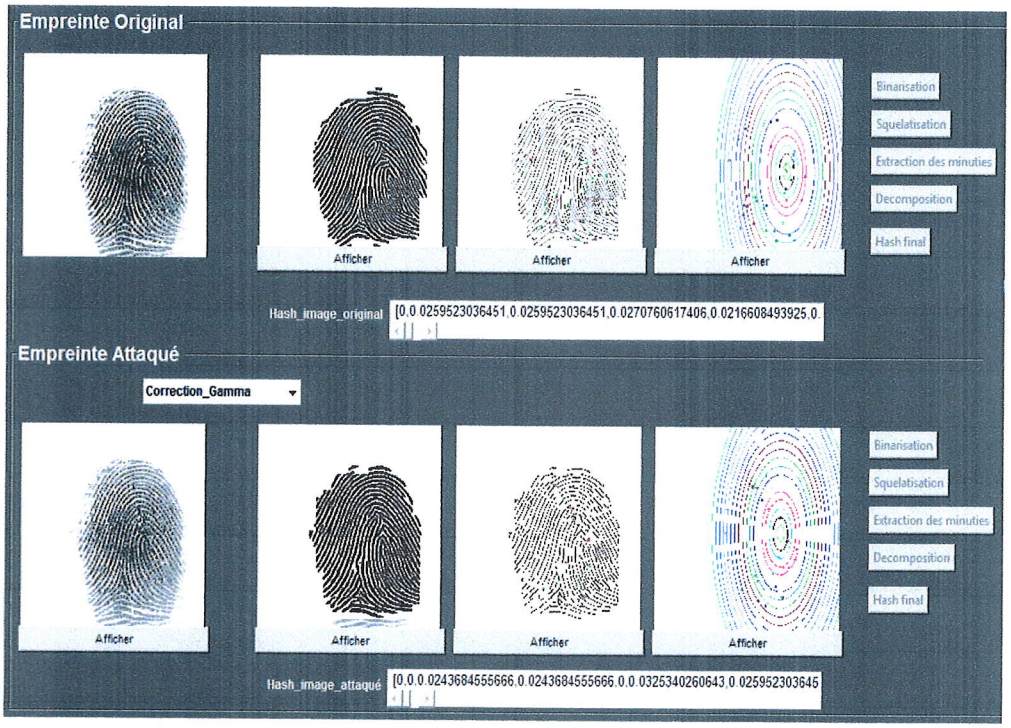


FIGURE 4.10 – Les résultats obtenus de la partie d'image originale et la partie d'image attaquée.

- Pour chaque image originale et pour chaque image attaquée, les résultats obtenus doit

être sauvegardés dans un dossier renommé par le nom d'image pour image originale et renommé par le nom d'image et l'attaque choisi pour l'image attaquée.

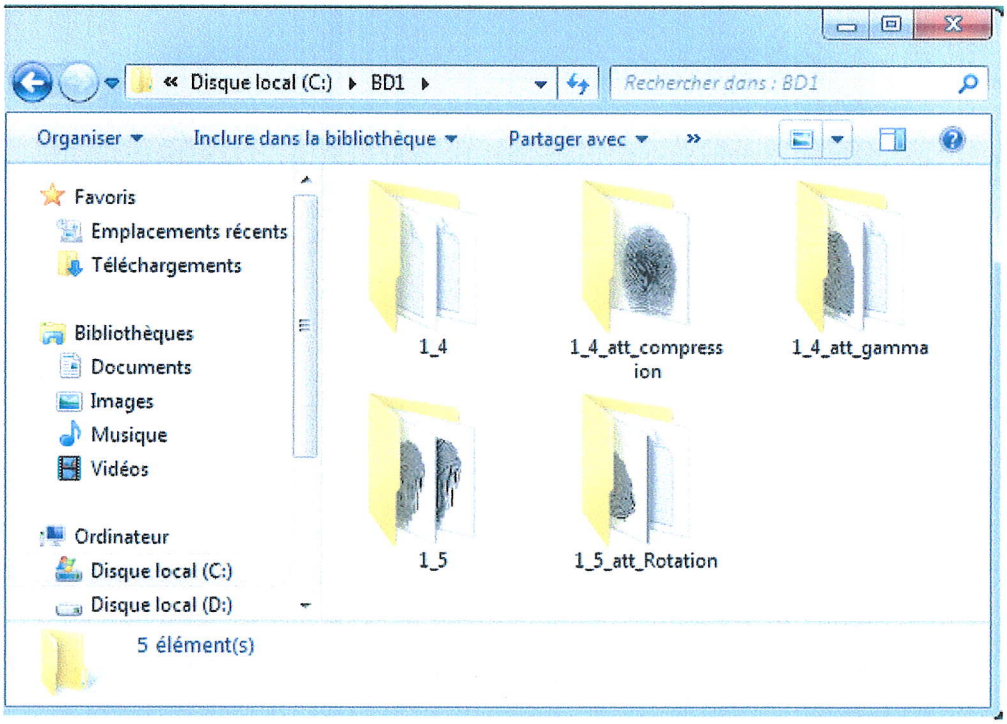
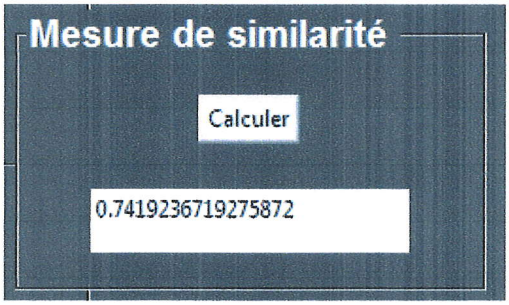
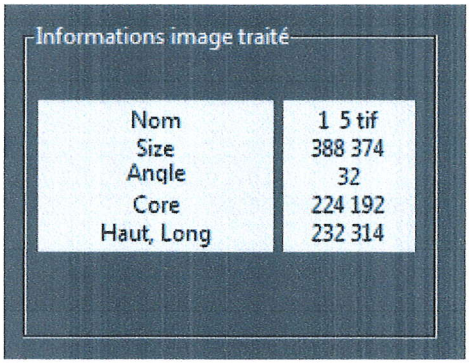


FIGURE 4.11 – Les résultats savaugardés pendant le traitement.

❒ Pour la partie de la mesure de similarité nous avons :



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



❒ Dans cette partie, nous affichons les informations sur l'image qui sera traitée.

4.3.3 Les attaques acceptables utilisés

Dans ce test, nous appliquons quelques manipulations acceptables tel que la rotation la compression les deux bruits (gaussien et sel et poivre) la correction gamma et scaling. Le tableau 4.2 montre les paramètres de chaque manipulation acceptable.

Attaque	Paramètre							
Rotation	± 1	±3	± **5**	± 10	± 30	45	± 90	
Scaling	0.5	**0.75**	0.9	1.1	1.5			
Compression JPEG	**10**	20	30	40	50	60	70	80	90
Correction Gamma	0.75	0.9	1.1	1.25	**1.3**			
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 – Les paramètres utilisés pour chaque manipulation

➢ Pour calculer le TPR et FPR, nous appliquons les attaques présenter dans le tableau 4.2 sur la base d'image (104 empreintes), en utilisant le paramètre en gras pour chaque attaque.

4.4 Analyse est interprétation des résultats

Il n'existe pas une méthode, qui remplit tous les critères d'une méthode parfaite, car lors de l'implémentation de ces méthodes nous sommes toujours confronté à un compromis, où chaque méthode a ces inconvénients et ces avantages, l'équilibrage entre les points forts et les points faibles dépend du domaine d'application.

4.4.1 La robustesse perceptuelle

Pour évaluer la robustesse de notre méthode contre les différentes manipulation 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 et l'écart type. Notre teste est appliqué sur 104 images choisies de la base FVC set A, puis nous comparons les résultats avec notre seuil choisi (0.5).

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

Compression :

Attaque	Paramètre	Moyenne-mesure	Écart-Type
Compression	10	0.632	0.013
	20	0.643	0.013
	30	0.69	0.012
	40	0.665	0.013
	50	0.644	0.012
	60	0.665	0.012
	70	0.692	0.013
	80	0.689	0.012
	90	0.719	0.012

TABLE 4.3 – La moyenne de similarité et l écart-type pour chaque paramètre de la compression.

Rotation :

Attaque	Paramètre	Moyenne-mesure	écart-type
Rotation	1	0.604	0.013
	3	0.623	0.013
	5	0.616	0.013
	10	0.582	0.012
	30	0.52	0.012
	45	**0.295**	0.012
	90	0.507	0.012

TABLE 4.4 – La moyenne de similarité et l'écart-type pour chaque paramètre de la Rotation.

Scaling :

Attaque	Paramètre	Moyenne-mesure	Écart-Type
Scaling	0.5	**0.353**	0.01
	0.75	0.544	0.012
	0.9	0.589	0.012
	1.1	0.589	0.013
	1.5	**0.414**	0.012

TABLE 4.5 – La moyenne de similarité et l'écart-type pour chaque paramètre de Scaling.

Correction-Gamma :

Attaque	Paramètre	Moyenne-mesure	Écart-Type
Correction-Gamma	0.75	0.616	0.012
	0.9	0.672	0.013
	1.1	0.683	0.012
	1.25	0.673	0.013
	1.3	0.663	0.013

TABLE 4.6 – La moyenne de similarité et l'écart-type pour chaque paramètre de Correction-Gamma.

Bruit-Gaussien :

Attaque	Paramètre	Moyenne-mesure	Écart-Type
Bruit-Gaussien	0.001	0.642	0.013
	0.005	0.617	0.013
	0.010	0.653	0.013
	0.015	0.634	0.013
	0.020	0.641	0.013
	0.025	0.655	0.013
	0.030	0.655	0.013
	0.035	0.653	0.013

TABLE 4.7 – La moyenne de similarité et l'écart-type pour chaque paramètre de Bruit-Gaussien.

Bruit-Sel-Poivre :

Attaque	Paramètre	Moyenne-mesure	Écart-Type
Bruit-Sel-Poivre	0.01	0.632	0.013
	0.02	0.609	0.013
	0.03	0.631	0.013
	0.04	0.623	0.013
	0.05	0.614	0.013
	0.06	0.612	0.013
	0.07	0.614	0.013
	0.08	0.603	0.013

TABLE 4.8 – La moyenne de similarité et l'écart-type pour chaque paramètre de Bruit-Sel-Poivre.

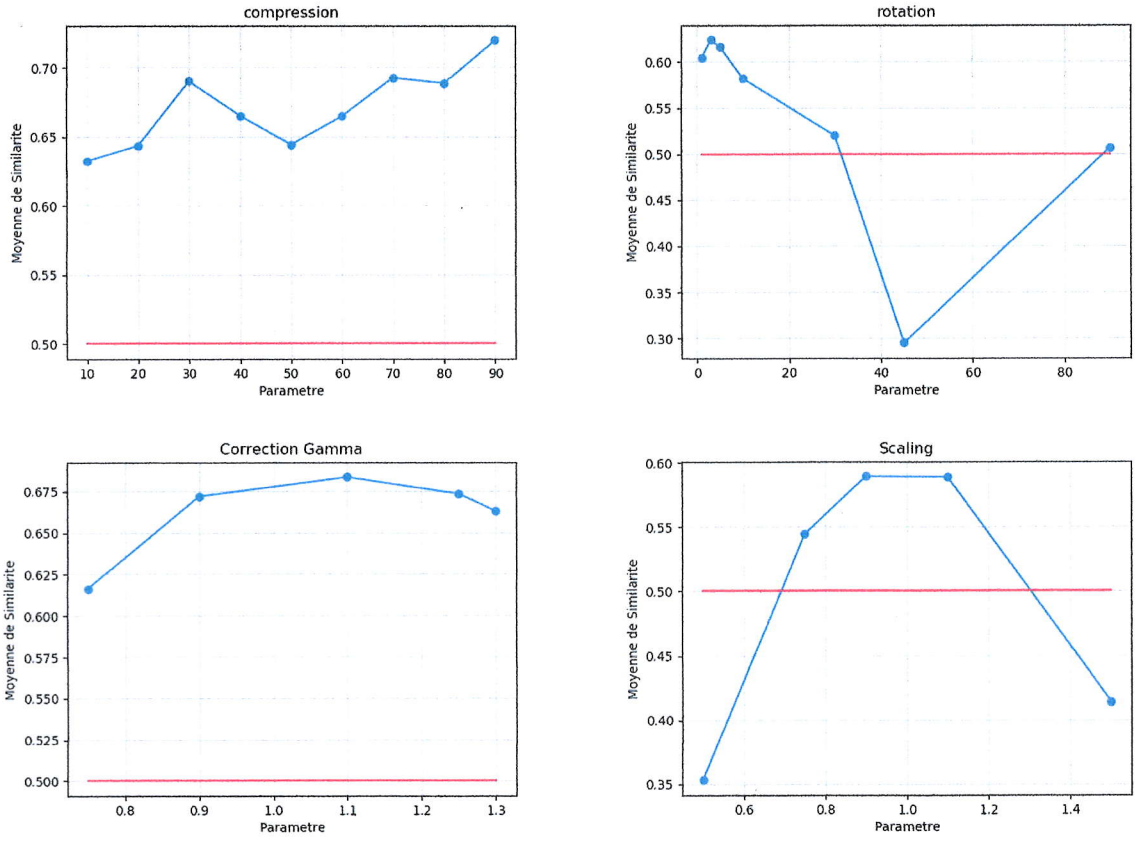
• La moyenne de mesure de similarité générale et l'écart-type globale pour toutes les attaques sont illustré dans le tableau suivant :

Attaques	Moyenne de similarité	Écart-Type
Compression	0.671	0.0125
Rotation	0.535	0.0123
Scaling	0.50	0.0121
Correction-Gamma	0.661	0.0125
Bruit-Gaussien	0.644	0.0127
Bruit-Sel-Poivre	0.617	0.0129

TABLE 4.9 – La moyenne générale de la mesure de similarité et l'écart-type globale pour chaque attaque.

• Pour voir comment la corrélation linaire évolue avec la variation des différentes manipulations, nous avons représentés les résultats sous forme des courbes.

- Les figures suivantes montre les différentes courbes représentatives de la robustesse de notre méthode :



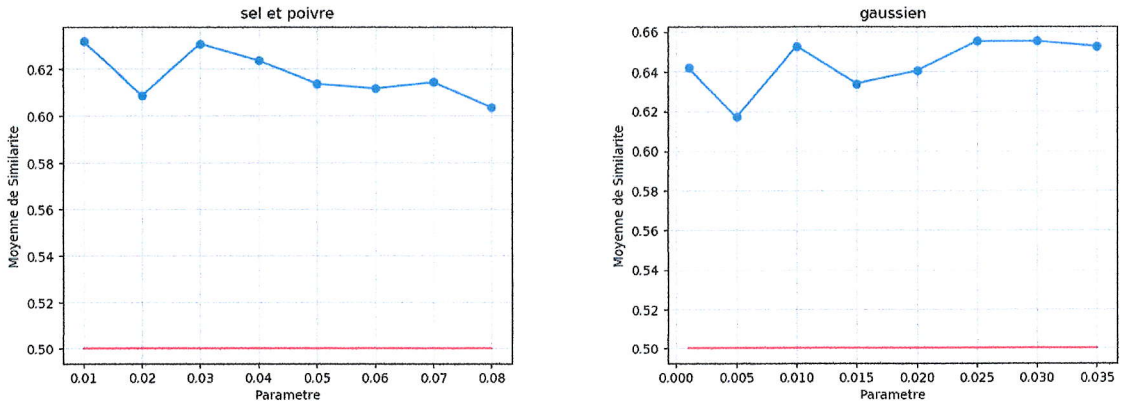


FIGURE 4.12 – Évaluation de la robustesse par les différentes manipulations acceptables

➢ Les résultats expérimentaux montrent que notre méthode est avantageux à la robustesse vue que la moyenne de mesure de similarité calculée entre le hach des images originaux et le hach des images infectée par des attaques acceptables est supérieure à 0.5 pour la totalité des manipulations, sauf dans le cas de la rotation 45, de scaling 0.5 et 1.5 ne donne pas des bons résultats (Moyenne de similarité $\prec$ 0.5), ce qui implique que la robustesse de cette méthode n'est pas si parfaite envers le degré 45 de la rotation et les deux degrés de scaling (0.5, 1.5). - Les résultats montre aussi que la moyenne de mesure pour toutes les attaques est supérieure à 0.5, et différents écarts types de 0.0121 à 0.0129, donc la méthode proposée est robuste face à la totalité des attaques acceptable.

4.4.2 La capacité de discrimination

Pour tester la capacité de discrimination de notre méthode, nous calculons la mesure de similarité entre chaque image originale d'empreinte digitale et les autres images originaux qui n'appartient pas aux empreintes de la même personne, les résultats sont représentés dans l'histogramme 4.13 où l'axe des X sont les valeurs de la mesure de similarité obtenu et l'axe des Y la fréquences d'apparition de chaque valeur.

La méthode proposée est appliquée à 104 images différentes d'empreintes digitales de la même base de données, et la distance entre chaque paire de hach est calculée pour obtenir 9984 résultats, comme indiqué dans la Figure 4.13.

- Les résultats obtenus sont illustré dans l'histogramme suivant :

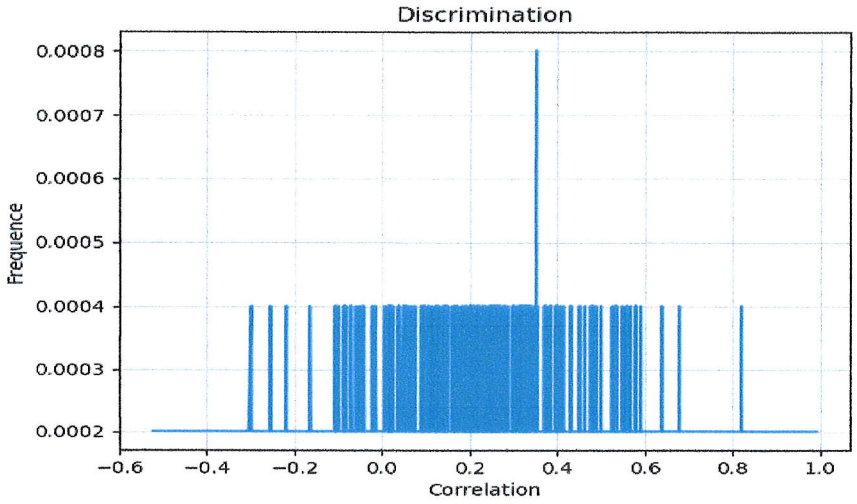


FIGURE 4.13 – Evaluation de la discrimination.

➢ Notre seuil défini est 0.5, d'après l'histogramme nous obtenons un taux d'erreur pour les images différentes qui sont fausse identifiée comme des images similaire égale à **TEE = 0.3**. Vu que le taux d'erreur est faible et la figure montre que l'histogramme se situe presque complètement au dessous du seuil 0.5, alors les résultats expérimentaux montre que notre méthode est discriminante, ce qui indique que la méthode proposée est capable de distinguer entre différentes images avec un pourcentage de 70%.

4.4.3 TPR et FPR

Pour visualiser les performances de classification d'algorithme évalué, où le True positifs rate (TPR) et False positifs rate (FPR) sont respectivement des indicateurs de l'endurance et de la capacité discriminante. Ils sont définis comme suit :

❒ Pour calculer le **TPR** (True positifs rate), nous testons la mesure de similarité entre le hach de chaque image d'empreinte originale et les hashs de cette image avec un paramètre standard pour chaque attaque, donc nous calculons la corrélation linéaire entre le hach d'une image et les six hashs manipulés de cette dernière, nous comparons le résultats avec certains seuils, dans ce cas nous obtenons le nombre de paire d'image visuellement identique considérées comme des images similaire (N similaire). Le nombre totale de paires d'image visuellement identique est 624 (n identique).

❒ Pour calculer le **FPR** (False positifs rate). Nous calculons la mesure de similarité entre le hach d'un échantillons d'empreinte original est les 96 hashs attaqués (nous ne comparons pas avec les échantillons de la même personne), les résultats obtenus nous lui comparons avec les différents seuils, nous obtenons le nombre de paires d'images différentes considérées comme des images similaires(N distinct) pour chaque seuils. Le nombre total de

paires d'images différentes est 59904 (n différent).

• Le tableau 4.10 montre les résultats obtenu pour le P_{TPR} et P_{FPR} pour des différents seuils :

Les seuils	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
P_{TPR}	0.958	0.907	0.841	0.732	0.629	0.512	0.367	0.229	0.110
P_{FPR}	0.066	0.146	0.245	0.369	0.471	0.666	0.809	0.922	0.981

TABLE 4.10 – Les résultats de P_{TPR} et P_{FPR} par rapport au différents seuils

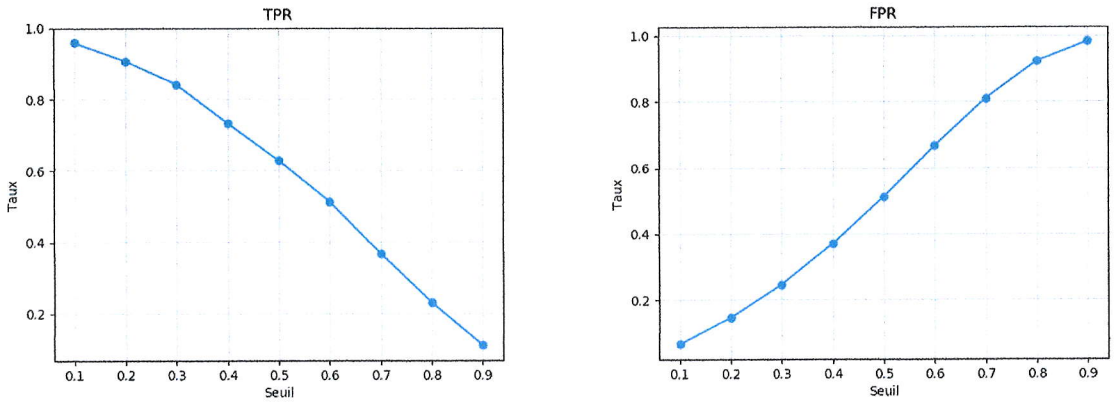


FIGURE 4.14 – Evaluation de TPR et FPR par rapport aux différentes seuils.

➢ Les résultats montrent que le TPR et FPR calculer se change d'un seuil à un autre, en outre dans notre méthode le résultat avec une valeur proche à 1 est meilleur que celui avec un petit TPR, de même, la méthode avec un petit FPR est plus performante que avec un grand FPR. ce qui implique que notre méthode est performante pour le seuil 0.5 car le TPR et FPR est acceptable.

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 d'empreintes digitales, nous avons arrivé à conclure que notre méthode est robuste contre quelques attaques acceptables et elle est discriminante contre les différentes empreintes qui n'appartient pas à la même personne. Également nous concluons que notre méthode ne donne pas des bons résultats dans le cas de la rotation 45, scaling 0.5 et 1.5. Ce qui implique que la robustesse de cet algorithme n'est pas si parfaites envers ces trois paramètres. Les résultats expérimentaux ont montré que notre hachage est robuste contre les manipulations qui préservent le contenu normal.

Conclusion générale et perspective

L'objectif principale de ce projet était l'application de l'une des approches de sécurité des informations visuelles les plus sophistiquées, à savoir le hachage perceptuel afin d'améliorer la sécurité des images d'empreinte digitale.
Le projet réalisé dans ce mémoire a abouti à la création d'une application permettant d'appliquer un algorithme de hachage perceptuel à base d'entropie et la décomposition elliptique sur les images d'empreinte digitale pour garantir leur sécurité contre les différentes modifications acceptable comme la rotation scaling correction gamma compression bruit gaussien et bruit sel et poivre.

Nous avons vu les étapes utilisées dans le système de reconnaissance biométrique d'empreinte digitale 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.
Le troisième chapitre a été consacré à une description détaillée de notre méthode de hachage perceptuel d'image d'empreinte digitale à base d'entropie et la décomposition elliptique, où nous avons expliqué ses différentes phases.

Enfin, nous avons essayé d'illustrer l'importance de cette technique en montrant les résultats de l'algorithme de hachage par l'utilisation de la corrélation linéaire qui est basé sur la similarité de contenu visuelle entres les images d'empreinte digitale, afin de construire un hach qui permet d'assuré la robustesse et la discrimination de ces images. Nous conclu que notre méthode appliqué dans ce mémoire est robuste et sur contre quelques attaques acceptables comme la compression, bruit sel et poivre bruit gaussien correction gamma à l'exception de la rotation et scaling.

A l'issue de ce travail, les perspectives suivantes peuvent être proposées pour poursuivre les recherche dans ce domaine :
➢ Prendre en compte d'autres types d'attaques pour analyser les signatures perceptuelles.
➢ Étudier la robustesse à d'autres types de caractéristiques extraites.
➢ Améliorer le temps d'exécution.

Bibliographie

[1] L. Hong A.K. Jain and S. Pankanti. Biometrics : Promising frontiers for emerging identification market comm. acm, pp. 91-98. (February 2000).

[2] J.-L. Dugelay et al. *Recent Advances in Biometric Person Authentication, IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP).* Orlando, Florida, (May 2002.).

[3] International Biometric Group. The henry classification. `www.biometricgroup.com`.

[4] FBI. Federal bureau of investigation. `homepage:www.fbi.gov`.

[5] biométrie. biométrie online. `http://www.biometrie-online.net`, (Consulté le 1/12/2018).

[6] L. Hong A.K. Jain and S. Pankanti. A prototype hand geometry-based verification system, in proc. of 2nd int l conf. on audio and video-based biometric person authentication, pp. 166 171. (March 1999).

[7] P.J. Phillips W. Zhao, R. Chellappa and A.Rosenfeld. Face recognition : A literature survey, acm computing surveys (csur), volume 35, issue 4. (December 2003).

[8] W.A. Barrett. *A survey of face recognition algorithms and testing results, Conference Record of the Thirty-First Asilomar Systems Computers, pp. 301-305.* Conference on Signals, (1997.).

[9] R.P. Wildesr. *Iris Recognition : An Emerging Biometric Technology , Proceedings of the IEEE, Volume 85, Issue 9, pp. 1348 -1363.* (Sept 1997).

[10] S.J. Vaughan-Nichols. *Voice authentication speaks to the marketplace.* Computer, Volume : 37, Issue 3, pp. 13-15, (March 2004).

[11] Ibrahim Henawy, Magdy Rashad, Omaima Nomir, and Kareem Ahmed. Online signature verification : State of the art. *INTERNATIONAL JOURNAL OF COMPUTERS TECHNOLOGY*, (2013).

[12] Fedias Meriem. *Combinaisons de données d'espaces couleurs et de méthodes de vérification d'identité pour l'authentification de visages.* Université Mohamed Khider Biskra, (2013).

[13] Dang Hoang Vu. *Biométrie pour l'identification, Rapport final, Institut de la Francophonie pour l'informatique.* Hanoï, Vietnam, (07, 2005).

[14] Barka Khaled Boukhris Youcef. *Système d'identification biométrique à base d'un modèle flou.* Université KasdiMerbah Ourgla, (2015/2016).

[15] F. Perronnin and J.-L.Dugelay. *Introduction à la biométrie - Authentification des individus par traitement audio-vidéo. Traitement du signal, Vol. 19, No. 4.* (2002).

[16] Morpheus From The Matrix. the history of fingerprint. `http://onin.com/fphistory.html`.

[17] Francis Galton. *Fingerprint, McMillan.* London, (1892).

[18] S.Prabhakar. *Fingerprint classification and Matching using filterbank.* these de Doctorat en Informatique, Université de Michigan, (2001).

[19] Anil K. Jain SalilPrabhakar MaltoniDavide, Dario Maio. *Handbook of fingerprint recognition.* Springer, New York, (2003).

[20] X. Tang X. Tong, J. Huang and D. Shi. *Fingerprint minutiae matching using the adjacent feature vector.* Pattern Recognition Letters, (2005).

[21] Anil Jain et al. *Automated Fingerprint Identification and imaging.* (2000).

[22] Francis Galton. *Fingerprint.* McMillan, London, (1892).

[23] H. Ailisto and M. Linholm. A reviex of fingerprint image enhancement methods. *Image and Graphics, Vol. 3, No. 3, pp. 401-424, 200,* (2003).

[24] H. Ailisto and M. Linholm. A reviex of fingerprint image enhancement methods. *Image and Graphics, Vol. 3, No. 3, pp. 401-424, 200,* (2003).

[25] Venugopal.K.R** Ravi.J*, K.B.Raja**. fingerprintre cognition using minutiae score matching. *Journal of Engineering Science and Technology, Vol.1(2), 35-42,* (2009.).

[26] M. SezginetB.Sankur. "survey over image thresholding technique and quantitative performance evaluation". *Journal of Electronic Imaging, vol. 13, p. 146-165,* (2004).

[27] S.W. Lee et C.Y. Suen L.Lam. Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, vol. 9, p. 869-885,* (1992).

[28] Arcelli C. and Baja G.S.D. A width independent fast thinning algorithm. *IEEE Transaction on pattern Analysis and Machine Intelligence, vol. 4, no. 7, pp. 463-474,* (1984).

[29] N.Galy. *Etude d'un système complet de reconnaissance d'empreintes digitales pour un capteur microsystème à balayage, p 80.* (2005).

[30] J. Kim S.Kim, D.Lee. *Algorithm for Detection and Elimination of False Minutiae in Fingerprint Image, Lecture Notes in Computer Science.* Spring Verlag, vol. 2091, p. 235-240, (2001).

[31] W.Shu Z.Bian, D.Zhang. Knowledge-based fingerprint post-processing. *International Journal of Pattern Recognition and artificial Intelligence, vol.16, no.1,p. 53-67,* (2002).

[32] A. Leone A. Farina, Z.M. Kovacs-Vajna. *Fingerprint minutiae extraction from skeletonized binary images.* Pattern Recognition,vol.32, p. 877-889, (1999).

[33] Francesco Turroni. *Fingerprint Recognition : Enhancement, Feature Extraction and Automatic Evaluation of Algorithms.* dottoratodiricerca in InformaticaCiclo XXIV Universita di Bologna, (2012).

[34] D. Maltoni D. Maio. *Direct Gray-Scale Minutiae Detection In Fingerprints.* IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 1, p. 27-39, (1997).

[35] A. Rosenfel R.Stefanelli. *Some parallel thinning algorithms for digital pic.* Journalof the ACM, Vol.18, No2, pp.255-264, (April 1971).

[36] The Prisoners G. J. Simmons. *Problem and the Subliminal Channel. In CRYPTO.* pages 51.67, (1983).

[37] S. Katzenbeisser et F. A. Petitcolas. *Hiding Techniques for Steganography and Digital Watermarking.* Artech House, Inc, Norwood, MA, USA, 1st édition, ISBN1580530354, (2000).

[38] Vincent Martin. *Contribution Des Filtres Lptv Et Des Techniques D'interpolation Au Tatouage Numérique.* (2006).

[39] 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, (September 2007).

[40] C. Paul. Oorschot A. Scott Menezes, J. Alfred. Vanstone. handbook of applied cryptography.boca raton. *CRC Press*, (1998).

[41] Brahim Ait Es Said Azhar Hadmi, William Puech and Abdellah Ait Ouahman. *Perceptual Image Hashing, Watermarking -Volume 2.* Dr. Mithun Das Gupta (Ed.), ISBN : 978-953-51-0619-7, InTech, DOI : 10.5772/37435, (2012).

[42] Azhar HADMIA. *protection des données visuelles Analyse des fonctions de hachage perectuel.* Laboratoire d'informatique de robotique et de Microelectronique de Montpellier, (26 octobre 2012).

[43] Ton Kalker Jaap Haitsma and Philips Job Oostveen. *Robust Audio Hashing for Content Identification.* Research International Workshop on Content-Based Multimedia Indexing (CBMI'01), (2001).

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

[45] Brahim Ait Es Said William Puech, Azhar Hadmi and Abdellah Ait Ouahman. *Perceptual Image Hashing, Watermarking - Volume 2.* Dr. Mithun Das Gupta (Ed.), ISBN : 978-953-51-0619-7, InTech, DOI : 10.5772/37435, (2012).

[46] Yong Wang. *New Way to Construct Cryptographic Hash Function.* (16 February 2014).

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

[48] Ricardo Antoio Parrao Hernandez and Brian M. Kurkoski. *Robust Image Hashing Using Image Normalization and SVD Decomposition.* IEEE Transactions on Information Forensics and Security, (10 August 2011).

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

[50] F. Gu B.Yang and X. Niu. *Block mean value based image perceptual hashing.* Proceedings of the International Conference on Intelligent Information Hiding and Multimedia

Multimedia Signal Processing (IIH- MSP), 172. IEEE, 2006, ISBN 0- 7695-2745-0 p. 167.

[51] Christoph Zauner. *Implementation and Benchmarking of Perceptual Image Hash Functions.* eingereicht am Fachhochschul-Masterstudiengang Sichere Informationssysteme, in Hagenberg, (Juli 2010).

[52] Anand Rajaraman Leskovec, Jure and Jeffrey D. Ullman. *Mining of Massive Datasets.* Stanford : Stanford University, (2010).

[53] V. MONGA and B EVANS. *Perceptual image hashing via feature points : Performance evaluation and tradeoffs. Image Processing, IEEE Transactions on 15, 11.* (Nov 2006).

[54] D.G Lowe. *Distinctive image features from scale invariant key-points. Int. J. Comput. Vis, 60, 91,110.* (2004).

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

[56] Yachun Feng Hong Zhang Hao Chen Ding Yuan and Helong Wang. *Visual tracking via multi-experts combined with average hash model.* (2015).

[57] Fourier-Mellin. *Transform for Robust Image Hashing.* Fourth International Conference on Emerging Security Technologies, (2013).

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

[59] Vishal Monga and Brian L. Evans. Perceptual image hashing via feature points : Performance evaluation and tradeoffs. (Novembre 2006).

[60] Xu Ming Lan Jin Bo, Tang Hua Ping. Fingerprint singular point detection algorithm by poincaré index. (December 2008).

[61] V. Govindaraju S.Chikkerur, N.Cartwright. *Fingerprint image enhancement using STFT analysis IEEE Proceedings in ICAPR, pp. 20-29.* (2005).

[62] Christophe. [En ligne] Bernard. la mesure du flot optique et l'interpolation irrégulière. `http://www.cmap.polytechnique.fr`.

[63] Liyan Huang Yumin Dai Rajaraman Zhenjun Tang*, Xianguan Zhang. *Robust image hashing using ring-based entropies.* Departement of Computer Science,Guangxi Normal University, Guilin 541004 PR China, (2013).

[64] C.E. Shanno. A mathematical theory of communication. *Bell System Technical Journal 27 379.423, 623.656,* (1948).

[65] AK Jain et S. Prabhakar D. Maltoni, D. Maio. Fingerprint verification competition 2002 (fvc2002). `http://bias.csr.unibo.it/fvc2002`, Springer, Londres, 2009.

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 à authentifier. Les manipulations acceptables (compression JPEG, bruit Gaussien, rotation...) préservent l'aspect visuel de l'image à authentifier, par contre, les manipulations malicieuses (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 les données d'empreintes digitales.

Ce travail vise à la construction d'un hache perceptuel robuste à base d'entropie qui repose sur une décomposition elliptique de l'image d'empreinte digitale. Ce hache est utile pour protéger le Template des données biométriques (empreintes digitales) sauvegardé dans la base des données.

Mots clés : Empreint digitale, Hachage perceptuel, Integrité, Robustesse, Discrimination, Entropie, Mesure de similarité.

ABSTRACT

Perceptual hash functions are strongly inspired by cryptographic hash functions, they are based on the visual aspect of the data to be hashed, allowing a perceptual correspondence to be established between the original image and the image to be authenticated. Acceptable manipulations (JPEG compression, Gaussian noise, rotation, etc.) preserve the visual aspect of the image to be authenticated, but malicious manipulations (adding new objects, deleting or major modification of existing objects, for example) change the semantic content of the image. In recent years, many researchers have focused on this new approach to multimedia data security and fingerprint data.

This work aims to build a robust entropy-based perceptual axe that is based on a circular decomposition of the fingerprint image. This axe is useful to protect the Biometric Data Template (fingerprints) saved in the database.

Keywords : Fingerprint, Perceptual hashing, integrity, Robustness, Discrimination, Entropy, Similarity measurement.