

Promotion 2025 – 4<sup>ème</sup> année cycle ingénieur  
Majeure Ingénierie Financière, Parcours Recherche



Pôle Judiciaire de la Gendarmerie Nationale  
Service Central de Renseignement Criminel  
Pontoise

Tuteur école : Jean-Philippe Lelièvre  
Tuteur entreprise : Frédérick Réhault

## Création d'un outil logiciel de traitement d'images faciales

RAJESWARAN Apisan

Date de la soutenance : 12/09/2024  
Document non confidentiel

## **Sommaire**

## **I-Positionnement du stage**

### **A) Pôle Judiciaire de la Gendarmerie Nationale**

La Gendarmerie Nationale est une force armée instituée pour veiller à l'exécution des lois. La police judiciaire constitue l'une de ses missions essentielles. La gendarmerie nationale est destinée à assurer la sécurité publique et l'ordre public, particulièrement dans les zones rurales et périurbaines, ainsi que sur les voies de communication. Elle contribue à la mission de renseignement et d'information des autorités publiques, à la lutte contre le terrorisme, ainsi qu'à la protection des populations. L'ensemble de ses missions civiles et militaires s'exécute sur toute l'étendue du territoire national, ainsi qu'en haute mer à bord des navires battant pavillon français. À l'étranger, la gendarmerie nationale a des missions en lien avec les engagements internationaux de la France.

Le Pôle judiciaire de la Gendarmerie nationale (PJGN) est une entité spécialisée dans les domaines de la criminalistique et de l'intelligence judiciaire, ayant une compétence qui couvre l'ensemble du territoire français. Créé le 1er janvier 2011, il regroupe environ 600 professionnels (civils et militaires) et dispose de la capacité d'intervenir directement sur le terrain lors des incidents les plus graves. Le PJGN intervient principalement à la demande des unités de gendarmerie, telles que les brigades territoriales, les brigades de recherches et les sections de recherches, afin de résoudre des affaires judiciaires complexes ou nécessitant des investigations approfondies. De manière exceptionnelle, il peut également prêter assistance à une unité de la Police nationale, sur demande de celle-ci ou d'un magistrat.

Initialement basé au fort de Rosny à Rosny-sous-Bois, le PJGN a été relocalisé en mai 2015 dans de nouvelles installations situées au quartier militaire Lange à Pontoise, dans le Val-d'Oise. Depuis le 1er août 2022, le PJGN est dirigé par le général de division Gilles Martin.

Le PJGN se subdivise en plusieurs services : l'Institut de Recherche Criminelle de la Gendarmerie Nationale (IRCGN), le Service de Renseignement Criminelle de la Gendarmerie Nationale (SCRCGN), le Centre forensique d'Intelligence Artificielle (CFIA), l'état-major et l'administration.

Ainsi, les missions effectuées par le Pôle sont très diverses et recouvrent la totalité des domaines scientifiques en jeu dans les enquêtes les plus poussées. Pour avoir une idée de cette étendue voici une liste non exhaustive des expertises présentes : micro-analyse, incendies, explosifs, toxicologie, signal, imagerie, parole, véhicule, anthropologie, balistique. D'ailleurs pour cette dernière le PJGN abrite aussi une « bibliothèque » un peu particulière. Plus de 10 000 armes à feu y sont stockées, dont la plus ancienne date de 1720 et la plus insolite est une arme faite maison et utilisée par son inventeur pour se suicider. Y sont aussi conservées deux millions de munitions qui servent aux gendarmes pour identifier celles trouvées sur les scènes de crimes.

## **A) Le Service Central de Renseignement Criminel**

Le Service central de renseignement criminel (SCRC), successeur du service technique de recherches judiciaires et de documentation (STRJD) créé en 1976, a su évoluer pour s'adapter aux mutations de la délinquance, aux changements législatifs concernant les fichiers, ainsi qu'aux avancées des méthodes de traitement des données et des statistiques. Le SCRC est aujourd'hui à la pointe de technologies telles que l'intelligence artificielle et le big data.

Il constitue la composante d'intelligence judiciaire du Pôle judiciaire de la Gendarmerie nationale (PJGN), aux côtés de l'institut de recherche criminelle (IRCGN), qui représente la composante d'expertise scientifique. Mis à la disposition de l'ensemble des forces de sécurité intérieure, le SCRC joue un rôle crucial dans l'élaboration et la mise en œuvre des stratégies de lutte contre la criminalité de la gendarmerie.

Le SCRC remplit quatre missions principales au profit direct des unités territoriales et de recherche de la gendarmerie nationale, ainsi que des services de la police nationale :

### **1. Administration :**

- Gestion des fichiers communs à la police et à la gendarmerie nationale, tels que ceux relatifs aux antécédents judiciaires, aux empreintes digitales, aux personnes recherchées, et aux véhicules et objets volés.
- Gestion des bases de connaissances mises en œuvre par la gendarmerie nationale.

### **2. Élaboration du renseignement criminel :**

- Collecte, enrichissement, et analyse des informations centralisées via le réseau des unités d'appui judiciaire régionales et départementales de la Gendarmerie nationale.
- Compréhension des phénomènes criminels et des groupes impliqués, analyse de leur impact sur les territoires, anticipation de leur implantation, et conseil aux commandants territoriaux pour réguler ou neutraliser l'action des groupes criminels organisés.

### **3. Projection opérationnelle :**

- Engagement dans des investigations complexes, tant sur le terrain physique que dans le cyberspace.

- Mobilisation de capacités d'investigation et de techniques spécifiques pour soutenir les unités de recherche de la Gendarmerie nationale confrontées à des affaires de criminalité organisée.
- Utilisation de techniques avancées telles que l'analyse comportementale, l'investigation biométrique, et le traitement de la vidéo de masse, ainsi que l'expertise dans des domaines spécialisés comme les armes à feu, les explosifs, les trafics de véhicules, les affaires non élucidées, et les fraudes à l'identité.

#### 4. **Innovation :**

- Le SCRC, en tant que pôle d'innovation, dispose de capacités de recherche et développement, notamment grâce à des datascientists et ingénieurs mais surtout grâce à ses liens étroits avec le Centre Forensique d'Intelligence Artificielle (CFIA).
- Collaboration avec le monde de la recherche et partenaires externes pour développer des solutions techniques innovantes dans les domaines des sciences des données, de la maîtrise de l'espace numérique, et du véhicule connecté, au service des enquêtes et des opérations de sécurité intérieure.

L'organisation du SCRC se structure autour de quatre divisions et d'un centre spécialisé :

- La **Division des fichiers (DF)**, chargée de l'administration des fichiers nationaux de police judiciaire.
- La **Division des affaires non élucidées (DiANE)**, qui regroupe les compétences criminalistiques de l'IRCGN et les capacités d'investigation du SCRC pour travailler sur les affaires non élucidées dites « Cold Case ».
- La **Division des opérations (DO)**, qui répond aux demandes de rapprochements, détecte les phénomènes criminels, et appuie les unités en enquête.
- La **Division du renseignement (DR)**, qui élabore et diffuse le renseignement criminel.
- Le **Centre des sciences de la donnée (CSD)**, sollicité pour les traitements de données relatifs au renseignement criminel, visant à soutenir les enquêtes et à aider à la prise de décision pour les commandants confrontés à des phénomènes criminels complexes.

C'est au sein de la Division des fichiers du SCRC qu'a eu lieu mon stage même si j'ai été sous le patronage technique du CFIA la majeure partie du temps.

### **B) Le Centre Forensique d'Intelligence Artificielle**

Le Centre Forensique d'Intelligence Artificielle (CFIA) du PJGN est une unité innovante dédiée à l'application de l'intelligence artificielle dans les enquêtes criminelles, la forensique désignant l'ensemble des méthodes d'analyse fondées sur la science afin de servir au travail d'investigation judiciaire. Ce centre se focalise donc sur le développement et l'intégration de technologies avancées pour améliorer les capacités d'analyse et d'investigation de la Gendarmerie. Il est composé de deux officiers commissionnés : le LCL Daniel CAMARA et le CNE Pierre FICHEPOIL, ainsi que d'un personnel civil : Camille MAGNOSI.

Le CFIA collabore étroitement avec d'autres services de la Gendarmerie ainsi que des entités externes, tant au niveau national (Police Technique et Scientifique) qu'internationale (Bundespolizei), pour partager des connaissances et des technologies. Il représente une avancée significative dans la manière dont les forces de l'ordre utilisent les technologies de pointe pour résoudre les crimes et améliorer la sécurité publique.

Le CFIA a été distingué à plusieurs reprises pour ses contributions exceptionnelles dans le domaine de l'intelligence artificielle appliquée aux enquêtes criminelles notamment en tant que « Plus grand contributeur » à la plateforme Europol pour le développement d'outils comme RosetAI permettant de fournir une transcription de bonne qualité et une traduction multilingue et a aussi obtenu un Prix d'Excellence Europol du projet le plus innovant pour la création d'une plateforme d'outils d'intelligence artificielle qui inclut des outils pour l'analyse d'empreintes digitales, la détection d'armes et de drogues sur des photos ou encore la transcription automatique de la parole en texte (lors des dépôts de plaintes par exemple).

### **C) Institut de Recherche Criminelle de la Gendarmerie Nationale**

L'IRCGN est une unité spécialisée en matière de criminalistique et d'expertise de police scientifique (étant le seul des 7 laboratoires de police scientifique français sous les couleurs de la Gendarmerie). Il est composé de plusieurs laboratoires spécialisés dans divers domaines tels que la biologie, la chimie ou encore la balistique. L'IRCGN intervient dans la collecte, l'analyse et l'interprétation des preuves scientifiques, apportant une aide précieuse aux enquêtes criminelles et aux magistrats. Grâce à ses équipements de pointe et à son expertise technique, il joue un rôle crucial dans la résolution des affaires complexes de premier plan.

### **D) Problématique**

Au sein du SCRC le Département du fichier d'antécédents judiciaires de la Division des fichiers est en charge de la manipulation d'une application appelée TAJ (pour Traitement d'Antécédents Judiciaires). Cette application possède notamment un logiciel de reconnaissance faciale intégré développé par Cognitech, une entreprise allemande pionnière dans ce domaine depuis 20 ans maintenant. Pour ce faire le Groupe « image et rapprochement par photographies » dispose de photos faciales (aussi appelées photos anthropométriques) prises par les unités de terrain que l'on peut envoyer au TAJ. Le TAJ va ensuite comparer cette image d'interrogation à toutes les images anthropométriques dont dispose la Gendarmerie, cette base d'images étant composée de toutes les personnes ayant des antécédents judiciaires et donc connues des services de police et de Gendarmerie ou alors de personnes sous enquête. Le TAJ, sous réserve d'acceptation de la photo par le logiciel, sort une liste triée de 200 personnes susceptible de correspondre avec pour chaque proposition un score de correspondance, les Gendarmes ne gardant en général que les propositions avec un score supérieur à 98 %.

Il est important d'avoir en tête que le logiciel de Cognitech est une boîte noire pour la Gendarmerie : la Gendarmerie a acheté ce logiciel sous licence mais ignore tout de son fonctionnement et de ses défaillances.

Malheureusement, les opérateurs du TAJ ne disposent pas toujours de photos de grandes qualités. En effet, bien que la plupart des photos dont il dispose soient des « mugshots »

(photos d'identification normées, centrées sur le visage, prises lors des interrogations et des procès), il leur arrive également de travailler avec des photos de terrain issue d'enquête comme lors de filature ou des images de vidéosurveillance de piètre qualité. Lorsque les enquêteurs de la Gendarmerie rentrent ce type d'image dans le TAJ ce dernier a tendance à rejeter l'image en affichant une erreur. Les Gendarmes doivent donc retravailler sur un logiciel de retouche d'image (comme GIMP qui est libre et gratuit) la photo en ajustant notamment la luminosité, le contraste, la netteté ou plus simplement en recadrant l'image autour du haut du corps de la personne d'intérêt. Le problème pour la Gendarmerie étant la perte de temps engendrée par ce type de manipulations et que malgré les retouches certaines photos restent inexploitable pour l'enquête.

## **F) Responsabilité sociale et légale**

La Gendarmerie en tant qu'institution publique de premier de plan de la République et en tant que force armée chargée de missions de police s'engage fortement dans une politique de responsabilité sociétale et surtout légale, en particulier le PJGN. En effet, les magistrats et juges accordant une grande confiance envers les expertises apportées par le PJGN, il vient naturellement aux membres de ce dernier de fournir un travail exempt de tout reproche et d'être irréprochable sur le plan individuel pour éviter une perte de confiance envers l'institution. L'un des leitmotiv du Colonel Réhault qui était mon tuteur était de « ne rien faire qu'on ne puisse expliquer ». Dans le contexte du traitement d'images il s'agit là d'une mise en garde contre l'IA générative qui ne peut être utilisée pour faire un rapprochement par TAJ et utilisée lors d'un procès par exemple.

## **II-Présentation du stage**

### **A) faceComparison**

Mon stage n'est pas le premier à avoir lieu sur cette thématique bien précise du traitement d'images en vue de reconnaissance faciale. Il y a de cela plusieurs années un stagiaire du nom de Mohand a eu pour tâche de concevoir un logiciel permettant aux Gendarmes de retravailler les images manuellement plus rapidement et simplement, leur épargnant la maîtrise d'un outil lourd comme GIMP. Ce logiciel m'a servi de modèle pour comprendre ce que je devais automatiser et l'architecture d'une telle application (n'ayant jamais programmé l'entièreté d'un logiciel cela m'a bien orienté). Je vais donc le présenter pour que le lecteur ait une meilleure appréciation du déroulé de mon stage ainsi qu'une justification des choix que j'ai fait pour mon propre logiciel.

En apparence l'application s'ouvre comme une page Web nous laissant le choix entre deux fonctionnalités : Modification d'image et Comparaison d'image.

#### **Comparaison d'image :**

Il s'agit là du volet le moins intéressant pour nous car c'est une fonctionnalité qui n'a pas vraiment de lien avec le TAJ ou la reconnaissance faciale par ordinateur. En fait, il a été conçu pour les équipes de l'IRCGN qui dispose d'une unité spécialisée dans la reconnaissance faciale humaine. Ces experts vont en effet regarder des caractéristiques faciales comme la forme du nez, des lèvres ou des oreilles ainsi que des marqueurs comme des cicatrices, des tâches de naissance ou des grains de beauté pour rapprocher ou non deux individus. Il est important de noter qu'il s'agit d'une expertise purement humaine ne faisant intervenir aucun ordinateur. Ce type d'expertise peut par exemple être utilisé pour savoir si une personne une fois adulte correspond à un enfant qui a été kidnappé des décennies avant car ses caractéristiques faciales ne changent pas avec l'âge (sauf s'il y a des traumatismes physiques importants).

La page nous propose 2 zones de dépôt d'image différentes sur lesquelles nous pouvons soit cliquer et choisir les images d'intérêts dans le système de navigation soit faire un glisser-déposer. A noter qu'il est également possible de fournir les chemins absolus vers les deux images si nécessaire. Ensuite nous appuyons sur le bouton « Analyse » et les deux images apparaissent côte à côte pour pouvoir faciliter le travail des experts. Il y a également possibilité de zoomer sur chacune des images et même une fonctionnalité de zoom synchronisé.

#### **Modification d'images :**

Il s'agit de la fonctionnalité que je suis censé automatiser d'ici la fin de mon stage et celle qui a été conçue pour le SCRC en particulier. Encore une zone de dépôt fonctionnant de la même manière que décrite précédemment, sauf qu'il y en a qu'une. Il y a toujours la possibilité de rentrer le chemin absolu à la place de l'image directement. Il existe également des boutons permettant certaines fonctionnalités intéressantes comme un bouton de normalisation permettant d'ajuster la taille de l'image aux normes ISO utilisées par la Gendarmerie ou la possibilité de rentrer un facteur de zoom, initialisé par défaut à 1,3.



Lorsqu'on appuie sur « Analyse », l'image apparaît en dessous avec un menu déroulant comportant 5 barres dont on peut ajuster le curseur de -100 à +100 : Contraste, Luminosité, Exposition, Netteté, Saturation. Il y a également des boutons dans le menu permettant de rétablir tous les paramètres à 0 ou d'enregistrer l'image modifiée.

Sous le capot du logiciel, pour la partie technique, nous avons affaire à une application Python pour le backend utilisant le framework Flask pour communiquer avec le frontend qui est une page HTML animée par du JavaScript.

## **B) Bibliothèques utilisées**

**OpenCV** (Open Source Computer Vision Library) est une bibliothèque open-source largement utilisée pour le traitement d'images et la vision par ordinateur. Développée à l'origine par Intel, elle est maintenant maintenue par une communauté active de développeurs. En Python, OpenCV est particulièrement prisée pour sa simplicité d'utilisation et sa large gamme de fonctionnalités, allant des opérations de base sur les images (comme la conversion en niveaux de gris, le redimensionnement et la détection de contours) à des tâches plus complexes telles que la reconnaissance faciale, le suivi d'objets et la reconstruction 3D.

Grâce à son intégration fluide avec d'autres bibliothèques Python, telles que NumPy pour les opérations sur les matrices, OpenCV permet de créer des pipelines de traitement d'images efficaces et performants. De plus, elle offre des interfaces pour le traitement vidéo en temps réel, ce qui la rend idéale pour des applications telles que la surveillance, la robotique et l'analyse vidéo. Son adoption massive dans les domaines académiques et industriels en fait un outil incontournable pour tout projet impliquant la vision par ordinateur.

**RetinaFace** est un modèle de détection de visage basé sur les réseaux de neurones convolutifs (CNN), conçu pour détecter les visages dans des images ou des vidéos. Malgré son ancienneté (il a été développé en 2019), ce modèle reste particulièrement performant dans des applications de reconnaissance faciale et de suivi des visages montrant une robustesse exceptionnelle face à des conditions difficiles telles que les variations d'angle, les occultations partielles et les conditions d'éclairage non idéales.

RetinaFace s'inspire de l'architecture de RetinaNet, un modèle célèbre pour la détection d'objets. Cependant, il intègre des améliorations spécifiques pour la détection de visages, notamment la supervision multi-niveaux, qui permet une meilleure reconnaissance des visages de différentes tailles.

En plus de la détection des landmarks faciaux (yeux, nez, bouche), RetinaFace prédit également la boîte englobante du visage (les coordonnées du point en bas à gauche et du point en haut à droite du rectangle dans lequel est inscrit le visage) ce qui le rend extrêmement intéressant pour faire de l'extraction et du zoom sur les visages.

**Rembg** est une bibliothèque Python spécialisée dans la suppression automatique des arrière-plans d'images. Elle utilise des techniques de vision par ordinateur et des réseaux de neurones pour segmenter et isoler les sujets principaux des arrière-plans, produisant ainsi des images avec des arrière-plans transparents ou unis. Cette fonctionnalité est particulièrement utile pour des applications dans les domaines du commerce électronique, de la création de contenu visuel, et de la réalité augmentée, où il est souvent nécessaire de détacher des objets ou des personnes de leur environnement d'origine mais il semblerait que cette fonctionnalité ait également un intérêt en reconnaissance faciale.

Rembg se distingue par sa simplicité d'utilisation : une seule commande suffit pour traiter une image. Grâce à son algorithme basé sur des modèles d'apprentissage profond, elle offre une précision remarquable, même dans des situations complexes où les arrière-plans sont variés ou les contours des objets sont difficiles à distinguer.

La bibliothèque est également facilement extensible, ce qui permet de l'intégrer dans des pipelines de traitement d'image plus larges ou des applications en temps réel. En outre, elle supporte une large gamme de formats d'image et peut être utilisée pour traiter des lots d'images de manière efficace.

### **C) Objectif du stage**

Ce stage a pour objectif principal le développement d'un outil logiciel innovant dédié au traitement d'images faciales. Cet outil vise à répondre aux besoins spécifiques du DFAJ dont j'ai explicité les problématiques en amont. Le logiciel à concevoir devra ainsi permettre une exploitation efficace de des images rejetées par le TAJ à travers plusieurs fonctionnalités clés, tout en étant adaptable aux différentes plateformes (fixes ou mobiles) et sans dépendance à un système d'exploitation spécifique (les Gendarmes travaillant aussi bien sous Linux que sous Windows).

L'une des missions essentielles du projet consiste à évaluer le potentiel d'exploitation d'une image faciale en la comparant à une autre. Cela inclut l'évaluation de la qualité, de la netteté et de la conformité de l'image avec les standards requis pour une comparaison fiable. Le logiciel devra également intégrer une fonctionnalité semi-automatisée d'amélioration de la qualité des images. Cette étape est cruciale pour optimiser les images en vue d'une utilisation efficace, en corrigeant les défauts tels que le bruit, le flou ou les variations de luminosité.

Une autre mission clé consiste à redimensionner et calibrer les images faciales. Ce processus inclut le cadrage autour du visage, l'ajustement de la résolution et la normalisation des dimensions afin de garantir une comparaison précise. Le logiciel devra être capable de détecter des visages dans des photos ou des vidéos, puis de les extraire pour un traitement ultérieur. Cette extraction automatisée facilitera le travail de préparation des images pour la comparaison, rendant le processus plus efficace, moins coûteux en temps et en ressources humaines et moins sujet à l'erreur humaine.

Les livrables attendus de ce stage comprennent le développement du logiciel complet de traitement d'images faciales, une documentation technique détaillée pour assurer la maintenance et l'évolution du produit, ainsi qu'un rapport de stage et une présentation des résultats. Ces documents seront essentiels pour évaluer l'atteinte des objectifs du projet et pour en faciliter le déploiement au sein de la Gendarmerie.

### **III-Déroulement du stage**

#### **A) Découverte du Pôle, du projet et de la vision par ordinateur**

La première semaine de mon stage a été marquée par une introduction au Pôle au sein du CFIA dans le bâtiment du SCRC et au projet sur lequel j'allais travailler. Une fois arrivé j'ai tout de suite été immergé dans l'esprit militaire de haute sécurité qui règne par la découverte de toutes les procédures propres au Pôle : accès via un badge sécurisé physique, création d'un profil dans l'Intranet du Pôle, accès totalement interdit au Wifi, interdiction de certains sites Internet sensibles etc.

Une fois toutes ces consignes transmises par les officiers encadrant l'on m'a installé à mon poste de travail qui était un PC fixe, que le CFIA utilisait comme serveur de calcul pour entraîner leur modèles d'IA, équipé notamment de 3 GPU Nvidia GeForce RTX 4050 de dernière génération. Comme le Colonel Réhault était en déplacement toute la semaine je n'ai pas pu discuter plus en profondeur sur le projet avec lui mais le LCL CAMARA m'a chargé de prendre connaissance du logiciel faceComparison qui représentait alors le socle de ce que devait être le nouveau logiciel et surtout d'apprendre la programmation en HTML/CSS/JavaScript. En effet, les deux logiciels (l'ancien et le nouveau) étaient conçus comme des applications Web avec un traitement backend combinant le langage Python et le framework Flask. J'ai donc passé une bonne partie de ma première semaine de stage à apprendre les rudiments de ces langages et technologies et à les implémenter dans un petit projet (qui n'avait rien à voir avec le contenu du stage) pour manipuler et me familiariser avec la logique d'une telle application mêlant formulaire, requête et échange de fichier avec une page Web (en l'occurrence des images surtout).

La seconde partie de cette phase introductive a été la prise en main du logiciel faceComparison et l'immersion dans la documentation et le code fournis par le SCRC. La première problématique qui s'est présentée à moi fut celui des dépendances entre les bibliothèques utilisées et la version de Python sur ma machine (à savoir 3.10.12). En effet, Python étant un langage de programmation largement open-source et communautaire (basé sur une communauté de développeurs travaillant ensemble et souvent bénévolement à son développement), les mises à jour ne prennent pas en compte les dépendances avec les autres bibliothèques qui elles même ne se préoccupent guère des problèmes de connectiques avec les autres bibliothèques. Sachant qu'il est peu recommandé de changer de version de Python il ne nous reste plus que l'option de trouver une combinaison de versions des bibliothèques en jeu compatibles entre elles d'abord, puis avec notre version de Python. Avec l'aide du LCL CAMARA j'ai appris à gérer ce problème et nous avons trouvé un environnement compatible pour lancer faceComparison sous Python 3.10.12 comme voulu.

C'est donc en essayant de comprendre les fonctionnalités backend de faceComparison que j'ai découvert les principales bibliothèques de traitement d'images et de vision par ordinateur que sont respectivement OpenCV (Open Computer Vision) et Pillow, héritière de PIL (Python Imaging Library), qui servent toutes deux à lire des fichiers images en Python et à les manipuler. En cherchant à comprendre les différences entre ces deux outils qui semblent avoir les mêmes objectifs j'ai dû plonger en profondeur dans le domaine du traitement d'images et de la vision par ordinateur pour comprendre les fondements mathématiques de ces domaines.

Une image numérique matricielle (ou image bitmap) se représente mathématiquement comme une matrice (un tableau de nombres) ou un ensemble de 3 matrices aussi appelé tenseur. Prenons le cas d'une image en niveau de gris : chaque case de la matrice représente la valeur qu'est censé prendre le pixel correspondant à cette position, un pixel étant un petit rectangle constituant l'unité de base d'une image numérique et pouvant s'illuminer d'une unique couleur à un instant T. La valeur que peut prendre cette case (et le pixel associé) est un entier compris entre 0 (noir absolu) et 255 (blanc absolu). L'ordinateur en lisant case à case la matrice peut donc afficher l'image sur l'écran comme le montre l'exemple.

En ce qui concerne les images couleurs nous avons en réalité 3 matrices correspondant chacune à une des 3 couleurs primaires dont les cases prennent encore une fois valeur parmi les entiers de 0 (noir absolu) à 255 (couleur primaire en question à intensité maximale). L'ordinateur pour une case en particulier va lire les valeurs dans les 3 matrices de couleur à cette position et synthétiser par superposition une couleur à afficher sur l'écran.

Ainsi toutes opérations sur ces images représentent d'un point de vue formel une opération matricielle ou tensorielle comme une extraction de sous-matrice pour un recadrage ou même parfois un produit matriciel pour certaines opérations dites de filtrage.

Il existe de plus plusieurs modes de lecture des images numériques les plus courantes étant BGR (Blue Green Red) pour OpenCV et RGB (Red Green Blue) pour PIL qui constitue la principale différence entre ces deux bibliothèques et, comme nous le verrons plus tard, représente également un problème. En effet, toutes les bibliothèques et modèles de traitement d'image s'attendent à recevoir leurs images d'entrée sous un mode de lecture particulier expliquant pourquoi des erreurs de compilation peuvent apparaître dans le programme en fonction de la bibliothèque de lecture de ces mêmes images. On peut également citer d'autres espaces de couleur plus exotiques comme le HSV pour Hue (Teinte), Saturation et Value (valeur de luminosité) que nous n'utiliserons pas.

Il existe également une autre représentation des images numériques dite vectorielle composée d'objets géométriques individuels (segments de droite, arc de cercle, polygones etc) définis chacun par des différents attributs (forme, position, remplissage etc). Ces images ont l'avantage par rapport aux images matricielles de pouvoir être agrandies sans perte d'information ce qui est très intéressant pour zoomer par exemple. Par contre, il est très difficile de garder des informations importantes comme la couleur ou des traits mineurs qui sont pourtant extrêmement importants pour la reconnaissance faciale expliquant pourquoi je n'ai pas utilisé ce type de conversion pour ce stage. Les images vectorielles sont en revanche très utilisées en cartographie ou pour la réalisation de schémas.

Cette première période d'introduction s'est complétée par la visite de l'IRCGN, ce qui m'a donné une vision globale du rôle que joue la gendarmerie dans le traitement scientifique des affaires criminelles, et surtout d'un premier contact avec l'équipe du DFAJ. Un Gendarme du DFAJ a d'ailleurs pris le temps de me montrer le TAJ et avait préparé une liste d'images qui justement ne passaient pas, puis m'a montré en direct les transformations qu'il avait faites pour les faire passer, me donnant ainsi des pistes de réflexion sur l'automatisation du procédé (recadrage, ajustement de la luminosité, retirer le fond de l'image etc).

Finalement j'ai pu rencontrer le COL REHAULT et discuter du sujet et des attendus du stage. Le but est d'automatiser toute la partie de pré-traitement des images faciales pour faire gagner du temps aux équipes d'enquêteurs mais en respectant plusieurs critères. Tout d'abord, comme aime à le répéter le Colonel, il ne faut « rien que l'on ne puisse expliquer » signifiant

qu'aucune des opérations faites par le logiciel ne doit représenter une boîte noire pour la Gendarmerie qui doit être capable d'expliquer et de justifier ces mêmes opérations, devant la loi si nécessaire. Ainsi, cela exclu de facto tous les modèles d'IA générative d'image faisant de la Super Resolution ou de l'amélioration de qualité d'image par exemple. Le deuxième principe étant celui de la simplicité d'utilisation avec une véritable approche UX (User eXperience) : les Gendarmes qui seront amenés à utiliser ce logiciel n'ont pas forcément de formation en informatique et ne sont pas non plus forcément très à l'aise sur un ordinateur mais doivent quand même être capables de manipuler l'outil. Ce stage représente une baisse de complexité par rapport au logiciel faceComparison qui était assez complet mais également assez compliqué à utiliser pour un non-initié ce qui a empêché la mise en utilisation massive de ce dernier au sein des forces de l'ordre.

## **B) Travail de recherche académique et d'état de l'art**

Mon premier objectif a donc été de rassembler et de tester un maximum de technique d'amélioration d'image pour pouvoir faire une présélection avant de les tester. Il s'agit donc d'un travail d'état de l'art et de recherche académique sur le sujet.

Mon premier réflexe a donc était de tester les fonctionnalités de retouche natives des deux grandes bibliothèques de manipulations d'image à savoir OpenCV et PIL. J'en profite pour rappeler ici qu'il s'agit des mêmes fonctions qui permettent au logiciel faceComparison de modifier la luminosité, la saturation ou encore le contraste. Après plusieurs tests sur des images libres de droit récoltées sur Internet il apparaît que ces fonctions donnent des résultats satisfaisants à condition de bien choisir le ou les paramètres d'entrée. En effet, toutes ces fonctions natives ont un grand défaut : celui d'être paramétrique, c'est-à-dire que l'utilisateur doit rentrer manuellement un coefficient. Dans ma démarche d'automatisation cela représente un obstacle car les opérateurs ne peuvent pas perdre leur temps à trouver la bonne valeur de paramètre par tâtonnement d'autant plus que la plage de ces bonnes valeurs est très large et dépend énormément de la photo (d'après les tests que j'ai effectués). Mais j'ai quand même décidé de garder ces fonctions de côté pour une autre approche que je détaillerai plus tard.

Ensuite je me suis dirigé vers des modèles de Deep Learning issue de papiers de recherche (je cherchais des papiers intéressants sur Google Scholar et je récupérais le modèle correspondant sur le GitHub des chercheurs) pour voir ce qui existait.

Tout d'abord je suis tombé sur des modèles qui faisaient de la Super Résolution c'est-à-dire de l'amélioration de détails en utilisant de l'IA générative. J'ai rapidement dû abandonner cette voie à cause de la nullité juridique et éthique d'une image modifiée de la sorte. En effet, le modèle en augmentant la résolution va compléter les pixels manquants en proposant la valeur plus probable en fonction du jeu de données sur lequel il a été entraîné. Or cela signifie que le visage que l'on obtiendrait serait donc un mélange entre le visage cible et un « visage moyen », et donc pas notre cible.

Finalement, je suis tombé sur des méthodes d'amélioration de contraste et de luminosité automatiques en fouillant un peu les bibliothèques de vision par ordinateur et les forums dédiés. Deux procédés ont particulièrement éveillé mon attention.

**Approche par histogramme :**

L'histogramme d'une image numérique est un graphique qui représente la distribution des niveaux de gris ou des couleurs d'une image. Chaque niveau de gris (ou de couleur, dans le cas d'images en couleur) est comptabilisé, et le nombre de pixels correspondant à chaque niveau est tracé sur l'axe vertical, tandis que les niveaux de gris sont représentés sur l'axe horizontal. Par exemple, pour une image en niveaux de gris, l'histogramme montre combien de pixels de l'image ont une intensité de 0 (noir), de 1, de 2, etc., jusqu'à 255 (blanc) dans une image sur 8 bits.

Un histogramme permet d'analyser la luminosité et le contraste d'une image. Un histogramme concentré à gauche indique une image sombre, tandis qu'un histogramme concentré à droite indique une image claire. Un histogramme bien réparti suggère un bon contraste, tandis qu'un histogramme très étroit indique un faible contraste, en général.

**L'égalisation d'histogramme** est une technique de traitement d'image visant à améliorer le contraste d'une image en redistribuant l'intensité des pixels. L'objectif est de produire une image dont l'histogramme est aussi plat que possible, c'est-à-dire où les niveaux de gris sont répartis uniformément sur toute la gamme des valeurs possibles. Cela permet de révéler plus de détails dans les zones sombres et claires de l'image. Comme mentionné plus ce type d'égalisation d'histogramme n'est possible que pour les images en niveau de gris ce qui suppose une conversion en amont pour les images en couleurs.

**L'égalisation d'histogramme adaptative** (AHE pour Adapted Histogram Equalization) est une extension de l'égalisation d'histogramme simple (cette fois applicable aux images colorées) qui vise à améliorer localement le contraste de l'image, en particulier dans des zones où le contraste est faible. Plutôt que d'appliquer l'égalisation à l'ensemble de l'image, l'AHE divise l'image en petites régions, ou *tiles*, et applique l'égalisation d'histogramme à chacune d'elles séparément. Cela permet d'ajuster le contraste en fonction des caractéristiques locales de l'image.

Cependant, une limitation de l'AHE est qu'elle peut amplifier le bruit dans les régions uniformes de l'image. Pour pallier ce problème, une version améliorée appelée CLAHE (Contrast Limited AHE) est souvent utilisée. Le CLAHE limite l'amplification du contraste en appliquant un seuil à l'histogramme avant de réaliser l'égalisation, réduisant ainsi l'effet du bruit. Après plusieurs tests j'ai décidé de garder cette dernière au détriment de l'AHE basique.

### **Théorie Retinex :**

La théorie Retinex est un modèle perceptuel développé pour expliquer comment l'œil humain perçoit les couleurs de manière constante, malgré les variations d'éclairage. Proposée par Edwin Land dans les années 1960, cette théorie s'intéresse à la manière dont nous percevons les couleurs et à la constance des couleurs, c'est-à-dire notre capacité à percevoir une couleur comme étant la même, même sous des conditions d'éclairage différentes.

Le terme "Retinex" est une combinaison des mots "rétine" et "cortex", soulignant l'idée que la perception des couleurs résulte d'une interaction entre les processus qui se déroulent dans la rétine (l'œil) et ceux du cortex (le cerveau). Contrairement à l'idée que la perception des couleurs est simplement une réponse directe aux longueurs d'onde de la lumière captée par les photorécepteurs de la rétine, la théorie Retinex propose que le cerveau compare les informations de couleurs reçues de différentes parties d'une scène pour déterminer la couleur perçue d'un objet.

Un aspect fondamental de la théorie Retinex est le phénomène de constance des couleurs. Par exemple, une pomme rouge apparaît rouge aussi bien sous une lumière de jour blanche que sous une lumière artificielle jaunâtre. La théorie Retinex explique ce phénomène en suggérant que le cerveau évalue la couleur d'un objet en tenant compte de l'éclairage ambiant et en comparant la lumière réfléchie par cet objet à celle réfléchie par d'autres objets dans l'environnement.

Selon la théorie Retinex, le cerveau réalise ce traitement en analysant la scène à travers trois "chemins" ou canaux, correspondant aux trois types de cônes sensibles aux différentes longueurs d'onde (court, moyen, long) dans l'œil humain. Pour chaque canal, le cerveau compare l'intensité de la lumière réfléchie par un point particulier à celle des points environnants, ce qui permet de neutraliser les effets de l'éclairage variable et de maintenir une perception constante de la couleur.

La théorie Retinex a des applications dans le domaine de la vision par ordinateur et de l'imagerie numérique. Elle est utilisée pour développer des algorithmes de traitement d'image qui corrigent les couleurs afin de reproduire une perception humaine plus fidèle des couleurs, indépendamment des variations d'éclairage, corrigeant ainsi les photos sous-éclairées. Il apparaît également que ce procédé puisse servir à corriger des problèmes de surexposition dans certains cas. Pour notre projet nous nous sommes intéressés à deux implémentations différentes de cette théorie.

Le **Retinex à échelle unique** (SSR pour Single Scale Retinex) applique un algorithme de traitement d'image basé sur une seule échelle spatiale. Cela signifie qu'il utilise un filtre (généralement une fonction gaussienne) de taille fixe pour comparer les valeurs de luminosité d'un point spécifique de l'image avec celles des pixels environnants. Ce processus permet d'estimer l'éclairage et de normaliser la couleur d'un pixel en fonction de son environnement immédiat. Le seul argument d'entrée est la variance de la fonction gaussienne qui nous sert de filtre.

Le **Retinex à échelles multiples** (MSR pour Multi Scale Retinex), comme son nom l'indique, applique plusieurs SSR sur différentes échelles spatiales, c'est-à-dire avec des filtres de tailles variées. Chaque échelle traite l'image en capturant différents niveaux de détail (des plus fins aux plus grossiers). Les résultats obtenus de ces différentes échelles sont ensuite combinés pour produire l'image finale. Il faut donc 3 arguments correspondant chacun à la variance d'un filtre spatial.

Le lecteur aura sans doute noté qu'il s'agit là de deux méthodes paramétriques ce qui semble être contradictoire avec notre idée d'automatisation de la tâche mais ce n'est pas le cas ici. En effet, les chercheurs proposent des valeurs par défaut pour ces paramètres qui étonnamment fonctionnent très bien sur la plupart des images de tests. Souvent lorsque le SSR ne donne pas de résultats satisfaisants le MSR réussit, et vice versa. Il existe une littérature abondante sur la manière de calibrer le Retinex (trouver les valeurs optimales des paramètres en fonction d'un jeu de donnée) mais le modèle par défaut fonctionnant bien je n'ai pas creusé cette solution.

### **C) Mesure de qualité d'image et approche combinatoire**

Une alternative envisagée à l'utilisation de fonctions non-paramétriques était de générer à partir d'une image d'entrée des dizaines voire des centaines d'images modifiées par les fonctions paramétriques natives prenant des combinaisons différentes de paramètres (d'où

le nom de cette approche) et d'ensuite trouver un moyen de noter la qualité des images pour faire ressortir les 2 ou 3 meilleures seulement.

L'évaluation de la qualité d'image faciale (FIQA pour Face Image Quality Assessment en anglais) est un domaine assez récent de la recherche en vision par ordinateur appliquée à la reconnaissance faciale qui consiste à attribuer une note à une image biométrique pour évaluer sa qualité à être utilisée pour de la reconnaissance faciale. La plupart de ces systèmes de notations sont des modèles de réseau de neurones (donc des boîtes noires) entraînés sur des jeux de données spécifiques leur donnant leur singularité. Certains sont par exemple entraînés sur des jeux de données ne contenant que des visages comme FaceQnet et d'autres contenant des images de tout type (paysages, objets etc) comme le score BRISQUE.

Pour avoir une idée des potentialités de cette approche j'ai retenu ces deux scores particulièrement plébiscités par la communauté scientifique ainsi que OFIQ (Open Source Face Image Quality) qui est un score développé en source libre par une agence fédérale allemande (BSI) en collaboration avec des partenaires européens. Ce score m'a été proposé par le COL REHAULT lui-même car ce projet est accrédité par l'Union européenne et est voué à être largement utilisé par les forces de l'ordre des différents Etats membres dans les années à venir.

Malheureusement, comme nous le détaillerons un peu plus tard, l'état de l'art actuel des modèles FIQA fait que cette approche n'est pas très intéressante : les scores n'ont aucune consistance, donnant parfois des notations élevées à des photos médiocres et des notations basses à d'excellentes photos.

## **D) Arbre de traitement**

Une approche préconisée par le LCL CAMARA (et approuvée par le COL REHAULT) était de créer un arbre de traitement qui à chaque nœud calculer une métrique particulière (par exemple la luminosité) et appliquer ou non un traitement en fonction de cette métrique et dont l'intensité de traitement varierait en fonction de cette même valeur. J'ai abandonné cette idée car encore une fois les métriques proposées par la littérature scientifique ne sont pas fiables comme le montre l'utilisation de la variance du laplacien de l'image pour mesurer la netteté : une valeur haute peut à la fois signifier une image nette qu'une image fortement bruitée. De plus, il ne semble y avoir aucune corrélation entre cette mesure et l'intensité du filtre de débruitage à appliquer dans le cas où il s'agirait d'une image floutée, rendant la perspective d'un tel arbre compliqué.

## **E) Tests et implémentation**

Une fois cette pré-sélection de traitements effectuée (il y avait d'autres méthodes que je n'ai pas présentées car je ne les ai pas gardées finalement comme la correction gamma), et sous les conseils du LCL CAMARA, j'ai décidé de créer un dataset d'images libres de droit. Ce dataset comportait 50 images très diverses au niveau de la qualité (netteté, bruit, éclairage, luminosité etc), de l'angle (mugshot, caméra de surveillance, prise d'en bas, prise de loin etc) et de certaines spécificités (port d'une casquette, effet fisheye, une partie du visage assombrie ou surexposée, effet de halo, plusieurs personnes sur la photo, etc) me permettant d'avoir un échantillon représentatif de ce que le TAJ semble rejeter (ou du moins ce rapprocher de ces photos là).



Mon objectif était de mieux cerner le TAJ, comprendre le type d'images qu'il rejette et vérifier mes convictions quant aux traitements à appliquer pour faire accepter les images. J'ai donc donné cette base d'images aux équipes du DFAJ leur demandant de rentrer une à une chacune de ces images et de me retourner un tableau Excel dans lequel était noté « oui » en vert si l'image passait sans modifications, « non » en jaune si l'image passait après traitement manuel (et le cas échéant de préciser les modifications faites) et « non » en rouge si l'image ne passait pas malgré toutes les manipulations manuelles possibles. Les Gendarmes sont même aller plus loin en me fournissant également toutes les images jaunes après modifications me permettant de mieux comprendre l'étendu des transformations. J'ai donc obtenu cette distribution dans le dataset :

### **insérer camembert**

En me concentrant sur les images jaunes et les transformations nécessaires pour les faire passer il semblerait que 3 grands procédés se dégagent pour notre logiciel. Premièrement, la plupart de ses images passent après des modifications légères, à savoir recadrage de l'image sur le visage et ajustement de contraste léger. Deuxièmement, une portion non négligeable de ces images passent après des modifications spatiales plus ou moins lourdes notamment la rotation de l'image pour aligner les yeux avec la ligne d'horizon ou enlever le fond pour ne laisser que la personne d'intérêt. Troisièmement, une minorité d'images passent après un traitement lourd sur la luminosité impliquant des corrections maximales sur ce critère.

Je pouvais donc concevoir l'architecture du logiciel et les différentes « routes » de traitement. La page d'accueil était conçue le plus simplement possible : une zone de dépôt pour l'image avec un formulaire à sa droite contenant 3 cases à cocher ou non (« Détection automatique », « Augmenter la luminosité », « Retirer le fond ») et un bouton soumettre. A noter que la case « Détection automatique » est la seule pré-cochée quand on démarre l'application car il s'agit de la combinaison de transformations légères qui marche le plus souvent (recadrage sur le visage + ajustement de contraste adaptatif + alignement des yeux). Voici un arbre qui résume tous les chemins de transformations possibles :

### **insérer arbre de transformation**

C'est donc ainsi que j'ai implémenter une première version du logiciel, que j'ai nommé faceX pour faceExtractor, avec comme demandé dans le cahier des charges un front en HTML et un back en Python/Flask. Il m'a fallu environ 2 semaines pour avoir une première version fonctionnelle et présentable (avec les quelques éléments graphiques) car malgré la simplicité de son architecture mon manque d'expérience en développement web et les problèmes de dépendance entre les bibliothèques basées sur OpenCV et celles basées sur PIL lors des appels de certaines routes ont retardé la finalisation du logiciel.

C'est à ce moment que le LCL CAMARA et moi-même avons décidé de présenter cette version au COL REHAULT et à son adjoint le CDT MARTIN. Les retours ont été très positifs, les deux étant des officiers de carrière donc encadrant directement les Gendarmes de terrain ils ont particulièrement apprécié la simplicité d'utilisation et les résultats visuels des traitements que j'appliquais. Il m'ont également demandé de faire quelques ajustements sur le côté UX et d'ensuite charger l'application sur le système interne du Pôle. Entre temps j'ai également pu montrer le logiciel au COL BOISMOREAU, commandant en chef du SCRC, par l'entremise de mon tuteur le COL REHAULT, qui semblait lui aussi satisfait.

Me restant encore deux semaines de stage, le COL REHAULT m'a donc orienté vers une nouvelle mission dans la continuité de ce que j'avais déjà fait : rajouter une fonctionnalité d'extraction de visage à partir de vidéo à faceX.

## **IV-Bilan**

### **A) Compétences**

Au cours de ce stage, j'ai eu l'opportunité d'acquérir de nouvelles compétences techniques et pratiques dans le domaine de la programmation, plus spécifiquement en Python et HTML, avec un accent particulier sur le traitement d'image, l'intelligence artificielle (IA) et le développement logiciel.

Tout d'abord, j'ai pu renforcer mes connaissances en programmation Python, Python étant un langage de haut niveau polyvalent et largement utilisé dans le domaine du traitement d'image et de l'intelligence artificielle. J'ai notamment appris à manipuler des bibliothèques spécialisées en Python telles que OpenCV et RetinaFace pour effectuer diverses opérations sur les images comme la conversion d'espace de couleurs, la détection de visages ou des opérations de filtrage.

En parallèle, j'ai également eu l'occasion de véritablement apprendre sur le tas des compétences en développement web en apprenant les langages les plus répandus dans ce domaine comme le HTML/CSS pour l'architecture et le JavaScript pour les aspects dynamiques. J'ai donc appris à créer des pages web avec une approche réellement UX facilitant la tâche de téléchargement d'images, de sélection des options de traitement et de visualisation des résultats. J'ai également dû travailler sur l'intégration de scripts Python dans des applications web en utilisant des frameworks comme Flask, me permettant ainsi de créer une application web interactive où le traitement d'image est effectué côté serveur et les résultats renvoyés côté client via une interface web.

De plus, j'ai pu réellement implémenter des modèles d'IA pour un projet concret me permettant de mieux comprendre l'utilisation de ces technologies d'un point de vue pratique. A l'école, dans une optique de pédagogie, nos cours de IA/ML se concentrent beaucoup sur la partie théorique et nous passons nos TD à toujours réinventer la roue c'est-à-dire développer des modèles simples qui existent déjà pour assimiler les bases. Ici j'ai pu manipuler des modèles comme le ferait un professionnel du milieu à savoir chercher un modèle, charger le modèle pré-entraîné, le tester pour vérifier qu'il résout bien notre problème et éventuellement le fine-tuner ou le ré-entraîner complètement (ou ne pas garder le modèle). J'ai ainsi pu me familiariser davantage avec l'écosystème de ce milieu (sur des sites spécialisés comme Hugging Face ou d'autres blogs) et surtout plonger en profondeur dans l'architecture de ces technologies en manipulant par exemple des fichiers .h5 (fichier contenant les poids du modèle) dont j'ignorais même l'existence avant ce stage.

J'ai aussi eu la chance de découvrir le domaine de la vision par ordinateur et du traitement d'image qui sont des domaines d'études qui m'étaient totalement inconnus (n'ayant pas de lien avec ma spécialisation) et que je trouve très intéressant d'un point de vue mathématiques et informatique.

Au cours de ce stage, j'ai également développé des compétences en gestion de projet en apprenant notamment à planifier et à organiser mon travail de manière autonome, en respectant des échéances strictes et en gérant plusieurs tâches à la fois. D'ailleurs la capacité à prioriser les tâches en fonction de leur importance et de leur urgence a été essentielle pour mener à bien ce projet.

Enfin, le stage m'a permis de renforcer mes compétences en résolution de problèmes. Le développement de solutions automatisées de traitement d'images en utilisant de l'intelligence artificielle nécessite souvent une réflexion approfondie et méthodique à cause de la complexité du domaine. J'ai appris à aborder ce problème de manière structurée, en effectuant d'abord une analyse documentaire complète, puis en explorant différentes solutions possibles avant de choisir les plus appropriées en fonction de tests et de considérations empiriques.

Un point ambivalent de ce stage est la quasi absence de travail d'équipe : lors de ce projet j'avais un tuteur qui me donnait des consignes et m'orientait vers certaines voies qu'il pensait utile et un tuteur technique qui était une sorte de professeur me guidant et m'aidant lorsque je rencontrais des problèmes ou des interrogations techniques mais pas de véritable équipe programmant le projet avec moi. Cela a des points bénéfiques comme l'autonomisation et surtout la liberté. En effet, je pouvais avancer à mon rythme et je pouvais moduler mon planning selon mes choix. C'est d'ailleurs grâce à cela que j'ai pu explorer autant lors de ce stage et délivrer une première version du logiciel assez rapidement. L'inconvénient principale est que le milieu de la programmation est un milieu largement associatif et que les projets ne se font pas tout seul en général. Je n'ai donc pas pu approfondir mes connaissances en gestion de projet d'équipe, notamment l'utilisation de certains outils pour cela comme Git et Gitlab, et en livraison continue de projet.

## **B) Lien avec l'ingénierie financière**

Ce stage m'a permis de découvrir un univers (la Gendarmerie scientifique et plus généralement celui de l'application des sciences pour les forces armées) et un domaine scientifique (la vision par ordinateur) qui sont tous deux assez éloignés de la finance quantitative qui est ma spécialisation initiale.

Malgré cela j'ai pu développer des compétences techniques transverses fortes pas seulement pour l'ingénierie financière mais pour l'ingénierie en général, ou tout du moins toutes les branches de l'ingénierie mêlant mathématiques appliquées et programmation. J'ai notamment eu la chance de significativement améliorer mes capacités de programmation Python qui est un langage extrêmement utilisé en finance (et un peu partout) et je peux désormais comprendre l'architecture et le fonctionnement de certains logiciels financiers, m'ouvrant ainsi la voie vers des postes à mi-chemin entre l'ingénierie logiciel et la finance quantitative traditionnelle comme IT Quant, Développeur Quantitatif ou Analyste Quantitatif avec une forte compétence en programmation.

## **C) Difficultés rencontrées**

La première difficulté rencontrée a été celle de mon inexpérience en développement logiciel. En effet, ma méconnaissance du domaine a fortement retardé la livraison du logiciel même si grâce à l'aide du LCL CAMARA et à sa patience j'ai pu combler ce manque en m'autoformant sur Internet. Encore aujourd'hui j'ai d'énormes manques en HTML et JS mais je suis capable de concevoir un logiciel web simple comme cela s'est avéré lors du stage avec faceX.

La deuxième difficulté principale a été celle de la dépendance entre les bibliothèques s'appuyant sur OpenCV ou PIL. Certaines combinaisons de transformations devenaient impossibles à réaliser malgré les conversions d'espace de couleurs (RGB ou BGR) ce qui m'a amené à prendre une décision radicale : abandonné les routes de traitements les plus complexes (d'autant plus que les premiers tests envoyés au TAJ nous montraient que des routes simples étaient suffisantes) et orienter le développement du logiciel vers la bibliothèque

possédant les meilleures fonctionnalités et dépendances, à savoir OpenCV. J'ai donc quasiment abandonné PIL à l'exception du modèle rembg pour enlever le fond des images.

## **D) Analyse du stage**

De mon point de vue le stage est un succès tant d'un point de vue pédagogique pour moi que d'un point de vue opérationnel pour la Gendarmerie. Pour ma part j'ai énormément appris sur différents domaines très techniques qui ont également élargis mon champ des possibles quant à mon orientation de carrière. Pour la Gendarmerie, les premiers tests de faceX semblent indiquer qu'il remplit sa mission correctement : faire passer la majeure partie du temps les photos au TAJ. Ainsi, faceX va certainement faire gagner un temps précieux aux enquêteurs et leur permettre de traiter plus d'affaires.

Mais faceX n'est pas parfait. Premièrement, il ne permet pas de faire passer toutes les photos, sa simplicité qui est sa principale force représente également une limite. En intégrant des modèles plus complexes il serait peut-être possible de traiter ces cas particuliers mis en valeur par les tests ou relevés par les opérateurs du TAJ : effet fisheye, lunettes, surexposition, images prises de trop loin, couvre-chef etc. Pour ne pas altérer la simplicité de faceX une solution pourrait être de rajouter un formulaire caché (sous forme de menu déroulant) sur la page d'accueil contenant les boutons pour activer des routes de traitement très spécialisées comme « Enlever les lunettes ». Pour la partie backend de ce genre de traitement je suis tombé lors de ma phase de recherche sur plusieurs modèles d'IA qui justement proposent ce genre. Il faudra seulement faire attention car certains de ces modèles utilisent de l'IA générative donc il y a nécessité d'opérer une pré-sélection minutieuse pour éviter tout problème.

Le deuxième axe d'approfondissement reste la fonctionnalité d'extraction de visages depuis une vidéo directement qui se heurte à de nombreux problèmes techniques en terme de puissance de calcul par exemple. Il faudra soit faire un compromis entre la précision du modèle et le temps de traitement ou trouver un moyen pour faire du suivi de visage plus efficacement. On peut envisager un autre modèle spécifique à ce problème là ou peut-être envisager une approche reposant sur du calcul haute performance pour accélérer le temps de traitement.

Enfin, on peut également envisager de réessayer l'approche par arbre de traitement proposée par mes deux tuteurs mais sous la forme d'un réseau de neurones, une telle approche ayant déjà été implémentée (cf TreEnhance) mais que je n'ai malheureusement pas pu tester. Ce modèle n'étant pas spécifique aux visages il serait peut-être intéressant de reprendre son architecture et de l'entraîner sur des jeux de données libres de droits ne contenant que des visages comme FFHQ ou CelebA-HQ.