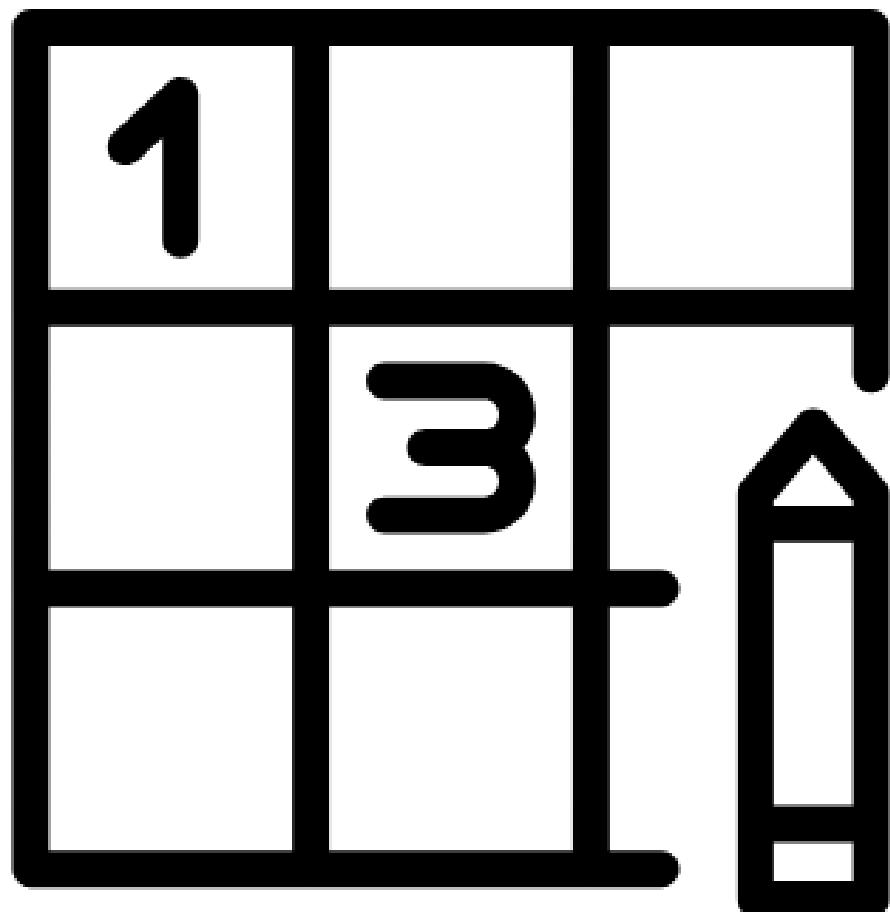


Rapport de projet

Les Hacheurs de C

Projet S3



Contents

1 Présentation OCR x Sudoku	4
1.1 Introduction au projet	4
1.2 Présentation des membres	5
1.2.1 Valentin VEINANTE (C1)	5
1.2.2 Alexandre BODIN (C1)	5
1.2.3 Mark DELALOY (C1)	6
1.2.4 Kevin PETIT (C1)	6
2 Etat du groupe (moral)	7
2.1 Objectifs / Ambitions	7
2.2 Gestion du travail d'équipe	8
3 Budget	9
3.1 Coûts prévus	9
3.2 Modèle économique (bénéfices attendus)	9
3.2.1 Vente aux Sites de Génération de Sudoku :	9
3.2.2 Intégration de la Publicité :	9
3.3 Coûts finaux	10
4 Découpage du projet	11
4.1 Les différentes parties du projet	11
4.2 Répartition des tâches	11
4.3 Planning visé au démarrage du projet	12
5 Développement et difficultés rencontrées	13
5.1 Traitement d'image	13
5.1.1 1ère soutenance (0% → 70%)	13
5.1.2 2ème soutenance (70% → 100%)	19
5.2 Grid Detection	20
5.2.1 1ère soutenance (0% → 90%)	20
5.2.2 2ème soutenance (90% → 100%)	21
5.3 Sudoku Solver	23
5.3.1 1ère soutenance (0% → 100%)	23
5.3.2 2ème soutenance (100% → 120%)	24
5.4 Réseau neuronal	25
5.4.1 1ère soutenance (0% → 35%)	25
5.4.2 2ème soutenance (35% → 100%)	26
5.5 Interface Graphique	28
5.5.1 2ème soutenance (0% → 100%)	28



6 Impressions du groupe	31
6.1 1ère soutenance	31
6.1.1 Valentin	31
6.1.2 Alexandre	32
6.1.3 Mark	33
6.1.4 Kevin	34
6.2 2ème soutenance	35
6.2.1 Valentin	35
6.2.2 Alexandre	36
6.2.3 Mark	37
6.2.4 Kevin	38
7 Conclusion	39
7.1 Bilan 1ère soutenance	39
7.1.1 Ce qu'il nous reste à faire...	39
7.2 Bilan 2ème soutenance	39
7.3 Bilan général	40



1 Présentation OCR x Sudoku

1.1 Introduction au projet

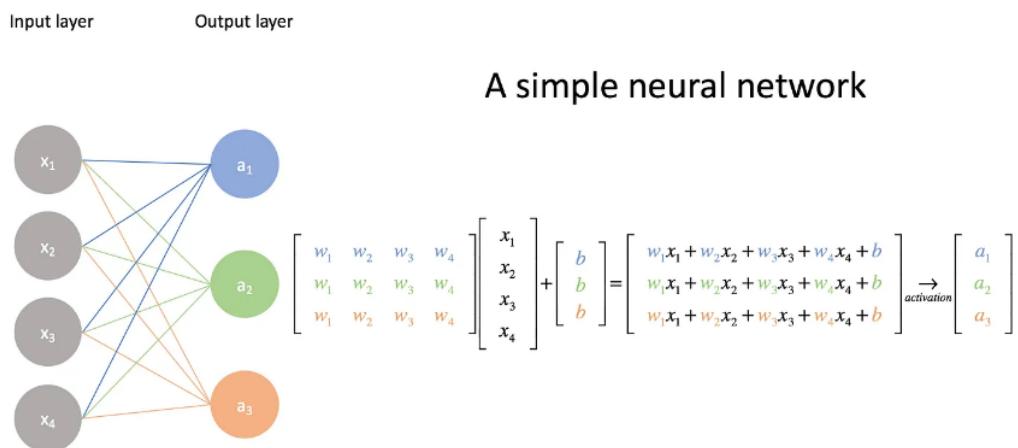
Le but fondamental de ce projet, baptisé "OCR Sudoku Solver", est de créer un logiciel sophistiqué capable de résoudre des grilles de sudoku à partir d'images.

Pour ce faire, le logiciel devra être en mesure de prendre en entrée une image de la grille de sudoku, de reconnaître les chiffres qui y sont présents, de résoudre le puzzle en appliquant des algorithmes spécifiques, puis d'afficher la grille résolue en sortie.

En somme, l'objectif est de développer un programme polyvalent qui peut effectuer toutes les étapes nécessaires pour passer d'une image de sudoku à une grille résolue, en utilisant des techniques d'OCR, de traitement d'image, d'apprentissage automatique (avec un réseau de neurones), et de résolution d'énigmes de sudoku.

Le projet inclut également la création d'une interface graphique conviviale pour faciliter l'interaction avec l'utilisateur. Ce projet nécessitera une collaboration efficace de quatre personnes pour atteindre ces objectifs ambitieux.

La difficulté est d'apprendre à l'ordinateur à identifier/reconnaitre les composants nous intéressants. Il devra les virtualiser pour pouvoir les manipuler afin de proposer une solution au problème posé, tout en gardant un aspect de simplicité pour l'utilisateur.



1.2 Présentation des membres

1.2.1 Valentin VEINANTE (C1)

Anciennement en Terminal Math/Physique-Chimie, je fais de la programmation depuis plusieurs années pour divers projets.

Ce projet S3 m'interresse grandement car ce dernier nous permet d'avoir une introduction à l'intelligence artificielle ainsi qu'au Deep Learning.

Comme le projet S2, je pense qu'évoluer dans un environnement moins restreint que les TPs nous permettra de progresser sur divers aspects tel que l'autonomie, la technique et la cohésion de groupe.

1.2.2 Alexandre BODIN (C1)

Ayant effectué une Terminale Math-NSI, j'ai eu l'occasion de faire beaucoup de programmation depuis un moment. Cette 1ere année à EPITA a été source de nombreuses découvertes sur mes capacités en matière de réalisation de projet, de travail de groupe ou encore de connaissances acquises.

J'ai toujours été curieux de savoir comment une IA fonctionnait et la difficulté d'en réaliser une. J'ai donc beaucoup d'attentes envers ce projet.

Premièrement je m'attend à acquérir des compétences dans des domaines auxquels je ne me serait pas particulièrement intéressé de base.

Mais surtout j'espère qu'il nous permettra de nous dépasser et de finaliser un projet dont nous serons tous fiers.

J'aimerai également qu'il nous permette de nous améliorer non seulement techniquement mais aussi dans l'organisation de groupe.

Je suis assez curieux de voir combien de nos objectifs nous pourrons atteindre et je reste confiant dans l'issue de notre projet. Je nous pense capable de réussir ce défi comme nous avons su le faire jusque là !



1.2.3 Mark DELALOY (C1)

Anciennement en Terminal Spécialités Mathématiques/Physique-Chimie, option Maths Expert, section anglophone, au lycée St Vincent de Senlis (60300), je suis un grand passionné de volley, de psychologie mais surtout d'informatique.

Ce monde m'est cher car il a bercé ma plus tendre enfance. Mes premiers souvenirs étant de regarder mon père monter/démonter des ordinateurs et d'essayer, tant bien que mal, de l'assister.

De plus, l'univers des jeux vidéos m'avait tellement plu que l'un de mes plus grand rêve était de travailler en tant que Game Designer ou dans le Game Art. Cependant, je me suis rendu compte que la plupart des éditeurs utilisaient notre passion pour nous exploiter et que ces métiers étaient trop spécialisés.

C'est donc après cette désillusion que je me suis tourné vers l'EPITA et le métier d'ingénieur informatique, bien plus polyvalent et libre.

Vous pouvez donc imaginer que c'est avec énormément de motivation que je vais m'impliquer dans ce projet, afin de développer mes connaissances dans un domaine que je ne connais pas encore et qui permettra de développer cette fameuse polyvalence dont il est question.

Je suis persuadé que ce travail en groupe nous permettra de sortir plus grands, nous aidera à développer notre capacité d'adaptation, à faire des compromis, et finalement notre qualité d'écoute.

1.2.4 Kevin PETIT (C1)

Ancien élève de terminale Maths/NSI (avec option Maths Expertes), je suis paradoxalement plus intéressé par ce projet de S3 que par le projet de S2. Malgré son aspect restreint pouvant être repoussant au premier abord, avec un cahier des charges déjà fixé, je trouve que ce cadre nous permet de nous concentrer intégralement sur la résolution du problème posé. L'an dernier se posait parfois la question du cadre (que nous nous imposions à nous même), selon nos objectifs de développement (pouvant grandement différer d'un groupe à l'autre en fonction des concepts de jeu), ici le cadre est rigoureusement identique pour tout le monde.

En cela, ce projet est peut-être aussi plus proche du contexte de travail auquel nous serons confrontés plus tard en entreprise : un cahier des charges a été établi et nous devons nous assurer d'y répondre le mieux possible avec le programme que nous développons.



2 Etat du groupe (moral)

2.1 Objectifs / Ambitions

Le premier objectif du projet pour tous les membres du groupe est d'apprendre plus sur la création d'un OCR. Etant des personnes de nature extrêmement curieuse, nous voyons ce projet de S3 comme étant une réelle opportunité de comprendre le fonctionnement de cet outil que l'on peut déjà avoir été amené à manipuler par le passé.

Le second objectif, plus scolaire, consiste à consolider nos bases en programmation. Avant de commencer notre projet, nous avons déjà la certitude que le temps passé sur ce dernier sera bénéfique à notre apprentissage et permettra de consolider considérablement nos bases.

De plus, pour la majorité d'entre nous, les connaissances/notions utiles pour la création de l'OCR, le Sudoku Solver et le Traitement d'Image étaient inexistantes, obligeant un réel travail de recherche en amont.

Pour la plupart d'entre nous, ce projet représente une seconde opportunité exigeant autant de travail sur une si courte période. Expérimenter de nouveau le travail en équipe, contrainte à laquelle nous serons exposés tout au long de notre future carrière d'ingénieur en informatique est nécessaire. Cela sera aussi une opportunité pour tous, d'apprendre à travailler la coordination d'un groupe tout en tenant les échéances imposées par un planning donné en amont.

Nous avons donc de grandes ambitions pour ce projet dans lequel nous nous investissons pleinement.



2.2 Gestion du travail d'équipe

Pour travailler sur chacun des points du projet nous avons formé 2 duos : Mark/Kevin et Alexandre/Valentin.

Chacun des duos a développé un point qu'ils avaient en commun puis à intervalle régulier, les deux équipes se sont retrouvées pour mettre faire un point le travail effectué.

Ainsi chacun a pu se concentrer sur son travail personnel tout en faisant avancer l'ensemble du projet.

La méthode par duo a permis de se motiver mutuellement car la solitude pouvait facilement nous décourager. Cela permettait également d'alléger la charge de travail assez lourde sur les points principaux de développement et de la répartir équitablement en subdivisant alors un travail, énorme en apparence, en de plus petites tâches. Le duo permettait, en plus, un contrôle mutuel.

Cette méthode a permis ainsi de progresser à intervalles réguliers et d'avancer sur l'ensemble des points du programme simultanément et en cohésion avec l'équipe.



3 Budget

3.1 Coûts prévus

Le tableau ci-dessous n'est pas exact, certes, mais permet d'avoir une approximation de la valeur de nos dépenses à prévoir pour le projet. Ce dernier sera complété au fur et à mesure de l'avancement du projet, si d'autres ressources sont jugées nécessaires.

Services	Budget
Transports	$210 \times 4 = 840$
Internet	$140 \times 4 = 560$
Stockage	1 To - SSD 500 Go SSD 512 Go
OS	Linux Windows 11 Windows 10

3.2 Modèle économique (bénéfices attendus)

Ce plan d'affaires se concentre sur la monétisation de l'OCR x Sudoku Solver, en mettant en avant deux principaux axes :

3.2.1 Vente aux Sites de Génération de Sudoku :

- L'OCR Sudoku Solver sera proposé sous forme d'offre premium aux sites web qui proposent des générateurs de sudoku.
- Tarifs compétitifs basés sur le nombre d'utilisateurs potentiels des sites partenaires.
- Personnalisation de l'application pour refléter la marque du site web partenaire.

3.2.2 Intégration de la Publicité :

- Intégration de publicités non intrusives dans l'application pour la version gratuite.
- Sélection de publicités en adéquation avec les intérêts des utilisateurs.
- Optimisation constante de l'emplacement et du type de publicités pour maximiser les revenus.



3.3 Coûts finaux

Pour faire un bilan final de nos bénéfices il nous faut d'abord nous rendre compte de ce que notre projet nous aura coûté en tout.

Services	Budget
Caféine	15€ x 4 = 60€
Électricité + chauffage	80€ x 4 = 320€
Transports	85€ x 2 = 170€
Heures de sommeil	40h x 4 = 160h

Si nous considérons qu'une heure de sommeil revient à 14€ le coût total de notre projet nous revient donc à $14\text{€} * 160 + 60\text{€} + 320\text{€} + 170\text{€} = 2\,790\text{€}$.

En combinant la vente aux sites de génération de sudoku avec l'intégration de publicités, nous cherchons à maximiser les revenus tout en offrant une expérience de qualité aux utilisateurs. Cette approche offre une opportunité solide de monétisation pour l'OCR Sudoku Solver.

Cependant, l'aspect extrêmement spécifique et niche du programme peut entraîner un problème de demande. Il risque d'y avoir trop peu de demandes pour que ce système soit viable. Il faudra alors l'adapter aux vues de la situation.



4 Découpage du projet

4.1 Les différentes parties du projet

- Traitement d'image: appliquer différents filtres de convolution + identifier le sudoku dans l'image + découper le sudoku en 81 cases.
- Réseau de neurones : réseau neuronal permettant d'identifier les numéros dans les 81 cases découpées + recréer le sudoku de l'image dans un tableau de 81 cases.
- Sudoku Solver : algorithme permettant de résoudre le sudoku donné en paramètre et renvoyant le résultat sous la forme d'un tableau de 81 cases.
- Interface : esthétique du logiciel + permettre d'appliquer tous les filtres de façon simple et intuitive.

4.2 Répartition des tâches

Nous avons essayé de nous répartir au mieux les tâches citées précédemment en fonction de leur complexité et de la durée qu'elle prennent à être effectuées: des ajustements dans la répartition pourront également être faits au cas où certaines tâches s'avèrent plus difficiles ou chronophages que prévues.

Voici ci-dessous notre planning de répartition actuel.

	Valentin	Alexandre	Mark	Kevin
Traitement d'Image			X	♡
Réseau neuronal	♡	♡		
Sudoku Solver			♡	
Interface	X	♡	X	X

Légende : ♡ = Responsable de la tâche ; X = Travaille sur la tâche



4.3 Planning visé au démarrage du projet

Nous avons essayé de répartir les tâches de manière équitable, non pas en termes de complexité, mais en termes de temps de travail.

	Soutenance 1	Soutenance 2
Site Web	0%	100%
Traitement d'Image	90%	100%
Réseau neuronal	30%	100%
Sudoku Solver	100%	100%
Interface	0%	100%

Légende : *Espérance d'avancement sur chaque tâche avant la première soutenance*



5 Développement et difficultés rencontrées

```
alexandre@alexandre-ASUS-TUF-Gaming-F17-FX706HCB-TUF706HCB:~/Documents  
./pretreatment img5.jpg u --autorotate  
Rotation Angle: -1266272024573309301035106304.00 degrees  
*** stack smashing detected ***: terminated  
Abandon (core dumped)
```

Démonstration d'une journée type à EPITA

5.1 Traitement d'image

5.1.1 1ère soutenance (0% → 70%)

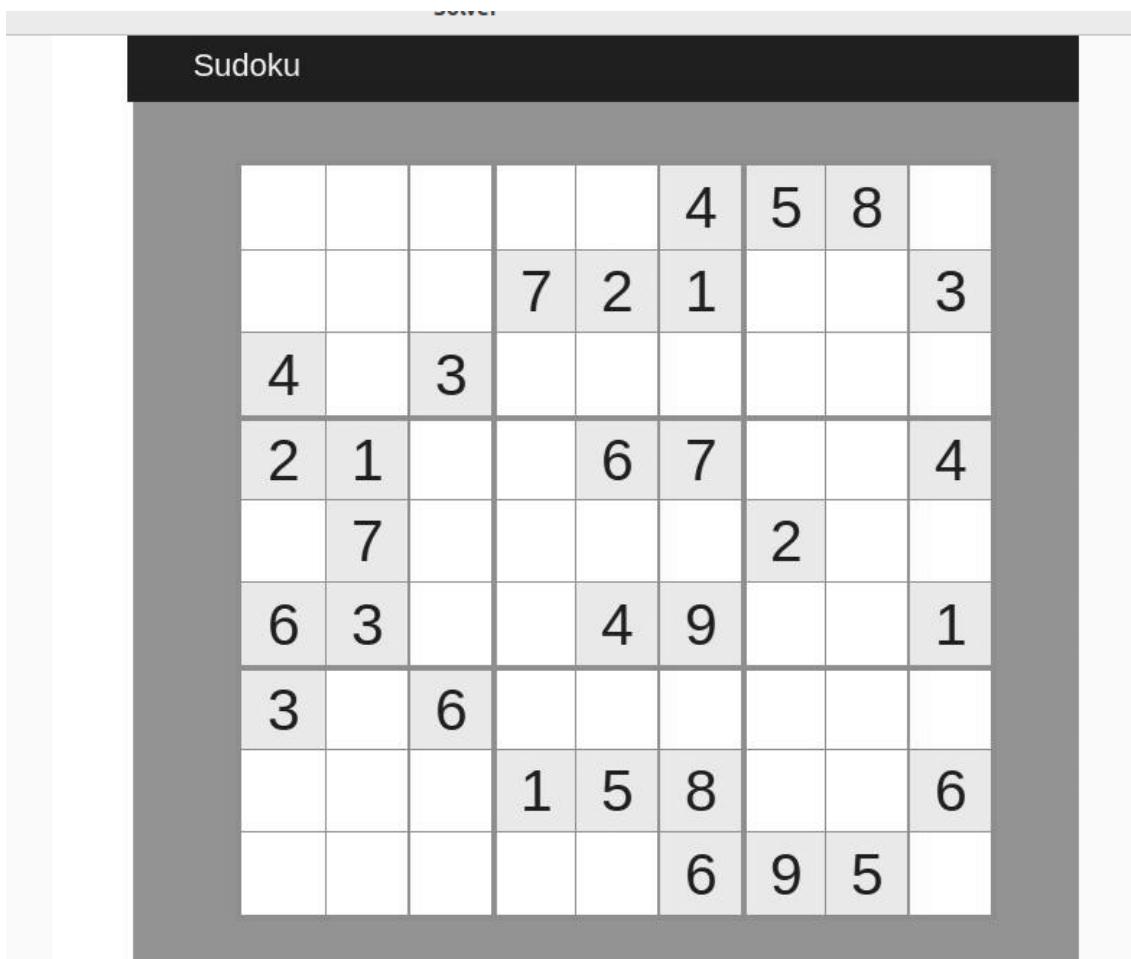
Pour implémenter les traitements nécessaires à notre projet, nous avons retenu la bibliothèque SDL Image. Il a donc tout d'abord fallu mettre en place des fonctions de chargement d'images, permettant de stocker celles-ci dans des surfaces SDL préalablement à l'application des traitements.

Une fois le terrain correctement préparé, la conversion en niveaux de gris a été la première fonction à coder. Son implémentation a été rapide, ne nécessitant que l'application d'une formule pondérant l'intensité des composantes rouge, verte, et bleue de chaque pixel.

En revanche, la conversion en noir et blanc, pouvant pourtant paraître simple au premier abord, a nécessité bien plus de réflexion afin de faire face aux problématiques rencontrées.

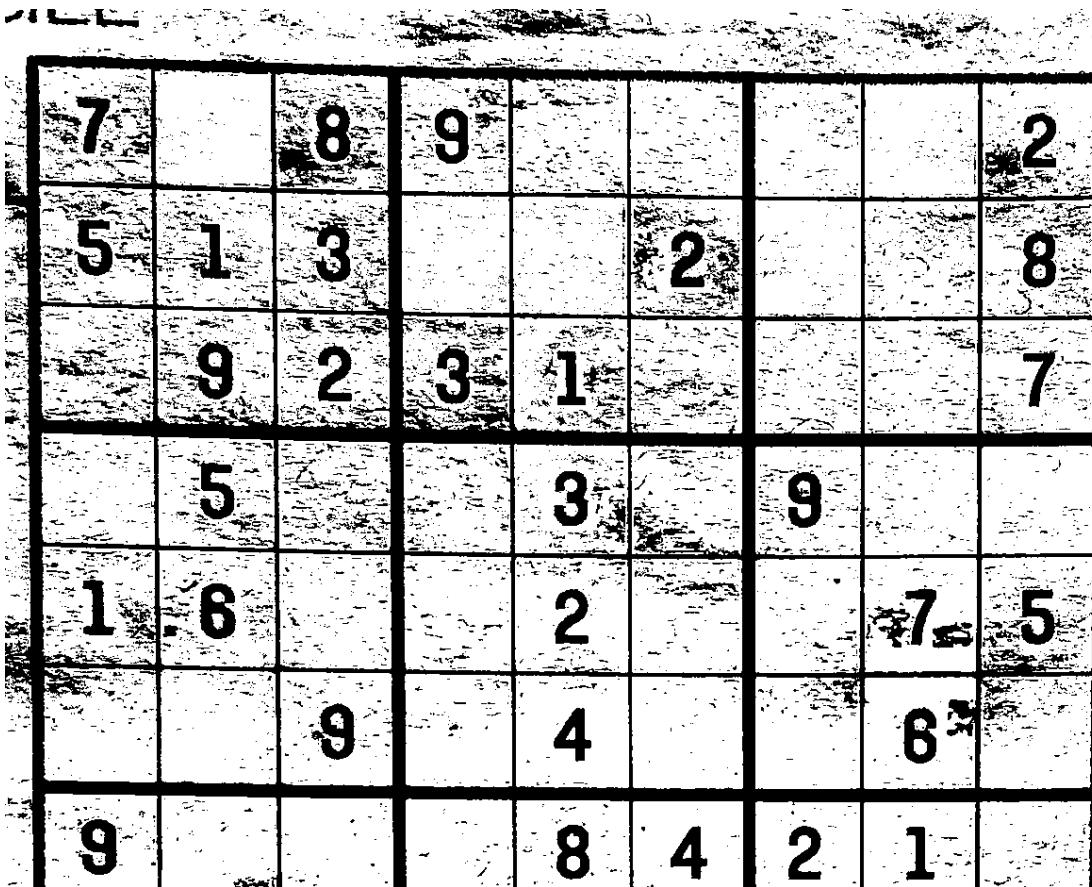
Une première version naïve avec un threshold fixe a été implementée : tout pixel inférieur au threshold devenait noir, sinon il devenait blanc. Cette version (avant tout réalisée à des fins de test) a montré ses limites évidentes. Seules les premières images bénéficiaient d'une conversion correcte, le threshold étant alors tantôt trop bas, tantôt trop haut pour les images plus complexes.





La méthode d’Otsu a alors été choisie pour améliorer la conversion en noir et blanc. Cette méthode procède à une analyse de l’image et détermine, grâce à un histogramme des niveaux de gris, quel est le threshold optimal à appliquer sur l’image. Cette méthode a alors abouti sur des résultats corrects sur toutes les images citées en exemple, sauf la 4e. En effet, la méthode d’Otsu ne fait que déterminer un threshold global à appliquer à toute l’image. Or, l’image 4 présente une grande disparité de contraste et de luminosité, ce qui produit alors du noir indésirable au niveau des zones d’ombre après application d’Otsu.



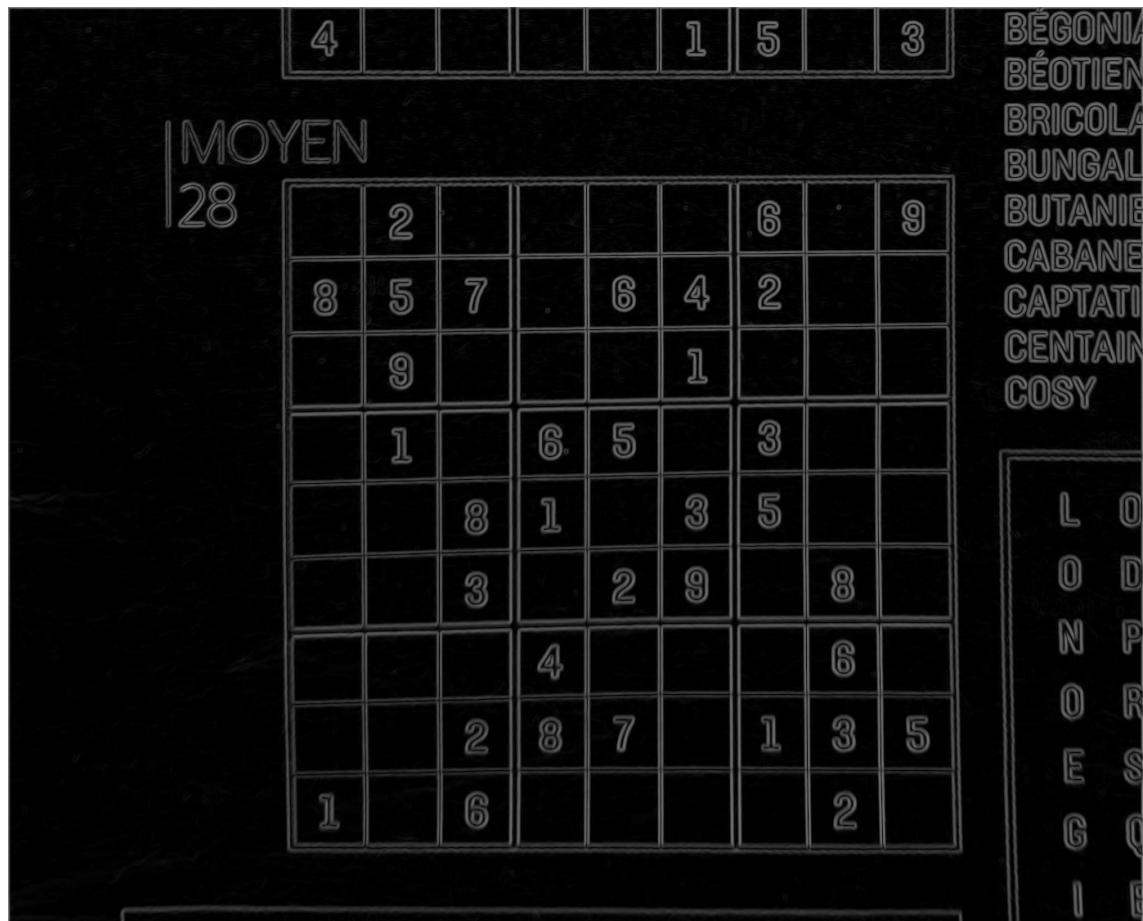


Filtre de otsu avec seuil adaptatif

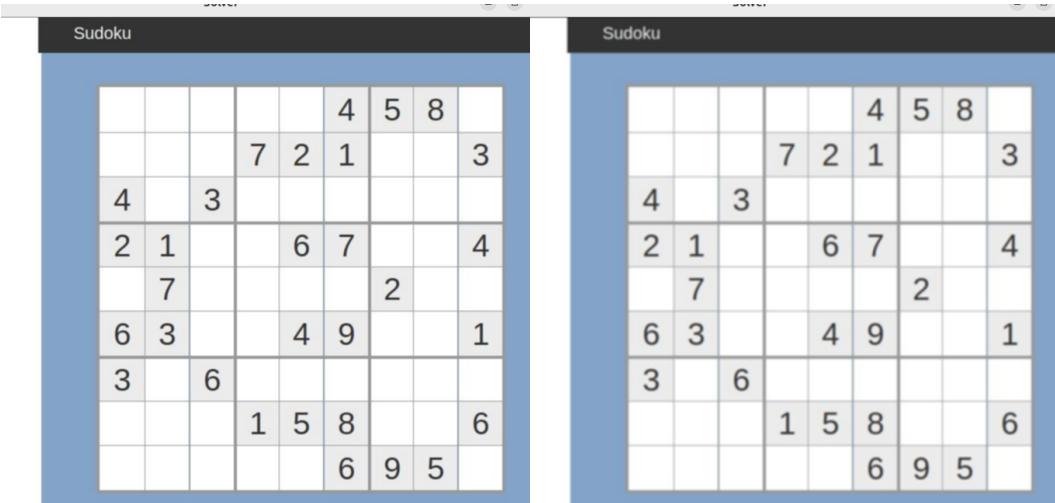
Une fonction de normalisation a donc été implémentée, dans l'espoir de mieux répartir les niveaux de gris sur l'ensemble du spectre pour améliorer le traitement. Cette fonction analyse l'histogramme des niveaux de gris de l'image et "normalise" ces niveaux de gris en les étalonnant de manière plus équilibrée entre 0 et 255, améliorant ainsi le contraste de l'image. Cependant, cela n'a pas permis d'améliorer la conversion en noir et blanc. (Cette fonction a tout de même été conservée dans le code source pour sa possible utilité dans de futurs traitements/filtres).

La fonction désormais retenue pour le noir et blanc est la méthode de Sauvola. Contrairement à Otsu qui ne définit qu'un threshold global, Sauvola détermine un threshold par bloc de pixels analysées. Cela permet ainsi de séparer le traitement de zones aux luminosités différentes et de rendre un meilleur résultat malgré les "défauts" de l'image originale.





Pour améliorer la conversion en noir et blanc et également pour se préparer à la détection des bordures, deux fonctions de flou gaussien ont été réalisées. Ces deux méthodes de flou gaussien se basent sur le principe des filtres de convolution en opérant sur des matrices : l'une en 3x3, pour un léger flou, l'autre en 5x5, pour un flou un peu plus intense. Le flou gaussien nous permet de réduire le bruit de l'image préalablement à l'application des traitements nommés précédemment. Le résultatat de la chaîne de traitement est ainsi amélioré.



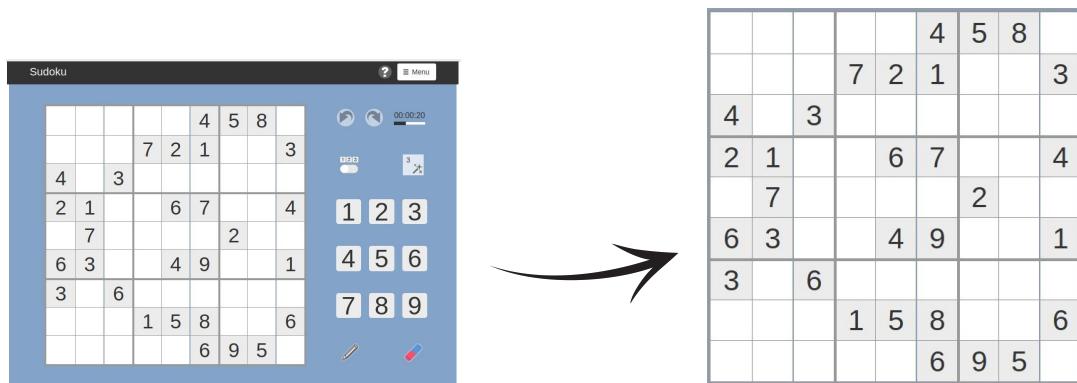
Flou gaussien léger 3x3 et fort 5x5

Une fonction de rotation et une fonction de redimensionnement ont été implémentées. La fonction de rotation permet notamment le traitement de la grille d'exemple n°5. La rotation crée une nouvelle surface de la même dimension que l'originale, puis applique des règles de trigonométrie afin de localiser l'emplacement du pixel source à copier dans le pixel de l'image destination. Ainsi l'on est sûr que, malgré les arrondis des fonctions trigonométriques, tous les éléments de la destination auront forcément une source.





rotation d'une image sur un angle donné



découpage d'une image pour isoler la partie avec la grille

La rotation faisant disparaître une partie des bords de l'image, le redimensionnement permet à l'utilisateur d'élargir les bordures si nécessaire avant la rotation, afin de conserver la totalité de la grille de Sudoku dans le cadre de l'image.

Enfin, pour effectuer la détection de la grille (et par la suite le découpage de celle-ci en images individuelles), la méthode de la transformation de Hough a été retenue.



5.1.2 2ème soutenance (70% → 100%)

Pour cette deuxième soutenance, un nouveau filtre a été ajouté afin de parfaire la qualité des images après extraction. En effet, ce filtre sobrement nommé "Algo Master Clean" est le seul à être appelé après le découpage de la grille du sudoku et ne trouve donc pas la vocation de son existence dans l'amélioration de l'image afin de faciliter la détection de la grille. Son but est de nettoyer de tout bruit l'ensemble des images extraites afin de faciliter la reconnaissance des caractères par l'OCR, en utilisant la toute-puissance millénaire d'Algo Master.

Le fonctionnement de l'Algo Master Clean est relativement simple. Lors du traitement d'une image, nous allons d'abord appliquer un filtre noir et blanc afin d'améliorer les résultats du filtre Algo Master. Ensuite, nous allons considérer chaque pixel noir comme le noeud d'un graphe. En faisant cela, nous allons réaliser un parcours en profondeur sur le graphe de l'image afin de trouver la plus grande composante connexe du graphe. Sachant que le noeud d'un graphe est considéré lié à un autre noeud lorsque le pixel noir qu'il représente touche un deuxième pixel noir. Pour finir, nous allons parcourir le vecteur de marque du graphe et transformer en pixel blanc la totalité des pixels n'appartenant pas à la plus grande composante connexe.

Chaque composante connexe est représentée dans le vecteur de marque par un noeud "représentant de la composante connexe". Ce représentant est choisi lors du parcours si le noeud est le premier pixel de la composante connexe.

En ce qui concerne les avantages de cet algorithme, nous pouvons noter sa grande rapidité d'exécution (temps d'exécution bien inférieur à 0,1 seconde pour le traitement de 81 images), l'hommage rendu au savoir millénaire qui nous a été transmis par Algo Master ainsi que sa grande efficacité. Le seul défaut qui peut lui être trouvé est qu'il doit s'appliquer seulement sur des images déjà découpées (les cases extraites du sudoku), mais ce défaut n'est pas un problème car ce filtre n'a pas vocation à être utilisé dans cette situation.



Image d'un huit avant puis après application de la puissance d'Algo Master



Son incapacité à être utilisé sur une image non découpée est inhérente à son mode de fonctionnement. En effet, cette fonction se base sur la recherche du plus grand graphe connexe. Sur une image découpée, ce dernier sera forcément le chiffre et les autres graphes seront le bruit. Dans l'image non découpée, l'utilisation de ce filtre n'a juste aucune raison d'être car le plus grand graphe connexe ne sera pas les chiffres.

Pour conclure sur le filtre Algo Master Clean, nous pouvons dire que ses nombreux avantages et sa nature de filtre applicable en "Post-Traitement" nous permettront d'améliorer considérablement la fiabilité de l'OCR.

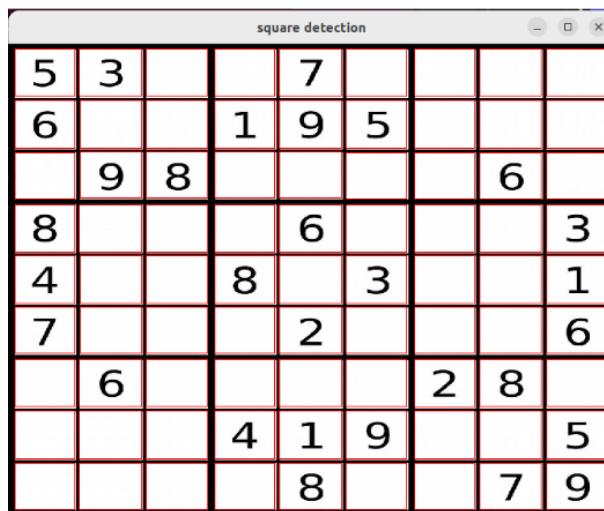
5.2 Grid Detection

5.2.1 1ère soutenance (0% → 90%)

Nous avons réalisé un programme permettant de faire le lien entre l'ensemble du prétraitement, qui nous permet d'obtenir une image exploitable, et l'OCR qui reconnaît le chiffre d'une image.

Pour ce faire nous avons opté pour une simple détection de carrés.

A l'aide de 2 fonctions, nous pouvons détecter les contours d'un carré dans l'image et lister tous les carrés que nous avons trouvés. Il nous suffit ensuite de parcourir cette liste et de garder les carrés faisant une certaine taille (pour qu'ils correspondent à des cases du sudoku).



Démonstration de la détection de grille sur l'image 1



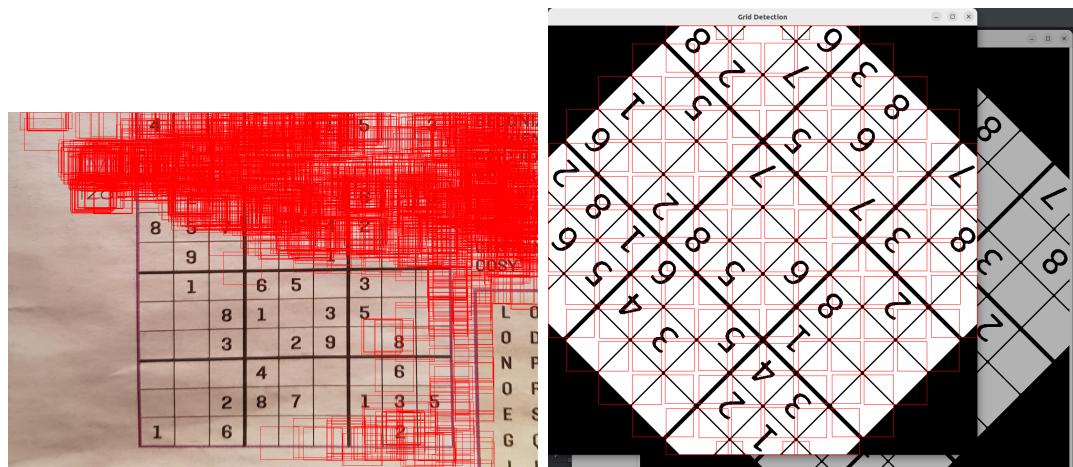
Une fois ces carrés détectés, nous prenons les tailles du plus petit (pour simplifier les images) et nous sauvegardons toutes les sous-images contenues dans les carrés dans un dossier d'extraction. Ces images pourront donc être exploitées par l'OCR par la suite.

Si jamais il y'en a plus ou moins que 81, nous pourrons ajuster les détections ou les filtres mais pour le moment les paramètres sont identiques pour toutes les cases.

5.2.2 2ème soutenance (90% → 100%)

Pour cette seconde soutenance, le défi ambitieux consistait à transformer et affiner notre processus de détection de grille, une entreprise qui s'est révélée être un exercice particulièrement exigeant. Notre objectif central était d'étendre la capacité du système à reconnaître des sudokus plus complexes, caractérisés par des inclinaisons subtiles sur les cases, ainsi que la possibilité que des lignes soient partiellement occultées en raison de filtres appliqués ou de la qualité médiocre de l'image. Bien que ces défis aient constitué des obstacles significatifs, il était impératif de perfectionner notre algorithme pour garantir la détection précise de ces cases, même dans des conditions adverses.

Les ajustements apportés ont eu un impact considérable sur le nombre de rectangles identifiés dans les sudokus, rendant nécessaire une révision approfondie du processus de tri. Les tailles minimales et maximales d'un carré ont été ajustées avec minutie, améliorant ainsi la phase de filtrage et réduisant le risque d'erreurs dans le processus de détection. En parallèle, la conception de l'algorithme de détection a été repensée pour une interaction plus harmonieuse avec les filtres du prétraitement, garantissant ainsi des détections plus précises en alignement avec les caractéristiques spécifiques des images d'entrée et anticipant les anomalies potentielles.

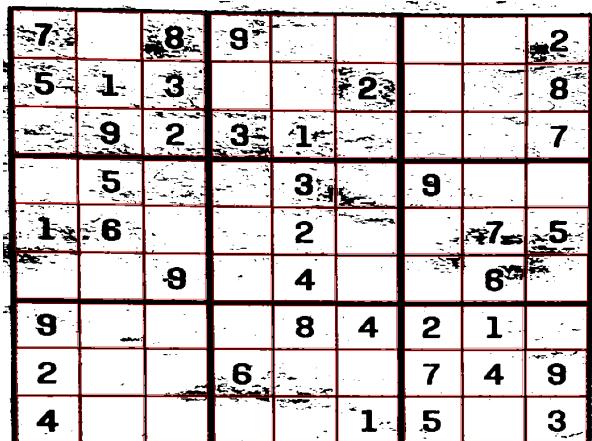


Exemples de difficultés rencontrées

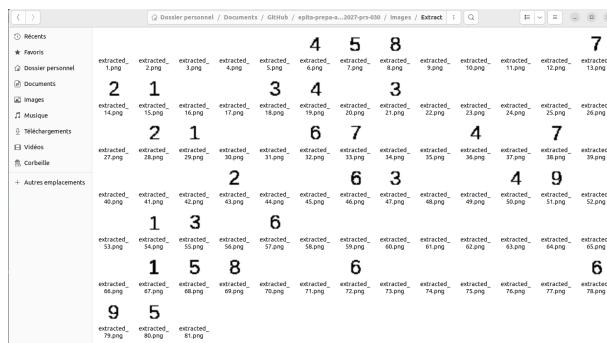


Ces modifications ont revêtu une importance cruciale dans le renforcement de la robustesse de notre système de détection de grille, prenant en considération une multitude de variations possibles dans la présentation des sudokus. Elles ont favorisé une adaptation plus agile aux défis inhérents aux inclinaisons, aux coupures de lignes et aux fluctuations de la qualité d'image. Ce travail a indéniablement contribué à l'amélioration globale des performances de notre projet. Nous sommes fermement convaincus que ces ajustements consolident la capacité de notre solution à traiter des cas plus complexes et à fournir des résultats précis dans une gamme étendue de conditions. L'aboutissement de ces refinements constitue une étape significative dans la réalisation de notre objectif, et nous sommes confiants quant à la pertinence et à la fiabilité accrues de notre système dans des situations diversifiées.

Voici quelques images d'illustration pour que vous vous fassiez une idée des avancements effectués :



Démonstration d'une détection de grille sur l'image 4 (avec bruit) après application de filtres



Exemple du dossier d'extraction des cases



5.3 Sudoku Solver

5.3.1 1ère soutenance (0% → 100%)

Afin de réaliser le Sudoku Solver, nous nous sommes basés sur le principe de "backtracking" traduisible par "retour en arrière".

Le principe de l'algorithme est plutôt simple à comprendre.

Le backtracking consiste dans la résolution par approche récursive afin de remplir les cases vides de la grille. Pour cela, L'algorithme commence par choisir un chiffre possible pour une case vide, en veillant à ce qu'il n'entre pas en conflit avec les chiffres déjà présents dans la même ligne, colonne et carré tous associés. Ensuite, il explore de manière récursive les cases suivantes, répétant ce processus.

Lorsque l'algorithme détecte une erreur, par exemple, un chiffre en conflit avec un autre dans la même unité (ligne, colonne ou carré), ou lorsqu'il atteint une impasse, il revient en arrière (backtrack) à la dernière décision prise, c'est-à-dire la case précédemment remplie.

À ce stade, il modifie le chiffre choisi et reprend l'exploration récursive à partir de ce point. Cette séquence se répète jusqu'à ce que la grille soit complètement remplie sans erreur ni impasse, assurant ainsi une solution valide respectant les règles du sudoku soit retenue.

En créant un algorithme basé sur cette méthode, nous nous sommes assurés d'obtenir des résultats cohérents et rapides.

Given Sudoku:	Solved Sudoku:
... .4 58.	127 634 589
... 721 ..3	589 721 643
4.3	463 985 127
21. .67 ..4	218 567 394
.7. ... 2..	974 813 265
63. .49 ..1	635 249 871
3.6	356 492 718
... 158 ..6	792 158 436
... ..6 95.	841 376 952

Avant/Après Sudoku Solver 9x9



Bien qu'il existe d'autres algorithmes un peu plus optimisés que celui-ci, le gain de temps d'execution a été jugé négligeable pour une implémentation qui se serait avérée bien plus complexe.

5.3.2 2ème soutenance (100% → 120%)

Afin de parfaire notre sudoku solver, quelques détails ont été revus/ajoutés. En effet, il est maintenant possible pour ce dernier de résoudre des problèmes de 16 par 16.

Au départ, le sudoku solver n'était capable de résoudre uniquement des tableau de 9x9 mais il est désormais possible de résoudre des 16x16. Cela n'aura pas été très long à implémenter car nous nous sommes calqués sur le principe de backtracking de son prédecesseur (voir ce qui a été réalisé pour la soutenance 1).

Une fonction a été également ajoutée afin de permettre au sudoku solver de savoir lequel du programme fait pour la résolution du 9x9 ou du 16x16 il doit utiliser. Il fera son choix en fonction de la taille du tableau donné en paramètre. Cela peut paraître n'être pas réellement important, mais c'est ce genre de détails qui fait la différence sur le produit final et qui contribue à l'expérience utilisateur.

Given Sudoku:	Solved Sudoku:
.F.1 .2AE C...	8FB1 62AE C7D3 G945
.63G C.84 EF1. 2...	A63G C584 EF19 2B7D
E.97 B3F.	E597 B3FD 82G4 CA16
4D2C 6... .F..	4D2C 197G 6A5B 3F8E
.... E1B7 35A. .8.C	926F E1B7 35AG 48DC
3G.. 24.. .E7D ..5F	3GC8 2469 BE7D A15F
B.5.94. .6..	BA5D 8C3F 1942 76EG
.... D.G5 F..C	147E DAG5 F68C 923B
.... 9.1C .83A B.F.	D7G5 961C 283A BEF4
2C.B ..E3 54.. ..9.	2C8B 7GE3 546F 1D9A
63.4 ..D. .B91 .CG2	63E4 AFD8 7B91 5CG2
..A9C. 8.67	F1A9 4B52 DGCE 8367
C8.. G..A .D.. .5..	C843 G72A 9DE6 F5B1
5... 3.46 .1F.	5BD2 3846 A1F7 EGC9
.916 .E.B ..2. ..A8	7916 FECB G325 D4A8
.E.. .D9. 4CB8 ..2.	GEFA 5D91 4CB8 6723

Avant/Après Sudoku Solver 16x16



Cependant la méthode du backtracking commence à montrer ses limites. En effet, pour le 16x16 nous passons de 81 cases à 256, augmentant la quantité de calcul à réaliser de façon significative. La résolution du sudoku peut alors entraîner quelques secondes d'attente. Nous avons estimer que cela était une durée acceptable mais nous recommanderions vivement d'optimiser voir de changer de méthode s'il venait à résoudre des sudokus plus important en taille.

5.4 Réseau neuronal

5.4.1 1ère soutenance (0% → 35%)

Pour le réaliser, le travail a été décomposé en plusieurs étapes dont la première était bien entendu la compréhension du fonctionnement d'un réseau neuronal (Racourci NN). Cela étant passé par la lecture de multiple articles et documentations afin de pouvoir appréhender ce sujet vaste et complexe.

Après avoir compris les grandes lignes nous nous sommes donc lancés dans la conception puis l'implémentation d'un NN permettant de réaliser une fonction Xor. Ce réseau est composé de trois couches distinctes, l'Input, une couche cachée et l'Output. De par sa simplicité, l'implémentation de ce réseau se fit rapidement et sans grande difficulté.

Nous nous sommes donc lancés avec confiance dans l'implémentation du NN servant d'OCR et ce en partant de la base solide que représentait le Xor. Nous l'avons donc adapté afin que ce réseau puisse fonctionner avec des arrays, nous permettant de gérer un grand nombre de poids et de neurone. Ce dernier est, lui aussi, composé de trois couches distinctes dont une couche cachée et ce dernier est réalisé sur une base de fully-connected-network.

C'est à dire que l'entièreté des noeuds d'une couche sont connectés avec l'entièreté des noeuds de la couche suivante.

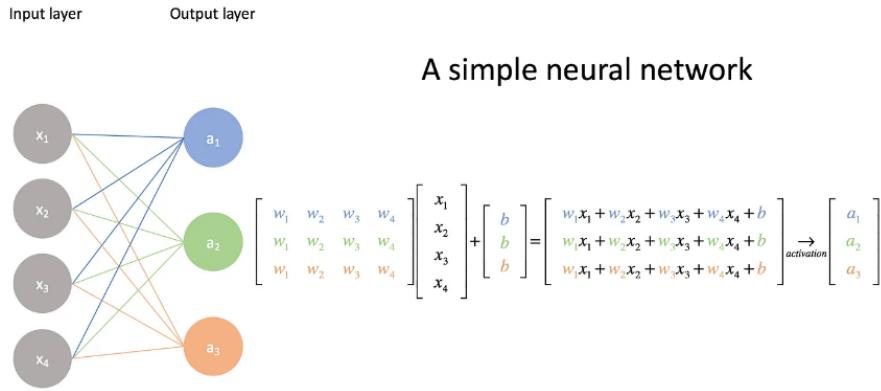
Quant à la fonction d'activation, celle actuellement utilisée est une simple sigmoid (ainsi que sa dérivé pour la correction des poids lors de la phase de Back propagation).

Malheureusement ce réseau est beaucoup plus complexe à implémenter et son développement c'est avéré extrêmement chronophage à cause du fait que le débogage d'un réseau neuronal est très fastidieux et ce malgré sa proximité de conception avec le réseau neuronal Xor vu plus haut.

Le débogage du NN de l'OCR est d'abord passé par l'ajout de biais, qui n'étaient pas utilisés auparavant. Ces derniers ont permis à l'OCR de reconnaître un chiffre unique lorsque l'IA à été entrainé dessus mais cela ne lui a malheureusement pas octroyé la capacité de faire une prédiction viable lorsqu'elle a été entraînée sur un ensemble de chiffres.

A ce jour, nous n'avons malheureusement pas encore réussi à réaliser une prédiction viable avec cette IA mais nous avons la conviction d'y parvenir prochainement.

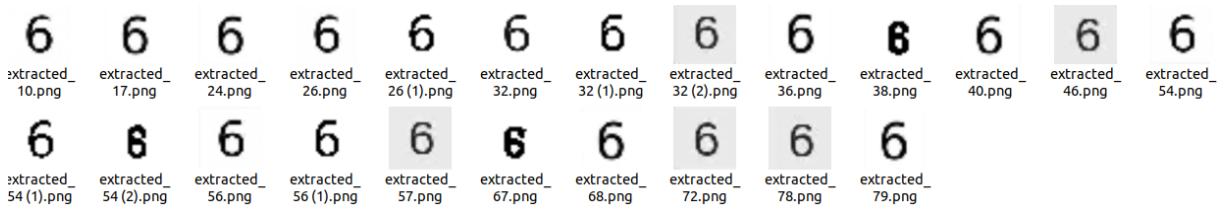




5.4.2 2ème soutenance (35% → 100%)

Beaucoup de choses se sont passées entre la première et la deuxième soutenance pour notre fantastique réseau neuronal. Ce dernier a subi de nombreuses modifications lui permettant d'atteindre un niveau d'efficacité excellent.

Depuis la dernière soutenance, l'OCR a subi de nombreux changements l'ayant d'abord amené à se séparer de l'architecture du réseau XOR. En effet, à ses débuts, l'OCR était basé sur le réseau XOR et suivait ses évolutions. Désormais, notre OCR a sa propre architecture ainsi que sa propre fonction. Cette séparation de l'OCR par rapport au XOR passe par sa fonction d'activation qui est passée de la fonction sigmoïde (utilisée de base sur le XOR) à une fonction Softmax. Le changement d'architecture passe aussi par l'utilisation d'une structure pour le réseau de neurones au lieu d'une liste de poids et de biais, rendant la gestion du réseau plus simple et lisible. Du côté de l'apprentissage, nous sommes restés sur une classique front/back propagation, utilisant donc notre structure remise à neuf. Grâce à notre base de données, nous réussissons à obtenir d'excellents résultats de manière très rapide (100% de réussite au bout de 200 itérations), et pour tous les exemples que nous avons pu tester, des résultats similaires ont pu être constatés.



Exemple d'une partie de la database



Notre base de données a été constituée par nos soins en récupérant diverses cases de sudoku trouvées sur internet (découpage fait par notre Grid détection). Cela a permis d'obtenir des chiffres diversifiés de par leur taille, police d'écriture, position dans l'image, ...

Au niveau de sa constitution, nous avons décidé de simplifier sa structure. En effet, l'ancienne version du réseau de l'OCR possédait un hidden layer, ce qui rendait ce réseau beaucoup plus gourmand en puissance de calcul. Après divers essais, nous nous sommes rendu compte que le retirer optimisait notre réseau mais améliorait aussi ses prédictions. Nous avons donc décidé de garder ce réseau.



Exemple d'une prédition : probabilité d'un chiffre

Au final, à travers les connaissances acquises lors de la réalisation d'un réseau XOR, puis de nos multiples échecs, nous avons su nous adapter et rebondir afin de créer un réel réseau de neurones fiable capable de détecter des chiffres à coup sûr pour résoudre un Sudoku, et nous sommes fiers que cela soit mission accomplie!

```

Step 0998      Average Loss: 0.26      Accuracy: 0.905
Step 0999      Average Loss: 0.38      Accuracy: 0.908
Predicted 0007
Predicted 0002
Predicted 0001
Predicted 0000
Predicted 0004
Predicted 0001
Predicted 0004
Predicted 0009
Predicted 0006
Predicted 0009
alexandre@alexandre-ASUS-TUF-Gaming-F17-FX706HCB-TUF706HCB:~/

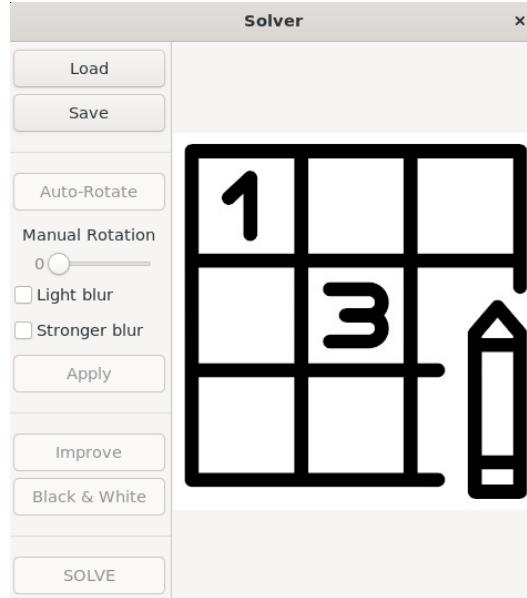
```

Phase d'apprentissage du réseau de neuronne



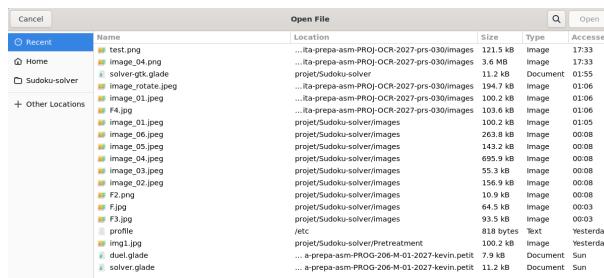
5.5 Interface Graphique

5.5.1 2ème soutenance (0% → 100%)



Interface avec logo à l'ouverture

L'interface graphique a été entièrement réalisée lors de cette seconde période. Elle nous avait été inutile lors de la première soutenance, les parties n'étant encore qu'indépendantes les unes par rapport aux autres. Cette interface graphique a été réalisée à l'aide de la bibliothèque GTK, mais également de la bibliothèque SDL pour une partie spécifique.



Explorateur GTK pour LOAD/SAVE

La fenêtre de l'interface se compose tout d'abord des boutons LOAD et SAVE, éléments primordiaux de cette application. Ces deux boutons ouvrent un explorateur de fichiers GTK permettant à l'utilisateur d'importer son image (.png, .jpeg, .jpg) dans la fenêtre principale, ou de sauvegarder dans l'emplacement de son choix l'image actuellement affichée dans l'application.



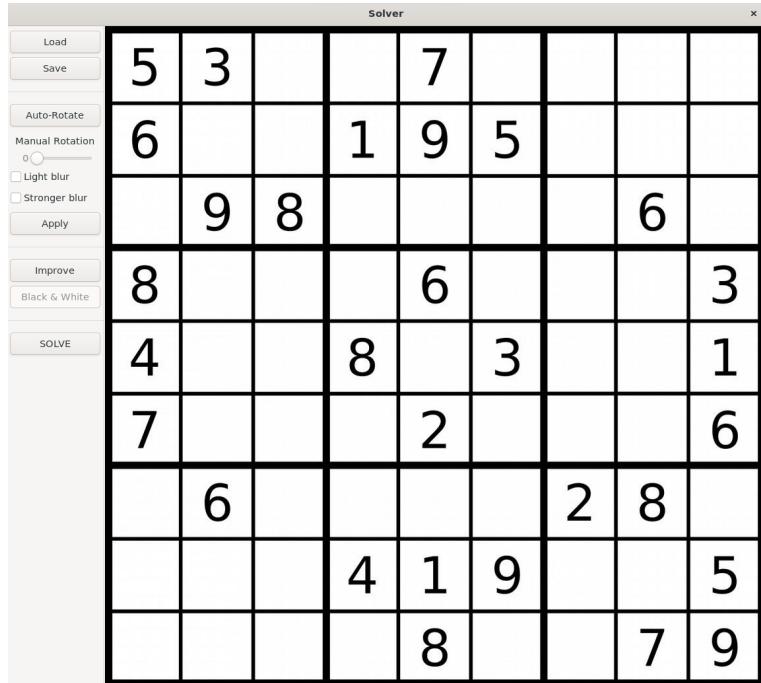


Image chargée

Viennent ensuite les traitements manuels que l'utilisateur peut appliquer à son image afin d'améliorer la détection et le découpage de la grille. L'utilisateur peut alors appliquer une rotation de n'importe quel angle entre 0 et 360 à l'aide d'un slider, incrémentable de 1 en 1 à l'aide des flèches directionnelles GAUCHE/DROITE et de 10 en 10 à l'aide des touches PAGE UP/PAGE DOWN.

L'utilisateur peut également appliquer les traitements de flou gaussien. Le premier, un flou assez léger, utilise une matrice de convolution de Gauss de 3x3, tandis que le second, plus puissant, utilise une matrice de Gauss en 5x5. La sélection du flou le plus puissant des deux entraîne automatiquement le blocage de la sélection du plus léger, qui ne sera donc pas applicable tant que le flou 5x5 est demandé.

L'application de la rotation manuelle et du flou de Gauss se font à l'aide du bouton **Apply** situé juste à la fin de ce bloc de l'application.

Viennent ensuite les boutons **IMPROVE** et **BLACK AND WHITE**. Ces traitements sont quasi obligatoires afin de résoudre l'image correctement lors de l'appel du **SOLVER**.

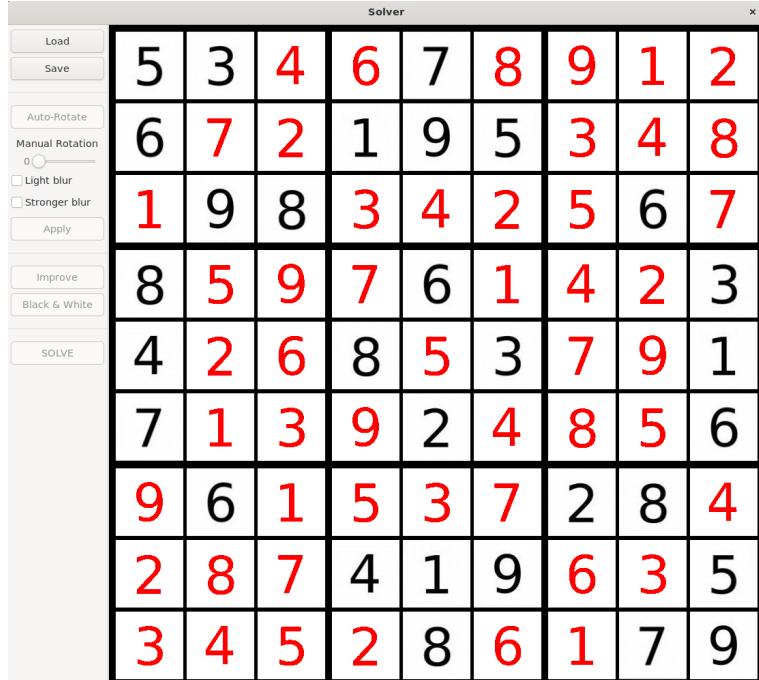
Le bouton **IMPROVE** applique un grayscale normalisé à l'image. C'est à dire qu'en plus de convertir l'image en niveaux de gris, cette fonction améliore les contrastes grâce à un réétalonnage de l'histogramme des niveaux de gris. Il est obligatoire d'appliquer ce traitement pour avoir le droit d'appliquer le traitement **BLACK**



AND WHITE, qui reste grisé tant que l'IMPROVE n'a pas été réalisé.

Le BLACK AND WHITE transforme (comme son nom l'indique de manière assez explicite) l'image de la fenêtre pour la convertir en noir et blanc à l'aide du filtre d'Otsu. Elle marque la fin de ce troisième bloc de commandes de l'interface GTK.

Enfin, le bouton SOLVE, le plus important, réalise la réalisation du Sudoku en appelant successivement toutes les étapes restantes : détection de la grille, extraction des cases, reconnaissance des chiffres, résolution du sudoku reconstitué et affichage de la grille reconstituée une fois résolue. La phase de détection de la grille est affichée dans une fenêtre "pop-up" SDL, permettant à l'utilisateur de s'assurer que cette étape s'est correctement effectuée (et dans le cas inverse d'éventuellement revoir l'application de ses traitements optionnels). Une fois cette fenêtre fermée par l'utilisateur, l'interface GTK s'actualise et laisse apparaître (si la résolution a été possible) une affichage reconstitué de la grille, avec les nouveaux chiffres ajoutés lors de la résolution mis en évidence en rouge.



Grille complétée après SOLVE

Cette grille résolue présentée dans un affichage standardisé est alors enregistrable à l'aide du bouton SAVE présenté précédemment.



6 Impressions du groupe

6.1 1ère soutenance

6.1.1 Valentin

Sur la période précédant la première soutenance j'ai été chargé de l'élaboration et de l'implémentation des réseaux neuronaux, durant m'a tâche j'ai eu la chance d'être assisté par Alexandre qui c'est avéré être d'une grande aide.

Comme dit précédemment dans la partie sur le réseau neuronal, l'implémentation de celui ci réalisant une fonction Xor a été relativement simple.

La véritable difficulté résidant dans le coeur de ce projet : l'OCR.

Le réseau neuronal dédié à l'OCR a pour but de reconnaître un chiffre sur une image afin de le traduire en entier pour l'ordinateur.

Ce dernier a requis la majeure partie de notre temps à Alexandre et à moi de par les nombreux problèmes rencontrés et la difficulté de débogage.

Malgré toutes ces difficultées, je suis très heureux d'avoir été affecté à l'élaboration des réseaux neuronaux car cela m'a permis d'explorer un nouveau pan de l'informatique que je n'aurais sûrement pas exploré autrement.

Grâce à cela j'ai pu découvrir et apprendre énormément sur le fonctionnement des réseaux neuronaux, du deep learning et plus généralement des IAs.

Ce projet m'aura permis d'élargir mon champ de compétence et de compréhension de l'Informatique. surtout avec les possibilités infinies que représente l'IA et ce en peu de temps.

Le réseau neuronal mis à part, j'ai aussi participé, avec Mark et Alexandre à la détection de grille. Son élaboration était elle aussi assez compliquée et chronophage mais nous avons réussi (en grande partie grâce à la mise en commun des idées de chacun) à résoudre ce problème. Cela est très motivant pour la suite et j'espère que l'on réussira, pour la deuxième soutenance, à faire fonctionner l'OCR à l'instar de la détection de grille

Pour conclure, je peux dire sans crainte que, malgré les difficultés rencontrées, notre groupe arrivera à surmonter ces problèmes grâce à sa grande cohésion afin de fournir un travail pour la deuxième soutenance dont nous pourrons également être fier.



6.1.2 Alexandre

Mon rôle était le développement avec l'aide de Valentin d'un réseau neuronal capable de reconnaître une porte XOR, puis d'améliorer ce réseau afin de créer le centre de ce projet : l'OCR (Optical character recognition) soit un algorithme apprenant par lui-même à comprendre quel chiffre est représenté dans une image.

Avec Valentin nous avons été sujets à de nombreux problèmes, les réseaux neuronaux étant extrêmement complexes à corriger, réajuster et debugger s'avère être une tâche plus que difficile surtout pour nous qui n'avons jamais eu l'occasion de développer ce genre de logiciel.

Cela dit, cela nous a permis d'acquérir des compétences uniques et une compréhension des IAs étonnantes. Je fus ravi d'avoir pu aller jusque là pour cette 1ere soutenance même si les difficultés rencontrées m'ont un peu démoralisé.

Les réseaux de neurones restent passionnats et impressionnats et je suis ravi d'avoir pu les découvrir au travers de notre travail.

J'espère que nous allons aller de l'avant et utiliser ces erreurs comme une force pour avancer de plus belle et réaliser un projet que nous n'avions pas soupçonné etre capable de faire il y a quelques mois.

J'ai confiance en notre groupe et en notre perséverance pour la prochaine soutenance.



6.1.3 Mark

Pour cette première soutenance j'ai entièrement géré la partie Sudoku Solver.

Cela n'était pas d'une difficulté extrême mais il faut admettre que c'était une tâche extrêmement chronophage. En effet, j'ai du coder entièrement le SudokuSolver 2 fois car j'étais parti d'un mauvais axiome.

En effet, je pensais qu'il n'existe qu'une seule et unique façon de résoudre un sudoku.

Quelle ne fut pas ma surprise, une fois que je pensais en avoir terminé, que de me rendre compte que mon programme ne fonctionnait que dans très peu de cas.

Il était alors plus simple de tout recommencer plutôt que de réadapter entièrement ce dernier.

De plus, m'étant mieux renseigné et ayant gagné de l'expérience en C au cours de l'année, j'ai décidé d'utiliser la méthode de backtracking (expliquée précédemment) afin de refaire un code, cette fois-ci, entièrement fonctionnel.

Pour autant, ce premier échec n'a pas été perçu comme tel: me tromper m'a permis de mieux comprendre le fonctionnement de l'algorithme et m'a obligé à acquérir de nombreuses autres connaissances.

En plus de cela, j'ai dû m'intéresser au traitement d'image afin d'essayer de comprendre comment on pouvait reconnaître la grille du Sudoku, ce qui se révélera être une tâche beaucoup plus ardue. Nous nous sommes tous retrouvés à passer plusieurs nuits à réfléchir, et à essayer de trouver des solutions à ce problème.

Après être passé par de nombreux états d'âmes (désespoir, fatigue mentale, colère...) c'est la joie et la fierté de voir enfin notre travail fructifier qui restera la conclusion de cette première soutenance.

Voir le fruit de nos travaux mis bout-à-bout et se compléter les uns les autres a été une source de joie et d'enthousiasme telle, que la motivation n'est que plus grande. Faire notre propre OCR est un rêve à porté de main et je suis fier d'être arrivé jusque là.



6.1.4 Kevin

J'ai été pour cette première période de développement principalement en charge du traitement de l'image.

Ce travail a nécessité beaucoup de documentation en ligne sur les nombreux filtres et autres transformations graphiques existant déjà.

Il était inutile d'essayer de réinventer la roue quand tant de développeurs et de mathématiciens talentueux avaient déjà conçu et théorisé les algorithmes nécessaires pour résoudre notre tâche.

Une grande partie du travail a alors nécessité la compréhension de pseudo-code et de formules mathématiques afin de réaliser leur implémentation avec la bibliothèque SDL Image. Là où d'autres bibliothèques (non accessibles pour ce projet) telles qu'OpenCV offrent à l'utilisateur beaucoup de traitements pré-programmés, SDL ne donne que peu de solutions clés en main au programmeur. Il a donc fallu se charger d'implémenter les différentes méthodes de traitement d'images utiles à notre projet en manipulant les images et leurs composantes pixel par pixel.

La conversion en niveaux de gris a été très simple à implémenter, tout comme le flou de Gauss, qui reste assez trivial à comprendre. En revanche l'un des éléments donnant le plus de fil à retordre a été la conversion en noir et blanc, pour laquelle différents essais avec plusieurs algorithmes de traitements ont été nécessaires avant d'arriver à des résultats corrects sur la majorité des images.

L'autre élément complexe à gérer a été la reconnaissance de la grille et son découpage. Pour réaliser cette tâche difficile, j'ai donc été en partie assisté par mes coéquipiers.



6.2 2ème soutenance

6.2.1 Valentin

Pour cette deuxième soutenance, j'étais principalement chargé de l'implémentation de l'OCR avec Alexandre ainsi qu'à l'élaboration et l'implémentation du filtre Algo Master Clean présenté précédemment dans la partie sur le prétraitement.

L'implémentation de l'OCR était sans surprise relativement complexe. Beaucoup de changement ont été apporté et le fonctionnement de l'OCR c'est finalement séparé du fonctionnement du réseau Xor. L'OCR utilise dorénavant une struct pour l'implementation du réseau et sa fonction d'activation est devenue une softmax afin d'améliorer l'efficacité de cette OCR. Sa liaison avec les autres programmes a quant à elle été très rapide à réalisé car il suffisait de retourner les prédictions de l'OCR au lieu de les afficher dans la console.

La grande efficacité de l'OCR qui permet d'avoir un total de réussite de 100% avec à peine 200 itération d'apprentissage m'emplis de fierté car cet OCR est représentatif de notre travail fourni et représente une sorte de revanche sur l'OCR que nous souhaitions présenter lors de la première soutenance.

Du côté du filtre Algo Master Clean, son élaboration demanda de la réflexion de par la nature originale de ce filtre. Au final, malgré quelques ajustements ayant demandé de nombreux tests, le filtre Algo Master Clean est parfaitement fonctionnel et permet à l'OCR de réaliser une meilleure prédition que ce qu'il pourrait faire en recevant une image n'ayant subit qu'un prétraitement et aucun "Post-Traitement" par ce filtre.

Je suis très fier du travail accompli jusqu'à présent et je suis heureux d'avoir pu fournir, comme mes camarades, un travail digne de ce nom permettant à notre projet d'être efficace, fiable, lisible et fonctionnel, en bref, un produit fini.



6.2.2 Alexandre

Cette étape de soutenance représente pour moi un véritable défi à tous les niveaux de notre projet de développement. La pression était palpable, alimentée par notre désir de concevoir un projet complet qui exigeait non seulement le bon fonctionnement individuel de chaque composant, mais également leur cohésion harmonieuse. En collaboration avec Valentin, j'ai pris en charge la conception du réseau de neurones visant à reconnaître les caractères et à identifier les chiffres du sudoku. Par la suite, mon attention s'est portée sur l'adaptation de la détection de la grille, une tâche complexe qui impliquait de repérer les cases sur les images, même en présence de bruit ou lorsque ces dernières n'étaient pas suffisamment définies.

L'acquisition des compétences nécessaires pour mener à bien ce projet a constitué un véritable défi intellectuel. Nous avons plongé profondément dans les rouages de ce projet, initialement perçu comme abordable, pour découvrir qu'il exigeait en réalité une compréhension approfondie et détaillée de l'ensemble des fonctionnalités attendues.

Je suis convaincu que notre efficacité a été remarquable, et nous avons pleinement atteint notre objectif ultime : la création d'une interface graphique fonctionnelle. Cette interface permet d'ouvrir l'image d'un sudoku, d'appliquer les traitements nécessaires, et d'effectuer avec succès la détection et la résolution de la grille. Notre détermination s'est également étendue à la résolution d'images provenant de magazines, souvent de mauvaise qualité et présentant des cases plus difficiles à isoler et à résoudre.

Le sentiment de fierté est indéniable face au travail accompli. Nous avons non seulement atteint nos principaux objectifs, mais nous avons également développé des compétences uniques dans des domaines variés. Cette expérience enrichissante nous a permis de maîtriser des aspects techniques complexes, de perfectionner notre capacité à résoudre des problèmes et de renforcer notre expertise dans le domaine du traitement d'images et de l'apprentissage automatique. En regardant en arrière, je constate avec satisfaction la progression significative que nous avons réalisée et je suis confiant dans notre capacité à relever de nouveaux défis avec la même détermination et compétence.



6.2.3 Mark

Pour cette deuxième soutenance, mon travail aura plus été dans la supervision que dans la programmation en tant que telle. Je m'occupais surtout d'organiser le groupe, prévoir les réunions, donner les deadlines, et voir si tout se passait au mieux. S'il y avait des questions ou besoin de soutien pour une tâche, je venais assister la personne. J'ai donc dû un petit peu toucher à tout pour cette dernière soutenance, ce que j'ai trouvé très intéressant.

Globalement j'ai pu comprendre beaucoup mieux les tenants et aboutissants de ce projet. J'ai trouvé fascinant le fonctionnement et principes des filtres appliqués sur les images tout autant que d'en apprendre d'avantage sur les réseaux de neurones. Grâce à cela j'ai pu étendre mes connaissances sur beaucoup plus de sujets que je ne pensais.

De plus, quelques petits correctifs ont été aussi été ajoutés au sudoku solver (voir ce qui a été dit sur le sudoku solver pour la 2ème soutenance).

En résumé, cette position plus axée sur le management et de "touche à tout" m'a beaucoup plu lors de ce mois de novembre. Certes, travailler sur ce projet avec des délais aussi courts aura été extrêmement stressant et nous aura valu de longues heures de travail en dehors des cours, cependant, la quantité de connaissance que ce projet nous aura apporté en vaut la peine.

Avoir eu une vision d'ensemble sur le projet et de voir mon rôle évoluer dans le groupe aura été une réelle opportunité pour moi. Je suis très fier de notre équipe et du travail que nous avons réalisé.



6.2.4 Kevin

Le plus ardu dans cette phase de développement a été la planification du travail de groupe. En effet, ne disposant pas cette fois-ci d'une semaine libre de tout cours pour faire le projet, nous avons dû chercher à organiser notre temps de la manière la plus efficace afin de ne pas stopper la progression du projet (tout cela sans oublier de travailler de manière correcte dans le reste des matières). Cette planification n'a pas été toujours été parfaite, mais nous avons réussi malgré ces différents obstacles à obtenir une application de bonne qualité générale.

La gestion collective du travail du groupe a également été plus complexe que lors de la première soutenance. Là où la première période nous laissait développer chacun dans notre coin les différentes parties de l'application de manière indépendante, cette deuxième phase ne permettait plus cela et nécessitait la mise en commun de nos réalisations. Cette mise en commun ne s'est pas faite sans rencontrer quelques embûches. Me chargeant personnellement de l'interface graphique, il apparaît très clair dans cette situation que de bons commentaires de fonctions, détaillés tout en restant simples à comprendre, sont un bonheur en tant que programmeur. Après différents débbugages sur certaines parties interconnectées et concertation entre les membres du groupe chargés des parties concernées, nous avons pu corriger les problèmes majeurs qui nous empêchaient d'atteindre une réalisation fonctionnelle. Le dernier rush avant le rendu a également été un moment difficile, poussant à terminer rapidement l'implémentation de dernières fonctionnalités importantes au programme. Beaucoup de réunions de groupe ont alors été tenues afin de progresser un maximum malgré les contraintes de temps. La répartition décidée initialement s'est également modifiée au fil du temps, certains membres ayant acquis une expertise bien meilleure que d'autres dans leur partie ou s'étant trouvés une affinité particulière pour une tâche différente.



7 Conclusion

7.1 Bilan 1ère soutenance

7.1.1 Ce qu'il nous reste à faire...

En conclusion, nous pouvons dire qu'au travers de tous les points abordés, nous allons réussir à faire notre OCR, nous rendant fiers et étant suffisamment stable pour être considéré comme durable.

Chaque membre du groupe a décidé de se spécialiser en acquérant un maximum de connaissances pour une bonne réalisation et surtout un bon rendu du projet afin d'optimiser un maximum le temps qui nous était imparti.

Le projet est pour nous le moyen d'enrichir notre connaissance de la programmation tout en développant un concept. C'est avec une volonté de redoubler d'efforts que nous sommes certains de réaliser un OCR qui nous rendra fiers.

Nous prévoyons de développer pour la prochaine soutenance: Toute l'interface graphique du logiciel, améliorer nos filtres et, s'il nous reste du temps, créer le site internet présentant le projet.

7.2 Bilan 2ème soutenance

Au cours de cette seconde soutenance, notre projet a connu des avancées significatives à travers plusieurs aspects clés, démontrant l'évolution et l'adaptabilité de notre équipe de développement.

Un nouveau filtre, l'Algo Master Clean, a été introduit pour améliorer la qualité des images extraites du sudoku, contribuant ainsi à nettoyer le bruit et renforçant la fiabilité de l'OCR. La détection de grille a été transformée et affinée, permettant une identification plus précise des rectangles dans les sudokus, renforçant ainsi la robustesse de notre système face à des variations complexes dans la présentation des sudokus.

Des améliorations ont été apportées au Sudoku Solver, lui permettant de résoudre des sudokus de 16x16. En ce qui concerne le réseau de neurones, des modifications substantielles ont été apportées, aboutissant à la création d'un réseau fiable, capable de détecter avec précision les chiffres pour résoudre un Sudoku.

Ces ajustements ont renforcé la crédibilité et l'efficacité de notre projet, nous positionnant favorablement pour les étapes futures. Nous sommes fiers des compétences développées, des obstacles surmontés, et de la contribution significative de chaque membre de l'équipe à la réussite globale de notre initiative.



7.3 Bilan général

En conclusion de ce projet de semestre dédié à l'élaboration collaborative d'un OCR pour la résolution de Sudoku en langage C, l'aboutissement de cette entreprise revêt une signification particulière.

En tant qu'équipe, découvrir les mécanismes internes d'un outil que nous utilisons quotidiennement a été une expérience enrichissante et captivante. Comprendre le fonctionnement complexe d'un OCR, un élément omniprésent dans notre vie quotidienne, a ajouté une dimension pratique et concrète à notre apprentissage. Chacun des membres de l'équipe a pu apprécier la profondeur des défis liés à la reconnaissance optique de caractères et ressentir une connexion plus profonde avec la technologie que nous utilisons instinctivement.

Ce projet a ainsi transcendé le cadre académique pour devenir une exploration pratique, renforçant notre compréhension du monde numérique qui nous entoure. En somme, cette expérience a été une occasion unique d'associer la théorie à la pratique, nous offrant une perspective nouvelle sur la technologie omniprésente dans nos vies et renforçant notre appréciation pour la complexité qui se cache derrière les outils que nous utilisons quotidiennement.

