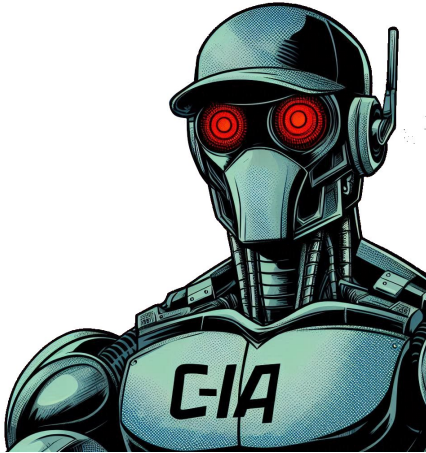


OCR Word Search Solver

Alexandre Divol
Théo Francois-Pichard
Andrew Youansamouth
Leopold Schneckenger

Novembre 2024



Contents

1	Introduction	3
2	présentation du groupe	4
3	Solveur de grille	6
4	Réseau de Neurones	11
4.1	Réalisation du PoC	12
5	Processus d'Entraînement	13
6	Interface GUI	16
7	Découpage des images	21
8	Problèmes rencontrés au cours du projet	25
9	Ressenti du groupe	28
10	Conclusion	30

1 Introduction

Bienvenue à la lecture de ce rapport sur le projet "OCR Word Search Solver". Ce document résume les efforts intensifs de notre équipe durant les deux derniers mois. À travers des heures d'acharnement et de recherche sur divers wikis et articles, nous avons cherché à comprendre des éléments techniques qui, aujourd'hui, nous paraissent plus simples mais qui ont été de véritables défis au départ. Ce rapport est destiné à détailler pas à pas notre développement et les solutions que nous avons apportées à chaque étape.

Notre projet a commencé par la création d'un solveur, un programme annexe crucial pour la résolution des grilles de mots. Ce solveur constitue la base de notre application, et sa conception a nécessité une compréhension approfondie des algorithmes de recherche et d'optimisation. Ensuite, nous nous sommes concentrés sur le réseau de neurones, un élément central de notre projet. Pendant un mois, nous avons exploré différentes architectures de réseaux, ajusté des paramètres et testé diverses approches pour obtenir les meilleurs résultats possibles en termes de reconnaissance de caractères.

En parallèle, deux membres de notre équipe ont travaillé sur un autre aspect fondamental : l'algorithme permettant de reconnaître une grille dans une image et de la découper correctement. Ce processus de découpage a été critique pour la précision du solveur, car il fallait s'assurer que chaque lettre soit correctement isolée et identifiée.

Enfin, nous avons développé le GUI (Graphical User Interface), une interface utilisateur ergonomique et intuitive qui simplifie l'utilisation du programme. Grâce à cette interface, même les utilisateurs non techniques peuvent facilement interagir avec notre application et résoudre des grilles de mots cachés en un clic.

Ce projet nous a également confrontés à de nombreux défis imprévus. De la gestion des erreurs dans le traitement d'image aux ajustements de l'algorithme de reconnaissance, chaque difficulté rencontrée nous a permis d'apprendre et de nous améliorer. Ce rapport vous offre un aperçu détaillé de ces défis et de la manière dont nous les avons surmontés.

Nous espérons que ce rapport reflète fidèlement le travail et la passion investis dans ce projet et qu'il servira de guide et d'inspiration pour de futurs développements dans le domaine de l'OCR et de la reconnaissance de grilles de mots.

2 présentation du groupe

Andrew (Chef de projet)

Bonjour à tous, je m'appelle Andrew et je suis actuellement étudiant à EPITA, où je me spécialise dans le développement logiciel et la gestion de projets informatiques. Mon intérêt pour l'informatique a commencé dès mon enfance, lorsque j'ai découvert la programmation à travers des jeux vidéo. J'étais fasciné par la manière dont des lignes de code pouvaient donner vie à des mondes virtuels et résoudre des problèmes complexes. Au fil des années, cette passion s'est transformée en un désir d'apprendre et de maîtriser les différentes facettes de l'informatique.

En tant que chef de projet sur notre initiative "OCR Word Search Solver", j'ai eu l'opportunité de coordonner une équipe talentueuse et de m'assurer que chacun puisse contribuer de manière significative. Mon rôle m'a permis de développer mes compétences en leadership, en communication et en gestion du temps. J'ai appris à établir des objectifs clairs, à répartir les tâches en fonction des forces de chaque membre et à maintenir un environnement de travail collaboratif et motivant.

Ce projet a été une expérience enrichissante, car il m'a permis de plonger dans des domaines variés tels que le traitement d'images, l'intelligence artificielle et l'optimisation d'algorithmes. J'ai particulièrement apprécié le défi de transformer une idée abstraite en une application fonctionnelle, tout en surmontant les obstacles techniques qui se sont présentés. Je suis convaincu que cette expérience me sera précieuse dans ma future carrière, où je souhaite continuer à innover et à relever des défis techniques.

En dehors de mes études, je suis également passionné par le développement personnel et la technologie. J'aime lire des livres sur l'entrepreneuriat et la gestion, et je m'intéresse aux nouvelles tendances technologiques. Je suis impatient de continuer à apprendre et à grandir dans le domaine de l'informatique, tout en contribuant à des projets qui ont un impact positif sur la société.

Léopold (Responsable de l'interface)

Salut à tous, je suis Léopold, étudiant à EPITA, et je suis passionné par le développement d'interfaces utilisateur et l'expérience utilisateur (UX). Mon parcours dans l'informatique a débuté avec une fascination pour la manière dont les utilisateurs interagissent avec la technologie. J'ai toujours cru que la technologie devrait être accessible et intuitive, et c'est ce qui m'a poussé à me spécialiser dans la conception d'interfaces.

Dans le cadre de notre projet "OCR Word Search Solver", j'ai eu la chance de travailler sur la création d'une interface graphique conviviale et ergonomique.

Mon rôle consistait à concevoir une interface qui non seulement soit esthétiquement plaisante, mais qui facilite également l'interaction des utilisateurs avec notre application. J'ai utilisé des outils comme GTK et Glade pour développer l'interface, ce qui m'a permis d'appliquer mes connaissances en design tout en apprenant de nouvelles compétences techniques.

L'un des aspects les plus gratifiants de ce projet a été de voir comment une interface bien conçue peut améliorer l'expérience utilisateur. J'ai pris en compte les retours d'expérience des utilisateurs pour affiner l'interface, en ajoutant des fonctionnalités et en simplifiant les processus. Cela m'a permis de comprendre l'importance de l'itération dans le design et de l'écoute des besoins des utilisateurs.

En dehors de mes études, je suis également passionné par le design graphique et la création de contenu numérique. J'aime explorer de nouvelles tendances en matière de design et expérimenter avec des outils de création. Je suis convaincu que l'informatique et le design vont de pair, et je suis impatient de continuer à développer mes compétences dans ces domaines. Mon objectif est de créer des solutions technologiques qui soient non seulement fonctionnelles, mais aussi agréables à utiliser.

Alexandre (Responsable des algorithmes)

Bonjour, je suis Alexandre, étudiant à EPITA, et ma passion pour l'informatique se concentre sur les algorithmes et le traitement des données. Mon intérêt pour ce domaine a été éveillé lorsque j'ai découvert la puissance des algorithmes pour résoudre des problèmes complexes et optimiser des processus. J'ai toujours été fasciné par la manière dont une bonne approche algorithmique peut transformer des données brutes en informations exploitables.

Dans le cadre de notre projet "OCR Word Search Solver", j'ai eu la responsabilité de concevoir et d'implémenter des algorithmes pour résoudre des grilles de mots. Ce défi m'a permis d'explorer des concepts avancés en recherche algorithmique, en optimisation et en traitement d'images. J'ai particulièrement apprécié le processus de débogage et d'amélioration continue, car chaque obstacle surmonté est une occasion d'apprendre et de grandir.

L'un des aspects les plus intéressants de mon rôle a été de travailler sur l'optimisation des performances de notre solveur. J'ai dû analyser et affiner les algorithmes pour garantir qu'ils fonctionnent efficacement, même avec des grilles de grande taille. Cela m'a permis de renforcer mes compétences en analyse de complexité et en gestion des ressources.

En dehors de mes études, je suis passionné par la programmation compétitive et les défis de codage. J'aime participer à des hackathons et à des compétitions de programmation, car cela me permet de tester mes compétences et d'apprendre

de nouvelles techniques. Je suis également un fervent défenseur de l'apprentissage continu et je m'efforce de rester à jour avec les dernières tendances en matière de technologie et d'algorithmique. Mon objectif est de continuer à développer mes compétences et de contribuer à des projets innovants qui ont un impact positif sur la société.

Théo (Responsable de l'intelligence artificielle)

Salut à tous, je suis Théo, étudiant à EPITA, et je suis passionné par l'intelligence artificielle et l'apprentissage automatique. Mon intérêt pour l'informatique a été éveillé par la découverte des possibilités offertes par l'IA pour résoudre des problèmes complexes et améliorer notre quotidien. J'ai toujours été intrigué par la manière dont les machines peuvent apprendre et s'adapter à leur environnement, et c'est ce qui m'a poussé à me spécialiser dans ce domaine.

Dans le cadre de notre projet "OCR Word Search Solver", j'ai eu la responsabilité de développer le réseau de neurones pour la reconnaissance de caractères. Ce travail m'a permis d'explorer des concepts avancés en machine learning, tels que la rétropropagation, l'optimisation des hyperparamètres et la gestion des ensembles de données. J'ai particulièrement apprécié le défi de concevoir un modèle capable de reconnaître des lettres dans des images, tout en surmontant les obstacles liés à la qualité des données d'entrée.

L'un des moments les plus gratifiants de ce projet a été de voir notre modèle s'améliorer au fil des itérations d'entraînement. Chaque ajustement que nous avons apporté a eu un impact direct sur la précision de la reconnaissance, et cela m'a permis de comprendre l'importance de l'expérimentation et de l'itération dans le développement de modèles d'IA.

En dehors de mes études, je suis également passionné par la recherche en intelligence artificielle et j'aime suivre les dernières avancées dans ce domaine. Je participe régulièrement à des conférences et des ateliers pour approfondir mes connaissances et échanger des idées avec d'autres passionnés. Mon objectif est de continuer à apprendre et à contribuer à des projets qui utilisent l'IA pour résoudre des problèmes réels, tout en explorant les implications éthiques et sociales de cette technologie.

3 Solveur de grille

Un programme en apparence simple, dont l'objectif est de retrouver des mots dans une grille de lettres. Cependant, un petit bémol : la grille est fournie sous forme de tableau. Il nous incombe donc d'identifier les mots à partir de ce tableau. Pour ce faire, nous nous sommes basés sur le pseudocode de l'article Wikipedia consacré au solving de mots. Cette méthode requiert de tester toutes

les possibilités, c'est-à-dire d'examiner chaque cas de figure : lorsqu'une première lettre est identifiée, il faut alors rechercher dans les huit directions possibles (haut, bas, gauche, droite et les quatre diagonales) pour déterminer si le mot continue ou non.

```
[alexdieu@hirohito-laptop Solver]$ ./solver
solver: Usage: ./solver <Text File> <Word to find> <Debug/Optional>
```

Figure 1: Fonctionnement du programme

L'élaboration de ce programme a commencé par une analyse approfondie du pseudocode. Nous avons dû adapter cet algorithme théorique à notre propre contexte, en tenant compte des spécificités de notre grille et des contraintes imposées par notre application. La phase de debugging a été cruciale pour garantir le bon fonctionnement du solveur. En mode debugging, notre programme permet de suivre pas à pas le processus de recherche des mots, en mettant en évidence les lettres testées et les directions explorées. Voici une démonstration du programme en mode débogage :

```
[alexdieu@hirohito-laptop Solver]$ ./solver grid horizontal 1
Welcome to solver.c 1.5.2 made by Theo and Alex [DEBUG MODE]!
File to look : grid
Word to find : HORIZONTAL
Launching solver ...
Opening the file grid
Grid is a 9 x 10
Transforming the file in an array
Array succesfully created
Here is the array:
[[ H O R I Z O N T A L ]
 [ D X R A H C L B G A ]
 [ D I K C I L E O K C ]
 [ I G A J H V L Y H I ]
 [ H G F G O D T I O T ]
 [ G D L R O W K B F R ]
 [ P L N R D N E R G E ]
 [ J H A I D U A J G V ]
 [ U K G F F O L L E H ] ]
Possible horizontal match on (0,0)
Possible horizontal match on (1,0)
Possible horizontal match on (2,0)
Possible horizontal match on (3,0)
Possible horizontal match on (4,0)
Possible horizontal match on (5,0)
Possible horizontal match on (6,0)
Possible horizontal match on (7,0)
Possible horizontal match on (8,0)
Possible horizontal match on (9,0)
It's a horizontal match ! on (0,0)(9,0)
Closing the file grid
Closed the file grid
(0,0)(9,0)
[alexdieu@hirohito-laptop Solver]$
```

Figure 2: Démonstration du programme en mode debugging

On peut voir ici que le debugging nous indique toutes les étapes effectuées pour trouver le mot. Sur un exemple plus complexe :

```

[alexdiou@hirohito-laptop Solver]$ ./solver grid world 1
Welcome to solver.c 1.5.2 made by Theo and Alex [DEBUG MODE]!
File to look : grid
Word to find : WORLD
Launching solver ...
Opening the file grid
Grid is a 9 x 10
Transforming the file in an array
Array succesfully created
Here is the array:
[[ H O R I Z O N T A L ]
 [ D X R A H C L B G A ]
 [ D I K C I L E O K C ]
 [ I G A J H Y L Y H I ]
 [ H G F G O D T I O T ]
 [ G D L R O W K B F R ]
 [ P L N R D N E R G E ]
 [ J H A I D U A J G V ]
 [ U K G F F O L L E H ] ]
Possible horizontal match on (5,5)
Possible bottom-left to top-right diagonal match on (5,5)
Possible bottom-right to top-left diagonal match on (5,5)
Possible bottom-right to top-left diagonal match on (4,4)
Possible backwards horizontal match on (5,5)
Possible backwards horizontal match on (4,5)
Possible backwards horizontal match on (3,5)
Possible backwards horizontal match on (2,5)
Possible backwards horizontal match on (1,5)
It's a backwards horizontal match! on (5,5)(1,5)
Closing the file grid
Closed the file grid
(5,5)(1,5)
[alexdiou@hirohito-laptop Solver]$

```

Figure 3: Démonstration du programme en mode debugging sur un exemple plus complexe

Une fois le debugging achevé et le programme ajusté, nous avons obtenu une version fonctionnelle et optimisée du solveur. La capture d'écran ci-dessous illustre le programme dans son état final, capable de retrouver avec précision les mots cachés dans la grille :

Pour transformer notre fichier texte en tableau, rien de plus simple : nous avons créé un tableau en itérant sur chaque caractère du fichier texte jusqu'au retour à la ligne, où nous passons à la ligne suivante du tableau. Il est évidemment crucial de vérifier que l'entrée est correcte, en s'assurant que chaque ligne contient le même nombre de caractères.

Ce projet, bien que simple dans son concept, a révélé de nombreux défis et a exigé une compréhension approfondie des algorithmes de recherche et d'optimisation. Il a également mis en lumière l'importance d'une phase de debugging rigoureuse


```

[alexdiu@hirohito-laptop Solver]$ cat grid
HORIZONTAL
DXRAHCLBGA
DIKCILEOKC
IGAJHYLYHI
HGFGODTIOT
GDLROWKBFR
PLNRDNERGE
JHAIDUAJGV
UKGFFOLLEH
[alexdiu@hirohito-laptop Solver]$ ./solver grid horizontal
(0,0)(0,9)
[alexdiu@hirohito-laptop Solver]$ ./solver grid vertical
(9,7)(9,0)
[alexdiu@hirohito-laptop Solver]$ ./solver grid diagonal
(0,1)(7,8)
[alexdiu@hirohito-laptop Solver]$ ./solver grid find
(4,8)(1,5)
[alexdiu@hirohito-laptop Solver]$ ./solver grid hello
(9,8)(5,8)
[alexdiu@hirohito-laptop Solver]$ ./solver grid world
(5,5)(1,5)
[alexdiu@hirohito-laptop Solver]$ ./solver grid goldorak
(8,1)(1,8)
[alexdiu@hirohito-laptop Solver]$ ./solver grid epita
Not found
[alexdiu@hirohito-laptop Solver]$ █

```

Figure 4: Programme fonctionnant correctement

pour assurer la fiabilité et l'efficacité du programme. La satisfaction de voir notre solveur fonctionner correctement après plusieurs heures de travail acharné est une récompense en soi, démontrant la valeur de l'effort et de la persévérance dans le développement de solutions algorithmiques.

Techniquement, le solveur utilise une approche basée sur la recherche en profondeur (Depth-First Search, DFS) pour explorer toutes les directions possibles à partir de chaque lettre de la grille. En intégrant des heuristiques pour réduire l'espace de recherche, nous avons pu améliorer la rapidité et l'efficacité du processus. De plus, nous avons mis en place un système de gestion des mots à rechercher, qui permet de filtrer les résultats en fonction de critères spécifiques, garantissant ainsi une précision optimale.

Cependant, le développement du solveur n'a pas été sans défis. L'un des principaux obstacles rencontrés a été la gestion des cas de bord, tels que les grilles de tailles variées et les configurations de lettres complexes. Nous avons dû ajuster nos algorithmes pour garantir qu'ils puissent traiter efficacement ces situations sans compromettre la performance. De plus, l'optimisation des temps de réponse a nécessité plusieurs itérations et tests, ce qui a parfois été frustrant,

mais a également conduit à des solutions innovantes.

Un autre défi majeur a été l'intégration du réseau de neurones pour la reconnaissance des caractères. Nous avons dû nous assurer que le solveur pouvait fonctionner de manière fluide avec le module de reconnaissance, ce qui a nécessité des ajustements techniques et des tests approfondis pour garantir une communication efficace entre les deux composants.

Malgré ces difficultés, l'équipe a su faire preuve de résilience et de créativité, ce qui a permis de surmonter les obstacles rencontrés. Grâce à une collaboration étroite et à des échanges d'idées constructifs, nous avons réussi à développer un solveur de grille performant et fiable, qui répond aux attentes de notre projet. Nous sommes fiers du résultat final et convaincus que notre travail contribuera à l'avancement des applications dans le domaine de l'OCR et de la reconnaissance de grilles de mots.

4 Réseau de Neurones

Nous entrons dans la partie la plus délicate du projet. Non pas parce qu'elle est particulièrement difficile en termes de codage, mais parce qu'elle est la plus complexe à comprendre. Simplifions tout cela. Saviez-vous que votre estomac abrite un réseau de neurones semblable à celui du cerveau d'un chien? Eh bien, oui! Lorsque votre estomac gargouille, ce n'est pas votre tête qui envoie l'ordre, mais bien un réseau de neurones dans votre système digestif. Certes, il ne contient "que" plusieurs centaines de millions de neurones, loin des milliards de notre cerveau, mais cela suffit amplement à réguler son fonctionnement.

Dans ce projet, nous allons imiter, de manière simplifiée, cette machine incroyablement complexe créée par la nature. Nous allons créer un réseau de neurones artificiel, lui fournir des entrées et lui permettre de générer plusieurs réponses possibles. Pour notre preuve de concept (POC), il devra être capable de comprendre des opérations logiques de base. Ensuite, nous développerons un réseau capable d'identifier les lettres.

Techniquement, le réseau a été entraîné sur un ensemble de données comprenant des images de lettres et de chiffres, ce qui lui a permis d'apprendre à reconnaître les différents caractères dans des conditions variées. Nous avons utilisé des techniques de prétraitement des images, telles que la normalisation et l'augmentation des données, pour améliorer la robustesse du modèle. De plus, nous avons mis en œuvre des couches de convolution et de pooling pour extraire les caractéristiques essentielles, suivies de couches entièrement connectées pour effectuer la classification finale.

Cependant, le développement du réseau de neurones a présenté plusieurs défis. L'un des principaux obstacles a été la nécessité de disposer d'un ensemble de données suffisamment diversifié et représentatif pour garantir une bonne généralisation du modèle. Nous avons dû passer du temps à collecter et à annoter des données, ce qui a été un processus laborieux mais essentiel pour la performance du réseau.

Un autre défi majeur a été l'optimisation des hyperparamètres du modèle. Trouver les bonnes valeurs pour des paramètres tels que le taux d'apprentissage, le nombre de couches et la taille des filtres a nécessité de nombreuses expérimentations et ajustements. De plus, nous avons rencontré des problèmes de surapprentissage, ce qui nous a conduits à intégrer des techniques de régularisation, telles que le dropout, pour améliorer la capacité de généralisation du modèle.

Enfin, l'intégration du réseau de neurones avec le solveur de grille a également nécessité des ajustements techniques. Nous avons dû nous assurer que les sorties du réseau pouvaient être correctement interprétées par le solveur, ce qui a impliqué des tests approfondis pour garantir une communication fluide entre les deux composants.

Malgré ces défis, l'équipe a fait preuve de détermination et de créativité, ce

qui a permis de surmonter les obstacles rencontrés. Grâce à une collaboration étroite et à des échanges d'idées constructifs, nous avons réussi à développer un réseau de neurones performant et fiable, capable de reconnaître avec précision les caractères dans les grilles de lettres. Nous sommes fiers du résultat final et convaincus que notre travail contribuera à l'avancement des applications dans le domaine de l'OCR et de la reconnaissance de grilles de mots.

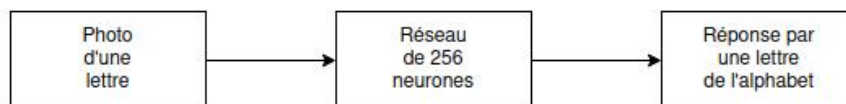


Figure 5: Diagramme représentant le fonctionnement de l'IA

4.1 Réalisation du PoC

Nous vous proposons un tour d'horizon complet de la réalisation d'un Proof of Concept (PoC) en intelligence artificielle. Que vous soyez novice ou expérimenté, vous trouverez ici toutes les étapes nécessaires pour maîtriser ce sujet passionnant.

Comprendre les Réseaux de Neurones

Un réseau de neurones est constitué de différentes couches de neurones, chaque couche ayant un rôle spécifique dans le traitement des données. On distingue principalement trois types de neurones : les neurones d'entrée, les neurones cachés et les neurones de sortie.

Les Neurones d'Entrée

Les neurones d'entrée sont la porte d'entrée des données dans le réseau. Ils sont initialisés avec les valeurs d'entrée que nous fournissons. Par exemple, si notre réseau comporte deux neurones d'entrée et que nos valeurs d'entrée sont 0 et 1, le premier neurone d'entrée sera initialisé à 0 et le second à 1. Ces neurones véhiculent les données brutes vers les couches suivantes du réseau.

Les Neurones Cachés

Les neurones cachés se situent entre les neurones d'entrée et les neurones de sortie. Leur rôle est de traiter les données transmises par les neurones d'entrée. Ils ajustent leurs poids (valeurs influençant leur comportement) au cours de l'apprentissage pour améliorer la précision des réponses du réseau. Ces neurones sont essentiels pour l'apprentissage des patterns complexes présents dans les données.

Les Neurones de Sortie

Les neurones de sortie constituent la dernière couche du réseau. Ils fournissent les résultats finaux. Lors des premiers entraînements, l'IA génère des

réponses aléatoires pour apprendre des feedbacks qu'elle reçoit. Après plusieurs tentatives, le réseau ajuste ses poids, détecte des patterns et améliore ses réponses.

Processus d'Entraînement

L'entraînement du réseau de neurones est une étape cruciale. Nous mesurons la fiabilité du réseau grâce au pourcentage d'erreur à chaque itération du processus d'entraînement. Comme montré dans la figure ci-dessous, nous observons une réduction progressive de l'erreur au fur et à mesure que le réseau apprend.

5 Processus d'Entraînement

```
[alexdieu@hirohito-laptop Neural Network]$ ./ocr_solver
Epoch 1/30
End of Epoch 1 - Average Loss: 0.018351
Epoch 2/30
End of Epoch 2 - Average Loss: 0.018131
Epoch 3/30
End of Epoch 3 - Average Loss: 0.018816
Epoch 4/30
End of Epoch 4 - Average Loss: 0.018131
```

Figure 6: Pourcentage d'erreur à chaque itération du processus d'entraînement.

Facteurs Déterminants la Qualité du Réseau

La qualité finale de notre réseau de neurones dépend de trois facteurs principaux :

La qualité du dataset (ensemble de données) : Un dataset pertinent et bien annoté est essentiel pour un apprentissage efficace. Ce sont nos valeurs permettant d'entraîner notre réseau. Par exemple, pour l'OCR, on retrouve de multiples images de A, B, C, etc...

La taille du réseau : Un réseau plus grand avec plus de neurones et de couches peut offrir de meilleures performances, mais il est aussi plus coûteux en termes de calculs et de ressources.

Le temps d'entraînement : Plus le réseau s'entraîne longtemps, mieux il ajuste ses poids et plus il réduit les erreurs. Un bon compromis doit être trouvé pour éviter le surapprentissage où le modèle devient trop spécifique aux données d'entraînement et perd sa généralité.

Résultats Finaux

Pour notre PoC, nous avons réalisé 10 000 itérations d'entraînement avec un réseau de 5 neurones. Cette configuration est suffisante pour traiter des logiques simples impliquant des 0 et des 1. Voici l'output final de notre PoC :

```
[alexdieu@hirohito-laptop Neural Network (Proof of Concept)]$ ./reseau_neural
Entraînement ET...
Test ET...
Entrée : (0, 0) - Sortie prédite : 0.000125 - Sortie attendue : 0
Entrée : (0, 1) - Sortie prédite : 0.002643 - Sortie attendue : 0
Entrée : (1, 0) - Sortie prédite : 0.002640 - Sortie attendue : 0
Entrée : (1, 1) - Sortie prédite : 0.996209 - Sortie attendue : 1
Entraînement OU...
Test OU...
Entrée : (0, 0) - Sortie prédite : 0.003740 - Sortie attendue : 0
Entrée : (0, 1) - Sortie prédite : 0.997511 - Sortie attendue : 1
Entrée : (1, 0) - Sortie prédite : 0.997464 - Sortie attendue : 1
Entrée : (1, 1) - Sortie prédite : 0.999426 - Sortie attendue : 1
Entraînement NOT_A_AND_NOT_B_OU_A_ET_B...
Test NOT_A_AND_NOT_B_OU_A_ET_B...
Entrée : (0, 0) - Sortie prédite : 0.004624 - Sortie attendue : 0
Entrée : (0, 1) - Sortie prédite : 0.994778 - Sortie attendue : 1
Entrée : (1, 0) - Sortie prédite : 0.994779 - Sortie attendue : 1
Entrée : (1, 1) - Sortie prédite : 0.004849 - Sortie attendue : 0
```

Figure 7: Output du PoC après 10 000 itérations avec un réseau de 5 neurones.

Le réseau de neurones se révèle extrêmement précis dans ses prédictions, approximant avec une grande justesse les valeurs attendues. En effet, pour une entrée correspondant à 0, le réseau prédit une valeur de 0,0002, ce qui est très proche de la valeur correcte. De même, pour une entrée de 1, la prédiction du réseau atteint 0,99, démontrant ainsi sa capacité à s'ajuster et à fournir des réponses presque exactes. Ces résultats illustrent la performance remarquable du réseau dans l'apprentissage et la reconnaissance des patterns, validant son efficacité après l'entraînement.

Après plusieurs jours d'entraînement intensif, il apparaît clairement que l'IA OCR n'a pas encore atteint un niveau de performance optimal. Bien que des progrès aient été réalisés, certains défis restent à surmonter pour atteindre les résultats escomptés.

[illegible]

Figure 8: Test du réseau de neurones après 3 jours d'entraînement

On peut examiner les probabilités des 26 lettres dans les 26 neurones de sortie, et constater que l'IA n'a réussi qu'un seul test. Cela peut s'expliquer par la qualité insuffisante du dataset utilisé pour l'entraînement.

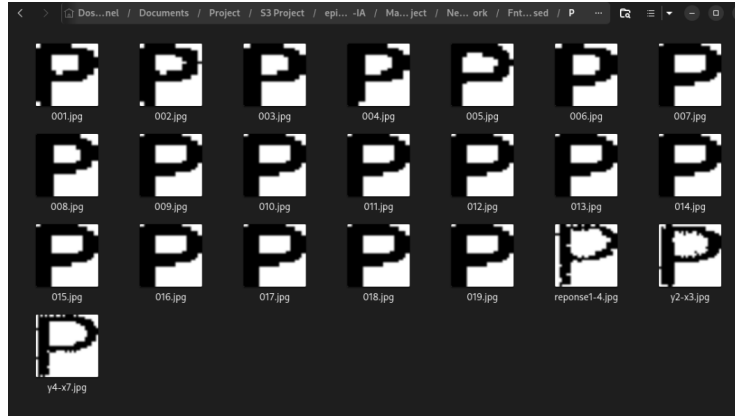


Figure 9: Voici un exemple de dataset pour la lettre P que nous avons utilisé afin d'entraîner le réseau de neurones



Figure 10: Voici un exemple de dataset pour la lettre A que nous avons utilisé afin d'entraîner le réseau de neurones

6 Interface GUI

L'avancement de l'interface graphique (GUI) n'avait pas été notre priorité pour la dernière soutenance. Nous avons concentré nos efforts sur le bon fonctionnement de chaque composant pris individuellement. Nous avons cependant réalisé un prototype ALPHA non-fonctionnel de l'interface, utilisant GTK et SDL, pour illustrer à quoi elle pourrait ressembler.

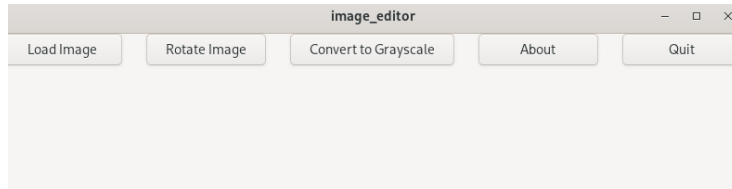


Figure 11: prototype de l'interface pour la première soutenance

L'interface graphique (GUI) de notre application a été conçue pour offrir une expérience utilisateur intuitive et agréable, facilitant l'interaction avec le solveur de grille et le réseau de neurones. Le principe de fonctionnement de l'interface repose sur la création d'un environnement visuel où les utilisateurs peuvent facilement charger des images de grilles de lettres, lancer le processus de reconnaissance et visualiser les résultats de manière claire et concise.

Techniquement, nous avons utilisé des bibliothèques telles que SDL et GTK pour développer l'interface. Ces outils nous ont permis de créer une interface réactive et esthétique, intégrant des éléments tels que des boutons, des menus et des zones d'affichage. L'interface a été conçue pour être modulable, permettant aux utilisateurs de personnaliser certains paramètres, comme le choix du modèle de reconnaissance ou le niveau de détail des résultats affichés.

L'un des principaux défis rencontrés lors du développement de l'interface a été de garantir une navigation fluide et intuitive. Nous avons dû réaliser plusieurs itérations de conception, en tenant compte des retours d'utilisateurs potentiels pour améliorer l'ergonomie et l'accessibilité de l'application. Cela a impliqué des tests utilisateurs pour identifier les points de friction et ajuster la disposition des éléments de l'interface. Un autre défi majeur a été l'intégration des différents composants de l'application, notamment le solveur de grille et le réseau de neurones, avec l'interface. Nous avons dû veiller à ce que les communications entre ces modules soient efficaces et sans faille, ce qui a nécessité des ajustements techniques pour garantir que les résultats de la reconnaissance soient affichés en temps réel et de manière cohérente. De plus, la gestion des erreurs et des exceptions a été un aspect crucial à prendre en compte. Nous avons mis en place des mécanismes pour informer les utilisateurs en cas de problème, comme un format d'image incorrect ou une absence de résultats, afin de garantir une expérience utilisateur positive et sans frustration.

Enfin, l'esthétique de l'interface a également été une priorité. Nous avons travaillé sur la palette de couleurs, la typographie et les éléments graphiques

pour créer une interface visuellement attrayante, tout en veillant à ce qu'elle reste fonctionnelle et facile à utiliser. Malgré ces défis, l'équipe a su faire preuve de créativité et de collaboration, ce qui a permis de surmonter les obstacles rencontrés. Grâce à un travail d'équipe efficace et à des échanges d'idées constructifs, nous avons réussi à développer une interface graphique performante et conviviale, qui répond aux attentes des utilisateurs. Nous sommes fiers du résultat final et convaincus que notre travail contribuera à l'amélioration de l'expérience utilisateur dans le domaine de l'OCR et de la reconnaissance de grilles de mots.

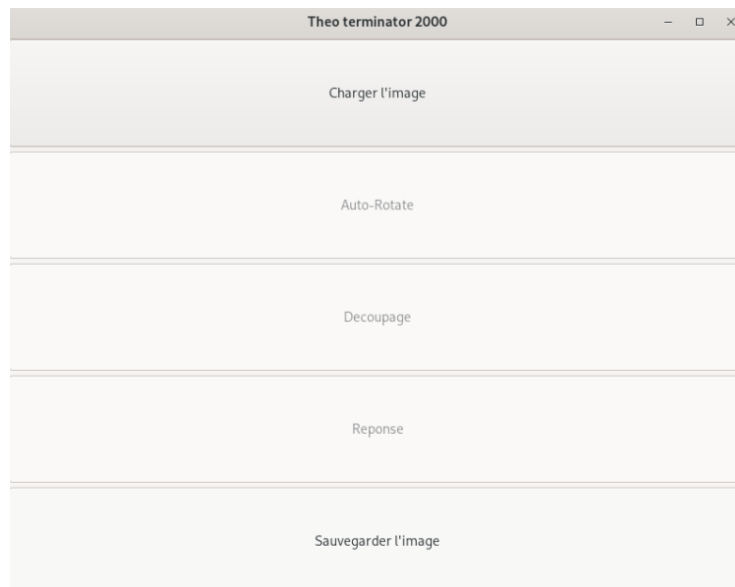


Figure 12: l'interface finie

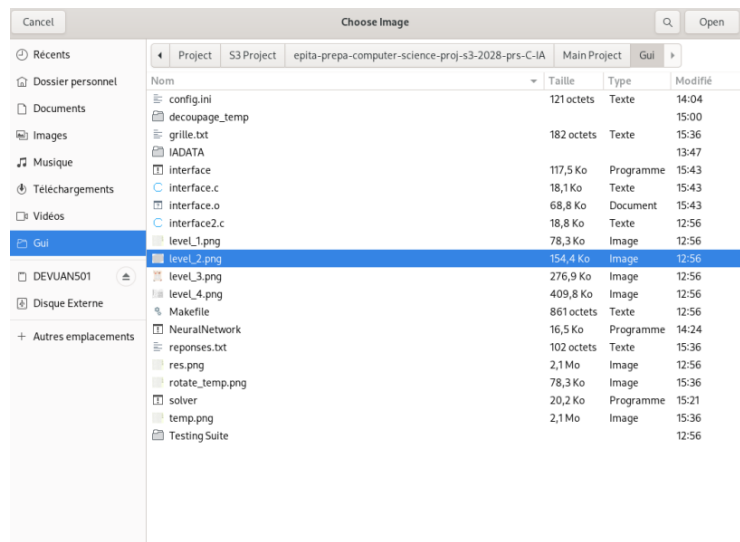


Figure 13: recherche de l'image voulue

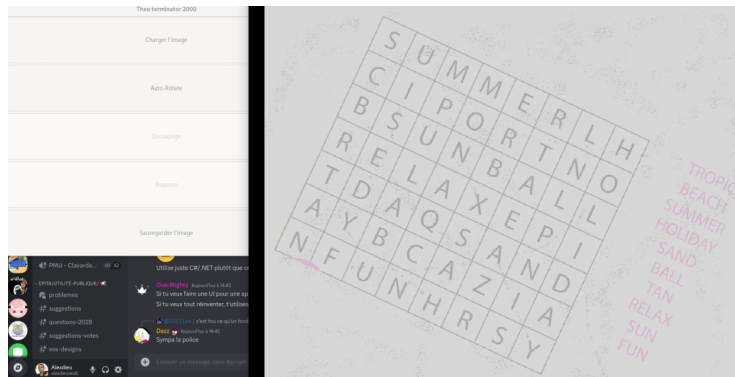


Figure 14: résultat final

Comme on peut le voir l'interface comporte plusieurs options comme le fait de charger une image parmi les fichiers locaux avec le bouton "Charger l'image", le bouton "Auto-Rotate" qui permet de faire rotationner l'image pour qu'elle soit bien droite, ainsi que le bouton "decoupage", le bouton "reponse" et le bouton "Sauvgarder l'image" qui ont leur fonction à l'image de leur nom.

7 Découpage des images

Le découpage des images a représenté la partie la plus complexe et technique du projet. Pour aborder ce défi, nous avons développé un algorithme relativement simple mais ingénieux. Voici les étapes détaillées de notre processus :

Tout d'abord, nous commençons par convertir l'image entière en noir et blanc, en appliquant un seuil de luminosité. Ce seuil est ajustable, ce qui nous permet de moduler les résultats obtenus en fonction de la qualité de l'image initiale. Ensuite, nous procédons à une rotation de l'image pour la redresser et la mettre à l'horizontale, garantissant ainsi une base de travail correcte pour le découpage.

L'algorithme de découpage suit les étapes suivantes : il commence par détecter toutes les masses de points noirs présentes sur l'image. Si ces masses sont trop grandes ou trop petites, elles sont ignorées, car elles ne correspondent pas aux critères d'intérêt. Une fois ces masses pertinentes identifiées, nous procédons à la détection des rectangles formant la grille.

La détection de ces lettres repose sur des principes clés. Nous les identifions grâce à leur espacement régulier et à leurs dimensions similaires en termes de longueur et de largeur. Cette uniformité nous permet de déterminer avec précision quels rectangles appartiennent à la grille. Les éléments qui ne respectent pas ces critères sont alors considérés comme des textes de réponse.

Ce processus, bien qu'il puisse paraître simple, se révèle extrêmement efficace. Il permet une analyse rigoureuse et précise, garantissant que chaque lettre et chaque caractère sont correctement détectés et traités.

Nous vous présentons ici deux exemples des résultats obtenus grâce à notre algorithme. Dans ces exemples, il est évident que toutes les lettres ont été détectées avec une grande précision, démontrant l'efficacité et la fiabilité de notre méthode :



Figure 15: Lettres détectés par l'algorithme

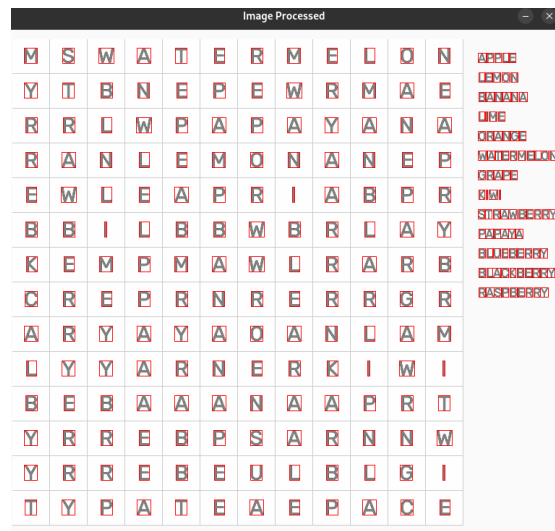


Figure 16: Lettres detectés par l’algorithme

Ensuite, le processus de découpage s'effectue en fonction de la catégorisation des éléments, selon qu'ils appartiennent à une réponse ou qu'ils font partie du tableau :

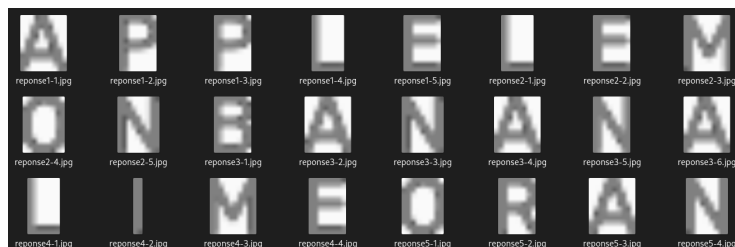


Figure 17: Résultat du découpage

Nous pouvons observer plus en détail les résultats obtenus grâce à la grille reconstruite à l'aide de SDL :

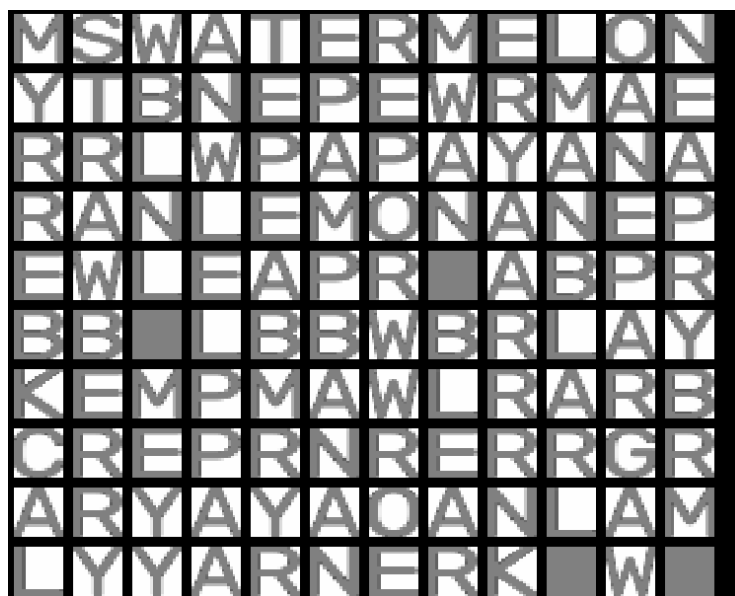


Figure 18: Grille reconstruite avec SDL

Ainsi que les réponses extraites :



Figure 19: Réponses extraites par l'algorithme

Dans notre plan de développer des outils de manipulation d'images, nous avons également développé un programme complémentaire intitulé rotate. Ce programme offre la possibilité de faire pivoter manuellement une image en spécifiant le degré de rotation souhaité.

Le programme rotate utilise les bibliothèques SDL et SDL-image pour charger différents formats d'images, y compris BMP, PNG et JPG. Voici un aperçu de son fonctionnement :

- Chargement de l'image : L'image à manipuler est chargée à partir d'un fichier spécifié par l'utilisateur.
- Calcul de la rotation : Grâce à un algorithme basé sur la trigonométrie, le programme calcule les nouvelles dimensions et les positions des pixels après rotation.
- Enregistrement de l'image : L'image pivotée est ensuite enregistrée sous un nouveau fichier, permettant à l'utilisateur de comparer facilement l'original et la version modifiée.

Le programme rotate est un outil pratique pour tous ceux qui ont besoin de manipuler les images de manière précise et flexible. Avec ce programme, la rotation des images devient simple et efficace.

Ici l'algorithme utilisé :


```
int nx = (x - cx) * cos(radians) - (y - cy) * sin(radians) + ncx;
int ny = (x - cx) * sin(radians) + (y - cy) * cos(radians) + ncy;
```

L'algorithme de rotation d'image repose sur des transformations trigonométriques pour recalculer la position des pixels. Voici une décomposition détaillée du processus:

Centre de l'Image:

- cx, cy : Coordonnées du centre de l'image originale.
- ncx, ncy : Coordonnées du centre de l'image après rotation.

Coordonnées Relatives :

- (xcx)(x - cx) et (ycy)(y - cy) sont les coordonnées du pixel par rapport au centre de l'image. Cela recentre le calcul autour du point pivot.

Imaginez que chaque pixel de l'image est attaché à une ligne qui part du centre de l'image. Lors de la rotation, chaque ligne pivote de l'angle spécifié, déplaçant ainsi le pixel à sa nouvelle position. C'est ce déplacement qui est calculé par les formules de rotation.

Ce processus est appliqué à chaque pixel de l'image pour obtenir l'image finale, pivotée selon l'angle désiré, tout en conservant la structure de l'image originale.

8 Problèmes rencontrés au cours du projet

Au cours de notre projet "OCR Word Search Solver", nous avons été confrontés à plusieurs défis techniques qui ont mis à l'épreuve notre capacité à travailler en équipe et à trouver des solutions créatives. Ces problèmes, bien que parfois frustrants, ont également été des occasions d'apprentissage significatives. Voici un aperçu des principaux problèmes rencontrés :

En outre, nous avons rencontré des problèmes liés à l'Address Sanitizer et aux erreurs de segmentation (segfaults). L'utilisation d'Address Sanitizer nous a permis d'identifier plusieurs failles de mémoire dans notre code, mais ces détections ont parfois retardé notre progression en raison du temps nécessaire pour localiser et corriger les erreurs. Les segfaults, en particulier, ont été une source fréquente de frustration, nécessitant une attention minutieuse pour tracer et éliminer les causes sous-jacentes de ces erreurs.

Crashes de l'Interface Utilisateur : L'une des difficultés majeures a été la stabilité de notre interface utilisateur. Nous avons rencontré plusieurs cas où l'application se fermait de manière inattendue, ce qui a perturbé le flux de travail et a frustré les utilisateurs lors des tests. Ces crashes étaient souvent liés à des erreurs de gestion de mémoire ou à des événements non gérés dans l'interface. Pour résoudre ce problème, nous avons dû revoir notre code, ajouter des vérifications d'erreurs et effectuer des tests approfondis pour identifier les causes sous-jacentes.

Bugs avec la Bibliothèque SDL : L'utilisation de la bibliothèque SDL pour le traitement d'images a également présenté des défis. Nous avons rencontré des bugs qui affectaient le rendu des images et la gestion des événements. Ces problèmes ont nécessité une recherche approfondie dans la documentation de SDL et des forums pour trouver des solutions. En fin de compte, nous avons dû ajuster notre approche et parfois contourner certaines limitations de la bibliothèque pour garantir un fonctionnement fluide.

Temps d'Exécution Trop Long pour le Découpage : Le processus de découpage d'image s'est avéré être un goulot d'étranglement en termes de performance. Les algorithmes que nous avons initialement implémentés prenaient un temps d'exécution excessif, ce qui ralentissait l'ensemble du traitement. Pour remédier à cela, nous avons dû optimiser notre code, revoir nos algorithmes et tester différentes approches pour réduire le temps de traitement. Ces optimisations ont nécessité des itérations et des tests rigoureux, mais ont finalement conduit à des améliorations significatives.

Problèmes de Détection : La détection des contours et des éléments dans les images a été un autre défi. Initialement, nos algorithmes de détection ne fonctionnaient pas comme prévu, ce qui entraînait des résultats erronés ou incomplets. Cependant, après plusieurs ajustements et tests, nous avons réussi à faire fonctionner ces algorithmes de manière satisfaisante. Ce processus a été long et a nécessité une collaboration étroite entre les membres de l'équipe pour identifier les problèmes et tester les solutions.

Dataset Non Adapté : Nous avons également constaté que notre dataset initial n'était pas adapté aux besoins de notre modèle d'apprentissage. Les images étaient souvent de mauvaise qualité ou ne représentaient pas suffisamment la diversité des caractères que nous devons reconnaître. Pour surmonter ce problème, nous avons dû élargir notre dataset en incluant des images supplémentaires et en veillant à ce qu'elles soient de haute qualité. Cela a nécessité un effort considérable, mais a finalement permis d'améliorer les performances de notre modèle.

IA qui N'Apprend Rien : Un des problèmes les plus préoccupants a été que notre modèle d'intelligence artificielle semblait ne pas apprendre correctement. Les résultats des tests étaient décevants, et nous avons dû examiner en profondeur notre approche d'entraînement. Cela a impliqué de réévaluer nos hyperparamètres, d'ajuster la structure du réseau de neurones et de s'assurer que les données d'entraînement étaient correctement étiquetées. Grâce à ces efforts, nous avons finalement réussi à améliorer les performances de notre IA, mais cela a nécessité un investissement de temps et d'énergie considérable.

En conclusion, bien que ces problèmes aient constitué des obstacles, ils ont également été des catalyseurs d'apprentissage et d'amélioration. Chaque défi rencontré a renforcé notre capacité à travailler ensemble, à innover et à trouver

des solutions efficaces. Ces expériences nous ont préparés à aborder des projets futurs avec une meilleure préparation et une plus grande résilience.

9 Ressenti du groupe

Andrew (chef)

En tant que chef de projet, cette expérience a été extrêmement enrichissante et stimulante. La coordination des différentes phases du projet et la gestion de l'équipe ont représenté un défi important, mais cela m'a permis d'améliorer mes compétences en leadership et en organisation. Observer l'évolution du projet et voir notre équipe surmonter les obstacles ensemble a été une source de grande satisfaction. Je suis particulièrement fier de notre capacité à rester soudés et à travailler de manière collaborative pour atteindre nos objectifs. Chaque réunion, chaque brainstorming a renforcé notre cohésion et a permis de créer un environnement où chacun se sentait valorisé et écouté. Cette expérience m'a non seulement appris à gérer un projet, mais aussi à inspirer et motiver une équipe vers un but commun.

Theo (responsable IA)

Travailler sur l'intelligence artificielle a été à la fois passionnant et complexe. Les défis liés à l'entraînement de notre modèle OCR et les ajustements constants nécessaires pour améliorer les résultats ont été très formateurs. Chaque itération a apporté son lot de leçons, et malgré les difficultés rencontrées, je suis fier des progrès réalisés et de notre persévérance. Collaborer avec une équipe dédiée et trouver des solutions ensemble a renforcé ma passion pour l'IA et m'a donné de précieuses leçons pour les projets futurs. J'ai particulièrement apprécié le processus d'expérimentation, où chaque échec était une opportunité d'apprentissage, et je suis convaincu que ces compétences me serviront dans mes futures aventures dans le domaine de l'intelligence artificielle.

Alex (responsable algorithmes)

En tant que responsable des algorithmes, j'ai eu l'opportunité de concevoir et d'implémenter des solutions techniques innovantes pour notre projet. Le développement de l'algorithme de découpage d'image a été particulièrement captivant et m'a permis d'approfondir mes connaissances en traitement d'image. Chaque étape de test et d'ajustement, bien que parfois frustrante, a été incroyablement gratifiante. Voir nos algorithmes fonctionner correctement après de nombreux essais a été une immense satisfaction. J'ai également appris à travailler sous pression et à respecter des délais serrés, tout en maintenant un haut niveau de qualité dans notre code. Cette expérience m'a non seulement permis de développer mes compétences techniques, mais aussi de renforcer ma capacité à collaborer efficacement avec les autres membres de l'équipe.

Leopold (responsable interface)

La conception de l'interface utilisateur a été un défi stimulant qui m'a permis d'explorer différentes approches pour rendre notre application à la fois intu-

itive et esthétiquement agréable. Travailler avec SDL et GTK pour créer un prototype fonctionnel m'a offert de nombreuses occasions d'apprentissage et de développement de compétences. Je suis très satisfait du résultat final et de la manière dont notre équipe a pu collaborer efficacement pour intégrer tous les composants. Ce projet a été une expérience précieuse qui a renforcé mes compétences en développement d'interfaces et en travail d'équipe. J'ai particulièrement apprécié le retour d'expérience des utilisateurs lors des tests, ce qui m'a permis d'affiner notre interface pour qu'elle réponde au mieux aux besoins de nos utilisateurs. Cette expérience m'a également appris l'importance de l'ergonomie et de l'esthétique dans le développement d'applications.

10 Conclusion

En conclusion, notre projet "OCR Word Search Solver" a été une expérience enrichissante et formatrice pour l'ensemble de l'équipe. Chacun des membres a pu apporter ses compétences et sa passion, contribuant ainsi à la réalisation d'un produit innovant et fonctionnel. À travers les différentes phases du projet, nous avons non seulement développé des compétences techniques en intelligence artificielle, en traitement d'images et en développement d'interfaces, mais nous avons également renforcé notre capacité à travailler en équipe, à communiquer efficacement et à surmonter les défis ensemble.

Les obstacles rencontrés tout au long de ce parcours ont été autant d'opportunités d'apprentissage, nous permettant d'affiner nos méthodes de travail et d'améliorer notre produit final. La collaboration entre les différents responsables de chaque domaine a été essentielle pour intégrer harmonieusement les différentes composantes de notre application, garantissant ainsi une expérience utilisateur fluide et intuitive.

Nous sommes fiers des résultats obtenus et des leçons tirées de cette aventure. Ce projet a non seulement renforcé notre passion pour l'informatique et l'innovation, mais il a également ouvert la voie à de futures explorations dans le domaine de l'intelligence artificielle et du traitement d'images. Nous espérons que notre application sera utile à de nombreux utilisateurs et qu'elle pourra être améliorée et étendue dans les années à venir.

Enfin, nous tenons à remercier tous ceux qui ont contribué à ce projet, que ce soit par leur soutien, leurs conseils ou leur collaboration. Cette expérience a été un véritable tremplin pour notre développement personnel et professionnel, et nous sommes impatients de relever de nouveaux défis ensemble.