Université Hassan Premier Faculté
Des Sciences et Techniques-Settat
Département Mathématiques et Informatiques

RAPPORT DE PROJET DE FIN D'ANNEE

SPECIALITE : MASTER RESEAU ET SYSTMES INFORMATIQUES

Système d'extraction automatique des informations depuis un extrait de naissance manuscrit

Réalisé par :

IKRAM AFAKHAR

Sous la direction de :

Mr ABDERRAHIM MARZOUK

Soutenu à la FST de Settat, le 10 juin 2025

JURY :

Mr ABDERRAHIM MARZOUK, professeur à La FST de Settat

Mr NASSERDDINE , professeur à La FST de Settat

قال الله تعالى :

﴿ يَرْفَعِ ٱللَّهُ ٱلَّذِينَ ءَامَنُواْ مِنكُمْ وَٱلَّذِينَ أُوتُواْ
ٱلْعِلْمَ دَرَجَٰتٍ ۚ وَٱللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿١١﴾ ﴾

Table des matières

Liste des figures

Dédicace

A ma très chère mère,

Quel que soit le geste ou la parole, je ne saurais jamais exprimer pleinement ma gratitude envers toi. Ton amour infini et ta bienveillance m'ont toujours entouré. Tes conseils sages, tes encouragements constants et tes prières affectueuses ont été des phares dans ma vie, me guidant à chaque étape. Ta présence bienveillante a été pour moi une source d'espoir, de force et de motivation pour surmonter les défis et atteindre les objectifs que je me suis fixés.

A mon très cher père,

Tu as toujours été un roc solide à mes côtés, m'apportant un soutien infaillible à travers vents et marées. Ce modeste travail est bien plus qu'une simple réalisation académique ; il est le reflet de ma profonde gratitude envers toi, pour tous les sacrifices consentis afin de tracer la voie de mon avenir. Tes conseils éclairés, ta sagesse inestimable et ton soutien indéfectible ont modelé la personne que je suis devenue aujourd'hui.

À mes chères sœurs,

Votre amour, votre soutien et votre complicité ont illuminé mon chemin de vie. Merci pour chaque moment où vous avez été présentes, m'insufflant de la force et de l'inspiration nécessaires pour surmonter les défis et persévérer dans la réalisation de mes rêves. Votre présence précieuse a enrichi mon parcours et a donné un sens plus profond à ma vie.

À mes chers professeurs,

Ceux qui se consacrent sans cesse à m'éclairer sur le chemin et les immenses horizons du savoir.

À mes chers amis,

. Votre amitié et soutien inestimables ont été des sources d'inspiration et de force, rendant les moments difficiles plus légers et les victoires plus douces. Je suis reconnaissant de vous avoir à mes côtés, enrichissant ma vie de votre présence chaleureuse et de votre amitié sincère.

Remerciements

C’est avec une immense gratitude que je dédie cette page à toutes les personnes qui ont, de près ou de loin, contribué à la réalisation de ce projet de fin d’année.

Avant toute chose, je remercie **Dieu**, pour sa guidance, sa miséricorde et les innombrables bénédictions qui m’ont accompagnée tout au long de mon parcours académique et personnel.

Je tiens à exprimer mes remerciements les plus sincères à Monsieur **Abderrahim Marzouk**, mon encadrant pédagogique, pour m’avoir proposé ce projet enrichissant, et pour son accompagnement tout au long de cette expérience. Son expertise, ses conseils pertinents et sa disponibilité ont grandement contribué à l’avancement de ce travail. Je lui suis profondément reconnaissante pour sa confiance et son soutien.

J’adresse également mes vifs remerciements à Madame **Sara Arezki**, cheffe de filière, pour son engagement, sa bienveillance et ses efforts constants en faveur de la qualité de notre formation.

Je n’oublie pas de remercier l’ensemble de mes professeurs pour la richesse de leurs enseignements, leur sérieux et leur implication. Grâce à eux, j’ai pu acquérir les connaissances et les compétences nécessaires pour mener à bien ce projet.

Enfin, je souhaite exprimer toute ma reconnaissance à **mes parents**, pour leur soutien inconditionnel, leur patience, leurs encouragements constants et leur amour sans faille. Leur présence et leurs sacrifices ont toujours été une source de motivation précieuse dans mon parcours.

Résumé

Ce rapport présente le travail que j'ai réalisé dans le cadre de mon Projet de Fin d'Année (PFA), consacré à la conception et au développement d'une application d'extraction automatique d'informations à partir d'actes de naissance numérisés. L'objectif principal de ce projet est de faciliter la digitalisation et l'exploitation des documents d'état civil, en automatisant l'extraction des données essentielles telles que le nom, le prénom, la date et le lieu de naissance, ainsi que les informations parentales.

Pour atteindre cet objectif, j'ai développé une application en Python dotée d'une interface graphique conviviale, réalisée avec Tkinter et TTKBootstrap. L'architecture du projet est modulaire et comprend des modules pour la gestion de l'interface utilisateur, l'extraction des informations à l'aide de techniques de reconnaissance optique de caractères , et le stockage structuré des résultats. L'application permet à l'utilisateur de charger un acte de naissance, d'extraire automatiquement les informations pertinentes, puis de les sauvegarder dans un format exploitable.

Ce projet m'a permis de mettre en pratique les compétences acquises durant ma formation, tout en approfondissant mes connaissances en développement d'applications Python et en traitement de documents numérisés. L'application développée constitue une solution innovante pour la modernisation et la simplification des démarches administratives liées à l'état civil.

Abstract

This report presents the design and development of a Python-based application for the automatic extraction of information from digitized birth certificates. The main objective of this project is to facilitate the digitization and exploitation of civil status documents by automating the extraction of essential data such as name, surname, date and place of birth, as well as parental information.

The developed application features a user-friendly graphical interface built with Tkinter and TTKBootstrap. Its modular architecture includes components for user interface management, information extraction using Optical Character Recognition techniques, and structured storage of results. The application allows users to load a birth certificate, automatically extract relevant information, and save it in a usable format.

This project enabled me to apply the skills acquired during my studies, while deepening my knowledge in Python application development and document processing. The resulting application offers an innovative solution for the modernization and simplification of administrative procedures related to civil status.

Introduction générale

La transformation numérique occupe aujourd'hui une place centrale dans la modernisation des administrations publiques et privées. Parmi les documents administratifs les plus sollicités figurent l'acte de naissance, qui constitue la pièce maîtresse pour de nombreuses démarches civiles, juridiques et sociales : inscription scolaire, obtention de documents d'identité, mariage, héritage, etc. Cependant, la gestion traditionnelle de ces documents, souvent sous forme papier ou numérisée sans structuration, engendre de multiples difficultés : perte de temps lors de la recherche d'informations, erreurs de saisie, archivage complexe, et accès limité aux données.

Face à ces enjeux, l'automatisation de l'extraction d'informations à partir d'actes de naissance numérisés apparaît comme une solution innovante et efficace. Elle permet non seulement de gagner du temps, mais aussi d'améliorer la fiabilité et la traçabilité des données, tout en facilitant leur intégration dans des systèmes d'information modernes. Ce projet s'inscrit dans cette dynamique de digitalisation, en proposant le développement d'une application capable d'extraire automatiquement les informations essentielles contenues dans un acte de naissance scanné ou photographié.

L'objectif principal de ce travail est de concevoir et de réaliser une application conviviale, basée sur le langage Python, qui intègre des techniques de reconnaissance optique de caractères pour identifier et extraire les données clés : nom, prénom, date et lieu de naissance, ainsi que les informations relatives aux parents. L'application offre une interface graphique intuitive permettant à l'utilisateur de charger un document, de visualiser les résultats de l'extraction, et de sauvegarder les données structurées dans un format exploitable.

À travers ce projet, il s'agit de répondre à une problématique concrète rencontrée par de nombreuses administrations : comment passer d'un archivage papier ou d'images brutes à une base de données fiable, exploitable et facilement accessible ? Ce rapport détaille l'ensemble des étapes suivies, depuis l'analyse des besoins jusqu'à la réalisation technique, en passant par la conception, les choix technologiques, la mise en œuvre et l'évaluation des résultats obtenus.

Chapitre I :
Contexte Général

1. Introduction

La gestion efficace des documents administratifs représente un enjeu majeur pour les institutions publiques et privées. Parmi ces documents, l'acte de naissance occupe une place centrale, car il constitue la base de nombreuses démarches administratives et juridiques : inscription scolaire, obtention de documents d'identité, mariage, succession, etc. Avec l'augmentation constante du volume de documents à traiter et la nécessité de répondre aux exigences de rapidité, de fiabilité et de sécurité, la digitalisation et l'automatisation des processus administratifs deviennent aujourd'hui indispensables.
Cependant, la simple numérisation des actes de naissance sous forme d'images ou de fichiers PDF ne suffit pas à exploiter efficacement les informations qu'ils contiennent. Les données restent souvent non structurées, ce qui complique leur recherche, leur traitement et leur intégration dans des systèmes d'information modernes. Face à cette problématique, il devient essentiel de mettre en place des solutions capables d'extraire automatiquement les informations clés à partir de documents numérisés.
C'est dans ce contexte que s'inscrit ce projet, dont l'objectif principal est de développer une application permettant d'extraire automatiquement les données essentielles d'un acte de naissance numérisé, telles que le nom, le prénom, la date et le lieu de naissance, ainsi que les informations parentales. La solution proposée repose sur l'utilisation de techniques de reconnaissance optique de caractères (OCR) intégrées dans une application Python dotée d'une interface graphique conviviale. Cette application vise à faciliter la transition vers une gestion numérique des documents d'état civil, en offrant un accès rapide, fiable et structuré aux informations contenues dans les actes de naissance.

1. Contexte

Traditionnellement, les actes de naissance sont archivés sous forme papier, ce qui complique leur gestion : recherche manuelle, risques de perte ou de détérioration, et difficultés d'accès rapide à l'information. La numérisation des documents a permis de stocker ces actes sous forme d'images ou de fichiers PDF, mais les informations restent souvent non structurées et difficiles à exploiter automatiquement.
Dans ce contexte, il devient essentiel de développer des solutions capables d'extraire automatiquement les données clés des actes de naissance numérisés. Ce projet s'inscrit dans cette dynamique de modernisation, en proposant une application qui facilite l'accès, le traitement et l'exploitation des informations contenues dans les actes de naissance.

2. Problématique

Malgré les avancées dans la numérisation des documents administratifs, de nombreuses institutions continuent de rencontrer des difficultés dans l'exploitation des informations contenues dans les actes de naissance. En effet, la majorité de ces documents sont simplement scannés et conservés sous forme d'images, ce qui ne permet pas une recherche rapide ni une extraction automatique des données.
Cette situation engendre plusieurs problèmes :

- La recherche d'informations spécifiques dans un grand volume d'images reste une tâche manuelle, longue et sujette à l'erreur.
- L'intégration des données dans des bases informatiques nécessite une saisie manuelle, augmentant le risque d'erreurs et la charge de travail.
- L'accès aux informations pour les usagers et les agents administratifs est limité, ce qui ralentit les procédures et nuit à la qualité du service.

Face à ces défis, il devient nécessaire de développer une solution capable d'extraire automatiquement et de structurer les informations essentielles à partir d'images d'actes de naissance, afin de moderniser et d'optimiser la gestion des documents d'état civil.

3. Besoins fonctionnels

- Permettre à l'utilisateur de charger une image d'acte de naissance (formats courants : JPG, PNG).
- Extraire automatiquement les informations clés : nom, prénom, date de naissance, lieu de naissance, nom du père, nom de la mère, etc.
- Afficher les résultats de l'extraction de manière claire et lisible dans l'interface.
- Offrir la possibilité de corriger ou valider les informations extraites avant leur sauvegarde.
- Enregistrer les données extraites dans un fichier structuré (par exemple, CSV).

4. Besoins non fonctionnels

- Fournir une interface utilisateur simple, intuitive et ergonomique.
- Garantir un traitement rapide des images pour une utilisation fluide.
- Préserver la confidentialité des données traitées.

5. Objectifs du projet

- **Automatiser l'extraction des informations essentielles à partir d'images d'actes de naissance**

L'objectif principal est de développer une application capable de détecter et d'extraire automatiquement, à partir d'une simple image scannée ou photographiée d'un acte de naissance, les données importantes telles que le nom, le prénom, la date et le lieu de naissance, ainsi que les informations relatives aux parents. Cette automatisation permet de transformer des documents non structurés en données exploitables.

- **Réduire le temps et les erreurs liés à la saisie manuelle des données**

En remplaçant la saisie manuelle par un processus automatisé, le projet vise à accélérer considérablement le traitement des actes de naissance. Cela permet non seulement de gagner du temps, mais aussi de limiter les erreurs humaines qui peuvent survenir lors de la transcription des informations, garantissant ainsi une meilleure fiabilité des données collectées.

- **Faciliter l'exploitation et l'archivage des informations extraites**

Les informations extraites sont sauvegardées dans un format structuré (par exemple, un fichier CSV), ce qui facilite leur intégration dans des bases de données ou des systèmes d'information existants. Cela permet un archivage plus efficace, une recherche rapide des données, et une meilleure accessibilité pour les utilisateurs et les agents administratifs.

6. Solution proposée

Pour répondre aux besoins identifiés et atteindre les objectifs fixés, la solution proposée consiste à développer une application logicielle en Python permettant l'extraction automatique des informations à partir d'images d'actes de naissance.
L'application offre une interface graphique conviviale, réalisée avec Tkinter et TTKBootstrap, qui permet à l'utilisateur de charger facilement une image d'acte de naissance. Une fois l'image importée, le programme utilise des techniques de reconnaissance optique de caractères pour détecter et extraire automatiquement les données essentielles : nom, prénom, date et lieu de naissance, ainsi que les informations parentales.
Les résultats de l'extraction sont affichés à l'utilisateur, qui peut les vérifier et les corriger si nécessaire avant de les sauvegarder. Les données validées sont ensuite enregistrées dans un fichier structuré (CSV), facilitant leur exploitation et leur archivage.
Cette solution vise à simplifier et à accélérer le traitement des actes de naissance, tout en garantissant la fiabilité des informations extraites et en réduisant la charge de travail liée à la saisie manuelle.

7. Planification du projet : diagramme de Gantt

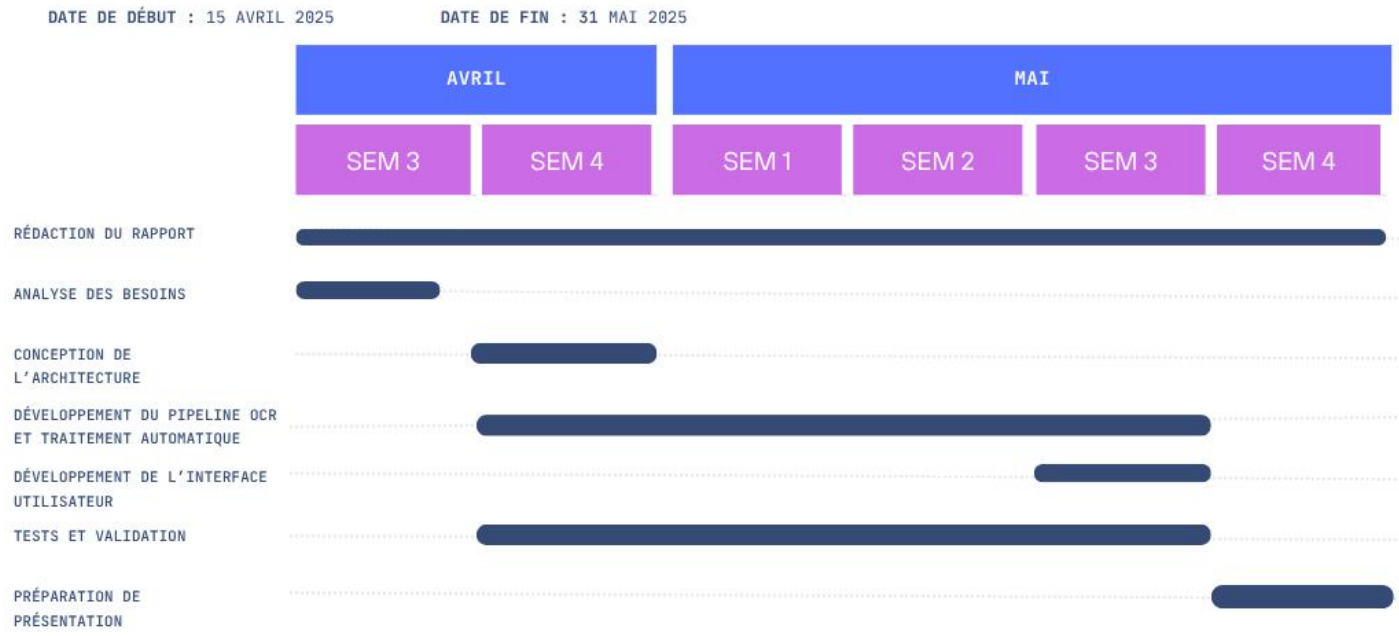

Figure 1 : Diagramme de Gantt

8. Conclusion

Ce premier chapitre a permis de présenter le contexte général du projet, en mettant en évidence l'importance de la digitalisation des documents administratifs et les défis liés à l'exploitation des actes de naissance sous forme d'images. Après avoir exposé la problématique, les besoins et les objectifs du projet ont été clairement définis, aboutissant à la proposition d'une solution adaptée basée sur l'automatisation de l'extraction des informations. Enfin, la planification du projet a été détaillée afin d'assurer une organisation efficace du travail. Les éléments abordés dans ce chapitre constituent ainsi la base sur laquelle reposent les étapes suivantes de conception, de réalisation et de validation de l'application.

Chapitre II :
Étude et Analyse

1. Présentation des documents d'état civil

1.1 Structure générale d'un acte de naissance

Un acte de naissance est un document administratif officiel qui contient des informations essentielles sur l'identité d'une personne. Il se compose généralement des sections suivantes :

- **En-tête administratif**

Cette section contient les informations relatives à l'administration qui a établi l'acte :

- Le nom de l'administration (mairie, préfecture, etc.)
- Le titre officiel du document ("Acte de naissance")
- Le numéro d'enregistrement de l'acte
- La date d'enregistrement de l'acte

- **Informations sur l'enfant**

Cette partie regroupe toutes les informations relatives à la personne concernée par l'acte :

- Le nom de famille
- Le(s) prénom(s)
- Le sexe
- La date de naissance (au format JJ/MM/AAAA)
- Le lieu de naissance (ville, département, pays)

- **Informations sur les parents**

Cette section contient les informations relatives aux parents :

- Le nom et prénom du père
- Le nom et prénom de la mère

- **Informations sur les déclarants**

Cette partie précise les détails concernant la déclaration :

- Le nom et prénom du déclarant
- La qualité du déclarant (père, mère, médecin, sage-femme, etc.)
- La date de la déclaration

- **Mentions légales**

Cette section finale contient les éléments de validation officielle du document :

- La signature de l'officier d'état civil
- Le cachet officiel
- Les mentions marginales éventuelles (mariage, divorce, décès, etc.)

Cette structure standardisée permet une identification claire et rapide des informations essentielles contenues dans l'acte de naissance. Elle facilite également le processus d'extraction automatisée des données, car chaque section occupe généralement une position déterminée dans le document et contient des informations spécifiques qui peuvent être identifiées et extraites de manière systématique.

1.2 Formats et supports

- **Formats d'images supportés**

L'application est conçue pour traiter les actes de naissance sous forme d'images numériques dans les formats suivants :

➢ **Format PNG (Portable Network Graphics)**

Extension de fichier : .png

Avantage :

- Compression sans perte de qualité
- Support de la transparence

- Idéal pour les documents scannés avec du texte
- Meilleure qualité pour les documents administratifs

➢ **Format JPG/JPEG (Joint Photographic Experts Group)**

Extensions de fichiers : .jpg ou .jpeg

Avantage :

- Taille de fichier réduite
- Format largement répandu
- Compatible avec la plupart des appareils photo numériques
- Adapté pour les documents photographiés

- **Sources des documents**

Les images peuvent provenir de différentes sources :

- Documents scannés à l'aide d'un scanner
- Photographies prises avec un appareil photo numérique ou un smartphone
- Images exportées depuis d'autres applications

- **Recommandations pour la qualité des images**

Pour optimiser les résultats de l'extraction, il est recommandé que les images respectent les critères suivants :

- Résolution minimale de 300 DPI (points par pouce)
- Bon contraste entre le texte et l'arrière-plan
- Éclairage uniforme pour les photographies
- Absence de reflets ou d'ombres
- Orientation correcte du document (pas de rotation)

- **Limitations actuelles**

L'application ne supporte pas actuellement :

- Les fichiers PDF
- Les documents au format texte
- Les images au format TIFF

Cette limitation aux formats PNG et JPG/JPEG a été choisie car ce sont les formats les plus couramment utilisés pour la numérisation de documents et ils offrent un bon compromis entre qualité d'image et taille de fichier. De plus, ces formats sont largement supportés par les bibliothèques de traitement d'image utilisées dans l'application.

2. Analyse des documents à traiter

2.1 Caractéristiques des documents

L'analyse des actes de naissance révèle plusieurs caractéristiques distinctives qui influencent directement le processus d'extraction automatisée. Ces particularités doivent être prises en compte pour assurer une extraction fiable et précise des informations.

- **Hétérogénéité des mises en page**

Les actes de naissance présentent une diversité significative dans leur présentation, principalement due à deux facteurs :

Variations administratives

- Chaque administration dispose de son propre modèle standardisé
- Les champs peuvent être positionnés différemment selon les modèles
- La typographie et la mise en forme varient selon les administrations
- Certains champs peuvent être présents ou absents selon le modèle utilisé

Évolution temporelle

- Les formats évoluent selon les périodes d'émission
- Les anciens actes présentent des caractéristiques distinctes des actes récents
- Les normes administratives influencent la présentation des documents

- **Qualité des supports numérisés**

La qualité des images numérisées constitue un facteur déterminant pour la précision de l'extraction :

Paramètres techniques

- Résolution variable (de 200 à 600 DPI)
- Niveaux de contraste divers
- Degrés de netteté différents
- Présence possible de distorsions

État de conservation

- Documents neufs ou récents
- Documents anciens avec altérations
- Documents partiellement détériorés

- **Éléments graphiques complémentaires**

Les documents peuvent inclure divers éléments qui enrichissent leur authenticité mais complexifient l'extraction :

Éléments de sécurité

- Filigranes de protection
- Filigranes d'identification
- Cachets officiels
- Tampons de validation

Éléments de mise en forme

- Cadres décoratifs
- Lignes de séparation
- En-têtes et pieds de page
- Éléments graphiques institutionnels

- **Typologie des textes**

Les documents combinent différents types de texte, chacun présentant ses propres caractéristiques :

Textes imprimés

- Textes dactylographiés
- Textes générés par ordinateur
- Caractères standardisés
- Différentes tailles et styles de police

Textes manuscrits

- Signatures officielles
- Mentions manuscrites
- Corrections à la main
- Annotations diverses

Cette diversité de caractéristiques nécessite une approche sophistiquée et adaptable pour l'extraction automatisée des informations. Le système doit être capable de :

- S'adapter aux différentes présentations de documents
- Gérer les variations de qualité des images
- Traiter les différents types de texte

- Maintenir un niveau élevé de précision malgré ces contraintes

2.2 Informations à extraire

L'analyse des actes de naissance a permis d'identifier les informations essentielles à extraire. Ces données sont structurées en plusieurs catégories pour assurer une extraction complète et organisée.

- **Informations d'identification de l'acte**

Numéro d'acte

- Numéro unique d'enregistrement
- Format variable selon les administrations
- Élément crucial pour la traçabilité

Date d'établissement

- Date de création de l'acte
- Format standardisé (JJ/MM/AAAA)
- Information importante pour la validité du document

- **Informations sur l'enfant**

Identité complète

- Nom de famille
- Prénom(s)
- Sexe

Données de naissance

- Date de naissance
- Lieu de naissance

- **Informations sur les parents**

Père

- Nom complet
- Prénom(s)
- Profession (si mentionnée)

Mère

- Nom complet
- Prénom(s)
- Profession (si mentionnée)

- **Informations complémentaires**

Les informations complémentaires enrichissent le document avec des détails importants sur le contexte de la déclaration. Cette section inclut l'identité et la qualité du déclarant, ainsi que la date de la déclaration. Les mentions légales, comprenant la signature de l'officier d'état civil, le cachet officiel et les mentions marginales éventuelles, authentifient le document et en garantissent la validité légale.

- **Critères d'extraction**

Le processus d'extraction doit respecter des critères stricts pour garantir la qualité des données. La précision est primordiale, nécessitant une extraction exacte des données et le respect des formats standards. La complétude exige la capture de toutes les informations requises et l'identification claire des champs manquants. La validation implique une vérification rigoureuse de la cohérence des données et la détection des anomalies potentielles.

Cette structuration détaillée des informations à extraire permet de standardiser le processus d'extraction tout en assurant une couverture complète des données essentielles. Elle facilite l'intégration des informations dans les systèmes de gestion et garantit la qualité et la fiabilité des données extraites. L'identification précise de ces informations est fondamentale pour le développement d'une solution d'extraction efficace et fiable, capable de répondre aux besoins des utilisateurs tout en respectant les

normes administratives en vigueur.

3. Contraintes techniques et fonctionnelles

Le développement de l'application d'extraction d'informations à partir d'actes de naissance doit répondre à un ensemble de contraintes techniques et fonctionnelles. Ces contraintes sont essentielles pour garantir la qualité, la fiabilité et l'utilisabilité de la solution.

3.1 Contraintes techniques

La performance constitue un aspect crucial de l'application. Le système doit être capable de traiter les images rapidement pour assurer une expérience utilisateur fluide et efficace. Cette exigence de performance s'applique à toutes les étapes du processus, du chargement de l'image à l'extraction des informations.

La précision de l'extraction est fondamentale pour la fiabilité du système. L'application doit garantir une extraction fiable et précise des informations, minimisant les erreurs et les omissions. Cette précision est particulièrement importante pour les données administratives qui peuvent avoir des implications légales.

La compatibilité du système est un autre aspect essentiel. L'application doit supporter les formats d'image courants, notamment JPG et PNG, pour assurer une large adoption et une utilisation pratique. Cette compatibilité doit être maintenue à travers les différentes versions du système.

La sécurité des données personnelles est une préoccupation majeure. L'application doit implémenter des mesures robustes pour protéger les informations sensibles contenues dans les actes de naissance. Cette protection inclut le chiffrement des données, la sécurisation des accès et la conformité aux réglementations sur la protection des données personnelles.

3.2 Contraintes fonctionnelles

L'interface utilisateur doit être intuitive et facile à utiliser. Cette intuitivité est essentielle pour permettre aux utilisateurs de maîtriser rapidement l'application et d'effectuer les opérations d'extraction sans difficulté. L'interface doit être claire, bien organisée et guidée par des instructions explicites.

La possibilité de correction manuelle est une fonctionnalité importante. Les utilisateurs doivent pouvoir vérifier et corriger les informations extraites avant leur validation finale. Cette fonction de correction permet d'assurer la qualité des données tout en offrant un contrôle à l'utilisateur.

L'export des données dans un format exploitable est une exigence fondamentale. L'application doit permettre l'exportation des informations extraites dans des formats standards, facilitant leur intégration dans d'autres systèmes ou leur utilisation dans différents contextes administratifs.

La gestion des erreurs d'extraction est un aspect crucial de l'application. Le système doit être capable de détecter, signaler et gérer les erreurs potentielles lors du processus d'extraction. Cette gestion des erreurs doit être transparente pour l'utilisateur et fournir des informations claires sur la nature des problèmes rencontrés.

Ces contraintes techniques et fonctionnelles guident le développement de l'application et définissent les critères de qualité auxquels la solution doit répondre. Leur prise en compte est essentielle pour assurer le succès du projet et la satisfaction des utilisateurs.

4. Approches existantes, choix méthodologiques et principes de fonctionnement

4.1 État de l’art et solutions existantes

L’extraction automatique d’informations à partir de documents administratifs constitue un domaine de recherche actif et un enjeu industriel majeur, notamment dans le contexte de la digitalisation des démarches administratives. Plusieurs approches et solutions ont été développées au fil des années, chacune présentant ses avantages et ses limites.

- **Reconnaissance Optique de Caractères (OCR)**

La reconnaissance optique de caractères (OCR, pour Optical Character Recognition) est une technologie qui permet de convertir des images de texte imprimé ou manuscrit en texte numérique exploitable. Les moteurs OCR analysent la structure des images, détectent les zones contenant du texte, puis identifient chaque caractère pour reconstituer le contenu textuel.

Parmi les solutions les plus connues, on trouve Tesseract, ABBYY FineReader ou Google Vision OCR. Ces outils sont capables d'extraire le texte brut à partir d'images ou de fichiers PDF.

Cependant, ils présentent plusieurs limites :

- **Structuration limitée :** Ils se contentent généralement de restituer le texte sans structurer les informations (pas de séparation automatique des champs comme nom, date, lieu, etc.).
- **Sensibilité à la mise en page :** Ils peinent à gérer la diversité des formats de documents administratifs, où la position et la présentation des champs peuvent varier considérablement.
- **Difficulté avec le manuscrit :** Les champs manuscrits, signatures ou mentions marginales sont souvent mal reconnus ou ignorés.

- **Deep Learning et vision par ordinateur**

Le deep learning (apprentissage profond) est une branche de l'intelligence artificielle qui utilise des réseaux de neurones artificiels composés de nombreuses couches ("deep" signifiant "profond"). Ces modèles sont capables d'apprendre des représentations complexes à partir de grandes quantités de données, ce qui les rend particulièrement efficaces pour le traitement d'images et la reconnaissance de formes.

Dans le contexte de l'extraction d'informations, le deep learning a permis de franchir un cap :

- **Détection d'objets :** Des modèles comme YOLO (You Only Look Once), Faster R-CNN ou SSD sont capables de localiser précisément les zones d'intérêt (champs à extraire) dans des documents complexes, même lorsque la mise en page varie d'un document à l'autre. Ces modèles sont entraînés à détecter des "boîtes" autour des éléments clés (noms, dates, signatures, etc.), facilitant ainsi l'extraction ciblée.
- **Segmentation sémantique :** D'autres approches utilisent la segmentation pour distinguer les différentes parties d'un document (en-tête, corps, mentions légales…).

- **Post-traitement intelligent et modèles de langage**

Pour aller au-delà de l'extraction brute, les solutions récentes intègrent des modèles de langage avancés (LLM) et des techniques comme le Retrieval-Augmented Generation (RAG) :

- **Structuration et validation :** Les LLM peuvent analyser le texte extrait, le structurer automatiquement (par exemple, en JSON), et valider la cohérence des informations (vérification de formats de dates, correspondance des noms, etc.).
- **Recherche documentaire :** RAG permet de croiser les résultats de l'OCR avec des bases de connaissances ou des exemples annotés, afin d'enrichir et de fiabiliser l'extraction.

Limites des solutions "clé en main"

Malgré ces avancées, les solutions toutes faites restent limitées pour les documents administratifs très hétérogènes :

- **Variabilité des formats :** Les actes de naissance, par exemple, présentent une grande diversité de modèles selon les administrations, les époques ou les pays.
- **Qualité variable des images :** Les documents scannés ou photographiés peuvent être de qualité inégale (bruit, distorsion, faible contraste…).
- **Présence d'éléments graphiques :** Cachets, filigranes, signatures, etc., compliquent l'extraction automatique.

4.2 Choix des solutions retenues et justification

Après une analyse approfondie des besoins du projet et des limites des solutions existantes, nous avons opté pour une approche hybride combinant plusieurs technologies complémentaires afin d'optimiser la précision, la rapidité et la robustesse de l'extraction d'informations à partir des actes de naissance.

- **Tesseract OCR**

Tesseract a été choisi comme moteur principal de reconnaissance optique de caractères pour sa robustesse, sa maturité et sa large adoption dans l'industrie. Un atout majeur de Tesseract est sa

capacité à être entraîné sur des jeux de données spécifiques. Dans le cadre de ce projet, nous avons réalisé un entraînement personnalisé sur des actes de naissance en français et en anglais, ce qui permet d'améliorer significativement la précision de la reconnaissance, notamment pour les termes administratifs, les mises en page particulières et les polices spécifiques à ces documents.

- **YOLOv8**

Pour la détection des champs d'intérêt (zones contenant les informations clés à extraire), nous avons retenu le modèle YOLOv8. Ce modèle de deep learning est reconnu pour sa rapidité d'inférence et sa précision dans la détection d'objets, même sur des images de qualité variable ou présentant des mises en page différentes. YOLOv8 permet ainsi de localiser automatiquement les zones pertinentes (noms, dates, lieux, etc.) sur les actes de naissance, facilitant une extraction ciblée et fiable.

- **RAG (Retrieval-Augmented Generation)**

Afin d'aller au-delà de l'extraction brute, nous avons intégré la technologie RAG, qui combine la recherche documentaire et la génération de texte. RAG permet de structurer, valider et enrichir les informations extraites en croisant les résultats de l'OCR avec des bases de connaissances ou des exemples annotés. Cette approche garantit une meilleure cohérence des données, réduit les erreurs et permet d'obtenir des résultats plus contextualisés et exploitables.

Ce choix technologique permet de répondre efficacement aux contraintes identifiées :

- **Précision :** grâce à l'entraînement spécifique de Tesseract et à la détection fine de YOLOv8.
- **Rapidité :** grâce à l'efficacité des modèles deep learning utilisés.
- **Adaptabilité :** grâce à la modularité de l'architecture et à la capacité d'intégrer de nouvelles règles ou bases de connaissances via RAG.
- **Fiabilité :** grâce à la validation croisée et à l'enrichissement des données extraites.

4.3 Fonctionnement des algorithmes utilisés

- **Tesseract OCR (LSTM)**

Tesseract est un moteur de reconnaissance optique de caractères (OCR) open-source, largement utilisé pour extraire du texte à partir d'images de documents.

Sa version moderne repose sur des réseaux de neurones récurrents de type LSTM (Long Short-Term Memory), particulièrement adaptés au traitement de séquences.

Principe mathématique :

Un LSTM traite une séquence d'entrées (ici, des pixels ou des segments d'image) en maintenant une mémoire interne qui lui permet de prendre en compte le contexte.

À chaque étape t, l'état caché ht et la cellule mémoire ct sont mis à jour selon :

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\tilde{c}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned}
$$

Où xt est l'entrée à l'instant t, σ la fonction sigmoïde, et * le produit élément par élément.

Entraînement personnalisé :

Dans ce projet, Tesseract a été entraîné sur un corpus spécifique d'actes de naissance en français et en anglais. L'entraînement utilise la fonction de coût CTC (Connectionist Temporal Classification), adaptée à la reconnaissance de séquences sans alignement parfait entre l'entrée (image) et la sortie (texte).

Application :

L'image est segmentée en lignes, puis en caractères. Chaque séquence de pixels est traitée par le LSTM, qui prédit la classe du caractère à chaque étape, en tenant compte du contexte (caractères précédents et suivants).

Cette approche améliore la précision, notamment pour des documents complexes ou peu standardisés.

- **YOLOv8 (You Only Look Once, version 8)**

YOLOv8 est un algorithme de détection d'objets basé sur les réseaux de neurones convolutionnels (CNN), conçu pour localiser rapidement et précisément des éléments dans une image.

Principe mathématique :

L'image d'entrée est divisée en une grille de S×S cellules. Pour chaque cellule, le réseau prédit :

- B boîtes englobantes (bounding boxes), chacune définie par (x,y,w,h) : centre, largeur, hauteur (normalisés)
- Une probabilité d'objet Pobj
- Une probabilité de classe pour chaque catégorie (softmax)

La sortie du réseau est donc un tenseur de dimension S×S×(B×5+C)

S×S×(B×5+C), où C est le nombre de classes.

Fonction de coût :

$$\text{Loss} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \dots$$

(avec des termes pour la localisation, la confiance, et la classification)

Application :

YOLOv8 analyse l'image en un seul passage ("You Only Look Once"). Après passage dans le réseau, les boîtes avec la plus forte probabilité sont conservées (suppression des doublons via NMS : Non-Maximum Suppression).

Dans le contexte des actes de naissance, YOLOv8 permet de détecter automatiquement les zones d'intérêt (noms, dates, lieux, etc.), même si la mise en page varie d'un document à l'autre ou si la qualité de l'image est variable.

- **RAG (Retrieval-Augmented Generation)**

RAG est une approche hybride qui combine la recherche documentaire (retrieval) et la génération de texte (generation) par intelligence artificielle.

Recherche documentaire :

- On encode la question q et chaque document di en vecteurs via un encodeur (ex : BERT).
- On calcule la similarité (cosinus) entre q et chaque di :

$$\text{sim}(q, d_i) = \frac{q \cdot d_i}{\|q\| \|d_i\|}$$

- On sélectionne les k documents les plus pertinents.

Génération :

- Un modèle génératif (LLM, ex : GPT) prend la question et les documents retrouvés comme contexte, et génère la réponse ou la structuration des données.
- La génération est basée sur la probabilité conditionnelle :

$$P(y|q, d_1, \dots, d_k) = \prod_t P(y_t|y_{<t}, q, d_1, \dots, d_k)$$

Où y est la séquence de sortie.

Application :

RAG permet d'obtenir des résultats plus précis et contextualisés, en s'appuyant à la fois sur les données extraites par l'OCR et sur des connaissances externes. Cela réduit le risque d'erreurs, d'oublis ou d'hallucinations, et permet de structurer automatiquement les informations extraites.

5. Choix des outils et technologies

5.1 Technologies de base

- **Python**

Le langage de programmation principal utilisé dans ce projet est Python. Ce choix s'explique par plusieurs raisons :

- **Facilité d'utilisation** : Python est reconnu pour sa syntaxe claire et intuitive, ce qui accélère le développement et la maintenance du code.
- **Richesse des bibliothèques :** Il dispose d'un écosystème très riche, notamment pour le traitement d'images, l'OCR, l'interface graphique et la manipulation de données.
- **Excellente communauté :** La communauté Python est très active, ce qui facilite la résolution de problèmes et l'accès à des ressources ou des exemples de code.

5.2 Technologies d'interface

- **Tkinter**

Tkinter est le framework GUI natif de Python. Il permet de créer des interfaces graphiques de manière simple et rapide, tout en étant multiplateforme.

- **TTKBootstrap**

Pour moderniser l'interface et la rendre plus attractive et responsive, TTKBootstrap a été utilisé. Il s'agit d'une extension de Tkinter qui applique des styles inspirés de Bootstrap, offrant ainsi une expérience utilisateur plus professionnelle et agréable.

5.3 Technologies d'extraction de texte

- **Tesseract OCR**

Tesseract est une solution open-source mature pour la reconnaissance optique de caractères (OCR).

Ses principaux avantages sont :

- **Support multilingue :** Il prend en charge de nombreuses langues, dont le français et l'anglais, ce qui est essentiel pour notre projet.
- **Bonne précision :** Il offre de très bons résultats sur des documents bien structurés, comme les actes de naissance.
- **Capacité à comprendre le contexte :** Grâce à ses modèles LSTM, Tesseract peut mieux interpréter le texte dans son contexte, ce qui permet une extraction plus intelligente des informations.
- **Personnalisation par entraînement :** Dans le cadre de ce projet, nous avons entraîné Tesseract sur un jeu de données spécifique (actes de naissance en français et en anglais). Cette étape d'entraînement permet d'améliorer la reconnaissance sur nos documents cibles, en adaptant le modèle aux particularités du vocabulaire, de la mise en page et des polices utilisées.

5.4 Bibliothèques de traitement d'image

- **OpenCV**

OpenCV est utilisé pour le prétraitement des images avant l'extraction de texte. Les principales opérations réalisées sont :

- Dénuage (suppression du bruit)
- Conversion en niveaux de gris
- Seuillage adaptatif (binarisation intelligente)

- Amélioration de la qualité (netteté, contraste, etc.)

- **PIL (Python Imaging Library)**

PIL, ou son fork Pillow, est utilisé pour la manipulation d'images (redimensionnement, rotation, conversion de formats, etc.), en complément d'OpenCV.

5.5 Outils de développement

- **Visual Studio Code (VS Code)**

Visual Studio Code est utilisé comme éditeur de code principal pour sa légèreté, ses nombreuses extensions et sa facilité d'utilisation.

- **PyCharm**

PyCharm est également utilisé comme IDE Python pour ses fonctionnalités avancées de développement, de débogage et de gestion de projets Python.

5.6 Intelligence artificielle et génération augmentée

- **RAG (Retrieval-Augmented Generation)**

Dans le cadre de ce projet, nous avons également exploré l'utilisation de la technologie RAG (Retrieval-Augmented Generation).

RAG combine la puissance des modèles de génération de texte (**LLM**) avec des capacités de recherche documentaire :

Principe : Lorsqu'une question est posée, le système commence par rechercher des documents pertinents dans une base de données ou un corpus, puis utilise un modèle de génération (comme GPT) pour produire une réponse en s'appuyant sur les documents retrouvés.

Avantage :

- Permet d'obtenir des réponses plus précises et contextualisées, car elles s'appuient sur des sources fiables.
- Réduit le risque d'hallucination des modèles génératifs purs.
- Idéal pour l'extraction d'informations précises à partir de documents administratifs ou juridiques.

Utilisation dans le projet :

RAG a été utilisé pour améliorer la qualité de l'extraction et de la restitution d'informations à partir des actes de naissance, en croisant les résultats de l'OCR avec des bases de connaissances ou des exemples annotés.

6. Architecture technique

6.2 Structure modulaire

L'application a été conçue selon une architecture modulaire afin de garantir la clarté, la maintenabilité et l'évolutivité du code. Chaque fonctionnalité principale est isolée dans un module dédié :

- **Module d'interface utilisateur**

Gère l'affichage graphique, la navigation et les interactions avec l'utilisateur via une interface moderne (Tkinter + TTKBootstrap).

- **Module de prétraitement d'image**

S'occupe du nettoyage, de l'amélioration de la qualité et de la préparation des images pour l'OCR (suppression du bruit, seuillage, redimensionnement…).

- **Module d'extraction de texte**

Utilise le moteur Tesseract OCR personnalisé pour extraire le texte brut à partir des images prétraitées.

- **Module de détection d'objets**

Implémente la détection des zones d'intérêt (champs à extraire) à l'aide d'un modèle YOLOv8, permettant de localiser précisément les informations clés sur le document.

- **Module de traitement intelligent**

Analyse et structure les textes extraits, applique des règles de validation, et peut intégrer des modèles de type RAG ou LLM pour une compréhension contextuelle avancée.

- **Module de post-traitement**

Effectue les dernières corrections, la validation des données extraites et la mise en forme finale des résultats.

- **Module de stockage**

Gère la sauvegarde des résultats, l'export des données structurées (CSV, JSON, base de données…), et l'archivage des images ou des logs.

6.3 Flux de traitement

Le traitement d'un document suit un pipeline clair, où chaque étape est prise en charge par un module spécifique :

i. Chargement de l'image

L'utilisateur sélectionne ou dépose une image via l'interface graphique.

ii. Prétraitement

L'image est nettoyée et optimisée (dénuage, conversion en niveaux de gris, seuillage adaptatif, amélioration de la netteté) pour maximiser la qualité de l'extraction.

iii. Détection des champs

Le module de détection d'objets (YOLOv8) identifie et localise automatiquement les zones d'intérêt (noms, dates, lieux, etc.) sur le document.

iv. Extraction du texte

Le texte est extrait à partir des zones détectées à l'aide de Tesseract OCR (modèle entraîné).

v. Structuration des données

Les textes extraits sont organisés en champs structurés (JSON, dictionnaire Python…), prêts à être exploités ou exportés.

vi. Post-traitement et validation

Les données sont vérifiées, corrigées si besoin (normalisation des dates, contrôle des formats…), et validées selon des règles métier.

vii. Export des résultats

Les résultats finaux sont exportés dans le format souhaité (CSV, JSON, base de données…) et peuvent être affichés à l'utilisateur ou transmis à d'autres systèmes

7. Conclusion

Ce chapitre a analysé en détail la structure et les particularités des actes de naissance, ainsi que les défis liés à l'extraction automatique d'informations à partir de ces documents. Après avoir étudié les limites des solutions classiques d'OCR, il a justifié le choix d'une approche hybride combinant Tesseract entraîné, YOLOv8 pour la détection des champs, et RAG pour le post-traitement intelligent. Cette architecture modulaire et innovante permet de garantir la précision, la fiabilité et l'adaptabilité du système, posant ainsi des bases solides pour le développement d'une application performante d'extraction de données à partir de documents administratifs numérisés.

Chapitre III : Architecture

1. Architecture générale de l'application

L'application a été conçue selon une architecture modulaire, favorisant la clarté, la maintenabilité et l'évolutivité. Chaque fonctionnalité principale est isolée dans un module dédié, ce qui permet de faciliter les évolutions futures et la correction des éventuels bugs.

- **Modules principaux**

L'architecture générale se compose des modules suivants :

i. Interface utilisateur

Permet à l'utilisateur de charger une image, de lancer l'extraction, de visualiser et de corriger les résultats.

Technologies : Tkinter, TTKBootstrap.

ii. Prétraitement

Améliore la qualité de l'image (dénuage, seuillage, conversion en niveaux de gris).

Technologies : OpenCV, PIL.

iii. Détection d'objets

Localise automatiquement les champs d'intérêt (noms, dates, lieux, etc.) sur le document.

Technologies : YOLOv8.

iv. Extraction OCR

Extrait le texte brut à partir des zones détectées.

Technologies : Tesseract OCR (entraîné).

v. Traitement intelligent

Structure, valide et enrichit les informations extraites, en s'appuyant sur des règles ou des modèles de type RAG.

Technologies : RAG, règles métier, LLM si applicable.

vi. Post-traitement

Effectue les dernières corrections, la validation finale et la mise en forme des résultats.

Fonctionnalités : Normalisation des dates, contrôle des doublons ou incohérences

vii. Stockage des résultats

Gère la sauvegarde des résultats extraits et l'export des données.

Technologies : CSV, JSON, base de données si besoin.

- **Schéma global de l'architecture**

Voici un exemple de diagramme d'architecture illustrant les interactions entre les modules :

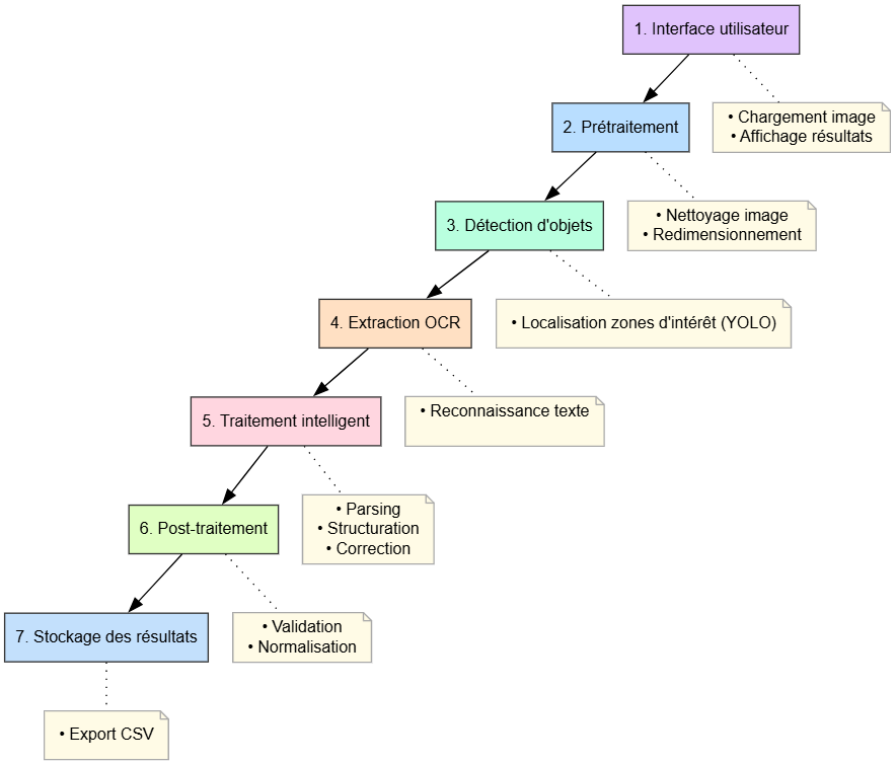

Figure 2 : Schéma global de l'architecture

2. Description des modules

2.1 Module Interface utilisateur

Rôle : Permet à l'utilisateur de charger une image, de lancer l'extraction, de visualiser et de corriger les résultats.

Technologies : Tkinter, TTKBootstrap.

Fonctionnalités :

- Sélection d'image (JPG, PNG)
- Affichage des résultats extraits
- Validation et correction manuelle des champs

2.2 Module Prétraitement

Rôle : Améliore la qualité de l'image (dénuage, seuillage, conversion en niveaux de gris) pour optimiser l'extraction de texte.

Technologies : OpenCV, PIL Numpy.

Fonctionnalités :

- Nettoyage de l'image
- Conversion en niveaux de gris
- Amélioration du contraste et suppression du bruit

2.3 Module Détection d'objets

Rôle : Localise automatiquement les champs d'intérêt (noms, dates, lieux, etc.) sur le document.

Technologies : YOLOv8.

Fonctionnalités :

- Détection des zones à extraire
- Génération de coordonnées pour l'extraction ciblée

2.4 Module Extraction OCR

Rôle : Extrait le texte brut à partir de l'image prétraitée à l'aide du moteur Tesseract OCR, spécifiquement entraîné sur le jeu de données du projet.

Technologies : Tesseract OCR (modèle personnalisé).

Fonctionnalités :

- Extraction automatique du texte sur l'ensemble de l'image.
- Utilisation d'un modèle Tesseract entraîné sur des actes de naissance pour une meilleure précision.
- Génération du texte brut pour l'analyse et la structuration.

2.5 Module Traitement intelligent

Rôle : Structure, valide et enrichit les informations extraites, en s'appuyant sur des règles ou des modèles de type RAG.

Technologies : RAG, règles métier, LLM si applicable.

Fonctionnalités :

- Structuration des données (JSON, dictionnaire)
- Validation des formats et cohérence des champs

2.6 Module Post-traitement

Rôle : Effectue les dernières corrections, la validation finale et la mise en forme des résultats.

Fonctionnalités :

- Normalisation des dates
- Contrôle des doublons ou incohérences

2.7 Module Stockage

Rôle : Gère la sauvegarde des résultats extraits et l'export des données.

Technologies : CSV, JSON, base de données si besoin.

Fonctionnalités :

- Export des résultats
- Archivage des images et logs

3. Diagramme de flux global

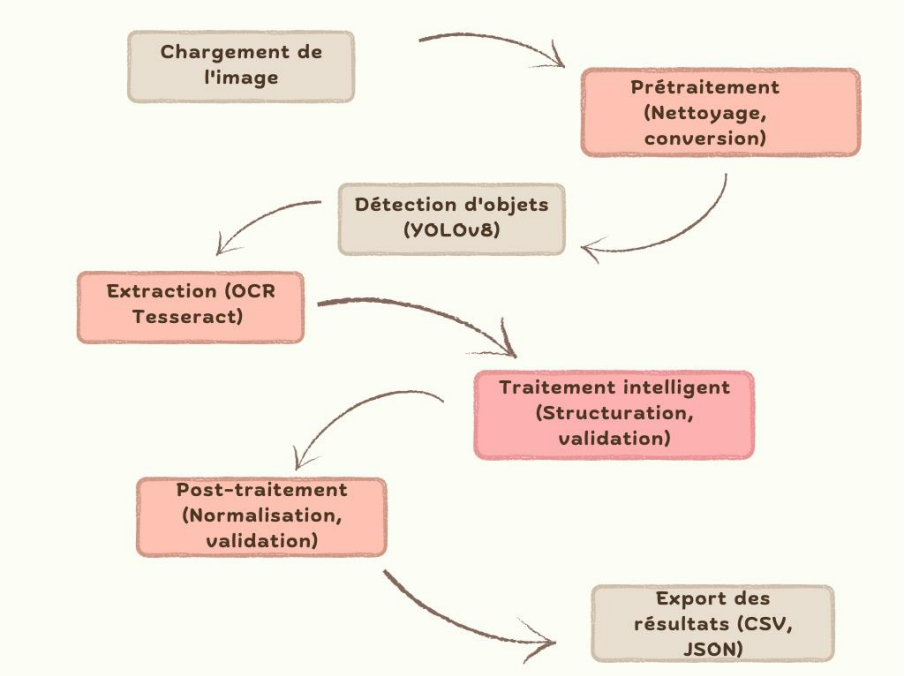

Figure 3 : Diagramme de flux global

- **Explication des étapes révisées :**

Chargement de l'image

- L’utilisateur sélectionne une image (JPG, PNG) via l’interface.

Prétraitement

- Nettoyage de l’image (dénuage, seuillage).
- Conversion en niveaux de gris pour optimiser la détection.

Détection d'objets

- Localisation des champs d’intérêt (noms, dates, lieux) via YOLOv8.
- Génération de coordonnées pour l’extraction ciblée.

Extraction

- Utilisation de Tesseract OCR uniquement sur les zones détectées.

Traitement intelligent

- Structuration des données extraites.
- Validation des formats et cohérence des champs.

Post-traitement

- Normalisation des dates.
- Contrôle des doublons ou incohérences.

Export des résultats

- Sauvegarde des données extraites en CSV ou JSON.

4. Conclusion

Ce chapitre présente l’architecture modulaire de l’application, conçue pour garantir clarté, maintenabilité et évolutivité. L’architecture se compose de sept modules principaux : l’interface utilisateur (Tkinter, TTKBootstrap), le prétraitement (OpenCV, PIL), la détection d’objets (YOLOv8), l’extraction OCR (Tesseract OCR entraîné), le traitement intelligent (RAG, règles métier), le post-traitement (normalisation, validation) et le stockage (CSV, JSON). Chaque module est détaillé en termes de rôle, technologies et fonctionnalités. Le chapitre inclut également un diagramme de flux global qui illustre le processus complet, depuis le chargement de l’image jusqu’à l’export des résultats, en passant par le prétraitement, la détection, l’extraction, le traitement et la validation.

Chapitre IV : Réalisation et Implémentation

1. Développement de l'interface utilisateur

L'interface utilisateur constitue le point d'entrée principal de l'application et a été conçue pour être à la fois ergonomique, moderne et accessible à tout utilisateur, même non technique.

Le développement a été réalisé en Python en s'appuyant sur la bibliothèque Tkinter pour la gestion des composants graphiques, et sur TTKBootstrap pour l'application de thèmes modernes et responsives.

Principales fonctionnalités de l'interface

- **Chargement d'une image d'acte de naissance**

L'utilisateur dispose d'un bouton dédié pour sélectionner un fichier image (formats JPG ou PNG) depuis son ordinateur. Le nom du fichier sélectionné s'affiche dans l'interface, confirmant la prise en compte du document.

- **Prétraitement de l'image**

Après le chargement, l'utilisateur peut lancer le prétraitement de l'image. Cette étape améliore la qualité du document (nettoyage, conversion en niveaux de gris, amélioration du contraste) afin d'optimiser la reconnaissance de texte.

- **Extraction automatique des informations**

Un bouton permet de démarrer l'extraction des données. L'application applique alors le moteur OCR sur l'image prétraitée, puis structure automatiquement les informations extraites.

- **Visualisation et correction des résultats**

Les résultats de l'extraction sont présentés dans une fenêtre de correction dédiée. L'utilisateur peut vérifier chaque champ extrait (nom, prénom, date, etc.), corriger les éventuelles erreurs ou compléter les informations manquantes avant validation.

- **Export des données validées**

Une fois les données vérifiées et corrigées, l'utilisateur peut les exporter au format CSV pour une intégration facile dans d'autres systèmes ou pour archivage.

- **Zone de logs en temps réel**

L'interface intègre une zone de texte dédiée à l'affichage des logs. Cette zone permet à l'utilisateur de suivre en temps réel l'avancement du traitement, d'identifier rapidement les éventuelles erreurs et de comprendre chaque étape du processus (chargement, prétraitement, extraction, parsing, export…).

Aspects ergonomiques et techniques

- Design moderne grâce à TTKBootstrap (couleurs, boutons, polices, etc.)
- **Navigation intuitive :** chaque étape est guidée, les boutons sont activés/désactivés selon le contexte (ex : extraction impossible tant qu'aucune image n'est chargée)
- **Gestion des erreurs :** en cas de problème (fichier non supporté, erreur d'extraction…), des messages clairs sont affichés à l'utilisateur

2. Intégration de l'OCR

L'extraction automatique du texte à partir des actes de naissance repose sur l'intégration du moteur Tesseract OCR, reconnu pour sa robustesse et sa capacité à être personnalisé. Dans le cadre de ce projet, Tesseract a été entraîné spécifiquement sur un corpus d'actes de naissance (en français et en anglais), ce qui permet d'optimiser la reconnaissance des termes administratifs, des mises en page variées et des polices spécifiques à ces documents.

Étapes du processus d'extraction OCR

i. Prétraitement de l'image

- Avant l'extraction, l'image subit une série d'opérations de prétraitement : suppression du bruit,

conversion en niveaux de gris, amélioration du contraste et binarisation. Ces étapes sont essentielles pour maximiser la qualité du texte reconnu par l'OCR, en réduisant les artefacts et en rendant le texte plus lisible pour le moteur.

- Les bibliothèques OpenCV et PIL sont utilisées pour appliquer ces transformations de manière efficace.

ii. Application de Tesseract sur l'image prétraitée

- Une fois l'image optimisée, elle est transmise au moteur Tesseract OCR. Grâce à l'entraînement réalisé sur des actes de naissance similaires à ceux à traiter, Tesseract est capable de reconnaître avec une grande précision les caractères, les mots et les structures propres à ces documents.
- Le résultat de cette étape est un texte brut, fidèle au contenu du document original.

iii. Intégration transparente dans le pipeline

- L'ensemble du processus d'OCR est entièrement automatisé et intégré dans le pipeline de l'application. L'utilisateur n'a pas besoin de configurer ou de lancer manuellement l'OCR : il lui suffit de cliquer sur le bouton d'extraction, et le système se charge de toutes les étapes, du prétraitement à la génération du texte.
- Cette automatisation garantit une expérience utilisateur fluide et réduit les risques d'erreur ou d'oubli.

Points forts de l'intégration

i. **Personnalisation** : L'entraînement de Tesseract sur des données spécifiques au projet permet d'atteindre un niveau de précision supérieur à celui d'un modèle générique.

ii. **Performance :** Le prétraitement adapté et l'automatisation du pipeline assurent une extraction rapide et fiable, même sur des documents de qualité variable.

iii. **Simplicité d'utilisation :** L'utilisateur n'a aucune manipulation technique à effectuer ; tout est géré en arrière-plan par l'application.

3. Traitement et structuration des données extraites

Après l'extraction du texte brut par l'OCR, l'application met en œuvre un module de parsing intelligent chargé d'analyser, de structurer et de valider les informations issues du document. Cette étape est cruciale pour transformer un texte non structuré en données exploitables et fiables.

Étapes du traitement et de la structuration

i. Identification des champs clés

- Le module de parsing parcourt le texte brut pour repérer et extraire les informations essentielles : nom, prénom, date de naissance, lieu de naissance, identité des parents, etc.
- Cette identification repose sur des expressions régulières, des règles linguistiques et, si besoin, des modèles de langage pour reconnaître les différentes formulations possibles dans les actes de naissance.

ii. Structuration des données

- Les informations extraites sont organisées sous forme de dictionnaire Python ou de structure JSON. Chaque champ clé du document (par exemple, "nom", "date_naissance", "lieu_naissance") devient une entrée distincte dans cette structure.
- Cette structuration facilite l'export, l'intégration dans des bases de données, ou l'utilisation dans d'autres systèmes informatiques.

iii. Validation et correction automatique

- Avant présentation à l'utilisateur, le module applique des règles de validation : vérification du format des dates, contrôle de la cohérence des champs (par exemple, la date de naissance ne peut pas être postérieure à la date d'établissement de l'acte), détection des champs manquants ou incohérents.
- Des corrections automatiques peuvent être proposées, comme la normalisation des formats de

date ou la correction de fautes courantes.

iv. Interface de correction utilisateur

- Une fois les données structurées, une fenêtre de correction s'ouvre dans l'interface. L'utilisateur peut alors :

- Visualiser chaque champ extrait
- Corriger les erreurs éventuelles ou compléter les informations manquantes
- Valider définitivement les données avant leur export ou leur sauvegarde

- Cette étape garantit la qualité finale des données et permet de pallier les éventuelles limites de l'OCR ou du parsing automatique.

Points forts de ce module

i. **Automatisation avancée :** la majorité des documents sont traités sans intervention manuelle, grâce à des règles et des modèles adaptés.

ii. **Souplesse :** l'utilisateur garde le contrôle final et peut corriger ou compléter les données si nécessaire.

iii. **Fiabilité :** la validation automatique réduit les risques d'erreurs et garantit la cohérence des informations extraites.

4. Gestion des erreurs et validation

La fiabilité de l'application repose sur une gestion rigoureuse des erreurs et une validation systématique des données à chaque étape du traitement. L'objectif est de garantir que les informations extraites soient correctes, complètes et exploitables, tout en offrant à l'utilisateur une expérience transparente et rassurante.

Gestion des erreurs à chaque étape

i. Chargement d'image

- Vérification du format et de la lisibilité du fichier sélectionné (JPG, PNG).
- Affichage d'un message d'erreur si le fichier est corrompu, non supporté ou inaccessible.
- Désactivation des boutons d'action tant qu'une image valide n'est pas chargée.

ii. Prétraitement

- Contrôle de la réussite des opérations de nettoyage, conversion et amélioration de l'image.
- Signalement immédiat en cas d'échec d'une opération (ex : image trop dégradée, erreur de conversion).

iii. Extraction OCR

- Détection des erreurs lors de l'appel à Tesseract (ex : absence de texte détecté, problème de configuration du moteur).
- Affichage d'un message explicite en cas d'échec de l'extraction.

iv. Parsing et structuration

- Gestion des cas où le texte extrait ne permet pas d'identifier certains champs clés.
- Signalement des champs manquants ou incohérents (ex : date de naissance absente ou au mauvais format).

Affichage des messages d'erreur

- L'interface intègre une zone de logs qui affiche en temps réel les messages d'information, d'avertissement ou d'erreur.
- En cas de problème bloquant, une fenêtre de dialogue (popup) informe l'utilisateur de la nature de l'erreur et des actions possibles (ex : recharger une image, corriger un champ, etc.).
- Les messages sont rédigés de façon claire et pédagogique pour guider l'utilisateur, même non technique.

Correction manuelle et validation utilisateur

- Après l'extraction et la structuration, l'utilisateur peut corriger manuellement les champs extraits via une interface dédiée.
- Cette étape permet de rectifier les erreurs résiduelles de l'OCR ou du parsing, et d'ajouter les informations manquantes.
- L'utilisateur valide ensuite les données avant leur export ou leur sauvegarde.

Validation automatique des formats et détection des incohérences

- Le système applique des règles de validation automatique :
- Vérification du format des dates (JJ/MM/AAAA)
- Contrôle de la cohérence des champs (ex : la date de naissance ne peut pas être postérieure à la date d'établissement)
- Détection des doublons ou des valeurs aberrantes
- Les champs non conformes sont mis en évidence pour attirer l'attention de l'utilisateur.

Robustesse et tests

- Chaque étape du pipeline a été testée et validée sur un ensemble varié d'actes de naissance (différents formats, qualités d'image, langues) afin de garantir la robustesse et la précision du système.
- Les tests ont permis d'anticiper et de gérer la majorité des cas d'erreur rencontrés en pratique.

5. Conclusion

Le chapitre 4 présente la mise en œuvre de l'application, depuis le développement d'une interface utilisateur intuitive jusqu'à l'intégration du moteur Tesseract OCR entraîné sur des actes de naissance. Il détaille le processus complet : prétraitement des images, extraction et structuration automatique des données, ainsi que la gestion des erreurs et la validation des résultats. L'ensemble garantit une extraction fiable, automatisée et facilement corrigeable par l'utilisateur, répondant ainsi aux objectifs d'efficacité et de robustesse du projet.

Chapitre V : Interface Utilisateur et Présentation

1. Présentation de l'interface graphique

L'interface graphique de l'application a été conçue pour offrir une expérience utilisateur moderne, intuitive et guidée, grâce à l'utilisation de Tkinter et TTKBootstrap. Elle permet de réaliser l'ensemble du processus d'extraction d'informations à partir d'un acte de naissance, depuis le chargement de l'image jusqu'à la correction et l'export des résultats.

1.1. Accueil et interface principale

L'écran d'accueil présente le titre de l'application et la section d'extraction depuis un fichier unique.

Figure 4 : Interface principale

1.2. Chargement d'un fichier image

- L'utilisateur clique sur « Charger un fichier image » pour sélectionner un acte de naissance au format JPG ou PNG.
- Une fenêtre de sélection de fichier s'ouvre, permettant de parcourir les dossiers du système

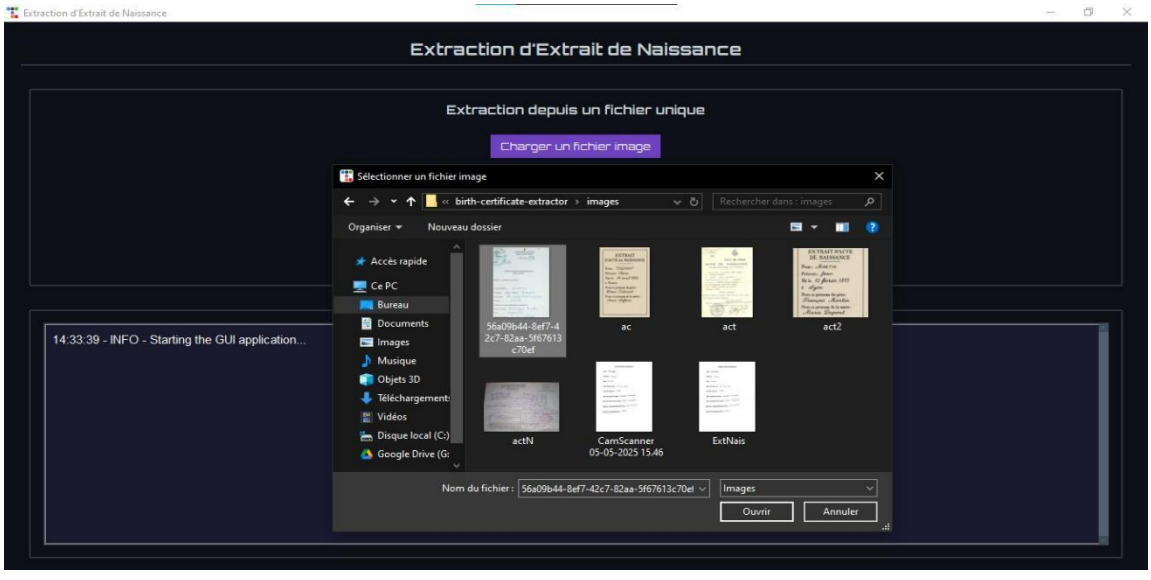

Figure 5 : Chargement d'une image

1.3. Affichage du fichier chargé et logs

- Une fois le fichier sélectionné, son nom s'affiche dans l'interface.
- La zone de logs affiche les messages relatifs au chargement et à la préparation de l'image.

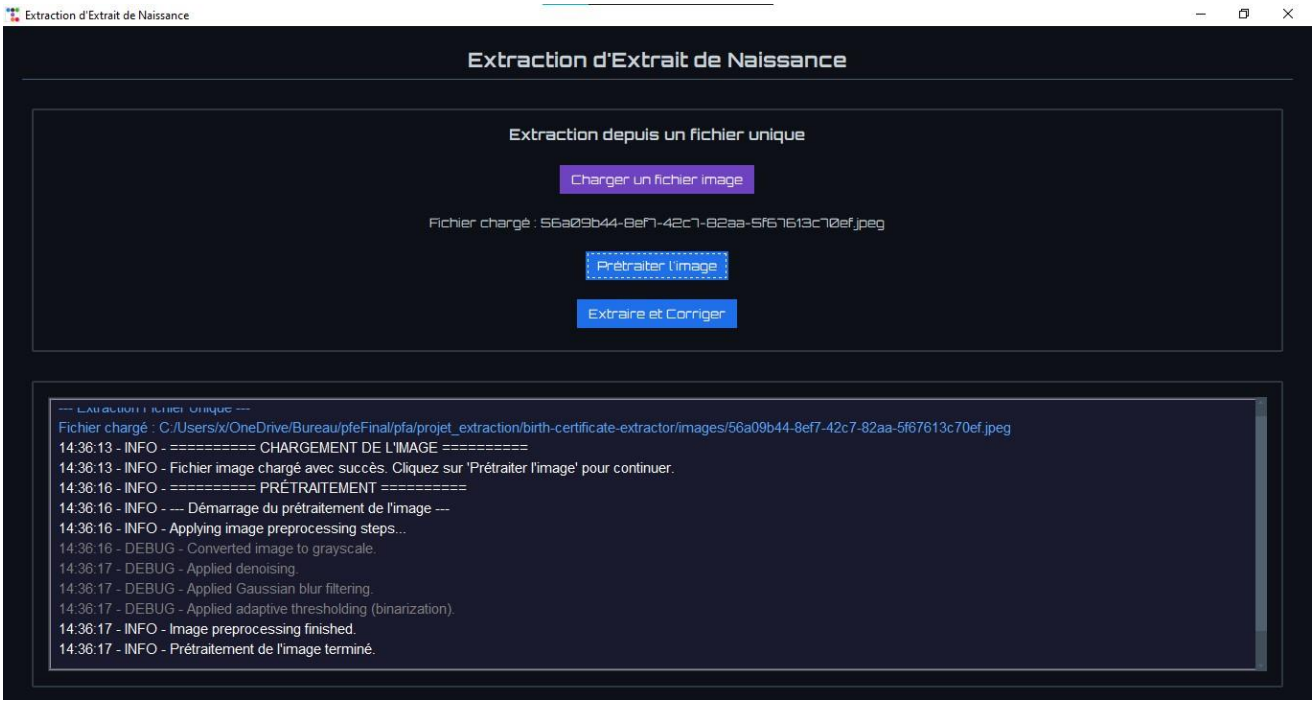

Figure 6 : Chargement d'une image

1.4. Prétraitement et extraction

- Après le chargement, l'utilisateur peut lancer le prétraitement de l'image, puis l'extraction des données.
- Une barre de progression et des messages d'état informent l'utilisateur de l'avancement du traitement.

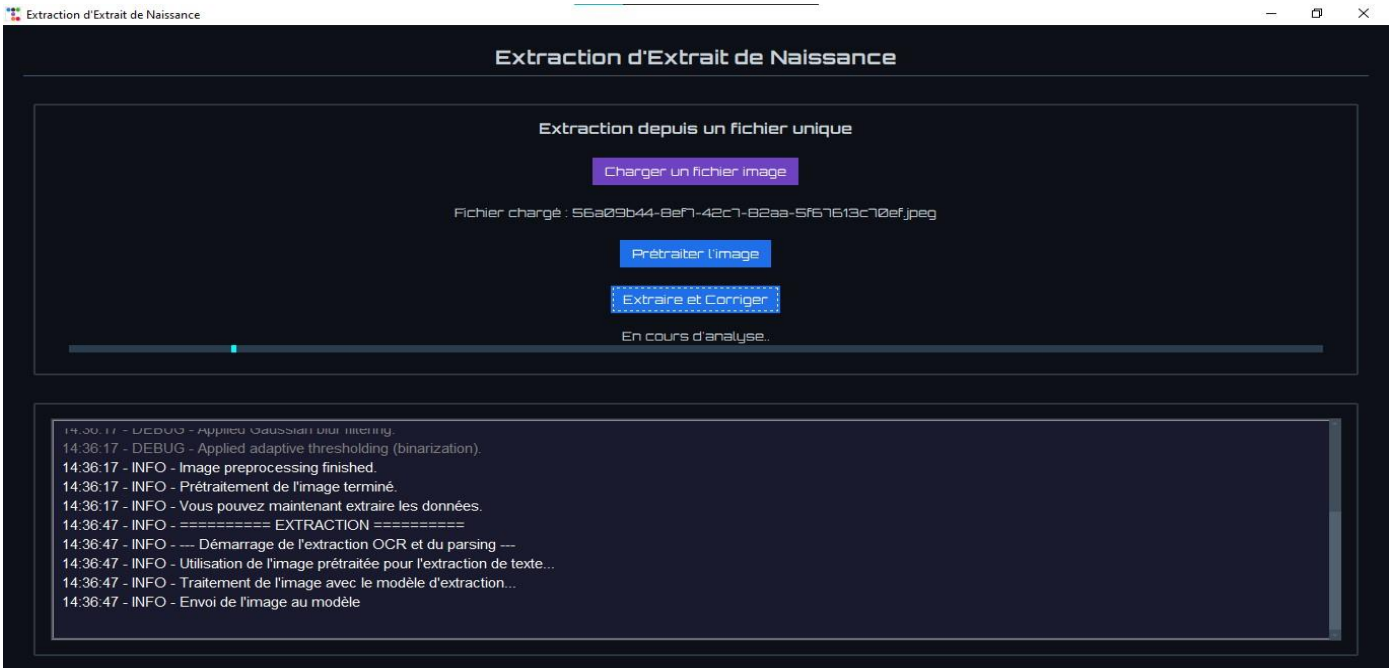

Figure 7 : Prétraitement et extraction

1.5. Correction des données extraites

- Une fenêtre modale s'ouvre pour permettre la correction manuelle des champs extraits (nom, prénom, date de naissance, etc.).
- Chaque champ est éditable, et l'utilisateur peut valider ou annuler les modifications.

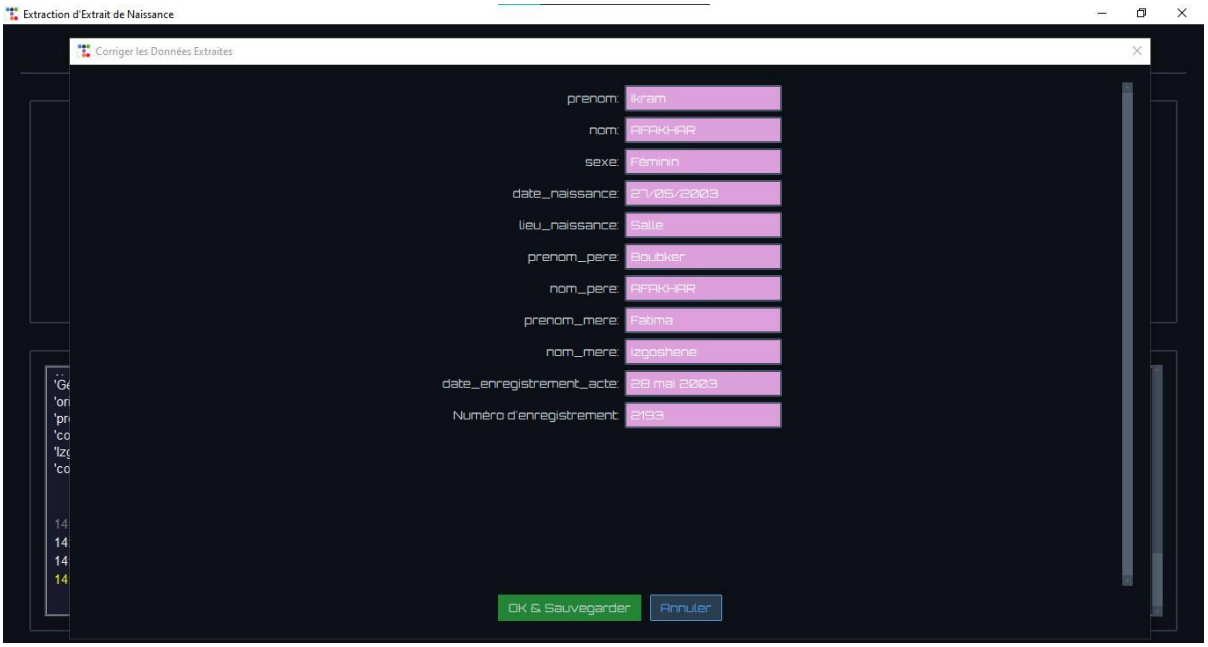

Figure 8 : Correction des données extraites

1.6. Enregistrement et export des données

- Après validation, les données corrigées sont automatiquement enregistrées dans un fichier CSV.
- Une fenêtre de confirmation s'affiche pour informer l'utilisateur du succès de l'opération et du chemin d'enregistrement du fichier.
- La zone de logs affiche également les détails de l'export (chemin, données sauvegardées, etc.).

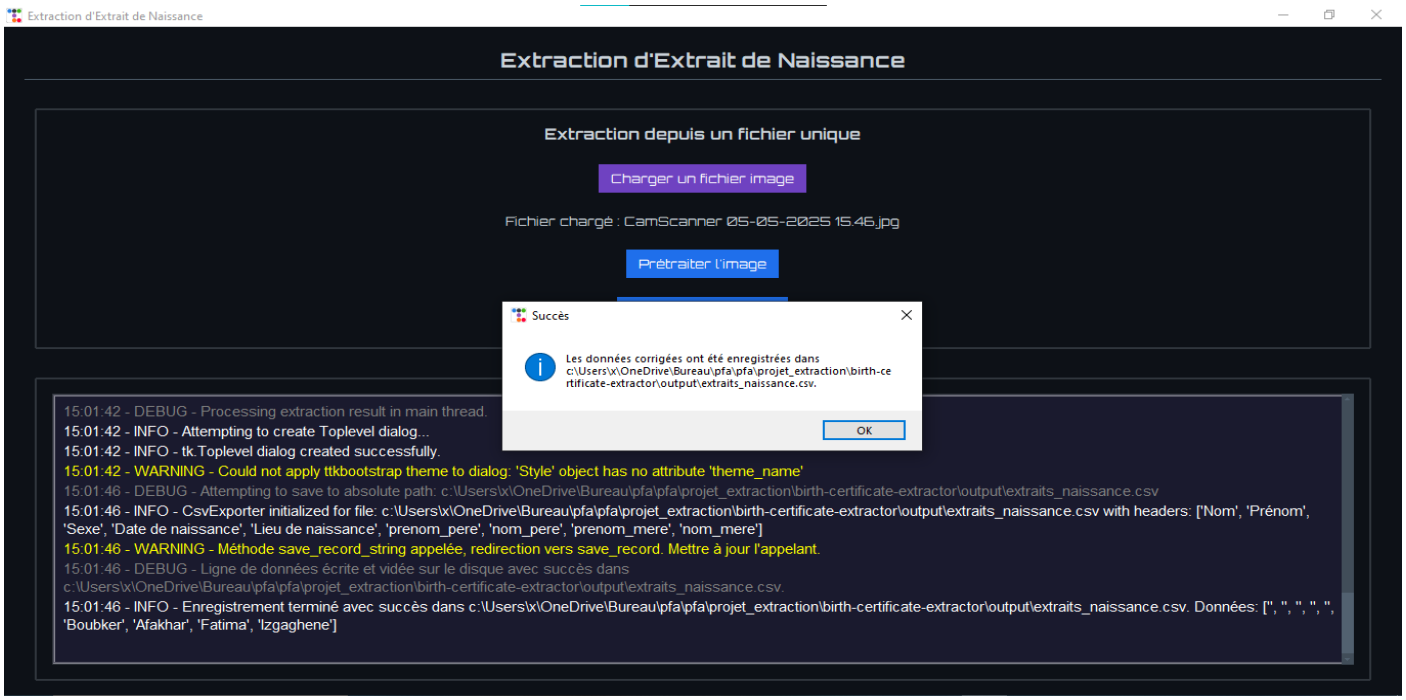

Figure 9 : Enregistrement et export des données

2. Fonctionnalités principales de l'interface

Chargement d'image : prise en charge des formats JPG/PNG, affichage du nom du fichier.

Prétraitement : amélioration automatique de l'image, logs détaillés.

Extraction : reconnaissance de texte, parsing et structuration automatique.

Correction : interface dédiée à la validation et à la correction des champs extraits.

Export : sauvegarde des résultats au format CSV, confirmation visuelle de l'enregistrement.

Logs en temps réel : suivi de toutes les étapes du traitement.

3. Outils et technologies utilisés

Afin de garantir la robustesse, la performance et la maintenabilité de l'application, de nombreux outils et bibliothèques open-source ont été sélectionnés et intégrés au projet. Chacun d'eux répond à un besoin précis, du traitement d'image à l'intelligence artificielle, en passant par l'interface utilisateur et la gestion des données.

3.1. Langage et environnement

Python

Langage principal du projet, choisi pour sa simplicité, sa lisibilité et la richesse de son écosystème scientifique et IA.

3.2. Interface graphique

Tkinter

Bibliothèque standard de Python pour la création d'interfaces graphiques.

TTKBootstrap

Extension de Tkinter permettant d'appliquer des thèmes modernes et responsives à l'interface.

3.3.Traitement d'images

opencv-python

Utilisé pour le prétraitement des images (nettoyage, conversion en niveaux de gris, seuillage, amélioration du contraste, etc.).

Pillow (PIL)

Pour la manipulation, la conversion et le redimensionnement des images.

numpy

Pour la gestion efficace des matrices d'images et l'interface avec OpenCV et PIL.

3.4.Extraction de texte et NLP

- **Tesseract OCR & pytesseract**

  Moteur OCR open-source, utilisé via la bibliothèque Python pytesseract pour extraire le texte des images.

- **spaCy**

  Pour le traitement avancé du langage naturel, la validation et la correction des champs extraits.

- **thefuzz[speedup]**

  Pour la comparaison floue de chaînes de caractères (fuzzy matching), utile pour corriger les erreurs d'OCR ou de saisie.

- **unidecode**

  Pour la normalisation des caractères accentués ou spéciaux.

3.5.Intelligence artificielle et deep learning

torch

Bibliothèque de deep learning utilisée pour l'entraînement et l'inférence de modèles personnalisés.

transformers, sentencepiece, accelerate, seqeval, datasets, sentence-transformers

Pour l'intégration de modèles de traitement du langage naturel (LLM, RAG, etc.), la vectorisation de texte, l'évaluation de séquences et la gestion de jeux de données.

langchain, faiss-cpu, llama-cpp-python

Pour la recherche sémantique, la gestion de bases de connaissances vectorielles et l'utilisation de modèles LLM locaux.

3.6.Manipulation et export de données

- **pandas**

  Pour la manipulation, l'analyse et l'export des données structurées (CSV, DataFrame).
- **openpyxl**

  Pour la gestion des fichiers Excel (lecture/écriture).
- **SQLAlchemy**

  Pour l'intégration et la gestion de bases de données relationnelles.

  Déploiement, configuration et utilitaires
- **Flask**

  Pour la création éventuelle d'une API web ou d'une interface complémentaire.
- **requests**

  Pour la gestion des requêtes HTTP (appels API, récupération de ressources).
- **python-dotenv**

  Pour la gestion sécurisée des variables d'environnement et des clés API.
- **protobuf**

  Pour la sérialisation efficace des données et la communication entre modules.

Remarque :

L'ensemble de ces bibliothèques est listé dans le fichier requirements.txt du projet, ce qui garantit la reproductibilité de l'environnement de développement et facilite l'installation sur d'autres machines.

4. Conclusion

L'interface utilisateur, moderne et ergonomique, guide l'utilisateur à chaque étape du processus d'extraction. Les différentes fenêtres et zones de logs assurent une transparence totale sur le déroulement des opérations, tandis que la correction manuelle garantit la qualité des données finales.

Les captures d'écran ci-dessus illustrent le parcours utilisateur type, de l'ouverture de l'application à l'export des résultats.

Conclusion Générale

La digitalisation des documents administratifs, et en particulier des actes de naissance, représente un enjeu majeur pour la modernisation et l'efficacité des services publics et privés. Ce projet s'est inscrit dans cette dynamique en proposant une solution complète et automatisée pour l'extraction, la structuration et l'archivage des informations essentielles à partir d'actes de naissance numérisés.

À travers une analyse approfondie du contexte, des besoins et des contraintes, nous avons conçu une architecture modulaire et robuste, intégrant des technologies avancées telles que Tesseract OCR entraîné, des modules de prétraitement d'image, de parsing intelligent et de correction automatique. L'interface graphique, développée avec Tkinter et TTKBootstrap, a été pensée pour offrir une expérience utilisateur intuitive, guidée et transparente, permettant à tout utilisateur de réaliser l'ensemble du processus, de l'import d'une image à l'export des données structurées.

La solution développée répond ainsi aux principaux objectifs fixés : automatiser l'extraction des données, réduire les erreurs et le temps de traitement, et faciliter l'intégration des informations dans des systèmes d'information modernes. La gestion des erreurs, la correction manuelle et la validation automatique garantissent la fiabilité et la qualité des résultats.

Ce travail ouvre la voie à de nombreuses perspectives d'amélioration, telles que l'extension à d'autres types de documents administratifs, l'intégration de modules de reconnaissance de texte manuscrit, ou encore l'enrichissement des fonctionnalités d'export et de recherche. En conclusion, ce projet démontre la faisabilité et l'intérêt d'une solution d'extraction automatisée, contribuant à la modernisation des pratiques administratives et à une meilleure valorisation des données d'état civil.