

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option 1: Génie Logiciel (G.L)

Option 2: Système d'Information et de Connaissances (S.I.C)

Thème

Réalisation d'une application desktop pour la
reconnaissance des zones de texte

Réalisé par :

- **ARBI Hamza**
- **BORSALI Fayçal**

Présenté le 03 juillet 2022 devant le jury composé de

- *Mr. BELABED Amine* (Président)
- *Mr. MATALLAH Houcine* (Encadrant)
- *Mme. CHAOUCHE RAMDANE Lamia* (Examinatrice)
- *Mr. BOUABDALLAH Karim* (Promoteur)

Année universitaire : 2021-2022

Remerciements

*Avant toute chose, nous rendons grâce à **Allah**, le tout puissant, de nous avoir donné la force et la patience de mener à bien ce travail, et nous le prions de nous dresser et nous éclairer un chemin vers la réussite.*

*Nous manifestons nos remerciements les plus sincères et les plus distinguées à **Mr. MATALLAH Hocine** pour son suivi, sa disponibilité et ses conseils tout au long du projet. Ce fut un honneur et un grand plaisir de vous avoir comme encadrant.*

*Nous adressons aussi nos plus profonds remerciements à **Mr. BOUABDALLAH Karim**, notre promoteur au sein de Naltis. Ses remarques, ses conseils et ses critiques constructives nous ont été d'une grande utilité pour mener à terme ce travail.*

Nous sommes reconnaissants à l'ensemble du corps professoral de l'Université Abou Bakr Belkaid de Tlemcen pour leurs enseignements et pour les valeurs qu'ils nous ont inculquées et tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail.

Un remerciement très particulier aux membres du jury qui ont accepté d'évaluer notre travail.

Enfin, nous remercions toute personne ayant contribué de près ou de loin à l'aboutissement de ce travail.

Dédicaces

Je dédie ce modeste travail

A mes très chers parents, qui m'ont tout donné dans ce monde

A mon adorable épouse, pour son inestimable support

A mon frère et mes sœurs, que je chérie profondément

A toute ma famille, qui sont toujours à mes côtés

A mes amis et mes proches, qui m'offrent toujours les plus beaux des souvenirs

A mes chers enseignants du primaire jusqu'à ma dernière année d'études

Et A tout ceux que j'aime.

Merci !

Fayçal Borsali

Dédicaces

A ma très chère mère

Quoi que je fasse ou je dise je ne saurai point de remercier comme il se doit, Ton affection me couvre ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon très cher père

*Tu as toujours été à mes côtés pour me soutenir et m'encourager.
Que ce travail traduit ma gratitude et mon affection.*

A mes sœurs « Kawther » et « Asma », A ma chère tante « Warda » et à sa petite famille, et à toutes les personnes qui ont participé à la cause de ma réussite.

A tous mes professeurs que j'estime, du primaire, lycées, jusqu'à mes dernières années d'université.

Qu'ALLAH vous garde en pleine santé et de bonheur.

Grand merci à vous.

Hamza ARBI

Résumé

Naltis est une entreprise de services et conseil informatique. Ayant plusieurs solutions sur le marché depuis des années et avec plein de clients permanents, l'entreprise travaille en continu à améliorer ses services et vérifier que tous les besoins et désirs de ses clients sont satisfaits. Pour cela, Naltis collabore avec ses clients pour détecter tout éventuel point d'amélioration.

Le présent projet de fin d'études traite un des besoins qui a été reconnu par plusieurs clients de différentes solutions de Naltis. Ce problème est la lecture automatique des feuilles de présence manuscrites. Utile pour les logiciels de gestion d'examen, de gestion de scolarité et potentiellement pour d'autres, Naltis a décidé de dédier tout un projet à résoudre ce problème pour tous ses clients actuels et futurs.

Notre solution est donc un module qui s'intègre dans quasiment n'importe quelle solution qui traite d'une manière ou d'une autre une liste de présence manuscrite. En plus d'une application de bureau qui démontre de manière ergonomique le cas d'utilisation typique de ce module tout en offrant aux futurs développeurs qui vont maintenir notre solution un environnement pour tester et démontrer leurs avancements. La valeur ajoutée de notre projet s'inscrit donc dans le cadre global du suivi qu'offre Naltis à ses clients.

Mots clés : Reconnaissance d'écriture, découpage automatique, feuille de présence, gestion d'examen

Table des matières

| | |
|---|-----------|
| TABLE DES MATIERES | 1 |
| I- INTRODUCTION | 5 |
| I-1 Contexte..... | 5 |
| I-2 Problématique..... | 5 |
| I-3 Objectifs | 6 |
| I-4 Organisation..... | 6 |
| II- ANALYSE..... | 8 |
| II-1 Introduction | 8 |
| II-2 Démarche d'analyse | 8 |
| II-3 Organisme d'accueil..... | 8 |
| II-4 Expression des besoins | 9 |
| II-4.1- Scolimed | 10 |
| Processus existant | 10 |
| Analyse du besoin | 10 |
| II-4.2- QCM Manager..... | 10 |
| Processus existant | 10 |
| Analyse du besoin | 11 |
| II-5 Exigences fonctionnelles..... | 12 |
| II-6 Conclusion..... | 13 |
| III- CONCEPTION | 15 |

| | | |
|--------------|--|-----------|
| III-1 | Introduction..... | 15 |
| III-2 | Approches envisagées..... | 15 |
| III-2.1- | Développement de plusieurs modules | 15 |
| III-2.2- | Développement d'un module intégrable..... | 16 |
| III-3 | Comparaison des approches..... | 16 |
| III-4 | Aperçu de la solution | 17 |
| III-5 | Conception détaillée..... | 18 |
| III-5.1- | Application console..... | 18 |
| a-1- | Reconnaissance des coins et de l'orientation de la feuille..... | 21 |
| a-2- | Détection des zones d'écriture des noms | 21 |
| a-3- | Extraction des zones remplies | 22 |
| a-4- | Reconnaissance des lettres des noms | 22 |
| a-5- | Calcul de la similarité entre les noms détectés et la liste des invités..... | 22 |
| III-5.2- | Application démonstrative..... | 23 |
| b-1- | Importation de la feuille de présence | 24 |
| b-2- | Configuration des options de la séance | 24 |
| b-3- | Impression des feuilles de présence à remplir..... | 24 |
| b-4- | Correction par l'administrateur | 24 |
| III-5.3- | Algorithme de reconnaissance des caractères latins | 26 |
| III-6 | Conclusion | 26 |
| IV- | REALISATION..... | 28 |
| IV-1 | Introduction | 28 |
| IV-2 | Outils et technologies..... | 28 |

| | | |
|-------------|---|-----------|
| IV-2.1- | Python | 28 |
| IV-2.2- | Tensorflow | 29 |
| IV-2.3- | C++ | 29 |
| IV-2.4- | Framework QT | 30 |
| IV-2.5- | Git | 30 |
| IV-2.6- | GitHub..... | 31 |
| IV-2.7- | Microsoft Office..... | 31 |
| IV-2.8- | Diagrams.net (draw.io)..... | 32 |
| IV-3 | Livable finale..... | 32 |
| IV-3.1- | Application console..... | 33 |
| IV-3.2- | Application démonstrative | 37 |
| IV-3.3- | Algorithme de reconnaissance des caractères latins..... | 39 |
| IV-4 | Conclusion..... | 39 |
| V- | CONCLUSION | 42 |
| V-1 | Conclusion générale..... | 42 |
| V-2 | Perspectives..... | 44 |
| | REFERENCES BIBLIOGRAPHIQUES | 45 |
| | LISTE DES FIGURES | 1 |
| | LISTE DES TABLEAUX | 3 |
| | LISTE DES SIGLES ET ABREVIATIONS | 4 |

Chapitre I

Introduction

I- Introduction

I-1 Contexte

Le développement et l'intégration des solutions informatiques a pour objectif principal d'automatiser le traitement des données, de dématérialiser la gestion documentaire et de permettre aux personnes d'être plus créatives, efficaces et productives. Il est donc intéressant pour les entreprises de services informatiques de comprendre le travail de leurs clients et de rapidement détecter les opportunités d'automatiser ou de faciliter une partie du travail en proposant une solution informatisée qui s'intègre dans le quotidien du client et le rend plus heureux.

Dans ce cadre-là, l'entreprise Naltis, l'organisme qui nous a accueilli dans le cadre de ce projet, a mis en place une démarche itérative en collaboration avec les clients de leurs différentes solutions développées. Cette démarche vise à détecter les besoins non satisfaits et les domaines d'amélioration qui sont liés aux processus visés par les solutions déjà acquises par les dits-clients. Cette démarche donne naissance à différents projets qui renforcent les produits offerts par l'entreprise ou donnent naissance à de nouveaux produits.

I-2 Problématique

Une des problématiques qui a découlé de la démarche introduite ci-avant est le besoin régulier et récurrent de gérer des listes de présences. Celles-ci sont manuellement renseignées sur différents logiciels et physiquement archivées. En conséquence, ce processus devient lourd et provoque un gaspillage de temps et d'énergie pour les employeurs et administrateurs tout en engendrant une traçabilité minime.

Notre projet de fin d'études s'est focalisé sur cette problématique, qui consiste à envisager une solution au problème de gestion automatisée des listes de présence afin d'éviter le gaspillage de ressources humaines et d'offrir une meilleure traçabilité de ce processus.

I-3 Objectifs

Le besoin étant présent chez différents clients qui utilisent différents logiciels fonctionnant sous différentes technologies et dans de différents environnements, nous avons fixé avec les parties prenantes au projet que la solution à développer doit être facilement portable et qu'elle soit suffisamment générique pour s'adapter aux différents cas d'utilisations exprimés.

1. Proposition d'une solution informatique portable qui automatise la gestion des listes de présence ;
2. Documentation de la méthode à suivre pour intégrer la solution dans un logiciel existant.

I-4 Organisation

Notre mémoire est organisé en cinq (05) chapitres. Tout d'abord, on va commencer par introduire la problématique et mettre le lecteur dans le contexte de notre PFE. Dans le deuxième chapitre, nous présentons l'analyse effectuée de la situation et des besoins recensés. Dans le troisième chapitre, nous évoquerons les différents choix auxquels on a fait face lors de la conception de notre solution en passant de l'approche générale à la conception détaillée des parties du système développé. Après cela, nous présenterons notre livrable final, ainsi que les outils et technologies utilisées pour réaliser la solution proposée. En conclusion du travail effectué, nous allons revenir sur les points importants de ce projet ainsi qu'un ensemble de perspectives à explorer dans le futur.

Chapitre II :

Analyse

II- Analyse

II-1 Introduction

Dans ce chapitre, nous allons d'abord présenter Naltis, l'entreprise qui nous a accueilli pour réaliser notre projet de fin d'études ; ensuite, nous allons présenter les besoins communs à plusieurs projets qu'on a recensé et qui ont fait l'objet de notre projet, en présentant leur état actuel et la valeur ajoutée qu'on espère apporter par notre solution.

II-2 Démarche d'analyse

Pour une première phase, nous avons commencé par découvrir l'entreprise, ses projets, ses clients et son approche relative à la problématique qui nous intéresse. Dans une deuxième phase, nous avons procédé à des entretiens avec les équipes qui collaborent avec les clients concernés par notre solution pour éclaircir davantage la problématique et mettre la lumière sur les points les plus importants. En dernier lieu, nous avons synthétisé notre compréhension en un ensemble d'exigences qui nous serviront de référence pour le développement de notre solution.

II-3 Organisme d'accueil

Naltis Communication est une entreprise spécialisée dans les nouvelles technologies. Fondée en 2005 par deux jeunes algériens, l'équipe s'est élargie à une trentaine de jeunes talentueux issus de l'université algérienne au fil des années, se situe à Tlemcen très précisément à Bab Wahren.



Figure II.1 Logo de Naltis

La particularité de Naltis réside dans sa capacité de développement des nouveaux supports de communication comme des sites Web, des plates-formes informatiques pouvant offrir de multiples services aux différents utilisateurs (managers, commerciaux, clients, etc.), en traitant l'information de manière poussée et efficace. Dans le cadre de son activité, elle intègre également un département de développement de logiciels, standards ou personnalisés, en fonction des besoins des professionnels.

Services

Naltis offre principalement deux services :

1. Communication d'entreprise

Naltis offre ses services de conseil en communication aux entreprises soucieuses de développer un marketing efficace et ciblé, afin de faire émerger efficacement leur activité au sein d'un environnement de travail actuellement très dense.

2. Soft Design

Du fait de sa longue expérience dans le développement informatique, Naltis est en mesure d'apporter des solutions fiables et stables, capables de répondre de manière spécifique aux besoins des clients tout en leur offrant une grande facilité d'utilisation.

II-4 Expression des besoins

Pour faire notre analyse du besoin, nous avons questionné notre client (représentant de l'entreprise d'accueil) pour établir une liste de projets qui ont le besoin commun 'gestion automatisée des feuilles de présence' ; nous avons réduit cela aux solutions suivantes :

II-4.1- Scolimed

Scolimed est un logiciel de gestion de scolarité développé par Naltis, dédié aux facultés de médecine. Parmi les fonctionnalités de ce logiciel, nous devons gérer la liste des étudiants présents aux TD et TP.

Processus existant

Le processus actuel pour la gestion des présences se présente comme suit :

- La liste des candidats devant assister au cours est établie par avance dans le logiciel avec le programme pédagogique ;
- Lorsqu'un TD ou TP est terminé, le responsable du module doit valider la liste des étudiants absents ;
- La validation des absents se fait manuellement en comparant la feuille de présence avec la liste des étudiants devant assister au TD ou TP correspondant.

Analyse du besoin

Ce processus prend au responsable du module en moyenne une dizaine de minutes par séance. Sachant qu'une semaine typique contient en moyenne 30 séances de présence obligatoire et qu'une année contient en moyenne 30 semaines de cours, le coût moyen annuel de ce processus est de $30 \times 30 \times 10 = 9000$ minutes = 150 heures ! Le client considère ce temps comme un énorme gaspillage de temps et espère le réduire à au moins 50 %.

II-4.2- QCM Manager

QCM Manager est un logiciel de gestion des examens au format QCM avec correction automatique. Parmi les étapes importantes du processus de gestion d'un examen, nous devons gérer la liste des candidats qui se sont absentés à l'examen concerné, il a été conçu et développé par Naltis.

Processus existant

Le processus actuel pour la gestion des présences se résume comme suit:

- Une liste des candidats participants à l'examen est renseignée dans le logiciel, soit manuellement, soit à travers une des différentes méthodes d'imports de liste (fichier, appel d'une API, ...etc.) ;
- Lorsque l'examen est terminé, un agent saisi manuellement la liste des absents en se référant aux feuilles de présence ;
- Après l'opération de correction automatique, une validation de la liste des présents se fait en vérifiant que toute personne non-signalée comme absente détient une feuille de réponses.

Analyse du besoin

Le processus introduit ci-avant souffre d'un défaut reconnu par les clients de la solution. Les clients considèrent l'étape de « saisie de la liste des présents » comme un goulot d'étranglement dans le processus de correction automatique.

En effet, cette saisie se fait en parallèle avec deux autres opérations, à savoir (1) le scan des copies d'examens et (2) le scan du corrigé type.

Les clients considèrent cela comme une vraie problématique car les deux autres opérations sont (1) plus rapides grâce aux différents automatismes mis en place et (2) plus importantes vu la criticité de leurs résultats pour le déroulement de la correction. De ce fait, les clients souhaiteraient pouvoir affecter toute la ressource humaine à ces opérations critiques et pourtant ils sont toujours ralentis à cause de la saisie des absents.

Pour quantifier le coût de ce processus, nous avons contacté trois des clients de la solution QCM Manager pour prendre leur estimation du coût temporel de chacune des opérations parallèles : d'un côté le scan des copies d'examens et corrigé type et d'un autre côté, la saisie de la liste des présents. En résumé, voici leurs réponses illustrées dans le Tableau II.1:

Tableau II.1 coût temporel de la saisie manuelle des absents

| Tâche / Client | Client 1 | Client 2 | Client 3 |
|---|--|---|--|
| <i>Saisie de la liste des présents</i> | 30 minutes (1 agent) | 30 minutes (1 agent) | 15 minutes (1 agent) |
| <i>Scan des copies d'examens & corrigé type</i> | 40 mn (2 agents pendant 30 mn / 3 pendant 10 mn) | 50 mn (1 agent pendant 30 mn / 2 pendant 20 mn) | 20 mn (2 agents pendant 15 mn / 3 pendant 05 mn) |
| Total | 40 minutes | 50 minutes | 20 minutes |
| <i>Processus sans saisie de la liste des présents</i> | 3 agents pendant 20 minutes | 2 agents pendant 35 minutes | 3 agents pendant 10 minutes |
| Gain estimé | 50% du temps | 30% du temps | 50% du temps |

II-5 Exigences fonctionnelles

Après études des cas d'exploitation de notre système, nous avons pu décrire les besoins de notre client selon l'ensemble des exigences fonctionnelles suivant :

- Le système doit permettre de reconnaître automatiquement la liste des personnes mentionnées, à l'écrit, sur une feuille de présence ;
- Le système doit permettre d'importer une liste de personnes concernées par la séance gérée (examen, TD, TP, réunion, ...etc.) ;
- Le système doit permettre à un utilisateur de corriger la liste d'absents détectée automatiquement ;
- Le système doit correctement reconnaître les noms sur la feuille de présence dans au moins 90% des cas ;
- Le système doit être facilement intégrable dans des solutions informatiques existantes ou en développement par l'entreprise.

II-6 Conclusion

Dans ce chapitre, nous avons présenté la démarche menée dans cette phase d'analyse de la situation actuelle, en commençant par lister un ensemble d'applications qui ont le besoin similaire de « traitement automatisé des feuilles de présence », et par la suite rassembler tout ce qu'on a appris dans une liste d'exigences fonctionnelles du système.

Chapitre III :

Conception

III- Conception

III-1 Introduction

Ce chapitre va être consacré à la conception de notre solution. Tout d'abord, nous présenterons les approches générales qu'on a considéré avant de les comparer et de présenter l'approche qu'on a retenu. Ensuite, on va présenter la conception détaillée des parties importantes de notre système.

III-2 Approches envisagées

La problématique qui nous a été confiée présente plusieurs défis, parmi les plus importants de ces défis, nous citons la diversité des logiciels qui souffrent de la problématique soulevée. En effet, ces logiciels utilisent des technologies différentes et sont développés pour des plateformes différentes (applications de bureau, applications web, applications mobiles, ...). Nous avons donc considéré deux approches différentes :

III-2.1- Développement de plusieurs modules

Dans cette approche, on a proposé de :

1. Développer le système de gestion des feuilles de présence dans une application pilote qui servira à développer les fonctionnalités essentielles de ce système ;
2. Faire un développement personnalisé pour chaque solution ayant besoin de la fonctionnalité « gestion des feuilles de présence intégrées » en exploitant les résultats de l'application pilote ;
3. Intégrer le tout en une seule solution, dans une prochaine étape.

Cette approche a pour avantage la simplicité de la réalisation. Cela est dû au fait qu'on ne considère qu'un seul cas d'utilisation à la fois, donc chaque module couvrira uniquement les fonctionnalités souhaitées dans ce cas particulier sans se soucier des autres fonctionnalités.

L'inconvénient de cette solution est qu'elle est coûteuse (plusieurs cycles de développement) et très difficilement maintenable vu que la même fonctionnalité sera

Conception Comparaison des approches
développée en utilisant différentes technologies et sur différents projets, cela va obliger les développeurs à fournir des efforts supplémentaires pour maintenir ces solutions.

III-2.2- Développement d'un module intégrable

Dans cette approche, on a proposé l'approche inverse à la première solution :

1. Développer un module indépendant ;
2. Développer une solution pilote qui encapsule le module développé et présente aux utilisateurs-type de la solution une interface à travers laquelle ils peuvent utiliser le module développé ;
3. Intégrer le module dans les différents logiciels qui souffrent du problème défini ;

L'avantage de cette approche est qu'une fois la solution développée et intégrée, il suffira de maintenir un seul projet sur une seule technologie et de lancer par la suite une multitude de très petits projets pour l'intégration de ce module.

Cette approche a pour inconvénient la complexité précoce à deux niveaux : (1) dans la conception de la solution en résumant différentes solutions avec des exigences potentiellement divergentes et (2) dans l'intégration du module dans les différentes solutions qui sont supportées par différentes technologies.

III-3 Comparaison des approches

Nous synthétisons la comparaison entre les différentes approches présentées ci-dessus dans le tableau Tableau III.1 :

Tableau III.1 comparaison des approches de la solution

| <i>Critère / Approche</i> | Plusieurs modules | Un seul module | Commentaire |
|---|------------------------|-----------------|---|
| <i>Complexité premier développement</i> | Simple | Plutôt complexe | - Il a été décidé de limiter les fonctionnalités du module pour éviter la complexité de l'abstraction |
| <i>Complexité intégration</i> | Complexe et répétitive | Plutôt simple | / |
| <i>Maintenabilité</i> | Complexe et répétitive | Simple | / |

III-4 Aperçu de la solution

La solution retenue est le développement d'un module indépendant et portable, en établissant la feuille de route suivante :

1. Développer une application console qui traite les feuilles de présence scannées d'un événement (examen, cours, TD, TP, réunion, ...etc.) ;
2. Intégrer le module dans une application Desktop présentant le cas d'utilisation typique de ce module ;
3. Développer des kits de développement (SDK) dans les langages et Frameworks utilisés par l'entreprise. Ces SDK encapsulent notre application console et rendent son intégration bien plus simple.

Après concertation avec l'entreprise, et en considérant les contraintes temporelles du projet, il a été décidé de laisser la dernière étape de la feuille de route (développement des SDK) en perspective et de concentrer les efforts de l'actuel projet à atteindre les deux premiers jalons de la feuille de route.

III-5 Conception détaillée

III-5.1- *Application console*

La console est définie par Gartner comme étant la partie d'un ordinateur utilisée pour communiquer entre l'utilisateur et le système. Le type de consoles le plus courant est un terminal à tube cathodique (CRT - Cathode Ray Tube) avec souris et clavier. [1]

Le choix d'une application console se justifie par le fait qu'elles soient utilisables par les humains (ayant un minimum de compétences techniques) et par d'autres logiciels de manière transparente aux utilisateurs finaux.

Notre module réalise une tâche atomique, c'est pour cela que nous avons conçu notre application console comme étant une application autonome qui peut et doit s'exécuter correctement sans avoir connaissance d'aucun contexte.

L'application prend en entrée les arguments suivants :

- Une liste de personnes (candidats, étudiants, invités, ... etc.) qui sont censées être présentes ;
- Un ensemble d'images de la (les) feuille(s) de présence ;
- Des paramètres de l'algorithme de détection [Optionnel].

L'application réalise la suite d'opérations atomiques suivantes :

1. Lecture des feuilles de présence une après une :
 - a. Reconnaissance des coins et de l'orientation de la feuille ;
 - b. Détection des zones d'écriture des noms ;
 - c. Extraction des zones remplies (avec des noms) ;
2. Reconnaissance des lettres des noms ;
3. Calcul de la similarité entre les noms détectés et la liste fournie en entrée ;

- Le système peut potentiellement contenir des personnes dans les noms sont homonymes (même nom et même prénom), cette exigence est satisfaite par l'ajout optionnel des dates de naissance comme troisième champ de reconnaissance.

Dans les paragraphes suivants, nous allons éclaircir quelques détails sur les étapes les plus importantes du processus détaillé ci-dessus :

a-1- Reconnaissance des coins et de l'orientation de la feuille

La première étape de la lecture des feuilles de présence est de s'assurer que la feuille scannée est bien dans les dimensions attendues dans le reste des étapes. Pour cela, nous devons concevoir une feuille de présence vide facile à scanner et détecter son orientation automatiquement. Le concept le plus important à introduire dans notre modèle de feuilles de présence est le point d'ancrage. Nous avons conçu notre modèle pour contenir sur 3 de ses coins, un point d'ancrage qui nous permettra de reconnaître où se trouvant ces coins.

Ces points d'ancrages sont des points noirs suffisamment gros pour ne pas être confondus avec des points dû aux anomalies du scan ou des tâches sur la feuille. Pour les reconnaître, nous allons suivre le processus suivant :

1. Parcourir les pixels de l'image scannée depuis les quatre coins, jusqu'à ce que nous retrouvions trois points d'ancrage ;
2. Tourner l'image de sorte que les trois points d'ancrage se positionnent aux coins (Supérieur Droit, Supérieur Gauche, Inférieur Droit) ;
3. Redimensionner l'image de sorte que les trois points d'ancrage soient dans leur position prédéfinie (distance absolue par rapport au coin supérieur gauche considéré comme le point (0, 0)).

a-2- Détection des zones d'écriture des noms

La détection des zones d'écriture des noms est une des étapes les plus simples. Une fois l'image bien orientée et redimensionnée, il suffit de récupérer les pixels qui représentent les cases à remplir dans le modèle généré pour obtenir les zones d'écriture des noms.

a-3- Extraction des zones remplies

Cette étape sert à éliminer les lignes vides en calculant le pourcentage de remplissage de chaque case de la zone. Si toutes les cases contiennent un pourcentage inférieur à 5%, on considère que la ligne est vide.

a-4- Reconnaissance des lettres des noms

Vu que chaque case représente une seule lettre latine, nous nous sommes retrouvés face au choix de développer notre propre modèle intelligent de reconnaissance des caractères latins ou bien d'utiliser des outils disponibles sur le Web. Les domaines de la « reconnaissance optique de caractères » (OCR) et de la « reconnaissance de texte manuscrit » (HTR) étant bien développés [2] [3], nous avons décidé d'intégrer un des outils disponibles en ligne. Ces outils reçoivent en entrée une image d'un caractère et retournent une liste de caractères candidats avec le degré de confiance en chacun d'entre eux. Nous prendrons toujours le meilleur résultat. Nous détaillons un peu plus les étapes qui ont mené au choix de cet algorithme dans la chapitre III-5.3- (p.26).

a-5- Calcul de la similarité entre les noms détectés et la liste des invités

A l'issue de l'étape précédente, on aura une suite de lettres détectées pour chaque nom, il suffira de concaténer tous les caractères reconnus pour avoir le nom prédit par notre algorithme. Cependant, il se peut qu'un des caractères soit mal reconnu, pour palier à cela, et étant donné que notre champ de recherche est réduit, nous procédons à un calcul de similarité entre le nom prédit et la liste de toutes les personnes invitées pour corriger la prédiction. Afin d'établir un score de similarité entre deux chaînes de caractères, il suffit de prendre l'inverse de leurs distances en utilisant une des d'algorithmes de distance existantes. Parmi les plus simples algorithmes, est la distance de Levenshtein qui commence à 0 et est incrémentée de 1 à chaque différence trouvée entre les deux chaînes de caractères.

Par exemple, supposons que notre algorithme détecte la chaîne de caractères 'Karjm' et que notre liste d'invités contient les 2 prénoms Karim, Fayçal et Hamza, alors on aura :

- Distance (Karjm, Karim) = 1 (il n'y a qu'un seul caractère différent entre les deux : j et i).
- Distance (Karjm, Hamza) = 4 (4 caractères différents, seul le a se trouve dans les deux chaînes)
- Distance (Karjm, Fayçal) = 5 (4 caractères différents + le terme Fayçal contient une lettre en plus)

Après avoir calculé cette distance, on prend son inverse (plutôt l'inverse de la distance plus 1 pour éviter des divisions par 0 en cas de similarité parfaite) et on prend le nom qui présente la meilleure similarité donc la plus petite distance avec le mot original.

III-5.2- Application démonstrative

L'application démonstrative est une application pilote qui sert, d'un côté, à présenter les scénarios typiques d'exploitation de notre solution ; et d'un autre côté, de cadre pour le développement continu de l'application console présentée ci-dessus. Dans ce qui suit, nous allons présenter le scénario principal d'exploitation du système développé :

– Avant la séance –

1. Création d'une séance ;
2. Importation de la liste des participants ;
3. Configuration des options de la séance ;
4. Impression des feuilles de présence à remplir ;

– Pendant la séance –

5. Remplissage des feuilles de présences ;

– Après la séance –

6. Scan des feuilles de présence ;
7. Importation des feuilles de présence scannées ;
8. Appel du module de reconnaissance (Application console présentée ci-dessus) ;
9. Correction si des noms sont mal reconnus ;
10. Affichage du résultat dans la liste des séances organisées.

Dans le chapitre suivant, nous allons détailler quelques étapes citées ci-dessus :

b-1- Importation de la feuille de présence

Les formats supportés sont : Excel ou CSV avec les colonnes suivantes :

- Nom ;
- Prénom ;
- Date de naissance [Optionnel].

b-2- Configuration des options de la séance

La seule option pertinente pour notre projet est le choix d'utilisation ou non-utilisation de la date de naissance pour renforcer la reconnaissance des noms. Cela affecte deux étapes :

- Le modèle des feuilles à imprimer sera différent, selon notre besoin ou non de l'information date de naissance ;
- L'algorithme de détection change légèrement selon ce paramètre.

b-3- Impression des feuilles de présence à remplir

Le nombre de feuilles à imprimer est automatiquement calculé selon le nombre de personnes invitées (10 par feuille si les dates de naissance sont utilisées, sinon 12 par feuille).

b-4- Correction par l'administrateur

Nous allons expliquer le fonctionnement de cette partie en s'appuyant sur les deux diagrammes suivants

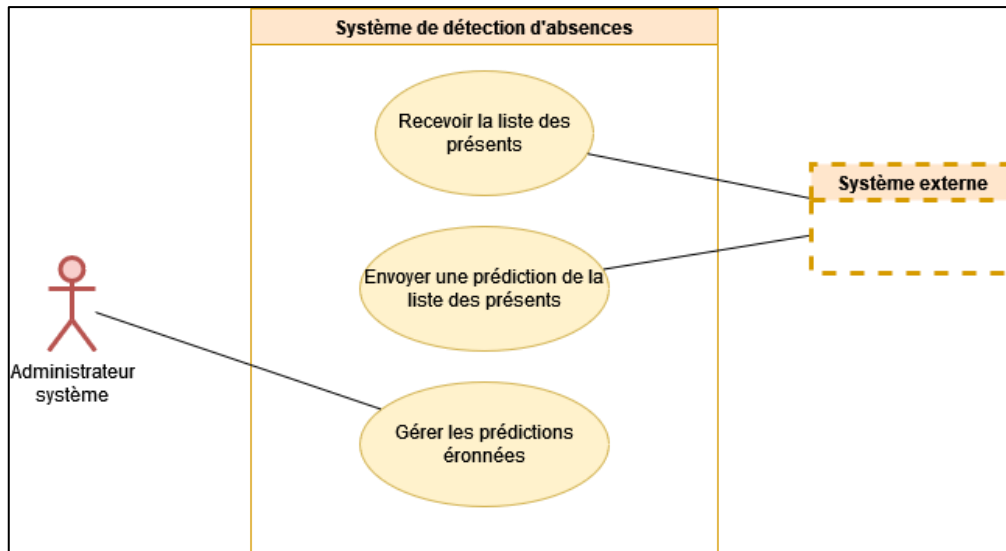


Figure III.3 Diagramme des cas d'utilisation de l'administrateur de l'application démonstrative

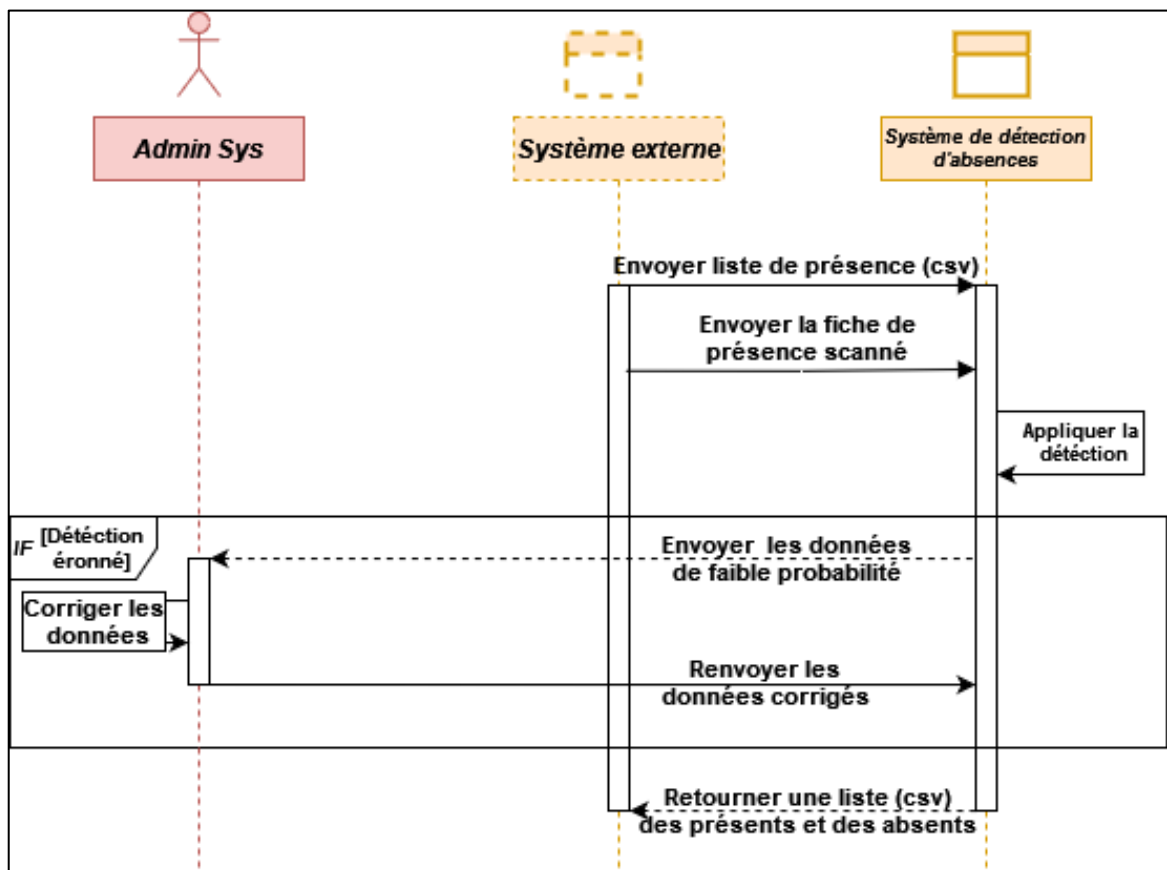


Figure III.4 Diagramme de séquence de l'administrateur de l'application démonstrative

III-5.3- *Algorithme de reconnaissance des caractères latins*

Afin de choisir un algorithme intelligent à utiliser, nous avons commencé à explorer les différents répertoires de logiciels open source (GitHub, Kaggle¹, ...etc.) puis les avons évalués en suivant l'approche suivante :

1. Nous nous intéressons aux données utilisées pour l'apprentissage ;
2. Nous nous intéressons à savoir si le projet contient un modèle pré-entraîné (qui est donc déjà assez intelligent pour plusieurs cas d'utilisation) ;
3. Nous évaluons le modèle entraîné (ou pré-entraîné) sur le jeu de données utilisé dans la phase d'entraînement ;
4. Si les résultats sont satisfaisants (>95%), nous allons tester le même algorithme sur un jeu de données que nous avons nous-même préparé ;
5. Si besoin, nous continuons l'entraînement de l'algorithme sur une partie du jeu de données que nous avons préparé.

Le jeu de données mentionné ci-dessus est extrait d'anciens projets de Naltis qui ont utilisé des feuilles similaires à notre feuille de présence (composé d'un ensemble de carreaux à remplir par des lettres). Nous avons-nous-même annoté les données avec la bonne lettre correspondante et les avons rassemblés dans un répertoire.

III-6 Conclusion

Dans ce chapitre, nous avons présenté la conception de notre solution, les différents choix que nous avons effectué, les fonctionnalités qu'on a décidé de développer et comment nous avons décidé d'affronter la problématique qui nous a été confiée. Dans le chapitre suivant, nous allons s'étaler sur la mise en œuvre de tout ce qui a été conçu.

¹ Kaggle est un site web qui rassemble une communauté de passionnés d'intelligence artificielle et machine learning, le site offre plein de défis, de bases de données de solutions open source

Chapitre IV :

Réalisation

IV- Réalisation

IV-1 Introduction

Dans ce chapitre, nous allons présenter les différents outils et technologies que nous avons utilisés pour réaliser notre projet en justifiant chaque fois les choix techniques effectués. Dans ce chapitre, nous allons aussi illustrer les interfaces les plus importantes de notre application.

IV-2 Outils et technologies

IV-2.1- *Python*

Python est un langage de programmation interprété, de haut niveau et polyvalent. Créé par Guido van Rossum et publié pour la première fois en 1991. Python est typé dynamiquement et doté d'un ramasse-miette. Il prend en charge plusieurs paradigmes de programmation, y compris la programmation structurée (en particulier, procédurale), orientée objet et fonctionnelle. Python est souvent décrit comme un langage « avec piles » en raison de sa bibliothèque standard complète. Python 3.0, sorti en 2008, était une révision majeure du langage et n'est pas complètement rétro-compatible ; une grande partie du code Python 2 ne s'exécute pas sans modification sur Python 3. [4]



Figure IV.1 Logo de python

Nous avons utilisé le langage Python dans le développement de notre application Console.

L'avantage de Python est sa simplicité, la facilité de l'installer sur différents systèmes et la facilité d'intégration des modèles intelligents requis pour la reconnaissance des caractères.

IV-2.2- *Tensorflow*

TensorFlow est une bibliothèque de logiciels gratuite et open source conçue pour l'apprentissage automatique et l'intelligence artificielle. Elle peut être utilisée dans plein de différents contextes, mais elle se concentre particulièrement sur la formation et l'inférence des réseaux de neurones profonds. [5]



Figure IV.2 Logo de TensorFlow

Nous avons utilisé TensorFlow dans l'exploitation du modèle de détection des lettres manuscrites et son ré-entraînement.

IV-2.3- *C++*

C++ est un langage de programmation à usage général créé par l'informaticien danois Bjarne Stroustrup comme une extension du langage de programmation C, ou « C avec classes ». [6]



Figure IV.3 Logo de C++

Nous avons utilisé le langage C++ dans le développement de notre application de bureau démonstrative.

IV-2.4- Framework QT

QT est un logiciel multiplateforme permettant de créer des interfaces utilisateur graphiques ainsi que des applications multiplateformes qui s'exécutent sur diverses plateformes logicielles et matérielles telles que Linux, Windows, macOS, Android ou même sur des systèmes embarqués avec un éventuel léger changement dans la base de code sous-jacente tout en restant une application native avec des capacités et une vitesse native.

*Figure IV.4 Logo de QT*

Nous avons utilisé le Framework QT dans le développement de notre application de bureau démonstrative.

IV-2.5- Git

Git est un logiciel décentralisé de gestion de versions. Il est conçu pour être efficace tant avec les petits projets, qu'avec les plus importants. Nous avons utilisé Git pour le versioning tout au long du projet.



Figure IV.5 Logo de git

Nous avons utilisé git pour collaborer pendant la phase de développement et garder un historique des versions du code.

IV-2.6- GitHub

GitHub est un service d'hébergement de référentiel Git, en ajoutant un bon nombre de ses fonctionnalités. Alors que Git est un outil en ligne de commande, GitHub fournit une interface graphique basée sur le Web. Il fournit également un contrôle d'accès et plusieurs fonctionnalités de collaboration.

*Figure IV.6 Logo de GitHub*

Nous avons utilisé GitHub pour partager nos répertoires Git et garder une trace de l'avancement du projet grâce aux différentes fonctionnalités que GitHub offre (comme l'onglet 'issues')

IV-2.7- Microsoft Office

Microsoft Office, ou simplement Office, est une famille de logiciels clients, de logiciels serveurs et de services développés par Microsoft. Il a été annoncé pour la première fois par Bill Gates en 1988.



Figure IV.7 Logo de Microsoft Office

Nous avons utilisé la suite Office pour la rédaction de ce rapport et des différents documents intermédiaires produits dans le cadre de notre projet (documents de conception, PV de réunions, ...etc.)

IV-2.8- Diagrams.net (draw.io)

Diagrams.net est un outil en ligne qui permet de créer des diagrammes UML et des schémas de bases de données, avec la possibilité de collaborer avec d'autres utilisateurs. Nous l'avons utilisé dans notre projet pour établir la conception et la création des différents diagrammes.



Figure IV.8 Logo de Diagrams.net

Nous avons utilisé le logiciel web 'Diagrams.net' pour générer les graphes insérés dans ce rapport.

IV-3 Livrable finale

Nous allons présenter quelques captures d'écran de notre version finale du livrable.

IV-3.1- *Application console*

Pour démontrer le fonctionnement de l'application console, nous allons présenter un scénario exemple. Supposons que nous avons la structure de fichiers suivante :

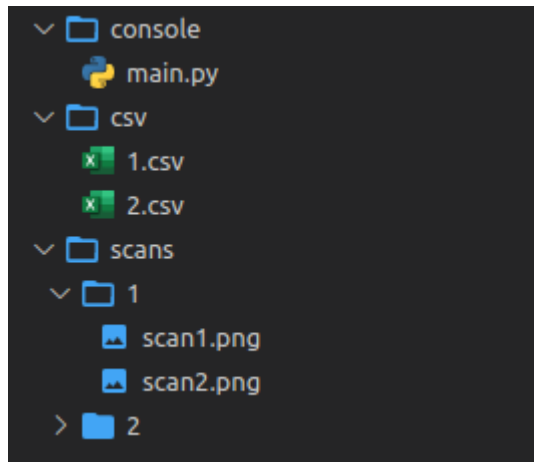


Figure IV.9 Structure des fichiers (exemple)

Le répertoire *console* contient le code de notre application console. Nous avons deux séances (1 et 2), mais nous allons montrer seulement le traitement de la séance n°1 comme exemple.

Le fichier *1.csv* contient la liste de invités suivante

| Nom ▼ | Prenom ▼ | Date de naissance ▼ |
|---------------|-----------------|---------------------|
| Bestaoui | Mohammed Nassim | 31/07/2004 |
| Kada | Mohammed Reda | 17/01/2004 |
| Baba Ahmad | Amanallah | 09/11/2004 |
| Benhadji | Ibrahim | 14/03/2005 |
| Benchira | Siraj | 22/05/2004 |
| Belarbi | Ayoub | 25/07/2004 |
| Tabet Zatla | Meriem | 17/05/2004 |
| Benzerhouni | Bouchra | 09/05/2005 |
| Hamdan | Neila | 09/11/2004 |
| Boublenza | Lilya Rym | 05/05/2003 |
| Bourki Hacene | Sirine | 07/09/2005 |
| Hadj Slimane | Abdel Illah | 10/03/2005 |
| Belarbi | Adem | 25/07/2004 |
| Elyebedri | Nour El Houda | 08/05/2005 |
| Hadj Slimane | Fadia | 04/01/2005 |

Figure IV.10 Liste des invités (exemple)

Le répertoire *scans/I/* contient les deux scans suivants :

Séance N° : 1

Liste de présence

Date : 27/05/2022

| | | | | | | | | | | | | | | | | | | | | |
|---|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 1 | Nom : | B | O | U | A | B | D | A | L | L | A | H | | | | | | | | |
| | Prénom : | S | A | L | I | M | A | | | | | | | | | | | | | |
| | Date de Naissance : | 0 | 5 | / | 0 | 4 | / | 1 | 9 | 9 | 2 | | | | | | | | | |
| 2 | Nom : | B | E | N | G | U | E | R | F | I | | | | | | | | | | |
| | Prénom : | W | A | L | I | D | | | | | | | | | | | | | | |
| | Date de Naissance : | 0 | 4 | / | 1 | 2 | / | 2 | 0 | 1 | 0 | | | | | | | | | |
| 3 | Nom : | B | E | L | A | I | Δ | I | | | | | | | | | | | | |
| | Prénom : | S | A | L | I | M | A | | | | | | | | | | | | | |
| | Date de Naissance : | 1 | 2 | / | 1 | 2 | / | 1 | 9 | 5 | 4 | | | | | | | | | |
| 4 | Nom : | A | I | S | S | A | O | U | I | | | | | | | | | | | |
| | Prénom : | F | A | T | I | N | A | Z | O | H | R | A | | | | | | | | |
| | Date de Naissance : | 0 | 7 | / | 0 | 8 | / | 1 | 9 | 9 | 4 | | | | | | | | | |
| 5 | Nom : | A | I | S | S | A | O | U | I | | | | | | | | | | | |
| | Prénom : | I | L | H | E | N | | | | | | | | | | | | | | |
| | Date de Naissance : | 1 | 2 | / | 1 | 0 | / | 1 | 9 | 9 | 7 | | | | | | | | | |
| 6 | Nom : | A | I | S | S | A | O | U | I | | | | | | | | | | | |
| | Prénom : | I | B | T | I | S | S | E | N | | | | | | | | | | | |
| | Date de Naissance : | 1 | 3 | / | 1 | 1 | / | 2 | 0 | 0 | 1 | | | | | | | | | |
| 7 | Nom : | S | A | L | A | H | | | | | | | | | | | | | | |
| | Prénom : | M | E | R | I | E | M | N | I | H | E | L | | | | | | | | |
| | Date de Naissance : | 0 | 1 | / | 0 | 7 | / | 1 | 9 | 9 | 9 | | | | | | | | | |
| 8 | Nom : | B | E | L | H | A | M | R | A | | | | | | | | | | | |
| | Prénom : | S | A | R | A | H | | | | | | | | | | | | | | |
| | Date de Naissance : | 1 | 3 | / | 0 | 8 | / | 1 | 9 | 9 | 3 | | | | | | | | | |
| 9 | Nom : | G | U | I | D | O | U | M | | | | | | | | | | | | |
| | Prénom : | A | B | D | A | L | L | A | H | A | B | D | E | L | K | A | D | E | R | |
| | Date de Naissance : | 1 | 6 | / | 0 | 4 | / | 1 | 9 | 9 | 5 | | | | | | | | | |

Figure IV.11 Scan 1 (exemple)

Séance N° : 1

Liste de présence

Date : 27/05/2022

| | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----------|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|
| 10 | Nom : | B | E | L | H | A | M | R | A | | | | | | | | | | | | | | | |
| | Prénom : | M | O | H | A | M | M | E | D | S | A | L | A | H | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 2 | 1 | / | 0 | 1 | / | 1 | 9 | 5 | 4 | | | | | | | | | | | | | |
| 11 | Nom : | B | E | N | G | H | E | R | R | A | | | | | | | | | | | | | | |
| | Prénom : | A | H | M | E | D | F | E | T | H | I | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 9 | / | 0 | 3 | / | 1 | 9 | 9 | 7 | | | | | | | | | | | | | |
| 12 | Nom : | K | H | O | D | J | A | | | | | | | | | | | | | | | | | |
| | Prénom : | A | M | I | N | A | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 5 | / | 0 | 8 | / | 1 | 9 | 9 | 8 | | | | | | | | | | | | | |
| 13 | Nom : | A | B | O | U | M | A | L | E | K | | | | | | | | | | | | | | |
| | Prénom : | T | A | R | E | K | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 3 | 0 | / | 1 | 2 | / | 1 | 9 | 9 | 9 | | | | | | | | | | | | | |
| 14 | Nom : | L | A | I | R | E | D | J | | | | | | | | | | | | | | | | |
| | Prénom : | M | O | H | A | M | M | E | D | A | M | I | N | E | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 5 | / | 0 | 4 | / | 2 | 0 | 0 | 1 | | | | | | | | | | | | | |
| 15 | Nom : | S | A | D | E | D | D | I | N | E | | | | | | | | | | | | | | |
| | Prénom : | S | A | F | A | A | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | 2 | 5 | / | 0 | 2 | / | 1 | 9 | 9 | 5 | | | | | | | | | | | | | |
| 16 | Nom : | | | | | | | | | | | | | | | | | | | | | | | |
| | Prénom : | | | | | | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | / | | | | | | | | | | | | | | | | | |
| 17 | Nom : | | | | | | | | | | | | | | | | | | | | | | | |
| | Prénom : | | | | | | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | / | | | | | | | | | | | | | | | | | |
| 18 | Nom : | | | | | | | | | | | | | | | | | | | | | | | |
| | Prénom : | | | | | | | | | | | | | | | | | | | | | | | |
| | | Date de Naissance : | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | / | | | | | | | | | | | | | | | | | |

Figure IV.12 Scan 2 (exemple)

Nous pouvons traiter cette séance en lançant la commande suivante

`python console/app.py --scan-dir ./scans/reunions/1/ --list ./csv/1.csv` où l'option *scan-dir* représente le dossier contenant les scans à traiter et l'option *list* représente la liste des invités en csv. Ci-après le résultat de l'appel de cette fonction :

```
INFO: Lancement de la détection des absents ...
INFO: Traitement des scans 1/2
INFO: Traitement terminé 1/2 (6 entrées)
INFO: Traitement des scans 2/2
INFO: Traitement terminé 2/2 (6 entrées)

INFO: Résultat de la détection des absents ...
```

| Nom | Prénom | Date nais. | Présence |
|---------------|-----------------|------------|----------|
| Bestaoui | Mogammed nassim | 31/07/2004 | 1 |
| Kada | Mohammed reda | 17/01/2004 | 1 |
| Baba Ahmad | Amanallah | 09/11/2004 | 1 |
| Benhadji | Ibrahim | 14/03/2005 | 1 |
| Benchira | Siraj | 22/05/2004 | 1 |
| Belarbi | Ayoub | 25/07/2004 | 1 |
| Tabet Zatla | Meriem | 17/05/2004 | 1 |
| Benzerhouni | Bouchra | 09/05/2005 | 1 |
| Hamdan | Neila | 09/11/2004 | 1 |
| Boublenza | Lilya Rym | 05/05/2003 | 1 |
| Bourki Hacene | Sirine | 07/09/2005 | 1 |
| Hadj Slimane | Abdel Illah | 10/03/2005 | 1 |
| Belarbi | Adem | 25/07/2004 | 0 |
| Elyebdri | Nour El Houda | 08/05/2005 | 0 |
| Hadj Slimane | Fadia | 04/01/2005 | 0 |

Figure IV.13 Résultat appel console (exemple)

IV-3.2- Application démonstrative

Ci-dessous le même scénario mais cette fois-ci utilisé avec l'application démonstrative (sans faire d'appel console direct).

| Séance : 01 | | | | |
|-------------|---------------|-----------------|-------------------|----------|
| Id | Nom | Prénom | Date de naissance | Présence |
| 1 | Bestaoui | Mohammed nassim | 2004-07-31 | - |
| 2 | Kada | Mohammed reda | 2004-01-17 | - |
| 3 | Baba Ahmad | Amanallah | 2004-11-19 | - |
| 4 | Benhadji | Ibrahim | 2005-03-14 | - |
| 5 | Benchira | Siraj | 2004-05-22 | - |
| 6 | Belarbi | Ayoub | 2004-07-25 | - |
| 7 | Tabet Zatla | Meriem | 2004-05-17 | - |
| 8 | Benzerhouni | Bouchra | 2005-05-09 | - |
| 9 | Hamdan | Neila | 2004-11-09 | - |
| 10 | Boublenza | Lilya Rym | 2003-05-05 | - |
| 11 | Bourki Hacene | Sirine | 2005-09-07 | - |
| 12 | Hadj Slimane | Abdel Illah | 2005-03-10 | - |
| 13 | Belarbi | Adem | 2004-07-25 | - |
| 14 | Elyebdri | Nour Elhouda | 2005-05-08 | - |
| 15 | Hadj Slimane | Fadia | 2005-01-04 | - |

Vérifier les absences

Figure IV.14 Liste des invités - avant séance (exemple)

| Séance : 01 | | | | |
|-------------|---------------|-----------------|-------------------|----------|
| Id | Nom | Prénom | Date de naissance | Présence |
| 1 | Bestaoui | Mohammed nassim | 2004-07-31 | Présent |
| 2 | Kada | Mohammed reda | 2004-01-17 | Présent |
| 3 | Baba Ahmad | Amanallah | 2004-11-19 | Présent |
| 4 | Benhadji | Ibrahim | 2005-03-14 | Présent |
| 5 | Benchira | Siraj | 2004-05-22 | Présent |
| 6 | Belarbi | Ayoub | 2004-07-25 | Présent |
| 7 | Tabet Zatla | Meriem | 2004-05-17 | Présent |
| 8 | Benzerhouni | Bouchra | 2005-05-09 | Présent |
| 9 | Hamdan | Neila | 2004-11-09 | Présent |
| 10 | Boublenza | Lilya Rym | 2003-05-05 | Présent |
| 11 | Bourki Hacene | Sirine | 2005-09-07 | Présent |
| 12 | Hadj Slimane | Abdel Illah | 2005-03-10 | Présent |
| 13 | Belarbi | Adem | 2004-07-25 | Absent |
| 14 | Elyebdri | Nour Elhouda | 2005-05-08 | Absent |
| 15 | Hadj Slimane | Fadia | 2005-01-04 | Absent |

Vérifier les absences

Figure IV.15 Liste des invités - après séance (exemple)

IV-3.3- *Algorithme de reconnaissance des caractères latins*

L'algorithme retenu de la démarche décrite en III-5.3- (p.26) provient d'un répertoire stocké sur la plateforme 'Kaggle'. Afin de le décrire, nous allons présenter les informations suivantes le concernant :

- Entrée acceptée par l'algorithme : image de taille 28x28 pixels en dégradé de gris ;
- L'algorithme essaye de détecter une des 26 lettres de l'alphabet (majuscule) ;
- L'algorithme est un projet Tensorflow ;
- Le répertoire contient un modèle entraîné stocké sous forme de fichier binaire ;
- Le modèle est un réseau de neurones composé des couches suivantes :
 - o Un réseau de neurones convolutifs [7] de taille 5x5, prenant en entrée l'image (de taille 28x28) et donnant des filtres de taille 32 pour chaque pixel traité (activation RELU) ;
 - o Une couche de MaxPooling [8] qui filtre et réduit la taille des données en remplaçant chaque matrice 2x2 par son maximum ;
 - o Une couche de dropout [9] qui enlève au hasard 30% données ;
 - o Une couche qui réduit la dimension des données à une seule dimension (flatening) ; [10]
 - o Deux couches de réseaux de neurones densément connectés qui transforme les données à une dimension de 128 (activation RELU [11]) puis à une dimension de 26 (le nombre de lettres majuscules) (activation softmax [12]) ;
- L'algorithme détecte les lettres du jeu de données total (jeu d'entraînement ainsi que le jeu de test) avec une précision avoisinant les 97% ;
- L'algorithme détecte les lettres de notre jeu de données (provenant de Naltis) avec une précision avoisinant les 70% ;
- Après un réentraînement sur notre jeu de données (séparé en jeu d'entraînement et jeu de test), la précision a été améliorée pour dépasser les 90%. Ce résultat a été jugé suffisant par le client.

IV-4 Conclusion

Dans ce chapitre, nous avons présenté notre livrable finale, les différentes technologies utilisés ainsi que quelques scénarios d'exploitation pour illustrer notre

Réalisation

Conclusion

travail. Ainsi, nous aurons conclu la présentation de notre travail depuis la toute première étape d'analyse à la phase finale de la réalisation.

Chapitre V :

Conclusion

V- Conclusion

V-1 Conclusion générale

Le projet qui nous a été confié par l'entreprise Naltis consistait à analyser un problème récurrent rencontré par plusieurs de leurs clients dans l'exploitation de leurs solutions informatiques. Notre projet consistait aussi à concevoir et réaliser une solution pratique, efficace et flexible.

L'objectif de la problématique traitée est de remédier au renseignement manuel des listes de présents ou d'absents dans certains logiciels qui consommait beaucoup de temps et d'efforts fournis inutilement par les agents ou utilisateurs chargés de cette tâche.

D'un autre côté, l'entreprise Naltis a voulu saisir cette opportunité pour se doter d'une solution de reconnaissance automatique des listes écrites à la main. Cette dernière sera utile dans plusieurs systèmes actuels et futurs.

Les principales contributions du système conçu et réalisé se situent à plusieurs niveaux :

- Tout d'abord, l'exploitation d'une collection de données (dataset) composé de photos de lettres manuellement écrites et de leur annotation automatique ;
- L'enrichissement de la collection de données et sa publication en tant que base de données open source ;
- La mise à la disposition de Naltis d'un module de détection automatique des lettres écrites ;
- La mise à la disposition de Naltis d'un module de lecture automatique des feuilles de présence qui exploite le module mentionné précédemment ;
- L'intégration du module dans une application démonstrative.

Nous sommes aussi conscients du fait que la réussite de la mise en place de la solution projetée repose sur :

- Le développement continu du modèle de reconnaissance des lettres manuscrites, en se basant sur les lettres mal-défectées récoltées par les différentes solutions qui exploitent le modèle ;
- Le développement de SDKs pour les différentes technologies utilisées pour faciliter l'intégration de la solution développée aux différentes solutions informatiques existantes.

Nous sommes par la même occasion doublement ravis d'avoir :

- Contribué à la stratégie de Naltis qui se veut comme une entreprise qui développe continuellement les solutions mises à la disposition de ses clients, et ce en utilisant les technologies les plus récentes et performantes ;
- Participé à renforcer le partenariat entre Naltis et le département Informatique de l'université de Tlemcen : d'un côté, l'université aide Naltis à développer ses différentes solutions informatiques à travers ses étudiants stagiaires et diplômés ; et Naltis participe à la formation des étudiants de l'université en les encadrant et les intégrant dans le monde professionnel.

Le présent projet a été pour nous, une expérience très riche en enseignements pour les différents raisons suivantes :

- Le fait d'avoir utilisé un mixte entre la démarche projet avec la méthode agile qui nous a permis de profiter des points forts des deux méthodes dans chaque utilisation ;
- Le fait d'avoir intégré un module d'intelligence artificielle dans un logiciel de développement nous a permis d'acquérir de nouvelles compétences dans le domaine et approfondir davantage nos connaissances suite à l'intersection faite entre ces deux mondes ;
- Ensuite, la méthode agile nous a appris à créer des fonctionnalités à grande valeur ajoutée avec une approche itérative et incrémentale et ceux en devançant nettement la livraison de valeurs prioritaires qui génèrent des résultats plus rapidement et maîtriser aisément les risques et les changements en cours de finalisation ;
- Sur le plan personnel, cette expérience était très bénéfique, enrichissante et instructive pour nous. Elle nous a permis de découvrir le monde professionnel de plus près et de

mettre en œuvre nos connaissances préalablement acquises par notre formation d'ingénieur en informatique. Plusieurs compétences ont été acquises durant ce stage comme le travail en groupe, l'esprit d'équipe, la gestion de projets, la bonne communication, le bon écoute, ...etc.

V-2 Perspectives

Le critère principal de notre réussite est la satisfaction de l'entreprise qui a proposé le sujet, néanmoins, il serait judicieux de contribuer à son amélioration continue en ouvrant les horizons vers de nouvelles méthodes et techniques. De ce fait, le système proposé reste flexible et ouvre sur des perspectives, notamment :

- Le développement de SDK encapsulant le module développé ;
- Continuer à enrichir la base de données d'images annotées ;
- Etendre le système pour qu'il détecte le texte écrit sans qu'il soit nécessairement mis dans des cases prédisposées à l'accueillir ;
- Développer le système pour reconnaître les caractères d'autres langues comme l'arabe.

Et bien plus de besoins qui peuvent être détectés en déployant la solution chez plus de clients et de continuer la démarche d'accompagnement des clients qui a permis à la base de détecter le besoin étudié dans le présent projet.

Références bibliographiques

- [1] Gartner, «Definition of Console,» [En ligne]. Available: <https://www.gartner.com/en/information-technology/glossary/console>.

- [2] D. & P. G. Impedovo, «Zoning methods for handwritten character recognition: A survey,» *Pattern Recognition*, pp. 969-981, 2014.

- [3] M. & T. S. P. Patel, «Handwritten character recognition in english: a survey,» *International Journal of Advanced Research in Computer and Communication Engineering*, pp. 345-350, 2015.

- [4] D. Kuhlman, *A python book: Beginning python, advanced python, and python exercises*, Lutz, 2009.

- [5] N. & F. K. Shukla, *Machine learning with TensorFlow*, Greenwich: Manning, 2018.

- [6] B. S. H. & D. R. G. Stroustrup, «A brief introduction to C++'s model for type-and resource-safety,» 2015.

- [7] R. Chauhan, K. K. Ghanshala et R. C. Joshi, «Convolutional neural network (CNN) for image detection and recognition.,» *First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 278-282, 2018.

- [8] A. C. D. C. M. J. G. L. M. & S. J. Giusti, «Fast image scanning with deep max-pooling convolutional neural networks.,» chez *IEEE International Conference on Image Processing*, 2013.
- [9] S. & K. N. Park, « Analysis on the dropout effect in convolutional neural networks,» chez *Asian conference on computer vision*, Cham., 2016.
- [10] S. & K. R. Tripathi, «Image classification using small convolutional neural network,» chez *9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2019.
- [11] A. F. Agarap, « Deep learning using rectified linear units (relu).,» arXiv preprint arXiv:1803.08375., 2018.
- [12] Y. L. J. K. Y. & F. Y. Wu, «Deep convolutional neural network with independent softmax for large scale face recognition.,» chez *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

Liste Des Figures

| | |
|---|----|
| Figure II.1 Logo de Naltis | 8 |
| Figure III.1 Liste de présence (avec dates de naissance)..... | 19 |
| Figure III.2 Liste de présence (sans dates de naissance) | 20 |
| Figure III.3 Diagramme des cas d'utilisation de l'administrateur de l'application démonstrative | 25 |
| Figure III.4 Diagramme de séquence de l'administrateur de l'application démonstrative | 25 |
| Figure IV.1 Logo de python | 28 |
| Figure IV.2 Logo de TensorFlow | 29 |
| Figure IV.3 Logo de C++ | 30 |
| Figure IV.4 Logo de QT | 30 |
| Figure IV.5 Logo de git | 31 |
| Figure IV.6 Logo de GitHub | 31 |
| Figure IV.7 Logo de Microsoft Office | 32 |
| Figure IV.8 Logo de Diagrams.net | 32 |
| Figure IV.9 Structure des fichiers (exemple) | 33 |
| Figure IV.10 Liste des invités (exemple) | 34 |
| Figure IV.11 Scan 1 (exemple) | 35 |
| Figure IV.12 Scan 2 (exemple) | 36 |
| Figure IV.13 Résultat appel console (exemple)..... | 37 |

| | |
|--|----|
| Figure IV.14 Liste des invités - avant séance (exemple)..... | 38 |
| Figure IV.15 Liste des invités - après séance (exemple)..... | 38 |

Liste Des Tableaux

Tableau II.1 coût temporel de la saisie manuelle des absents 12

Tableau III.1 comparaison des approches de la solution..... 17

Liste Des Sigles Et Abréviations

| | |
|-----|-----------------------------------|
| API | Application Programming Interface |
| CSV | Comma Separated Values |
| HTR | Handwritten Text Recognition |
| OCR | Optical Character Recognition |
| QCM | Questions à Choix Multiples |
| PFE | Projet de Fin d'Etudes |
| SDK | Software Development Kit |
| TD | Travaux dirigés |
| TP | Travaux pratiques |

Résumé

Naltis est une entreprise de services et conseil informatique. Ayant plusieurs solutions sur le marché depuis des années et avec plein de clients permanents, l'entreprise travaille en continu à améliorer ses services et vérifier que tous les besoins et désirs de ses clients sont satisfaits. Pour cela, Naltis collabore avec ses clients pour détecter tout éventuel point d'amélioration.

Le présent projet de fin d'études traite un des besoins qui a été reconnu par plusieurs clients de différentes solutions de Naltis. Ce problème est la lecture automatique des feuilles de présence manuscrites. Utile pour les logiciels de gestion d'examen, de gestion de scolarité et potentiellement pour d'autres, Naltis a décidé de dédier tout un projet à résoudre ce problème pour tous ses clients actuels et futurs.

Notre solution est donc un module qui s'intègre dans quasiment n'importe quelle solution qui traite d'une manière ou d'une autre une liste de présence manuscrite. En plus d'une application de bureau qui démontre de manière ergonomique le cas d'utilisation typique de ce module tout en offrant aux futurs développeurs qui vont maintenir notre solution un environnement pour tester et démontrer leurs avancements. La valeur ajoutée de notre projet s'inscrit donc dans le cadre global du suivi qu'offre Naltis à ses clients.

Mots clés : Reconnaissance d'écriture, découpage automatique, feuille de présence, gestion d'examen

Abstract

Naltis is an IT services and consulting company. Having several solutions on the market for years and with many permanent customers, the company works continuously to improve its services and ensure that all the needs and desires of its customers are met. For this purpose, Naltis collaborates with its customers to detect any possible opportunity for improvement.

This graduation project addresses one of the needs that has been recognized by several customers of different Naltis solutions. This problem is the automatic recognition of handwritten attendance sheets. Useful for exam management software, tuition management software and potentially for others, Naltis decided to dedicate a whole project to solve this problem for all its current and future customers.

Our solution is therefore a module that fits into almost any solution that processes in one way or another a handwritten attendance list. In addition to a desktop application that ergonomically demonstrates the typical use case of this module while providing future developers who will maintain our solution an environment to test and demonstrate their progress. The added value of our project is therefore part of the overall accompaniment that Naltis offers to its customers.

Keywords: Handwriting recognition, automatic cutting, attendance sheet, exam management

ملخص

نالتيس (Naltis) هي شركة استشارات وخدمات تكنولوجيا المعلومات تملك العديد من الحلول في السوق لسنوات ومع العديد من العملاء الدائمين، تعمل الشركة باستمرار لتحسين خدماتها والتأكد من تلبية جميع احتياجات ورغبات عملائها. لهذا الغرض، تتعاون Naltis مع عملائها لاكتشاف أي فرصة ممكنة للتحسين. يلبي مشروع التخرج هذا أحد الاحتياجات التي أقرها العديد من العملاء لحلول Naltis المختلفة. هذه المشكلة هي التعرف التلقائي على أوراق الحضور المكتوبة بخط اليد، حل هذه المشكلة يكون مفيداً لبرامج إدارة الامتحانات وبرامج إدارة الرسوم الدراسية وربما للآخرين، لهذا قررت Naltis تكريس مشروع كامل لحل هذه المشكلة لجميع عملائها الحاليين والمستقبليين.

وبالتالي، فإن الحل الذي نقدمه هو وحدة تتناسب مع أي حل تقريباً يعالج بطريقة أو بأخرى قائمة الحضور المكتوبة بخط اليد. بالإضافة إلى تطبيق سطح المكتب الذي يوضح بشكل مريح حالة الاستخدام النموذجية لهذه الوحدة مع تزويد المطورين المستقبليين الذين سيحافظون على حلنا ببيئة اختبار تقدمهم وإثبات تقدمهم. وبالتالي فإن القيمة المضافة لمشروعنا هي جزء من المرافقة الشاملة التي تقدمها Naltis لعملائها.

الكلمات المفتاحية: التعرف على خط اليد، القطع الآلي، ورقة الحضور، إدارة الامتحانات