RÉPUBLIQUE DU BÉNIN

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :
Godwine Papin **Houngavou**

Détection de l'usurpation d'identité sur les systèmes de reconnaissance faciale

Sous la supervision :
Lionel **Metongnon**, PhD Student

Année Académique : **2017 - 2018**

Table des matières

Liste des figures

Liste des Algorithmes

Sigles et Abréviations

API : Application Programming Language 14

HTTP : HyperText Transfert Protocol 15

OpenCV : Open Computer Vision 20
OTP : One-Time Password 7

SMS : Short Message Service 7
SQL : Structured Query Language 20

UML : Unified Modeling Language 20

Glossaire

API :

c'est un ensemble de fonction, de classes normalisées et d'outils utilisés pour la conception des applications. Une API spécifie comment les diverses fonctions sont manipulées par le programme client. Les APIs peuvent être utilisées pour créer des programmes d'interface graphique. L'objectif d'une API est de fournir une porte d'accès à une fonctionnalité en cachant les détails de la mise en œuvre à travers des blocs de code pré-faits. 14

biométrie :

Analyse mathématique des caractéristiques biologiques d'une personne, destinée à déterminer son identité de manière irréfutable. La biométrie repose sur le principe de la reconnaissance de caractéristiques physiques. Les empreintes digitales, et la gamme d'indices généralement visés par la biométrie, notamment l'iris, la rétine, la main et les empreintes vocales, offrent une preuve irréfutable de l'identité d'une personne puisqu'elles constituent des caractéristiques biologiques uniques qui distinguent une personne d'une autre et ne peuvent être associées qu'à une seule personne. 8

pirate informatique :

C'est un criminel informatique qui exploite les failles dans une procédure d'accès pour casser un système informatique, qui viole l'intégrité de ce système en dérobant, altérant ou détruisant de l'information, ou qui copie frauduleusement des logiciels. 4

reconnaissance faciale :

Un système de reconnaissance faciale est une application logicielle visant à reconnaître une personne à partir des caractéristiques de son visage de manière automatique. Ces systèmes sont généralement utilisés à des fins de sécurité pour confirmer une identité, mais aussi en domotique. 10

système d'information :

Un système d'Information (noté SI) représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein d'une entreprise. x

token :

Jeton en français, un token est un utilitaire matériel ou logiciel qui permet de générer des numéros ou des mots en fonction d'un algorithme précis, et donc permet une authentification sur des infra- structures informatiques. 7

usurpation d'identité :

L'usurpation d'identité consiste à utiliser, sans votre accord, des informations permettant de vous identifier dans un système d'information. Il peut s'agir, par exemple, de vos nom et prénom, de votre adresse électronique, ou encore de photographies (dans le cas de notre étude). Ces informations peuvent ensuite être utilisées à votre insu, pour commettre des actes répréhensibles ou nuire à votre réputation. 13

Dédicace

A
mon Père **Dominique HOUNGAVOU**
ma Mère **Clarisse KOUKE**
mes frères et soeurs

Remerciements

Nous exprimons notre vive gratitude aux personnes physiques et institutions qui ont contribué à rendre meilleur notre travail, notamment :

- **M. Lionel METONGNON**, notre Maître de mémoire et enseignant à l'Institut de Formation et de Recherche en Informatique (IFRI),
- **M. Djalil A. ASSOUMA** , Directeur Général de Bénin Télécoms Infrastructures, pour nous avoir accordé un cadre adéquat pour le stage académique,
- **M. Adrian ROSEBROCK** pour son aide
- **M. Salem AFFA** pour son aide
- Tout le corps professoral de l'**IFRI**, pour nous avoir transmis toutes les connaissances possibles que nous devrions avoir.

Que tous ceux qui nous ont aidé, de près ou de loin, trouvent ici l'expression de nos sentiments les meilleurs.

Résumé

La reconnaissance automatique des personnes est une technique de plus en plus utilisée avec l'augmentation constante des besoins en sécurité et les évolutions technologiques. Plusieurs méthodes sont utilisées pour authentifier des utilisateurs au sein d'un système d'information. Elle se fait par l'utilisation des mots de passe, des cartes à puce et aussi par des données biométriques. La reconnaissance faciale composante de la biométrie, est de plus en plus utilisée pour effectuer l'authentification au sein des systèmes d'information. Le visage fournit des informations sur un individu et permet de le reconnaître de façon unique. De part son taux d'utilisation, elle est devenue la cible de l'usurpation d'identité des utilisateurs légitimes. Ces différentes attaques utilisant des faiblesses des techniques utilisées pour reconnaître et authentifier un utilisateur, affaiblissent le taux de fiabilité des systèmes de reconnaissance faciale. Au regard de ces constats, nous avons proposé une solution pour pallier aux différentes attaques sur la reconnaissance faciale. La solution proposée dans un premier temps, fait la reconnaissance de l'utilisateur et dans un autre s'assure qu'il ne s'agit pas d'un cas d'usurpation d'identité de l'utilisateur à travers une photo ou un masque du visage de l'utilisateur ou encore une vidéo contenant le visage de l'utilisateur. La solution que nous nous proposons est une API libre qui regroupe un ensemble de fonctions qui permet aux développeurs d'implémenter de meilleurs systèmes de reconnaissance faciale dans leur processus d'authentification.

Mots clés : Biométrie, reconnaissance faciale, usurpation d'identité, API

Abstract

Automatic people recognition is a technique that is increasingly used with the constant increase in security needs and technological developments. Several methods are used to authenticate users within a information system. It is done through the use of passwords, smart cards and also biometric data. Facial recognition, a component of biometrics, is increasingly used to perform authentication within information systems. The face provides information about an individual and makes it possible to recognize him in a unique way. Due to its high usage rate, it has become the target of identity theft by legitimate users. These various attacks, which use weaknesses in the techniques used to recognize and authenticate a user, weaken the reliability rate of system of facial recognition. In view of these findings, we proposed a solution to overcome the various attacks on facial recognition. The proposed solution first recognizes the user and then ensures that it is not a case of usurpation of the user's identity through a photo or mask of the user's face or a video containing the user's face. The solution we propose is a free API that brings together a set of functions that allows developers to implement better facial recognition systems in their authentication process.
Key words: Biometrics, facial recognition, identity theft, API

Introduction

Le contrôle d'accès, défini comme l'ensemble des solutions ou techniques qui permettent de sécuriser et gérer les accès physiques à un bâtiment, ou les accès logiques à un système d'information, est indispensable pour l'évolution de toute entreprise. En effet, ces solutions se veulent être plus efficaces dans leur fonctionnement notamment en matière d'authentification. Plusieurs techniques ont donc été créées dans le but d'assurer une bonne authentification des utilisateurs. Les techniques d'authentification les plus utilisées sont entre autres, les mots de passe, les badges et cartes à puce, les certificats numériques. Suite à plusieurs problèmes de sécurité rencontrés (le cracking des mots de passe, le vol des badges, l'oubli des mots de passe longs...) avec ces méthodes d'authentification, il a été donc décidé l'utilisation des données biométriques dans les processus d'authentification. Considéré comme un facteur d'authentification de haute sécurité, la reconnaissance faciale (technique utilisant une composante de la biométrie), fait objet de cas d'usurpation d'identité. Avec l'avènement des médias sociaux et la recherche mondialisée, les images et vidéos de visages sont largement répandues sur Internet et peuvent être utilisées pour attaquer ces systèmes biométriques. Il est nécessaire de développer des applications plus sécurisées afin de réduire les cas d'usurpation d'identité.

0.1 Contexte et justification

Le contrôle des accès aux différentes ressources d'une entreprise constitue un point essentiel dans la sécurité de son infrastructure informatique. Ce contrôle d'accès passe par des systèmes d'authentification. Parmi ces systèmes d'authentification, la biométrie plus précisément la reconnaissance faciale parait mieux adaptée aux besoins grandissantes en matière de sécurité des données. Ceci est dû au haut niveau de sécurité qu'elle offre et à sa facilité d'accès. Mais ce dernier fait objet d'attaques qui visent à réduire son taux de fiabilité en visant l'usurpation des identités d'utilisateur légitime. Il est donc nécessaire de renforcer la sécurité au tour de ces systèmes de reconnaissance faciale. Le présent projet vise à développer une API qui se servira à, détecter l'usurpation d'identité sur les systèmes de reconnaissance faciale. Cette API sera libre et gratuite.

0.2 Problématique

La sécurité des infrastructures revêt une importance capitale pour le bon fonctionnement des entreprises. Dans cette grande famille d'outil d'authentification, nous avons, les systèmes de reconnaissance faciale qui sont sujets à des attaques visant à les tromper en exploitant des failles [2]. Ces systèmes sont souvent la cible de plusieurs techniques d'usurpation d'identité utilisant trois principaux types d'attaques connus dans ce domaine pour tromper la reconnaissance faciale :

- **les attaques photos :** en effet, une bonne méthode est de montrer une photo de l'utilisateur légitime à la caméra afin de tromper le système de reconnaissance faciale.
- **les attaques vidéos :** ce deuxième type d'attaque consiste à présenter une vidéo contenant un visage légitime d'un utilisateur au système de reconnaissance faciale qui considère ce dernier comme une personne légitime devant l'objectif de la caméra.
- **les attaques avec un masque :** la récente mise à disposition au grand public de l'impression tridimensionnelle crée également une faille pour les systèmes de reconnaissance faciale. En effet, cela permet de créer un masque 3D correspondant au visage de l'utilisateur.

Ces différentes attaques affaiblissent le taux de fiabilité de la reconnaissance faciale en touchant un point essentiel de la sécurité informatique. Cela nous amène donc à remettre en cause les systèmes utilisés pour effectuer la reconnaissance du visage. Quelles sont les mesures actuelles qui sont prises pour améliorer le taux de fiabilité de la reconnaissance faciale ? Les techniques actuelles utilisées sont elles performantes ? Sont-elles libre pour une implémentation par des développeurs ?

0.3 Objectifs

Notre objectif principal est de réaliser une API libre capable de détecter les tentatives d'usurpation d'identité sur les systèmes de reconnaissance faciale. Ainsi, nous voudrions renforcer la sécurité et réduire les intrusions frauduleuses dans les systèmes d'information utilisant la reconnaissance faciale. Plus précisément, il s'agira :

- de vérifier la source du flux de vidéo afin d'éviter les redirections de flux de vidéos en entrée de notre API ;
- de détecter le clignement des yeux et les micros mouvement dans l'environnement du sujet afin d'éviter les attaques photos ;

- d'analyser la texture du visage reçu depuis la source du flux de vidéo afin de savoir s'il s'agit d'une reproduction du visage de l'utilisateur légitime ;
- de détecter des objets dans l'environnement immédiat de l'utilisateur afin de détecter les cas de contrainte de soumission du visage sous menace d'un objet dangereux.

0.4 Environnement de stage

Notre stage de fin de cycle a été effectué au sein de **Bénin Télécoms Infrastructures** [2]. En effet, **Bénin Télécoms Infrastructures** est une société anonyme créée depuis 2005 par le décret n°2015- 116 du 10 mars 2015 portant modification statutaire. Elle a pour rôle de mettre à la disposition des autres opérateurs de la Téléphonie les ressources essentielles dont ils ont besoin pour la commercialisation des produits et services des télécommunications. A cet effet, Bénin Télécoms Infrastructures commercialise : les capacités internationales, les capacités nationales, les capacités et liaisons spécialisées numériques, la fibre optique noire, les circuits à fibre optique, les câbles en pairs de cuivres. Elle assure également l'acheminement des trafics d'interconnexions nationales et internationales.

0.5 Organisation du mémoire

Le présent travail est constitué en trois (03) chapitres. Le premier chapitre présente la revue de littérature sur la reconnaissance faciale, l'usurpation d'identité et la présentation des solutions existantes. Le deuxième chapitre met l'accent sur les choix techniques opérés en vue de la conception de notre solution baptisée **"Oudjat"** ainsi que sur le fonctionnement et l'implémentation de notre solution. Le troisième chapitre, quant à lui, fait une analyse critique des résultats issus de nos simulations après les avoir exposés.

[2] http://www.benintelecoms.bj/

Chapitre 1

Revue de littérature

Introduction

La réalisation d'un projet, part d'un état des lieux permettant d'avoir un bref aperçu sur le sujet du projet. Dans ce chapitre, nous ferons un état des lieux du sujet de notre travail. Dans un premier temps, nous ferons une revue de littérature sur le sujet, puis nous présenterons, dans un second temps, quelques solutions existantes.

1.1 Généralités

1.1.1 La sécurité informatique

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles d'une entreprise ou d'une personne permettant de les stocker ou de les faire circuler. Il est donc nécessaire de le protéger contre le pirate informatique

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation ou d'une personne sont utilisés dans le cadre prévu. Elle vise généralement cinq (05) principaux objectifs :

- **L'intégrité :** elle garantit que les données doivent être celles que l'on attend et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets.
- **La confidentialité :** elle consiste à assurer que seules les personnes autorisées aient accès aux ressources échangées.
- **La disponibilité :** le système doit fonctionner sans faille durant les plages d'utilisation prévues et garantir l'accès aux services et ressources installés avec le temps de réponse attendu.

- **La non répudiation :** elle permet de garantir qu'aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.
- **L'authentification :** elle consiste à assurer que seules les personnes autorisées aient accès aux ressources.

1.1.2 Notion de contrôle d'accès à un système d'information

Le contrôle d'accès désigne les différentes solutions techniques qui permettent de sécuriser et gérer les accès physiques à un bâtiment ou un site, ou les accès logiques à un système d'information. On distingue ainsi le contrôle d'accès physique et le contrôle d'accès logique exercés selon deux étapes.

La première étape : consiste dans l'audit et la configuration des droits d'accès attribués aux utilisateurs (on parle de "Gestion des identités et des habilitations" ou "Identity & Access Management").

La deuxième étape : consiste dans le contrôle des droits d'accès attribués aux utilisateurs au moment de l'accès au système.

A propos de l'authentification

L'authentification est une procédure par laquelle un système informatique certifie la demande d'accès faite par une entité (personne ou un ordinateur) afin d'autoriser l'accès de cette entité à une infrastructure [3]. Le but de ce processus est de vérifier l'identité de la personne ou de l'entité qui veut accéder à une ressource, qu'elle soit distante ou non. Pour ce faire, le système d'authentification compare les informations des utilisateurs autorisés stockées dans une base de données (en local ou sur un serveur d'authentification distant) à celles fournies par l'entité faisant la demande. Le processus d'authentification s'exécute avant que les vérifications d'autorisation et de limitation ne se produisent et avant que toutes autres actions ne puissent continuer. Ce processus d'authentification s'exécute suivant plusieurs facteurs qui permettent d'autoriser les accès.

1.1.2.1 Les facteurs d'authentification

Il existe 4 principaux facteurs d'authentification qui peuvent être utilisés dans le processus d'autorisation d'accès à des ressources sécurisées [7] :

- **Ce que l'on connaît (facteur mémoriel) :** une information que l'utilisateur a mémorisée et que lui seul connaît (exemple : un mot de passe, un nom, un code PIN).
- **Ce que l'on possède (facteur matériel) :** une information que seul l'utilisateur possède et enregistrée dans un support dont il est le seul détenteur (exemple : téléphone portable, carte à puce, badges).
- **Ce que l'on est (facteur corporel) :** une information qui caractérise l'utilisateur avec une empreinte qui lui est propre (exemple : biométrie).

- **Ce que l'on sait faire (facteur réactionnel) :** une information ou un acte que seul l'utilisateur peut produire (exemple : une signature).

Il existe également d'autres facteurs d'authentification présentés dans [20] qui s'appuient sur :

- **Où se situe-t-on ? :** la géolocalisation de l'entité qui cherche à s'authentifier peut être un facteur déterminant. Ce facteur permet d'assurer une authentification à partir de lieux définis, ou encore empêcher une nouvelle authentification si la précédente a été réalisée à des milliers de kilomètres au cours de la même heure.
- **A quelle date/heure est-on ? :** Certaines authentifications se voient autorisées qu'en période de travail, à des heures définies. En dehors de ces créneaux, elles ne doivent plus être autorisées.

1.1.2.2 Les méthodes de vérification

À partir des différents facteurs d'identification cités ci-dessus, des méthodes de vérification ont été mises en place pour qualifier le degré d'authentification, c'est-à-dire que l'on peut associer un ou deux facteurs pour renforcer la vérification.
Il existe 2 familles d'authentification présentées dans [20] :

- **L'authentification simple :** c'est un processus d'authentification qui ne requiert qu'un seul élément ou « facteur » d'authentification (exemple : l'utilisateur indique son mot de passe ou son pseudo) [10].
- **L'authentification multifactorielle :** encore appelée authentification à plusieurs facteurs ou authentification forte, elle consiste à effectuer l'authentification en passant par deux facteurs ou plus. Le principe de l'authentification forte est que l'utilisateur doit saisir, en plus de son mot de passe ou son pseudo, une deuxième information que lui seul possède ; on parle d'un deuxième facteur. Cela offre plus de sécurité au niveau de la vérification de l'identité de l'entité qui se connecte à la ressource informatique.

1.1.3 Les méthodes d'authentification

La biométrique est l'analyse des caractéristiques physiques strictement propres à une personne (voix, visage, iris, empreintes digitales...). Avant la biométrie, plusieurs solutions ou méthodes d'authentification existaient. Nous avons :

- **Login/Mot de passe**

  L'authentification par login/mot de passe est une méthode d'authentification pour accéder aux ressources sécurisées. Il s'agit d'une série de caractères utilisés comme moyen d'authentification pour prouver son identité lorsque l'on désire accéder aux infrastructures informatiques. Habituellement, il est associé à un identifiant qui peut être le nom de l'utilisateur.

- **Les OTP (One-Time-Password)**

  Un mot de passe à usage unique (siglé One-Time Password (OTP)) est un mot de passe qui n'est valable que pour une session ou une transaction. Les OTP permettent de combler certaines lacunes associées aux traditionnels mots de passe statiques, comme la vulnérabilité aux attaques par rejeu. Cela signifie que, si un intrus potentiel parvient à enregistrer un OTP qui était déjà utilisé pour se connecter à un service ou pour effectuer une opération, il ne sera pas en mesure de l'utiliser car il ne sera plus valide. En revanche, les OTP ne peuvent pas être mémorisés par les êtres humains, par conséquent, ils nécessitent des technologies complémentaires afin de s'en servir. Cela supprime les contraintes de :

  – Longévité du mot de passe. Le mot de passe est utilisé une seule fois
  – Simplicité du mot de passe. Le mot de passe est calculé par l'ordinateur et non pas choisi par un utilisateur
  – Attaque par dictionnaire ou par force brute : Pourquoi essayer de cracker un mot de passe obsolète ?
  – Reniflage et chiffrement du mot de passe : Le mot de passe à usage unique peut être envoyé en clair sur le réseau : Lorsqu'un renifleur en détecte un, il est déjà trop tard, car il est utilisé, et non réexpoitable.

  Ces OTP sont envoyés à l'utilisateur par différents moyens :

  – le code est reçu sur le téléphone de l'usager par mail ou Short Message Service (SMS). Il est souvent utilisé dans le secteur bancaire, notamment pour les achats effectués via l'Internet.
  – le mot de passe est parfois généré via une application sur le smartphone de l'utilisateur. Il nécessite donc qu'il installe sur son smartphone l'application contenant le soft token.
  – Dans d'autres cas, cela consiste à générer un mot de passe unique via un « token physique » . Le token RSA est un exemple célèbre de hard token. Le hard token génère un mot de passe aléatoire, valable pour une durée déterminée, qui est demandé lors de l'authentification de l'utilisateur.

- **Les certificats numériques**

  Le certificat numérique est une autre alternative aux solutions déjà présentées. Plutôt que d'être sollicité pour entrer un code OTP, une notification de « login » va être envoyé sur le smartphone. Une fois cette notification approuvée, le processus d'authentification se poursuit. Il emploie la cryptographie asymétrique (clé privée / clé publique) pour vérifier l'identité. Une clé privée est générée et stockée sur l'équipement. La clé publique, elle est stockée du côté serveur. À chaque login, un challenge est réalisé entre le serveur et le smartphone [20].

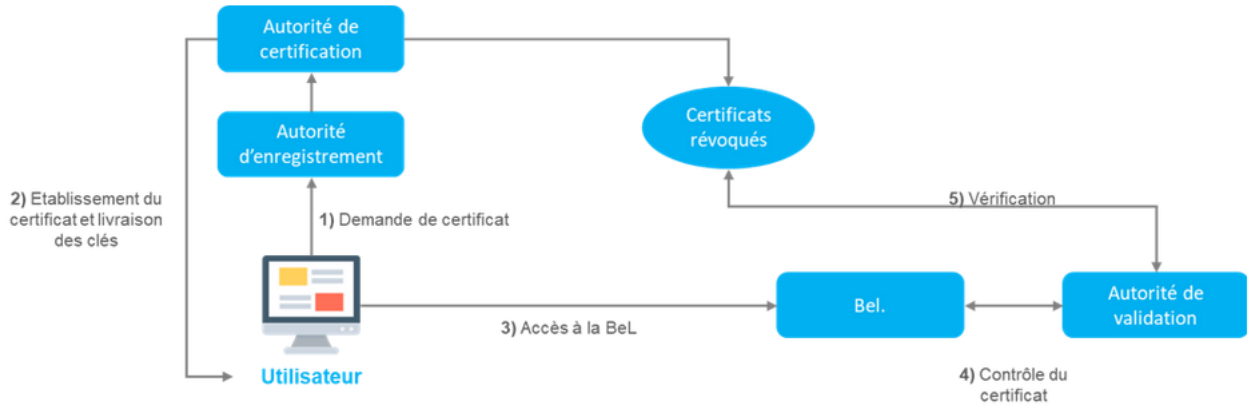


FIGURE 1.1 – Les certificats [1]

- **Les cartes à puce**

  Les cartes à puce sont des équipements munis d'un circuit intégré capable de mémoriser de façon sécurisée une série d'informations. Ce circuit intégré s'appelle une puce. Sur cette puce, peut être enregistrées des informations permettant d'authentifier un utilisateur. L'utilisation de ce système nécessite un deuxième équipement qui est le lecteur de la carte. La lecture des informations de la puce se fait avec ou non le contact avec la carte.

- **La biométrie**

  Considérée comme une des méthodes les plus prometteuses parce qu'elle fournit des caractéristiques physiques strictement propres à une personne, la biométrie est de plus en plus utilisée dans le processus d'authentification forte. On distingue quatre cas d'utilisation de la biométrie :

1. La reconnaissance digitale

   Elle consiste à reconnaître les différentes couches sur la paume de votre doigts après l'apposition sur un support. L'empreinte digitale est unique au niveau de chaque doigt de chaque individu.

2. La reconnaissance vocale

   Elle est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour une analyse de comparaison afin d'identifier un individu à partir des caractéristiques de cette voix.

[1] https ://weave.eu/cybersecurite-sommes-moyens-dauthentification/

3. La reconnaissance d'iris

   La reconnaissance de l'iris est une technique de biométrie permettant de reconnaître une personne par l'observation de son iris.

4. La reconnaissance faciale

   La reconnaissance de visage est un domaine de la vision par ordinateur consistant à reconnaître automatiquement une personne à partir d'une image de son visage.

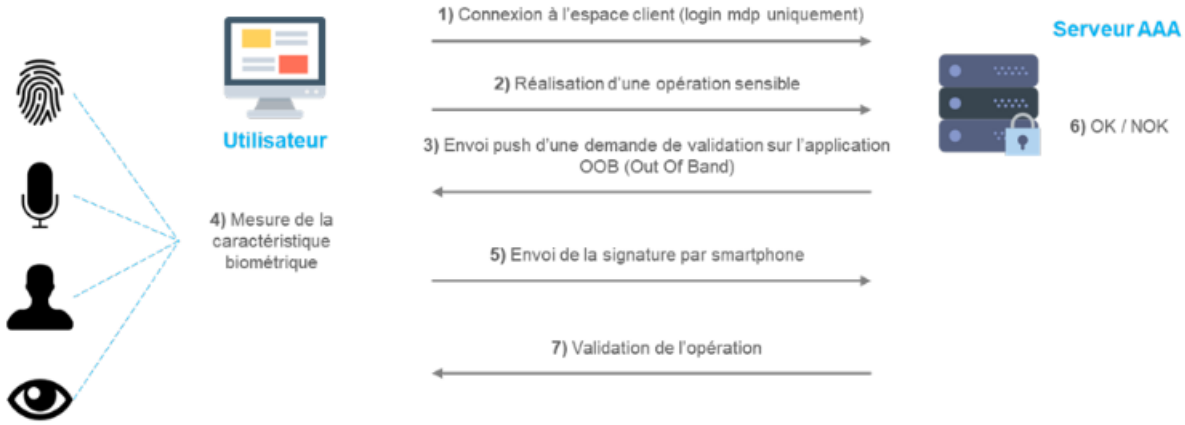


FIGURE 1.2 – Authentification via la biométrie

1.1.4 Critiques des méthodes existantes

Malgré l'aspect sans faille qui se dégage des solutions d'authentification existantes, ces solutions présentent également des insuffisances. Nous avons :

- Les problèmes de sécurité : les systèmes d'authentification par login/mot de passe n'offrent plus un haut niveau de sécurité pour la protection des infrastructures informatiques. Il existe plusieurs techniques qui permettent d'outre passer ces systèmes dont le cracking des mots de passe. Notons également certaines contraintes des mots de passe. Nous avons : la taille des mots de passe, l'utilisation des caractères spéciaux, la différence des mots de passe pour les diverses utilisations. Toutes ces contraintes interviennent dans la constitution des mots de passe.

- Les problèmes de réseau : les solutions d'authentification implémentant les OTPs par SMS/Mail nécessitent de la connexion internet pour pouvoir envoyer les OTPs par SMS/Mail. En cas d'indisponibilité de la connexion internet, les infrastructures seront inaccessibles par les utilisateurs car ils ne pourront pas recevoir les OTPs par SMS ou par mail.

- Les problèmes de synchronisation : les applications qui obligent des synchronisations régulières auprès des serveurs distants affectent la fiabilité du service. Pour récupérer les mots de passe à utiliser pour l'authentification, le serveur d'authentification doit synchroniser avec les serveurs de la solution d'authentification. En cas d'échec de synchronisation pour des problèmes de connexion internet, les infrastructures sécurisées ne seront pas accessibles.

- L'accessibilité des clés privées : les solutions implémentant les certificats numériques attribuent à chaque utilisateurs une clé privée. Cette clé privée est utilisée pour vérifier la légitimité de la demande d'accès faite par les utilisateurs. Habituellement, les clés privées sont stockées directement sur le disque dur de la machine comme un fichier. Cela pose un véritable problème de sécurité dans la mesure où un hacker qui arrive à accéder à la clé privée peut s'authentifier normalement.

- La perte de matériel : la perte des équipements comme les cartes à puces, les badges, les téléphones portables constitue un véritable problème de sécurité dans le cadre de l'utilisation de ces solutions d'authentification.

Au vu de toutes les critiques sur les solutions d'authentification existantes, nous avons opté pour l'authentification basée sur la biométrie. Notons que les données biométriques sont des informations uniques et propres à chaque utilisateur. Ainsi, nous pouvons, avec la biométrie, pallier à plusieurs problèmes de ces systèmes d'authentification existantes. Les éléments de la biométrie semblent donc être les plus sécurisés pour les systèmes d'authentification.

Dans le cadre de notre travail, nous avons opté pour la reconnaissance faciale parmi les quatre cas d'utilisation de la biométrie sus-cités. La reconnaissance faciale utilise le visage de l'utilisateur comme donnée d'authentification. Le visage d'un individu présente des traits qui permettent de le rendre unique et de le reconnaître parmi plusieurs autres visages. Dans les cas où nous sommes en présence des vrais jumeaux ou des sosies, les visages présentent quelques traits de ressemblance mais aucune correspondance n'est faite entre des visages de deux individus différents. Avec cette méthode, nous avons donc plus de fiabilité et de sécurité au niveau de l'authentification des utilisateurs.

1.1.5 La reconnaissance faciale

La reconnaissance faciale est une technique qui permet à partir des traits de visage :

- **d'authentifier une personne :** c'est-à-dire, vérifier qu'une personne est bien celle qu'elle prétend être (dans le cadre d'un contrôle d'accès)

- **ou d'identifier une personne :** c'est-à-dire, de retrouver une personne au sein d'un groupe d'individus, dans un lieu, une image.

Processus de la reconnaissance faciale

Un système de reconnaissance de visages est un système d'identification et de vérification d'individus, qui permet de vérifier si un personnel appartient à la base de données d'un système. Cette opération d'identification et de vérification se repose sur plusieurs étapes [16] :

- **Acquisition de l'image :** cette opération permet d'extraire du monde réel une représentation matricielle c'est-à-dire une image d'une personne, d'un objet, etc. Cette opération peut être statique (appareil photo, scanner,...) ou dynamique (caméra ou webcam...)

- **Prétraitements :** les données brutes issues des capteurs sont les représentations initiales des données, à partir desquelles des traitements permettent de construire celles qui seront utilisé pour la reconnaissance. L'image brute peut être affectée par différents facteurs causant ainsi sa détérioration, elle peut être bruitée, c'est à dire contenir des informations parasites à cause des dispositifs optiques ou électroniques. Pour pallier à ces problèmes, il existe plusieurs méthodes de traitement et d'amélioration des images, telle que : la normalisation, l'égalisation de l'histogramme, etc...

- **Détection puis localisation :** Les systèmes de reconnaissance de visages sont complexes. La difficulté réside notamment dans la partie détection automatique du visage, bien que nous développions surtout la partie reconnaissance, il est intéressant de parler de l'étape de détection automatique du visage qui est très importante dans un système de reconnaissance. Ce qui rend la détection de visages dans une image très difficile, c'est surtout la complexité du décor, les variations de poses, les conditions de lumières généralement inconnues, etc. Il existe plusieurs méthodes qui peuvent être appliquées à la détection automatique des visages. Il faut détecter la présence d'un visage dans l'image, ensuite le localiser en vue d'extraire les traits pour le caractériser et le différentier des autres. Le résultat de cette étape est l'obtention de la partie d'image à traiter.

- **Extraction des paramètres et Classification :** Dans cette étape on extrait de l'image les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de décision. Le choix de ces informations utiles revient à établir un modèle pour le visage, elles doivent être discriminantes et non redondantes. Ces informations seront ensuite classées, en d'autres termes, affectés à la classe la plus proche, les individus ayant des similarités sont regroupés dans la même classe. Ces classes varient selon le type de décision.

- **Apprentissage :** L'apprentissage consiste à mémoriser les modèles calculés dans la phase d'analyse pour les individus connus. Un modèle est une représentation compacte des images qui permet de faciliter la phase de reconnaissance mais aussi de diminuer la quantité de données à stocker ; en quelque sorte l'apprentissage est la mémoire du système.

- **Décision :** La décision est la partie du système ou on tranche sur l'appartenance d'un individu à l'ensemble des visages où pas, et si oui quelle est son identité. Donc la décision c'est l'aboutissement du processus. On peut le valorisé par taux de reconnaissance (fiabilité) qui est déterminé par le taux de justesse de la décision.

L'image 1.3 présente le processus de la reconnaissance faciale.

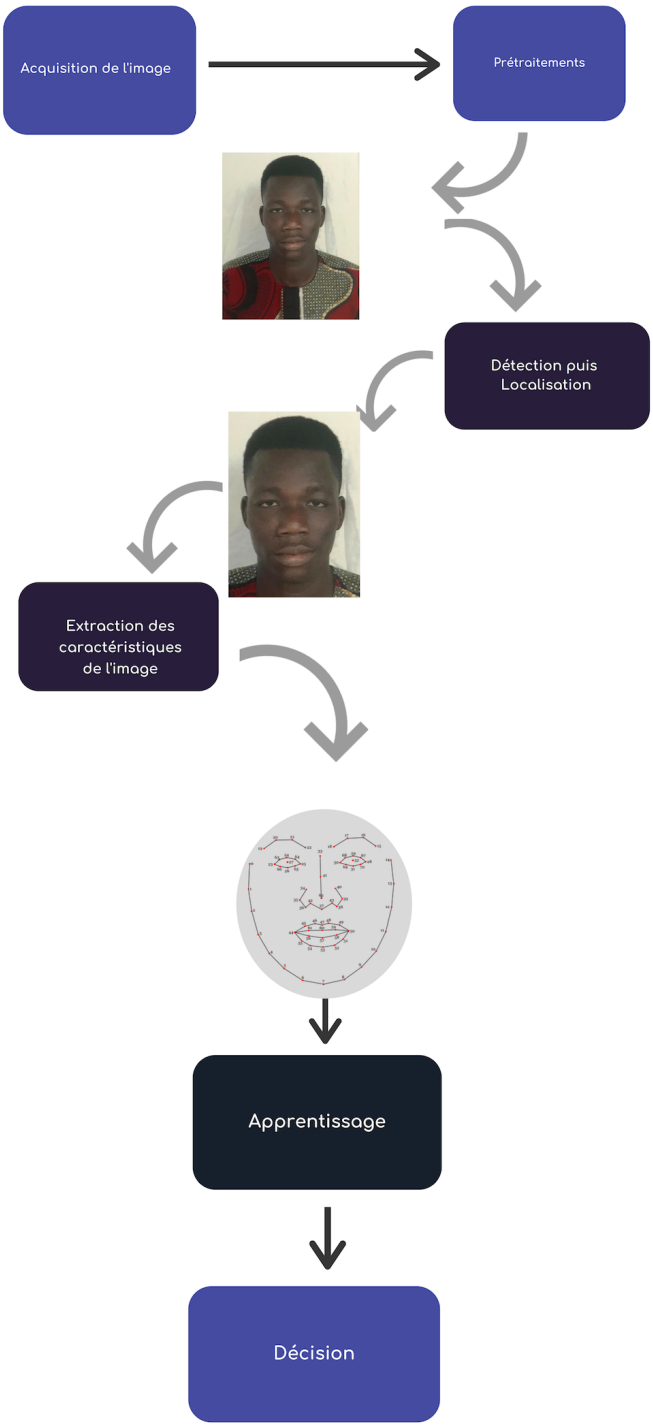


FIGURE 1.3 – Processus de la reconnaissance faciale

1.1.5.1 Avantages de la reconnaissance faciale

Depuis plusieurs années, les domaines d'applications de la reconnaissance faciale deviennent de plus en plus grandissants. Son utilisation se fait remarquer dans le domaine de l'aviation, le domaine militaire, aussi dans le domaine du marketing digital, etc. Le choix de l'utilisation de ces systèmes se base généralement sur les nombreux avantages qu'ils présentent. Ainsi nous pouvons identifier ces systèmes de reconnaissance faciale comme étant peu intrusive et qui peuvent également se dérouler à l'insu de l'utilisateur. Aujourd'hui, certains de ces systèmes sont capables de reconnaître des individus même avec des subterfuges (fausses moustaches, barbe, lunettes...). L'un de leur plus gros avantages est qu'il ne demande parfois aucune action de l'utilisateur et la reconnaissance se fait également sans un contact physique.

1.1.6 Quelques cas d'usurpation d'identité sur la reconnaissance faciale

Les systèmes de reconnaissance faciale sont la cible de plusieurs techniques d'usurpation d'identité utilisant trois principaux types d'attaques connus dans ce domaine pour tromper les systèmes de reconnaissance faciale [2] :

- **les attaques photos :** en effet, montrer une photo au système semble être une bonne méthode pour tromper la caméra. L'efficacité de ce type d'attaque a été aidée par l'arrivée d'imprimantes ayant une résolution élevée, aussi nommée haute définition. Ces imprimantes permettent en effet d'imprimer une image très qualitative, trompant aisément la vigilance des systèmes. Notons également la contribution des appareils photos de hautes résolutions dans cette attaque. Ce type d'attaque est devenu encore plus simple à effectuer avec l'apparition des réseaux sociaux avec les nombreuses photos postées par les utilisateurs [2].
- **les attaques vidéos :** ce deuxième type d'attaque consiste à présenter une vidéo contenant un visage légitime à l'objectif de la caméra afin de tromper le système de reconnaissance faciale. Des logiciels disponibles sur internet, appelés caméras virtuelles, tels que VirtualCamera ou Manycam permettent en effet de tromper le système de reconnaissance en lui présentant une vidéo comme si celle-ci provenait de sa propre caméra [2].
- **les attaques avec un masque :** l'impression tridimensionnelle permet de créer un masque correspondant au visage d'un utilisateur légitime. Il est alors facile d'imaginer l'utilisation d'un masque présentant le visage d'un individu pour tromper un système de reconnaissance faciale [2].

1.2 API : Application Programming Interface ou Interface de Programmation Applicative

On appelle Application Programming Language (API) un ensemble de fonctions, de classes normalisées et d'outils utilisés pour la conception des applications. Les APIs proposent aux développeurs la simplification des tâches lors de la programmation.

1.2.1 Caractéristiques des APIs

- **Dépendance au langage :** Une API peut être utilisable dans un unique langage de programmation ou être indépendante des langages. Dans le second cas un langage intermédiaire comme XML qui peut être utilisé comme format de données pour les requêtes aux fonctions et méthodes [11].
- **Licence d'utilisation :** l'API est sous licence libre et utilisable sans frais par tout programmeur, ou sous licence propriétaire et accessible uniquement à une communauté restreinte, ce qui le cas par exemple des API de jeux de consoles [11].
- **Niveau de langage :** On distingue d'une part l'API de haut niveau, en matière de langage de programmation, comme les APIs graphiques et d'autre part l'ABI (Application Binary Interface), proche du système, comme la Linux Standard Base ou les interfaces de pilotes de matériels.

Au delà de ces trois éléments, nous pouvons attribuer d'autres caractéristiques à une API. La bonne qualité d'une API est définie par les caractéristiques suivantes :

- **Simplicité :** dans la plupart de cas, les programmeurs ont tendance à résoudre des problèmes complexes par la proposition d'API. Ainsi ce problème doit paraître moins complexe c'est-à-dire plus simple pour l'utilisateur de l'API. Les APIs doivent offrir la simplicité et les programmeurs doivent déployer des efforts considérables pour préserver cette simplicité [14].
- **Fournir des abstractions utiles :** les exemples d'abstraction abondent dans notre monde. Les pilotes de périphérique résument les détails de la gestion du matériel du fournisseur. Même les langages de programmation font abstraction des détails de l'écriture du code machine. Un code bien écrit fournit des abstractions et les API ne font pas exception. Une API doit masquer les détails de ce qu'elle fait à ses utilisateurs tout en leur rendant service. Si les utilisateurs ont besoin de plonger dans le code de l'API ou dans son exécution pour la comprendre, le programmeur a donc fourni une mauvaise abstraction [14].
- **Cohérent et symétrique :** une API doit avoir une cohérence interne, un style de notation et nomination commun. La symétrie présente une considération légèrement plus

subtile. Lorsqu'on parle de symétrie, on parle d'une tendance à fermer les boucles ouvertes de la manière attendue par les utilisateurs. Par exemple, si on dispose d'une API d'accès aux fichiers qui permet d'appeler open(nom_fichier), on doit également proposer une méthode close(nom_fichier). Ouvrir et fermer ont une signification opposée, donc offrir les deux crée une symétrie en ce qui concerne cette opération. On doit généralement atteindre cette caractéristique dans une API en appliquant un regard critique sur ce qui est exposé aux utilisateurs [14].

- **Suivre le principe du moindre étonnement :** astucieusement nommé, le « principe du moindre étonnement », cela veut dire «un système doit se comporter de manière cohérente avec la manière dont les utilisateurs de cette composante vont probablement s'attendre à ce qu'il se comporte ; c'est-à-dire que les utilisateurs ne devraient pas être étonnés de la façon dont il se comporte. » Une API ne doit pas époustoufler ses utilisateurs. [14]

1.2.2 Types d'API

De nos jours, il existe plusieurs types d'API. On peut les classer en quatre (04) grandes catégories [13] :

- **Les services web et Websockets API :** Ce sont des parties d'un logiciel ou d'un système dont l'accès à leurs services se fait via une adresse sur le web. Elle constitue une technologie qui rend possible l'ouverture d'une session de communication interactive entre un navigateur et un serveur. Ils utilisent le protocole HyperText Transfert Protocol (HTTP) pour échanger les informations avec les applications clientes. Les web services et websocket API les plus connus sont : l'API Facebook , l'API Google Maps, WebSocket-Node et Caddy

- **Les APIs orientées Librairies et Classes :** Ce sont des APIs dont l'utilisation requiert une référence ou une importation d'un code ou de fichiers binaires. Les données et fonctionnalités fournient par ces types d'API sont organisées autour des classes comme dans les langages de programmation orientés objets. Les fonctions et routines de cette librairie sont utilisées pour créer de nouvelles fonctionnalités. L'API Twilio en bon exemple.

- **Les APIs d'objets distants :** Elles sont en général fournies au sein d'un intergiciel (Middleware) pour permettre la communication et les interactions pour des applications réparties. Elles utilisent un protocole de communication par exemple CORBA-Common Object Request Broker Architecture. Ces APIs fonctionnent en implémentant en local une représentation de l'objet distant puis intéragissent avec eux. Les interactions réalisées sont alors dupliquées sur l'objet distant via le protocole. La plupart de ces API sont maintenant considérées comme des logiciels hérités. Un exemple est .NET Remoting.

- **Les APIs matériels :** Elles sont utilisées dans l'adressage des pièces matérielles sur un appareil. Il s'agit par exemple des disques durs, des bus PCI.

1.3 API et programmes existants pour la reconnaissance faciale

---

A l'heure de nos recherches, il y a plusieurs solutions qui implémentent l'authentification par la reconnaissance faciale. Au nombre de ces applications, nous pouvons citer :

- **Face Recognition**
  Face Recognition est une application développée par Sensible Vision Inc pour les utilisateurs Android et iOS [19]. Il vous permet de déverrouiller facilement l'écran et d'autres applications sur votre smartphone en utilisant l'option de reconnaissance faciale. Vous n'avez pas besoin de vous souvenir de votre mot de passe pour déverrouiller votre smartphone pendant que vous utilisez cette application.

- **Blink**
  Blink est un logiciel qui vous permet de vous connecter à votre compte sur Windows Vista et 7. Le super avantage, c'est que vous n'aurez plus à taper votre mot de passe de connexion à la vue de tous. Il est possible d'enregistrer plusieurs visages différents, chacun correspondant à un compte différent sur l'ordinateur.

- **Keylemon**
  Keylemon est un logiciel de reconnaissance faciale qui permet de s'identifier au démarrage de l'ordinateur grâce à une webcam, au lieu de saisir ses identifiants d'ouverture de session [15]. Mieux, il est fourni avec un plugin pour Firefox qui garantit une connexion rapide à vos comptes Facebook, Twitter ou LinkedIn
  Bien conçu et entièrement traduit en français, Keylemon comprend un assistant de configuration qui vous aide à créer pas à pas un «modèle» de visage qui sera ensuite reconnu à chaque fois que vous allumerez l'ordinateur.

- **Gemalto Cogent :** c'est un système de reconnaissance faciale vidéo qui reconnaît automatiquement les visages dans une foule, même dans des environnements dynamiques et incontrôlés. Il envoie des alertes en temps réel pour pouvoir agir rapidement. Le système peut être intégré dans une large gamme d'équipements vidéo, et des algorithmes avancés augmentent la précision des correspondances.

- **Google Cloud Vision :** Cloud Vision est une nouvelle API propriétaire proposée par Google afin de permettre aux développeurs d'exploiter ses fonctionnalités pour intégrer l'analyse d'image à leurs applications. Elle permet de comprendre le contenu d'une image en intégrant de puissants modèles de machine learning dans une API

REST facile à utiliser. Cloud Vision classe les images rapidement dans des milliers de catégories (par exemple, "bateau"). Elle détecte également les objets et les visages et reconnaît les mots en caractères d'imprimerie contenus dans les images. Google Cloud Vision s'appuie notamment sur le machine learning et permet également la reconnaissance faciale.

- **Face ID :** est une technologie propriétaire de reconnaissance faciale de Apple. Elle se compose de matériel et des logiciels les plus évolués que nous ayons créés à ce jour. La caméra TrueDepth capture des données faciales précises en projetant et en analysant plus de 30 000 points invisibles, afin de créer une carte de profondeur et de capturer une image infrarouge de votre visage. Une partie du moteur neural de la puce A11, A12 Bionic et A12X Bionic (protégée au sein de la Secure Enclave) transforme la carte de profondeur et l'image infrarouge en une représentation mathématique et compare cette dernière aux données faciales enregistrées [17].
  Face ID s'adapte automatiquement aux changements de votre apparence, comme le maquillage ou la pilosité faciale. Si les changements sont plus significatifs, par exemple une barbe rasée, Face ID confirme votre identité à l'aide de votre code d'accès avant de mettre à jour vos données faciales. La technologie Face ID a été conçue pour détecter les couvre-chefs, les lunettes, les lentilles de contact et de nombreuses lunettes de soleil. En outre, elle a été pensée pour fonctionner à l'intérieur, à l'extérieur et même dans l'obscurité totale.

Après plusieurs analyses des différentes solutions existantes il est a noté à l'exception de la Face ID, que les autres solutions précédemment abordées sont faibles à l'usurpation d'identité. En ce qui concerne la face ID, la probabilité qu'une personne choisie au hasard dans la population puisse usurper de votre identité est d'environ 1 pour 1 000 000 avec une seule apparence enregistrée [17].

Face ID compare les informations de profondeur, absentes des photographies numériques en 2D ou imprimées. Cette fonctionnalité fait appel à des réseaux neuronaux sophistiqués pour empêcher tout individu portant un masque ou employant d'autres techniques d'usurper votre identité. La technologie Face ID est même sensible à l'attention. Elle reconnaît si vos yeux sont ouverts et si vous portez votre attention sur l'appareil. Dès lors, un tiers éprouvera d'avantage de difficultés pour déverrouiller votre appareil à votre insu (comme lorsque vous dormez). Il est très important de noter que les différentes techniques utilisées par Face ID ne sont pas exposées et Face ID reste propriétaire. Il s'avère donc nécessaire la mise en place d'un système de reconnaissance faciale efficace et libre qui permette de détecter des attaques photos, au masque et vidéos sur la reconnaissance. C'est dans cette optique que s'inscrit notre solution qui permet de détecter les cas d'usurpation d'identité.

Conclusion

Les facteurs d'authentification sont nombreux et de différentes formes. Dans ce chapitre, nous avons passer en revue les notions autour de l'authentification, les solutions existantes, leurs insuffisances et nos hypothèses de recherche autour desquelles seront centrées nos recherches et développements. Le chapitre suivant sera consacré à notre solution à travers sa conception et les outils utilisés pour sa réalisation.

Chapitre 2

Conception de Oudjat et présentation du matériel

Introduction

La réalisation d'une application requiert d'abord l'analyse et la conception des informations utiles à notre projet. La définition des modules induit les différentes étapes de sa réalisation. Il s'agit de la modélisation. Celle de notre travail est axée autour de du diagramme de cas d'utilisation. Et la conception se fera avec plusieurs outils de traitement d'images et de vision par ordinateur.

2.1 Méthodes

2.1.1 Méthode de résolution du problème

Pour le développement de notre application appelée **Oudjat**, nous avons utilisé la méthode agile comme méthode de travail [9]. La méthode agile a pour but, de réduire le cycle de développement des projets informatiques, de répondre plus rapidement aux évolutions des demandes de l'utilisateur final. Les projets informatiques agiles sont gérés de manière adaptative, incrémentale et itérative.

Les méthodes agiles visent à intégrer au cours du projet de manière continue le client final, c'est à dire le plus souvent l'utilisateur final à qui le nouveau logiciel est destiné. Les méthodes agiles permettent une plus grande réactivité aux demandes du client. Dans le cadre notre travail, nous avons donc opté pour la méthode Agile-Scrum principalement en raison de l'indépendance des fonctionnalités de notre solution. **Oudjat** intègre plusieurs fonctionnalités donc le développement peut être fait de manière indépendante.

2.1.2 Méthodes de modélisation

La modélisation des fonctionnalités de notre système se fera avec le langage Unified Modeling Language (UML) [5]. Il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un programme. UML offre un standard de modélisation, pour représenter l'architecture logicielle grâce à des représentations graphiques appelées diagrammes. Ces diagrammes nous ont permis de modéliser notre solution en utilisant le diagramme de cas d'utilisation.

2.2 Outils

Pour la mise en place des fonctionnalités précitées, Oudjat utilise plusieurs technologies et bibliothèques.

Python

Facile à apprendre et polyvalent, c'est un langage de programmation objet, multi-paradigme , multiplateformes et open source [1]. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Ce langage dispose de plusieurs bibliothèques spécialisées dans le traitement des images. Il est couramment utilisé dans la science de données et des fonctions avec moins de lignes de code.

SQLite

SQLite est un moteur de base de données Structured Query Language (SQL) autonome, de haute fiabilité, intégré, doté de nombreuses fonctionnalités. Il fournit une base de données légère sur un disque ne nécessitant pas de processus serveur distinct et permet d'accéder à la base de données à l'aide d'une variante non standard du langage de requête SQL. En plus de ses avantages, SQLite dispose de plusieurs plugins pour faciliter l'interaction avec le langage Python.

OpenCV

Open Computer Vision (OpenCV) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel [4]. Elle propose un ensemble de plus de 2500 algorithmes de vision par ordinateur pour le traitement de flux vidéo.

NumPy

NumPy est une bibliothèque Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle open source fournit de multiples fonctions permettant

notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

Face Recognition

Face Recognition est une API développé en Python qui permet de faire de la reconnaissance faciale. Il se base sur la bibliothèque Dlib de python et utilise du Deep Learning pour extraire les caractéristiques des visages sur une image [6]. Le visage d'un homme a des caractéristiques qui sont à l'avance encodées dans dlib et qui sont utilisés pour dire si un objet détecté dans une image est ou non le visage d'un Homme.

Mahotas

Mahotas est une bibliothèque de vision par ordinateur et de traitement d'images pour Python. Il comprend de nombreux algorithmes implémentés en C ++ pour la vitesse en fonctionnant avec des tableaux numpy et avec un très bonne Interface Python. Mahotas a actuellement plus de 100 fonctions pour le traitement des images et la vision par ordinateur et continue de croître. Il nous a été utile dans l'extraction de la texture en utilisant l'algorithme de Haralick basé sur les matrices de co-occurrence de niveau de gris.

Pillow ou PIL

Python Imaging Library (ou PIL) est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauver différents formats de fichiers graphiques. La bibliothèque est disponible librement selon les termes de la Python Imaging Library license . Plus maintenue depuis 2009, elle est remplacée à partir de 2010 par un dérivé proche, Pillow.

Scikit-image

Scikit-image est un ensemble d'algorithmes de traitement d'images. Il est disponible gratuitement et sans restriction. Elle a été conçu pour le traitement d'images pour SciPy. Scikit-image est une outil de bonne qualité, dont le code a été revu par des pairs, et écrit par une communauté active de bénévoles.

2.3 Principe de fonctionnement de notre API

Notre objectif dans ce travail, est d'arriver à détecter les cas d'usurpation de l'identité d'un légitime utilisateur pouvant faire la demande d'accès à une ressource informatique. Dans un premier temps, en se basant sur la reconnaissance du visage, identifier l'utilisateur légitime et dans un second temps s'assurer qu'il s'agit bien de l'utilisateur légitime qui fait la demande d'accès à la ressource informatique en se basant sur certains aspects du visage reconnu. La présentation des principes de fonctionnement se fera en deux étapes : le principe de fonctionnement des différentes méthodes fonctionnelles et générales de Oudjat et l'algorithme sur lequel est basée notre solution. L'ensemble de ces fonctions ont été regroupées pour donner le code source de l'API qui a été déposé dans un répertoire **Github** disponible sur le lien : https://github.com/godwineHoungavou/oudjat-API [12]

2.3.1 Principes de fonctionnement des méthodes de Oudjat

L'utilisation de Oudjat comme toute API, passe par la manipulation de ses différentes méthodes. L'implémentation de notre solution pour notre cas d'utilisation se base sur différentes méthodes qui seront présentées ci-dessous.

2.3.1.1 Méthodes fonctionnelles

- **webcam_list()**
  la méthode webcam_list permet de lister les différentes webcams connectées à l'ordinateur qui tourne le programme. Cette fonction ne prend rien en entrée et renvoie un tableau contenant la liste des différentes webcams connectées à l'ordinateur.

- **blink_detection()**
  Cette méthode reçoit en entrée un objet de type VideoStream qui est la variable d'initialisation du flux de la vidéo. Comme indiqué à la ligne 1 de l'algorithme 1, elle débute par la définition du temps de détection des clignement qui est par défaut 6 secondes et ensuite vérifie si le flux de la vidéo a bien démarré à la ligne 2. Ensuite , elle passe à la récupération du flux vidéo de la webcam à la ligne 3 et effectue des opérations sur ce flux pendant 6 secondes minimum comme indiqué à la ligne 4. En fonction des images qui forment la vidéo reçue (ligne 5), elle identifie le visage présent sur les images. A la ligne 7, le système extrait les yeux du visage pour compter le nombre de clignement présent. Ensuite, il vérifie si on détecte au moins un (01) clignement sur les yeux (ligne 8) et retourne **1** si oui (ligne 9). Dans le cas où aucun clignement n'est détecté après le temps défini, on renvoie **0** comme indiqué à la ligne 15 de l'algorithme de la méthode **blink_detection**. Dans le cas où le flux de la vidéo ne démarre pas, la méthode renvoie **-1**. La détection du clignement des yeux se base sur le calcul du rapport de l'aspect de l'œil. Ce rapport est une valeur constante lorsque l'œil est ouvert mais tombe rapidement à zéro lorsque l'œil est fermé [18]. Le calcul de ce rapport utilise les valeurs des

repères faciaux de l'œil. Ces repères faciaux sont présentés sur la figure. [8]

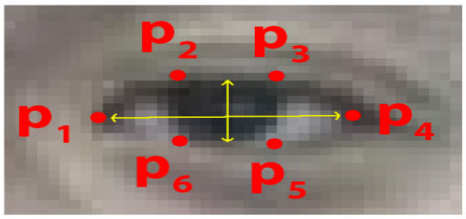


FIGURE 2.1 – Repères faciaux de l'œil

Le rapport de l'aspect de l'œil désigné par EAR dans ce présent document est obtenu par la formule suivante :

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||}$$

L'algorithme 1 de la **blink_détection** présente le déroulement de la détection du clignement des yeux.

Algorithme 1 : Algorithme de blink_detection

```
Input : fluxWebcam, times ;
Clignement();
if fluxWebcam exists then
    while video ← fluxWebcam do
        while times do
            for image in video do
                for visage in image do
                    for eye in visage do
                        if Clignement then
                            return 1
                        end
                    end
                end
            end
        end
        return 0
    end
else
    return -1
end
```

- **face_recognize()**
  Pour démarrer le processus de la reconnaissance de visage, comme l'illustre la ligne 1

de l'algorithme 2 (celui de la reconnaissance de visage), la méthode charge le visage de l'utilisateur connu depuis la base de données et définit un temps pour la reconnaissance du visage qui par défaut est 5 secondes (ligne 2). Ensuite, elle vérifie si le chargement du visage de l'utilisateur depuis la base de données et le démarrage du flux de la webcam se sont effectués avec succès. Elle récupère le flux de vidéo de la webcam et durant le temps défini, en fonction des images qui constituent la vidéo, le système identifie le visage présent sur les images pour une comparaison. Cela est décrit aux lignes 4, 5, 6 et 7 de l'algorithme 2. Le visage est par la suite comparé au visage de l'utilisateur chargé de la base de données pour faire ressortir les concordances. Cette opération s'effectue à la ligne 8 de l'algorithme. Dès qu'une correspondance est faite, on renvoie **1** (ligne 6) et **0** dans le cas contraire après le temps défini. (ligne 14). Dans le cas où le flux de la vidéo ne démarre pas ou le chargement du visage de la base de données échoue, la méthode renvoie **-1**.

L'algorithme 2 de la **face_recognize** présente le processus de reconnaissance de visage.

Algorithme 2 : Algorithme de face_recognize

```
   Input : fluxWebcam,times;
1  knowFaces ← load_from_db;
2  if fluxWebcam exists AND load_from_db success then
3      while video ← fluxWebcam do
4          while times do
5              for image in video do
6                  for visage in image do
7                      if visage in knowFaces then
8                          return 1
9                      end
10                 end
11             end
12         end
13         return 0
14     end
15 else
16     return -1
17 end
```

- **object_detection()**
  Cette méthode chargée de la détection des objets prend en entrée l'objet d'initialisation du flux de la vidéo et le temps de détection des objets. A la ligne 1, liste des objets

définis comme dangereux en fonction de l'environnement d'utilisation du programme est initialisé. Ensuite, la méthode vérifie l'initialisation de la liste des objets et la démarrage du flux de la vidéo à la ligne 3. Elle récupère le flux de vidéo de la webcam et durant le temps défini, en fonction des images qui constituent la vidéo, on commence la détection des objets présents sur les images. Ces opérations sont illustrées par les lignes 4, 5, 6 et 7 de l'algorithme 3 de l'**object_detection**. Ensuite, on prend chaque objet de la liste définie qu'on compare à chaque objet détecté dans l'environnement immédiat de l'utilisateur, c'est le rôle des lignes 8 et 9. Lorsqu'une correspondance est faite on renvoie **0** comme illustré à la ligne 7 et **1** dans le cas contraire après le temps défini (ligne 16). Dans le cas où le flux de la vidéo ne démarre pas ou l'initialisation de la liste des objets échoue, la méthode renvoie **-1**.

L'algorithme 3 de la **object_detection** présente le processus de détection des objets dangereux. On qualifie d'objets dangereux des objets ne devant pas se retrouver dans un environnement donné. A titre d'exemple, on peut citer un fusil dans une salle de classe.

Algorithme 3 : Algorithme de object_detection

```
   Input : fluxWebcam, times ;
1  listeObjetDangereux ← initialize;
2  if fluxWebcam exist AND initialization success then
3      while video ← fluxWebcam do
4          while times do
5              for image in video do
6                  for objetDetect in image do
7                      for objet in listeObjetDangereux do
8                          if objetDetect = objet then
9                              return 0
10                         end
11                     end
12                 end
13             end
14         end
15         return 1
16     end
17 else
18     return -1
19 end
```

- **texture_analysis() :**
  la méthode texture_analysis pour effectuer l'analyse de la texture du visage à détecter,

charge les données des textures de visage légitime (naturel) sur lesquelles se baseront les opérations de vérifications des textures. Comme indiqué à la ligne 3 de l'algorithme 4 de la **texture_analysis**, on vérifie le chargement des données des textures et la démarrage du flux de la vidéo à la ligne 3. Ensuite, on récupère le flux de vidéo de la webcam en live et durant le temps défini, en fonction des images qui constituent la vidéo, le système identifie le visage présent sur les images. Cela est indiqué à la ligne 4, 5, 6 et 7 de l'algorithme 4. On extrait ensuite, la texture du visage détecté ; sur cette texture, on effectue des opérations de concordance. A la ligne 9 de l'algorithme, dès qu'une correspondance est faite entre les données de la texture du visage détecté et celles du visage légitime stockées précédemment, on renvoie **1** comme indiqué à la ligne 10 et dans le cas échéant après le temps défini, on renvoie **0** à la ligne 16. Cette texture extraite sous la forme d'un histogramme, est comparé à des valeurs prédéfinies. Le résultat de la comparaison renvoyé en sortie, permet ainsi donc de décider s'il s'agit d'un masque, d'une vidéo ou du visage d'une personne physique présente devant la webcam. Dans le cas où le flux de la vidéo ne démarre pas ou le chargement des textures prédéfinies échoue, la méthode renvoie **-1**.

L'algorithme 4 de la **texture_analysis** présente le processus d'analyse de la texture.

Algorithme 4 : Algorithme de texture_analysis

```
   Input : knowTextures, fluxWebcam, times ;
1  knowTextures ← load_from_file;
2  if fluxWebcam exist AND load_from_file success then
3      while video ← fluxWebcam do
4          while times do
5              for image in video do
6                  for visage in image do
7                      for texture in visage do
8                          if texture in knowTextures then
9                              return 1
10                         end
11                     end
12                 end
13             end
14         end
15         return 0
16     end
17 else
18     return-1
19 end
```

2.3.1.2 Méthodes générales

- **check_number_face()**
  la méthode **check_number_face** permet aux utilisateurs de vérifier qu'on soumet exactement un visage à la webcam. Cette méthode reçoit l'objet de la source de vidéo en paramètre. Elle renvoie **1** si on détecte exactement un visage et **0** dans le cas contraire.

- **set_cam_index()**
  cette méthode sauvegarde dans la base de données l'index de la source de flux vidéo définir à partir.

- **video_record()**
  la méthode video_record permet d'enregistrer et de sauvegarder le flux vidéo sur un support de stockage. Elle sert essentiellement dans le cas où une tentative d'usurpation d'identité a été détecté. Elle permet ainsi d'avoir une trace de l'attaque.

- **video_saver()**
  elle permet d'enregistrer dans une base de données, le chemin vers le flux vidéo sauvegarder sur un support de stockage avec la fonction video_record().

2.3.2 Exemple de détection d'usurpation d'identité avec Oudjat

L'algorithme 5 présente un processus de détection d'usurpation d'identité sur la reconnaissance faciale.

Algorithme 5 : Algorithme de la détection d'usurpation d'identité

Input : $knowFaces, fluxWebcam$;
Output : $Authorized\_access, Access\_denied, Error\_to\_start\_detection$;

```
1  knowFaces ← load_from_db;
2  fluxWebcam ← initialize;
3  if fluxWebcam exist AND load_from_db success then
4      while video ← fluxWebcam do
5          // Fonction de l'algorithm 2
6          if faceRecognition() = 1 then
7              while 10 secondes do
8                  // Fonction de l'algorithm 1
9                  // Fonction de l'algorithm 3
10                 if blinkDetection() = 1 AND objectDetection() = 1 then
11                     // Fonction de l'algorithm 4
12                     if textureAnalysis() = 1 then
13                         return Authorized_access
14                     else
15                         return Access_denied
16                     end
17                 else
18                     return Access_denied
19                 end
20             end
21             return Access_denied
22         else
23             return Access_denied
24         end
25     end
26 else
27     return Error_to_start_detection
28 end
```

Pour démarrer, comme l'illustre la ligne 1 de l'algorithme, un modèle du visage de l'utilisateur enregistré dans la base de données est chargé en mémoire (ligne 1) pour accélérer le processus et le flux de la video est initialisé. Ensuite, elle vérifie si le chargement du visage de l'utilisateur depuis la base de données et le démarrage du flux de la webcam se sont effectués avec succès.

En effet, le système à travers la ligne 4 récupère le flux de vidéo de la webcam et en fonction des images qui forment la vidéo, il identifie le visage présent sur les images pour une comparaison à travers l'appel de la fonction faceRecognition() à la ligne 6. Lorsque le visage

détecté est reconnu, la programme poursuit son exécution, sinon le processus s'achève à la ligne 23.

Comme l'indique à la ligne 7, durant 10 secondes, on vérifie le clignement des yeux de l'utilisateur et la présence d'objet dit dangereux dans son environnement (ligne 10). En cas d'absence de clignement des yeux ou de présence d'objet dangereux, l'accès est coupé à l'utilisateur (ligne 18).

La dernière étape du processus de détection d'usurpation d'identité est l'analyse de la texture du visage extrait du flux de vidéo grâce à la fonction textureAnalysis() (ligne 12). L'accès est autorisé à l'utilisateur comme l'indique la ligne 13, lorsqu les données issues de l'analyse de la texture confirme qu'il s'agit de l'utilisateur et non d'une reproduction de son visage (masque ou vidéo contenant le visage). Dans le cas où il s'agit d'une reproduction, l'accès est refusé à l'utilisateur (ligne 15).

2.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Il permet de décrire l'interaction entre le ou les acteurs et le système comme l'illustre la figure 2.2.

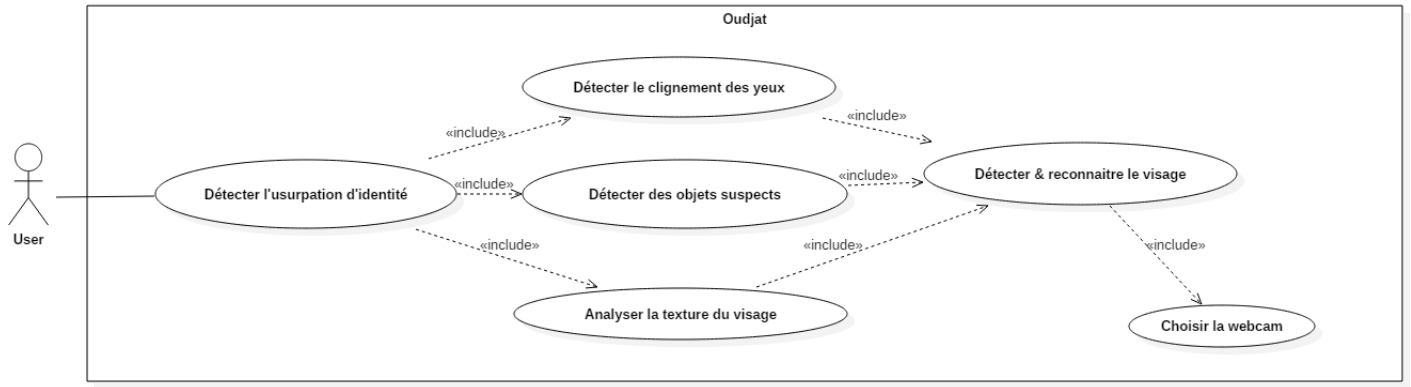


FIGURE 2.2 – Diagramme de cas d'utilisation

L'utilisateur pour effectuer de la détection d'identité, doit passer par la manipulation de certaines fonctions essentielles : la détection du clignement des yeux, la détection d'objets dangereux et l'analyse de la texture. La manipulation de ces fonctions nécessite deux étapes importantes : le choix de la webcam et le détection puis reconnaissance du visage. Ainsi l'utilisateur débute son travail par le choix de la webcam à utiliser. Ensuite, il détecte et reconnaît le visage soumit à la webcam. C'est donc sur ce visage que se fera la manipulation des fonctions sur-citées. Aucune opération n'est donc possibles si ces deux étapes échouent.

Conclusion

Dans ce chapitre, nous avons présenté les choix techniques opérés ainsi que notre solution à travers sa modélisation, son principe de fonctionnement et les outils utilisés. Le chapitre suivant exposera les différents résultats et quelques discussion.

Chapitre 3

Résultats et Discussions

Introduction

Dans ce chapitre, nous présenterons dans un premier temps, les résultats issus de l'utilisation des fonctionnalités de notre API pour détecter les cas d'usurpation d'identité et nous aborderons ensuite une discussion.

3.1 Authentification d'un utilisateur légitime avec Oudjat

L'utilisateur légitime du système accède à la ressource informatique en effectuant la reconnaissance faciale.

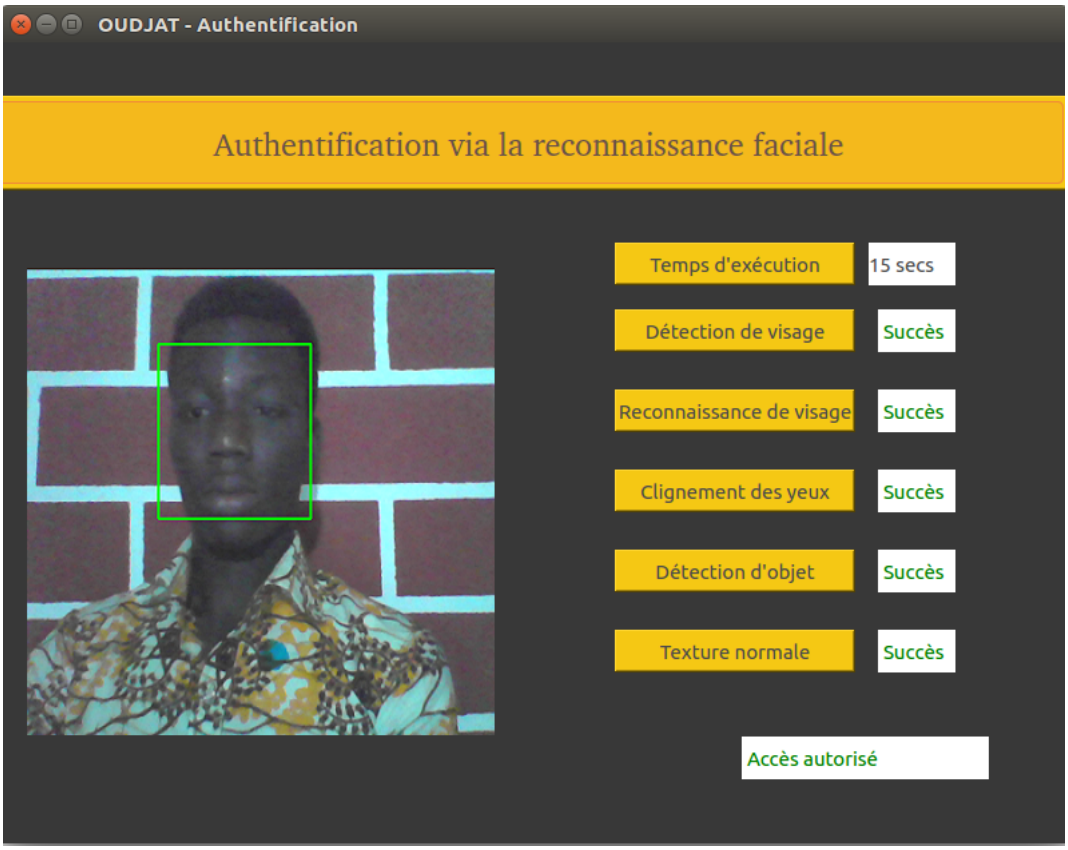
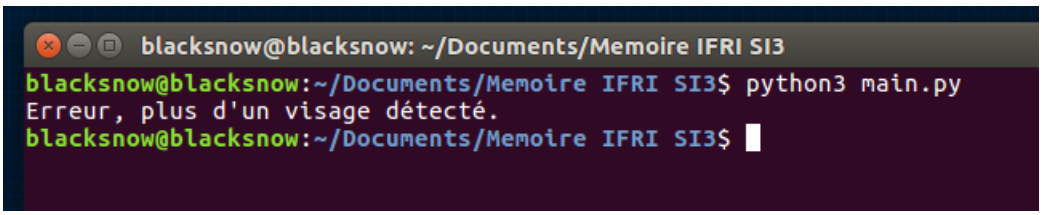


FIGURE 3.1 – Cas pratique- Authentification via la reconnaissance faciale

Dans le cas d'une intégration au sein d'un Système d'Information(SI), cette étape marque la fin du processus d'authentification et peut donc donnée accès à la ressource demandée.

3.2 Détection de l'usurpation d'identité avec Oudjat

3.2.1 Cas de détection de plus d'un visage



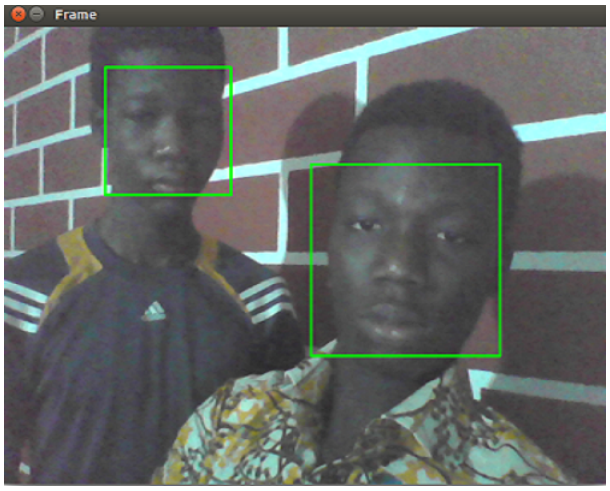


FIGURE 3.2 – Cas pratique- Détection de plusieurs visages

L'authentification de l'utilisateur légitime échoue lorsque plusieurs visage sont détectés. Dans le but de faciliter la compréhension de la présentation des résultats de la fonction en charge de la détection des visages, nous avons décidé d'afficher une frame de sortie du cas de la détection de plusieurs visages avec un message d'erreur. Notre application intègre cette fonctionnalité mais aucun affichage de cette frame n'est fait ; tout se passe en back-end.

3.2.2 Cas d'une usurpation d'identité avec une photo

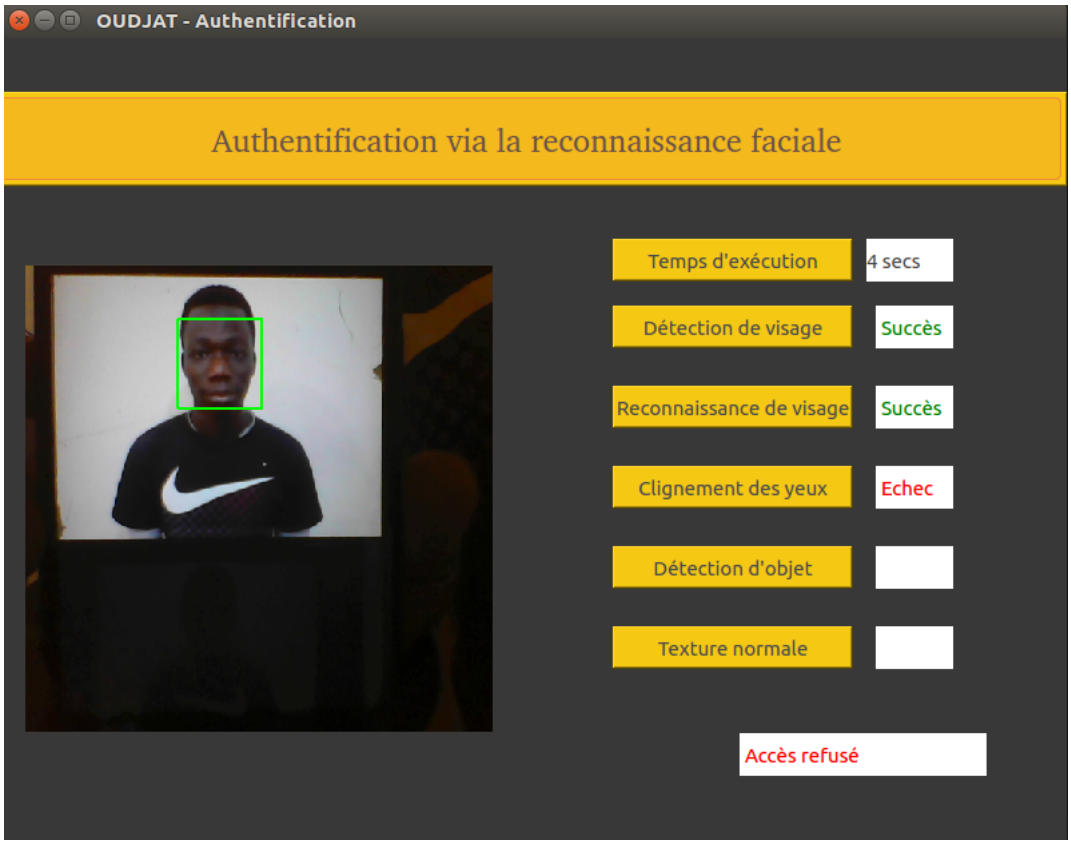


FIGURE 3.3 – Cas pratique- Détection d'une usurpation d'identité avec une photo

L'authentification de l'utilisateur a échoué parce qu'aucun clignement des yeux n'a été détecté. Il s'agit ainsi d'une photo du visage de l'utilisateur légitime qui a été soumit à la webcam.

3.2.3 Cas d'une usurpation avec une vidéo

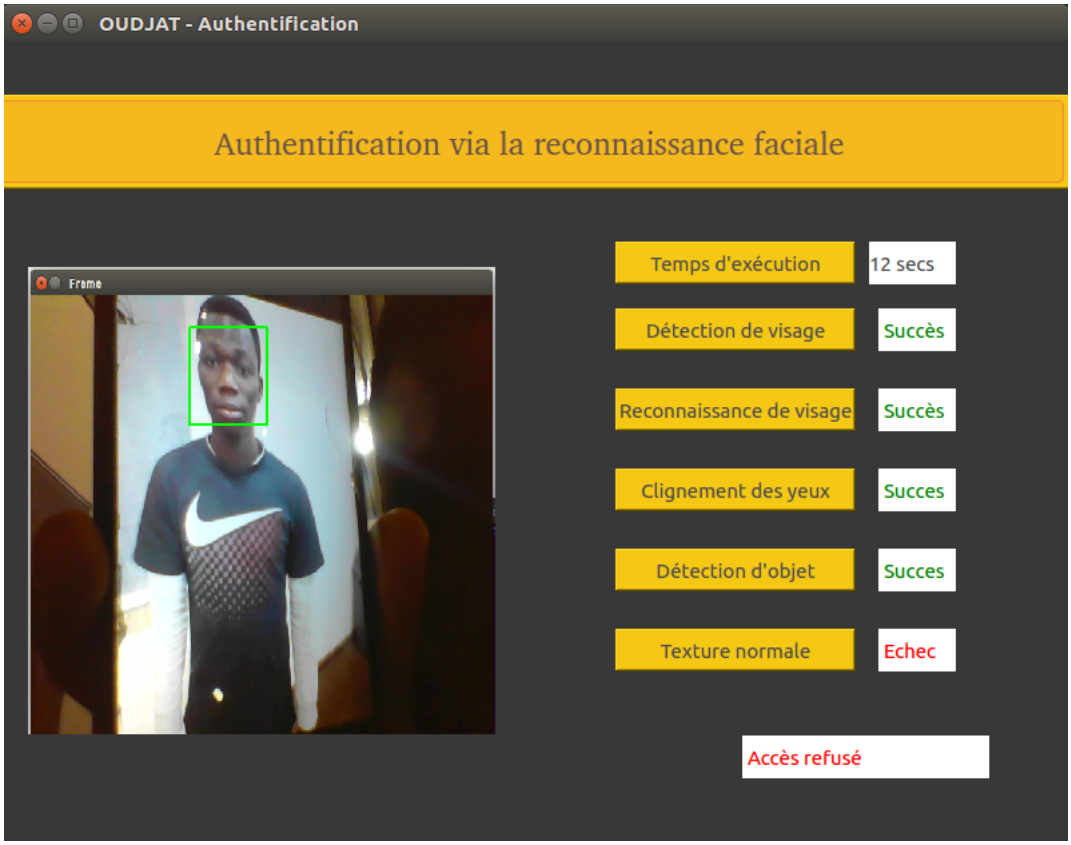


FIGURE 3.4 – Cas pratique- Détection d'une usurpation d'identité avec une vidéo

Dans ce cas d'utilisation nous testons chaque fonctionnalité de notre API. Ainsi l'authentification réussie lorsque toutes les étapes (fonctionnalités) réussissent. L'authentification de l'utilisateur a échoué parce que la texture du visage détecté correspond à celle d'un visage provenant d'un écran interposé entre le visage et la webcam. Il s'agit donc d'une vidéo contenant le visage de l'utilisateur légitime qui a été soumis à la webcam.

3.3 Discussion

Notre projet est né de plusieurs constats généraux de l'ordre de la sécurité informatique pour la protection des infrastructures utilisant la reconnaissance faciale comme moyen d'authentification.

Notre solution apporte une amélioration aux différents systèmes de reconnaissance faciale. Elle offre à travers plusieurs fonctions regroupées dans l'API, la possibilité de détecter

des clignements des yeux, d'analyser la texture d'une image et de vérifier l'attention du visage de l'utilisateur. Ainsi, notre solution est simple et facile à utiliser. Étant une API libre, elle est ouverte au grand public de développeurs implémentant le reconnaissance faciale dans leur projet. Elle implémente également une fonctionnalité de traçabilité des actions de tentatives d'usurpation d'identité.

Notre solution vient ainsi renforcer la sécurité autour de ces solutions existantes en permettant d'assurer la défense contre les différentes attaques d'usurpation d'identité notamment les attaques photos, attaques aux masques 3D et attaques vidéos sur les systèmes de reconnaissance faciale.

Néanmoins, loin de penser que notre solution est parfaite, nous sommes conscient qu'il reste encore des points à développer car certaines fonctions de la solution que nous avons proposée dans ce projet de mémoire, ne sont pas multiplateformes. Egalement, il reste à améliorer la performance en matière de la prédiction liées à la reconnaissance faciale et l'analyse de texture dans les conditions de changement de couleurs et de forte luminosité. Nous travaillerons également à améliorer l'apprentissage du module de détection des objets avec opencv et le deep learning. Une mesure de sécurité manquante à notre API est la vérification de l'attention de l'utilisation à travers la direction de son regard par la position de la pupille. Cette fonctionnalité a été implémentée mais n'a pas été intégré à notre API due à son instabilité.

Conclusion

Dans ce chapitre, nous avons exposé les résultats des tests de notre application et réalisé quelques critiques concernant ses performances et ses insuffisances. Ces insuffisances peuvent être perçues comme des perspectives afin d'améliorer le travail fait pour une utilisation plus efficiente.

Conclusion et Perspectives

La sécurité informatique passe par la sécurité de l'information. La sécurité des infrastructures informatiques n'est plus aujourd'hui un sujet inconnu. Aussi le hacking constitue aujourd'hui une arme de guerre assez dévastatrice et silencieuse. Dans cette vision, beaucoup d'outils de sécurité se développent de jour en jour afin de restreindre le champ d'attaque des pirates informatiques qui s'arment davantage.

L'objectif de ce mémoire a été de mettre en place un système de détection de l'usurpation d'identité sur les systèmes de reconnaissance faciale. Les solutions proposées proviennent des analyses effectuées sur les systèmes de reconnaissance faciale peu fiable. Ainsi les utilisateurs de notre application peuvent être protégés des différentes attaques que nous couvrons.

Dans ce mémoire, nous avons tout d'abord fait une revue de littérature autour de la sécurité informatique et plus précisément sur le concept de contrôle d'accès et d'authentification. Nous avons ensuite abordé les systèmes d'authentification existant tout en mettant en relief leurs insuffisances. Par la suite, nous avons réalisé la conception de la solution que nous avons proposée avant d'exposer les résultats et une discussion sur l'application que nous avons implémentée.

Bien que notre solution répond au besoin énoncé, il faut noter certaines insuffisances telles que la lenteur de l'application. Malgré que cela soit fait pour éviter l'accès frauduleux à toutes infrastructures informatiques via la reconnaissance faciale, il serait préférable de déployer la solution sur des ordinateurs spécifiques avec de bonne puissance de calcul et équipé de webcam de haute résolution.

Un autre axe de recherche en ce qui concerne les travaux futurs sera d'explorer la possibilité d'implémenter un module d'entraînement basé sur le deep learning afin de rendre plus efficace la détection d'objet et d'approfondir la vérification de la texture du visage photographié en se basant sur l'apprentissage approfondi et sur une grande base d'images. Dans les futurs travaux, nous aurions à améliorer la fonctionnalité de vérification de l'attention de l'utilisateur afin qu'elle soit intégrée aux prochaines versions de l'API.

Bibliographie

[1] A. R. A. Martelli and D. Ascher. *Python cookbook.* O'Reilly Media, Inc., 2005.

[2] M. M. C. André Anjos and S. Marcel. *Motion-Based Counter-Measures to Photo Attacks in Face Recognition*. Idiap Research Institute - Centre du Parc, rue Marconi 19, 1920 Martigny, Switzerland.

[3] J.-B. G. Beuque. *Authentification of data in a digital transmission system*. Google Patents, 2010.

[4] G. Bradski and A. Kaehler. *Learning OpenCV : Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[5] A. Gachet. *Introduction à UML*. O'Reilly, 2003.

[6] A. Geitgey. Face recognition. https://readthedocs.org/projects/face-recognition/downloads/pdf/latest/, Juin 2017.

[7] T. Pascal. *Methodes D'authentification*. Ed. Techniques Ingénieur, 2009.

[8] T. Soukupova and J. Cech. *Real-Time Eye Blink Detection using Facial Landmarks*. 2016.

Webographie

[9] M. Agile. https://blog.trello.com/fr/methode-agile-scrum-gestion-projet, consulté le 20 juillet 2018.

[10] T. I. Alerte à la fraude Phishing. https://www.thawte.fr/assets/documents/guides/new-phishing-tactics-2013.pdf, 2018.

[11] API. https://www.scriptol.fr/programmation/api.php, consulté le 25 octobre 2018.

[12] O. API. https://github.com/godwineHoungavou/oudjat-API, 03 décembre 2018.

[13] S. M. S. C. o. A. API Types. https://www.slideshare.net/sarahmaddox/api-types, consuté le 25 octobre 2018.

[14] E. D. Characteristics of Good APIs. http://www.monitis.com/blog/characteristics-of-good-apis/, consuté le 26 octobre 2018.

[15] Keylemon. https://keylemon.fr.softonic.com/, consulté le 12 septembre 2018.

[16] M. Online. https://www.memoireonline.com/02/11/4274/m_Authentification-de-visages-par-la-methode-danalyse-discriminante-lineaire-de-Fischer5.html, 03 décembre 2018.

[17] A. propos de la technologie avancée Face ID. https://support.apple.com/fr-fr/HT208108, consulté le 26 octobre 2018.

[18] PyImageSearch. https://www.pyimagesearch.com/, consulté en août 2018.

[19] F. Recognition. https://www.astucetech.net/top-10-applications-de-reconnaissance-faciale-pour-android-et-ios/, consulté le 13 septembre 2018.

[20] Weave. https://weave.eu/cybersecurite-sommes-moyens-dauthentification/, consulté le 7 Juin 2018.