

SOUTENACE INTÉRMEDIAIRE

Optical Character Recognition



BAME-OCR

Brieuc Crosson
Antoine Lassagne

Enzo Juhel
Maia Bucamp

31 Octobre 2023

Table des matières

1	Introduction	3
2	Les Membres de BAME-OCR	3
2.1	Brieuc CROSSON	3
2.2	Antoine LASSAGNE	4
2.3	Maia BUCAMP	4
2.4	Enzo JUHEL	4
3	Répartition des tâches	4
4	Traitemet d'images	5
4.1	Prétraitemet	5
4.1.1	Suppression des couleurs	6
4.1.2	Filtre Gaussien	7
4.1.3	Filtre de Canny	8
4.1.4	Seuillage par hystérésis	10
4.1.5	Réduction des bruits	11
4.1.6	Résultats	12
4.2	Traitemet	13
4.2.1	Transformée de Hough	13
5	Détection de la grille	16
5.1	Rotation automatique	17
5.2	Détection de toutes les cases de la grille	17
5.2.1	Tri des cases	17
6	Découpage de la grille	18
6.1	Adaptation pour le reseau de neurones	18
6.1.1	Redimension	18
6.1.2	Centrage des numéros	19
7	Reconnaissance de caractères	19
7.1	Structure	19
7.1.1	Propagation	20
7.1.2	Rétropropagation	20
7.2	Entraînement sur Image	20
7.3	Base de donne d'image	21
8	Résolveur de sudoku	21
8.1	Solubilité du sudoku	21
8.2	Solveur	21
8.3	Sauvegarde des résultats	22

9 Interface graphique	22
9.1 Terminal	22
9.1.1 Utilisation indépendante des filtres	23
9.1.2 Utilisation du réseau de neurones	23
9.2 GTK	24
9.2.1 Barre de Titre	25
9.2.2 Rotation et “Step by step”	25
9.2.3 Résolution des sudoku	26
9.3 Sauvegarde des résultats	27
10 Site internet	27
11 Gestion du Projet	28
11.1 Avancée du projet	28
11.2 Organisation du groupe	29
11.2.1 Outils de travail	30
11.2.2 UML	30
12 Conclusion	32
13 Sources	32

1 Introduction

Le groupe BAME-OCR progresse de manière dynamique dans son ambitieux projet de reconnaissance optique de caractères (OCR) au cours du semestre 3. Cette période, du 4 septembre au 9 décembre, a été marquée par un engagement constant dans divers aspects du projet, allant de la détection de lignes à la résolution de sudokus par intelligence artificielle, en passant par le traitement d'images et d'autres fonctionnalités cruciales liées à l'OCR.

Au fil des semaines, notre équipe a minutieusement exploré et mis en œuvre différentes méthodes, adoptant des approches innovantes et des algorithmes spécifiques pour résoudre les problématiques complexes rencontrées. Les choix technologiques stratégiques, les algorithmes clés, et les décisions cruciales prises lors du développement seront minutieusement détaillés dans les sections pertinentes de ce rapport, permettant ainsi une compréhension approfondie de notre démarche.

En outre, ce document ne se contente pas de présenter les réussites, mais il met également en lumière les défis significatifs auxquels le groupe BAME-OCR a été confronté au cours de cette période. Qu'ils soient d'ordre technique, conceptuel ou liés à la gestion du temps, ces obstacles seront exposés de manière transparente. De manière cruciale, les solutions créatives apportées à ces défis et les enseignements tirés de ces expériences seront également présentés, offrant ainsi une perspective précieuse sur l'évolution et la croissance de notre projet.

Dans l'ensemble, ce rapport se veut une fenêtre ouverte sur notre parcours au cours de ce semestre, capturant non seulement nos succès mais aussi nos apprentissages, dans le but de fournir une compréhension holistique de l'état actuel du projet BAME-OCR.

2 Les Membres de BAME-OCR

L'équipe de BAME-OCR est composée de membres passionnés et motivés, chacun apportant une expertise unique à ce projet. En tant que collectif diversifié, nous embrassons un éventail de compétences et de perspectives qui font de notre collaboration une aventure excitante. Faisons connaissance avec les membres clés de cette équipe dévouée qui se lance dans l'aventure du projet OCR, prêts à relever de nouveaux défis et à explorer de vastes horizons de connaissances.

2.1 Brieuc CROSSON

Brieuc Crosson ou le VIM Master pour les intimes. Je programme depuis des années, et je réalise des projets en parallèle des cours sur les sujets qui me passionnent, tels que la cybersécurité, l'IA et un peu le web, je vous invite à faire votre zwords du jour sur <https://zwords.mapland.fr/>. Ainsi ce projet OCR est pour moi une expérience de plus dans mon palmarès. Contrairement à d'habitude, nous allons devoir travailler en groupe, ce qui ne me déplait pas. Au contraire, cela me permettra d'enrichir mes compétences.

2.2 Antoine LASSAGNE

Chef de groupe de BAME-OCR, l'équipe reconnaît et récompense mon implication dans le projet et voit ma passion pour ce dernier. De plus, en tant qu'ex-chef du projet Piquette, l'équipe avait confiance en mes capacités de leadership. Toujours motivé et prêt à relever des défis, je m'engage avec enthousiasme dans ce nouveau projet de groupe. Étant très disponible pour les travaux d'équipe, aider les membres est l'une des priorités essentielles pour garantir le bon déroulement et la progression continue du projet. De plus, mon super clavier me permet d'atteindre une efficacité accrue de 110 % et de réduire les fautes de frappe de 75 %. Je pense que c'est l'une des raisons pour lesquelles on m'a choisi en tant que chef, à vrai dire.

2.3 Maia BUCAMP

Je suis une travailleuse acharnée, et je n'ai pas froid aux yeux (+turtle). Dans ce projet, j'aimerais m'intéresser et découvrir de nombreux domaines différents, sans avoir peur d'apprendre et de faire des erreurs. Grâce à cela, je vais acquérir de nouvelles compétences très utiles. Je pense être à la hauteur de ce projet fort intéressant et nouveau, qui diffère considérablement de ce que je fais habituellement. Je découvre de nombreux domaines, comme les réseaux de neurones. Cela représente pour moi un défi complexe mais rempli de possibilités d'apprentissage.

2.4 Enzo JUHEL

Pas présent durant le projet jeu vidéo de Piquette, mais toujours motivé au sein du groupe, j'ai pu apporter une nouvelle vision et des techniques de travail innovantes tout en m'intégrant à leur équipe. Je change constamment de système d'exploitation, ce qui m'a permis d'acquérir des connaissances sur tous les systèmes et de développer une forte adaptabilité, une grande patience, ainsi qu'une ténacité remarquable. Cela me permet donc de maîtriser rapidement tous les sujets sur lesquels je travaille.

3 Répartition des tâches

Tâche/ Membre	BRIEUC	ANTOINE	MAIA	ENZO
Traitement d'Image		Principal		Second
Solveur de Sudoku		Principal	Seconde	
Reconnaissance des caractères	Principal		Seconde	
Responsables des tâches spécifiques et rédaction des soutenances			Seconde	Principal
Site d'internet	Principal		Seconde	

Au début du mois de septembre, l'équipe a entrepris la répartition des tâches, optant pour une approche stratégique en divisant le travail en deux parties distinctes, chacune étant placée sous la responsabilité d'un membre principal, identifié comme "Principal" sur le tableau de répartition des tâches. Par la suite, des rôles secondaires ont été attribués pour assurer l'accomplissement de toutes les tâches nécessaires au projet, tout en établissant un suivi secondaire du projet.

À mesure que le projet progressait vers sa phase finale, une réévaluation minutieuse des tâches a été entreprise, entraînant une nouvelle répartition des responsabilités au sein du groupe. Bien que les rôles principaux aient été maintenus, une flexibilité a été introduite pendant la phase d'assemblage, permettant à chaque membre de l'équipe de contribuer à toutes les parties du projet.

Cette approche collaborative a favorisé une compréhension approfondie de chaque composant du projet par tous les membres de l'équipe. En permettant à chacun de s'impliquer dans l'assemblage final, nous avons renforcé la cohésion de l'équipe et amélioré la capacité de chaque membre à comprendre l'ensemble du projet. Cette polyvalence a également offert une flexibilité précieuse pour résoudre rapidement les éventuels problèmes d'intégration et garantir un résultat final homogène. Ce ci nous a permis d'assurer également une expertise approfondie dans des domaines spécifiques, cette phase d'assemblage a créé une synergie entre les membres, favorisant une collaboration étroite et une compréhension globale du projet.

4 Traitement d'images

4.1 Prétraitement

Afin de rendre les images compatibles avec le traitement par le réseau de neurones, il est impératif de leur attribuer un format spécifique, permettant une interprétation optimale par le réseau. Dans cette optique, l'application de divers filtres s'avère essentielle, et nous détaillerons ci-dessous plusieurs types de filtres qui ont été employés.

D'abord, on distingue deux catégories majeures de filtres : les filtres pixel par pixel et les filtres de convolution. Les filtres pixel par pixel, tels que le filtre noir et blanc ou les nuances de gris, agissent individuellement sur chaque pixel de l'image. En revanche, les filtres de convolution modifient un pixel en fonction de ceux qui l'entourent, impliquant une transformation plus globale de l'image. Parmi les filtres de convolution utilisés, on compte le flou gaussien ainsi que la plupart des filtres de Canny.

Cette combinaison de filtres permet d'adapter les images de manière à ce qu'elles soient mieux interprétées par le réseau de neurones, facilitant ainsi le processus d'apprentissage et d'analyse. L'utilisation judicieuse de ces filtres contribue à améliorer la qualité des données d'entrée du réseau, renforçant ainsi sa capacité à effectuer des prédictions précises et pertinentes.

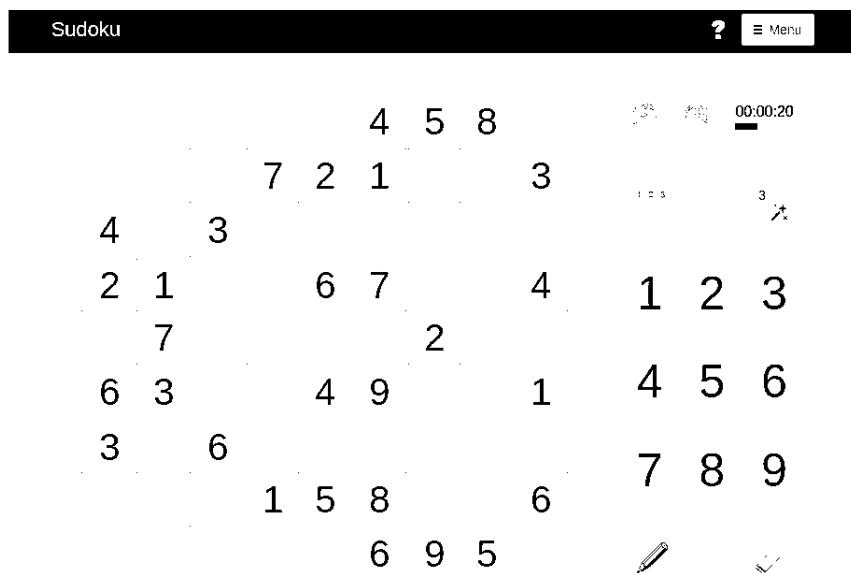
4.1.1 Suppression des couleurs

Pour optimiser la manipulation des images, il est essentiel de conserver uniquement les informations pertinentes, à savoir les droites et les caractères, qui ne requièrent pas la préservation des informations de couleur. C'est pourquoi nous procémons à une transformation de l'image en niveaux de gris, suivie d'une conversion en noir et blanc.

Le premier filtre, la conversion en niveaux de gris, vise à atténuer l'impact des couleurs en transformant tous les pixels en variations de gris. Cette opération s'effectue en ajustant directement les valeurs de rouge, vert et bleu (RGB) de chaque pixel, selon la formule suivante :

$$\text{Pixel Rouge} \times 0,3 + \text{Pixel Vert} \times 0,59 + \text{Pixel Bleu} \times 0,11$$

Par la suite, le filtre noir et blanc est appliqué dans certaines de nos fonctions spécifiques. C'est la raison pour laquelle la transformation complète de l'image n'est pas effectuée dès le début du traitement. Cette approche sélective nous permet de conserver la flexibilité nécessaire en fonction des besoins particuliers de chaque étape du processus de traitement d'image.



Conversion en noir et blanc résultat final

4.1.2 Filtre Gaussien

Le filtre Gaussien est un outil essentiel pour le floutage d'une image, visant à réduire les bruits présents et à atténuer les détails indésirables. Son mécanisme repose sur l'utilisation de matrices de convolution, permettant de déterminer la perte de précision et de détails dans l'image. Dans notre implémentation, nous avons choisi deux matrices : une de taille 5x5 et une de taille 3x3.

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Pour l'instant, nous effectuons des tests avec la matrice 3x3 et nous pensons que cela sera suffisant. Cependant, à tout moment, nous pouvons choisir d'utiliser la matrice 5x5 pour flouter davantage les images en éliminant plus de détails. Une fois que nous avons sélectionné la matrice, nous appliquons la convolution à chaque pixel de l'image. La convolution consiste à faire la somme des valeurs des pixels voisins, en utilisant les valeurs correspondantes de la matrice. Ensuite, nous plaçons le résultat obtenu dans une nouvelle image. Il est essentiel que l'image à partir de laquelle nous récupérons les valeurs soit en nuances de gris, et que nous ne la modifions pas pendant le processus, ce qui engendre des coûts en mémoire importants.



Filtre Gaussien résultat final

Actuellement, nos tests se concentrent sur l'utilisation de la matrice 3x3, et nous estimons que cela pourrait être suffisant pour nos besoins. Toutefois, nous conservons la flexibilité de choisir la matrice 5x5 à tout moment, offrant ainsi la possibilité de flouter davantage les images en éliminant encore plus de détails si nécessaire. Une fois la matrice sélectionnée, nous appliquons la convolution à chaque pixel de l'image. La convolution implique la sommation des valeurs des pixels voisins, en utilisant les valeurs correspondantes de la matrice. Le résultat obtenu est ensuite placé dans une nouvelle image.

$$\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

Matrice 5x5

Il est crucial que l'image à partir de laquelle nous récupérons les valeurs soit en nuances de gris, et que nous ne la modifions pas au cours du processus, ce qui pourrait entraîner des coûts importants en termes de mémoire. Cette précaution garantit la préservation de la qualité de l'image d'origine tout en bénéficiant des avantages du filtre Gaussien pour améliorer la robustesse du traitement d'image.

Après plusieurs tests, notre choix s'est finalement porté sur l'utilisation de la matrice 5x5 pour le filtre Gaussien. Cette décision a été prise en considération des résultats obtenus, visant à conserver un niveau adéquat de détail tout en réduisant efficacement le bruit présent dans l'image. L'application du filtre Gaussien avec une matrice 5x5 offre ainsi une solution équilibrée pour notre traitement d'image. Cette configuration particulière du filtre Gaussien s'avère particulièrement bénéfique dans le contexte du filtre de Canny, en facilitant son processus de détection de contours. En atténuant le bruit tout en préservant les détails essentiels, le filtre Gaussien prépare l'image de manière optimale pour le traitement ultérieur effectué par le filtre de Canny, renforçant ainsi la qualité globale du processus de détection de contours.

4.1.3 Filtre de Canny

Le filtre de Canny a été choisi pour mettre en évidence les informations utiles dans les images, en particulier les chiffres et les lignes. Ce filtre combine plusieurs filtres indépendants qui travaillent ensemble pour produire le résultat souhaité. Il est constitué d'un filtre de gradient d'intensité, d'un filtre d'orientation, d'un algorithme de suppression des maxima locaux, et d'un seuillage par hystérésis, que nous expliquerons dans la sous-partie suivante. Dans cette section, nous allons détailler les filtres de gradient d'intensité et d'orientation, ainsi que la suppression des maxima.

Pour commencer, nous calculons le gradient d'intensité sur une image en nuances de gris. L'objectif est de mettre en évidence les changements brusques de couleur ou de luminosité, c'est-à-dire les contours. Pour ce faire, deux matrices de convolution sont appliquées à chaque pixel de l'image. Une matrice est utilisée pour calculer le gradient le long de l'axe X, tandis qu'une autre matrice est utilisée pour le calculer le long de l'axe Y. Une fois que nous avons calculé les gradients X et Y d'un pixel, nous pouvons déterminer le gradient général en utilisant la formule suivante :

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Une matrice est utilisée pour calculer le gradient le long de l'axe X, tandis qu'une autre matrice est utilisée pour le calculer le long de l'axe Y. Une fois que nous avons calculé les gradients X et Y d'un pixel, nous pouvons calculer le gradient général en utilisant la formule suivante :

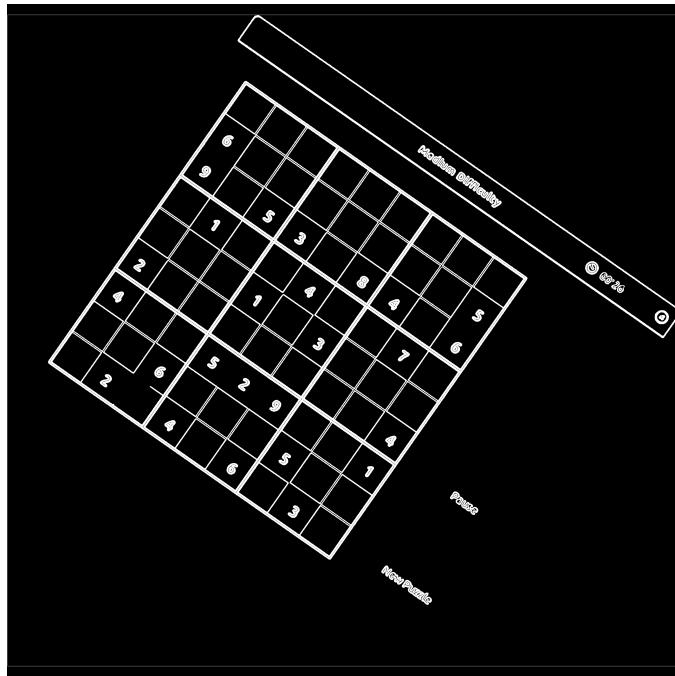
$$|G| = \sqrt{Gx^2 + Gy^2}$$

Le résultat obtenu grâce à cette formule est ensuite placé dans les composantes d'un pixel ayant les mêmes coordonnées sur une nouvelle image. L'image qui passe par ce processus devient une image en gradient d'intensité, mais elle peut encore contenir de nombreuses imperfections.

La deuxième étape consiste à effectuer le gradient d'orientation de l'image. Cela implique de déterminer un angle pour chaque pixel, ce qui sera utile lors de la suppression des maxima locaux. Pour réaliser ce filtre, nous parcourons l'image, et pour chaque pixel de l'image, nous calculons les gradients X et Y en utilisant les mêmes matrices que celles du filtre d'intensité. Cependant, à la place, nous utilisons la formule suivante :

$$\theta = \pm \arctan\left(\frac{Gx}{Gy}\right)$$

Comme pour le filtre d'intensité, nous plaçons le résultat obtenu dans une nouvelle image lors de la génération de l'image de gradient d'orientation. Une fois que nous avons calculé ces deux images à l'aide de ces filtres, nous pouvons les utiliser pour générer une image encore plus précise en supprimant les non-maximas. Cette méthode consiste à éliminer les pixels dont la valeur est inférieure à celle de leur voisin dans la direction de leur angle. Nous parcourons chaque pixel de l'image comme d'habitude, récupérons son angle à partir de la composante de l'image du filtre d'orientation, puis examinons dans cette direction sur l'image du filtre d'intensité s'il y a un voisin avec une valeur plus élevée. Si c'est le cas, nous supprimons le pixel actuel.



Grâce à ces premières étapes du filtre de Canny, nous obtenons une image beaucoup plus précise dans laquelle les bords sont plus visibles. Pour poursuivre le traitement avec ce filtre, nous devons effectuer un seuillage sur cette image. La méthode de seuillage choisie sera expliquée dans la sous-partie suivante.

4.1.4 Seuillage par hystérésis

Le seuillage par hystérésis est une technique visant à éliminer les pixels dont les valeurs se situent en dehors d'un seuil bas ou au-dessus d'un seuil haut. Après avoir appliqué le filtre de Canny à notre image, nous procédonnons au lissage du résultat en utilisant cette méthode. Pour amorcer ce processus, nous déterminons d'abord un seuil haut en employant la méthode d'Otsu. Cette méthode implique le calcul de l'histogramme de l'image, la détermination des poids initiaux, et l'exploration de tous les seuils possibles à travers l'histogramme. Nous mettons à jour les poids à chaque étape et calculons la variance de la classe, choisissant ainsi le seuil haut qui maximise la variance.

La formule littérale de la méthode d'Otsu guide cette détermination. Pour obtenir le seuil bas, nous divisons simplement le seuil haut par 2. Une fois les seuils déterminés, le programme parcourt tous les pixels de l'image. Si la valeur du pixel est supérieure au seuil haut, nous le "mettons en surbrillance" en attribuant à toutes ses composantes la valeur maximale. En revanche, si la valeur du pixel est inférieure au seuil bas, toutes ses composantes sont mises à 0. Si la valeur du pixel se situe exactement entre les deux seuils, il est marqué d'une valeur intermédiaire.

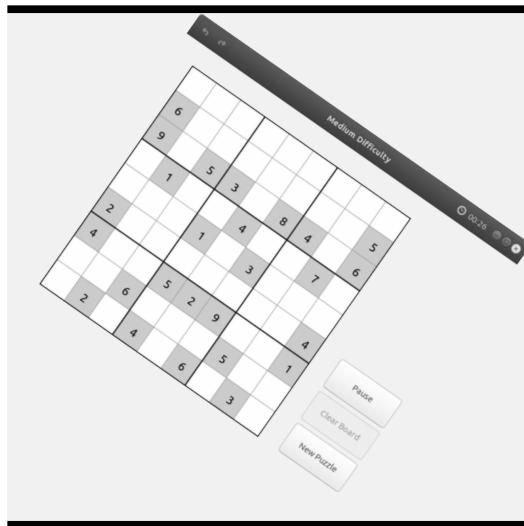
$$\sigma^2(t) = w_1(t)\sigma_1^2 + w_2(t)\sigma_2^2$$

Après le premier passage de l'image, un second passage est effectué pour éliminer les pixels marqués avec une valeur intermédiaire. Le programme effectue cette opération de manière simple : lorsqu'il rencontre un pixel intermédiaire, il vérifie s'il a des voisins maximaux. Si tel est le cas, les composantes du pixel sont mises au maximum ; sinon, elles sont mises au minimum. Une fois ces deux passages complétés, le seuillage par hystérésis est achevé, fournissant une image nette et épurée, prête à être utilisée pour la détection des contours.

4.1.5 Réduction des bruits

Le filtrage bilatéral constitue une avancée significative dans notre quête visant à réduire le bruit dans les images. Lors de la première soutenance, l'absence d'un algorithme efficace pour cette tâche constituait un défi majeur. Cependant, pour la deuxième soutenance, nous avons choisi d'adopter le filtre bilatéral, une approche qui s'est avérée être non seulement efficace, mais également exigeante en termes de temps de calcul. Néanmoins, nous sommes convaincus que son implémentation est cruciale en raison des résultats significatifs qu'elle offre.

Le filtre bilatéral, tout en utilisant le système de convolution similaire au flou gaussien ou aux différents gradients de Canny, se distingue par la particularité de son noyau adaptatif. Pour chaque pixel de l'image traitée, le filtre prend en compte divers paramètres, communément appelés sigma, afin de calculer des facteurs de distance et d'intensité. En combinant ces deux éléments, le filtre est capable de déterminer la nouvelle valeur du pixel, contribuant ainsi à l'élimination du bruit de manière efficace.



Filtre bilatéral

Bien que la différence puisse ne pas être immédiatement perceptible à l'œil nu, un examen approfondi de l'image révèle une nette amélioration. L'ensemble de l'image acquiert une texture lisse, résultant en une qualité visuelle supérieure. Cette caractéristique s'avère particulièrement avantageuse lors de l'application de techniques telles que le filtre de Canny, améliorant ainsi la détection des grilles de cases et, par extension, optimisant les performances de l'OCR dans son ensemble.

Zoomer dans l'image révèle la subtile mais significative transformation apportée par le filtre bilatéral, confirmant son rôle crucial dans notre démarche d'amélioration de la qualité des images et de l'efficacité globale du processus de reconnaissance optique des caractères (OCR). En dépit du coût en temps inhérent à cette approche, les bénéfices obtenus justifient pleinement son intégration dans notre arsenal d'outils pour le traitement d'images.

4.1.6 Résultats

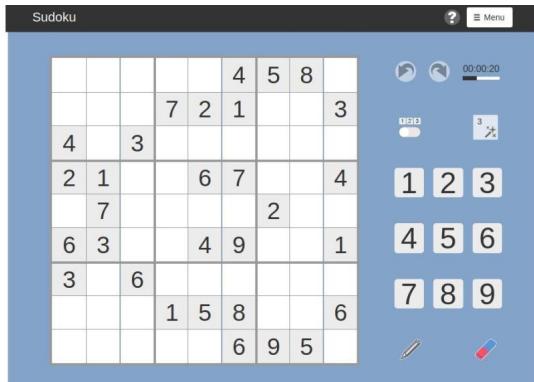
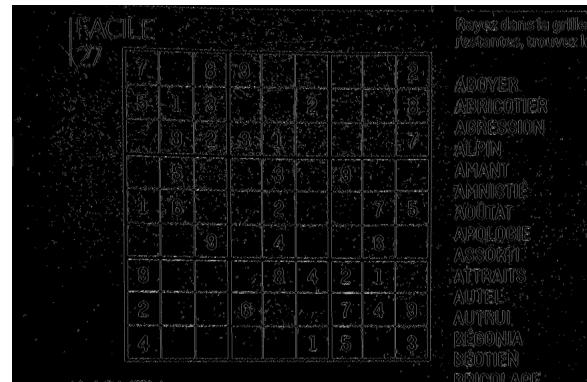
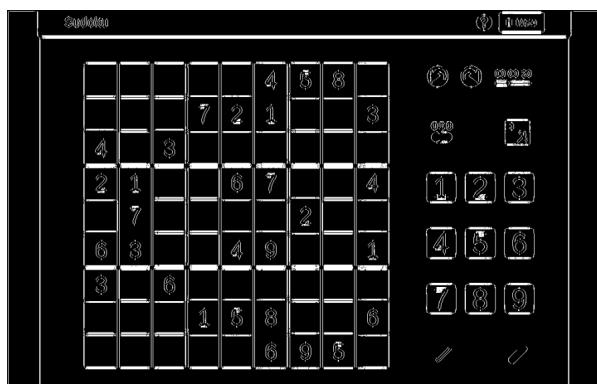


Image dans la base de données d'EPITA



Filtre de Canny avec seuillage automatique



Filtre de Canny résultat final

Le groupe se montre extrêmement satisfait des résultats obtenus avec l'application du filtre de Canny. Ce filtre a démontré une grande efficacité dans la détection des bords et des contours d'images, contribuant ainsi de manière significative à la qualité de nos traitements d'image. Les résultats obtenus avec le filtre de Canny ont dépassé nos attentes, fournissant des images nettes et précises, ce qui est essentiel pour notre projet de reconnaissance de caractères, de résolution de Sudoku et d'autres tâches liées au traitement d'images.

La précision de la détection des bords réalisée par le filtre de Canny constitue une ressource inestimable pour la suite du projet. Ces résultats seront exploités dans diverses applications, notamment dans la détection de contours de chiffres et de lignes dans les images de Sudoku, ainsi que dans la segmentation et l'identification des caractères lors de la reconnaissance de texte. En exploitant ces résultats de manière optimale, notre équipe est convaincue que nous serons en mesure de produire des solutions efficaces et précises pour les différentes tâches liées à l'OCR et au traitement d'images. Nous sommes enthousiastes à l'idée de continuer à explorer les capacités du filtre de Canny et d'améliorer les résultats des autres filtres grâce à celui-ci.

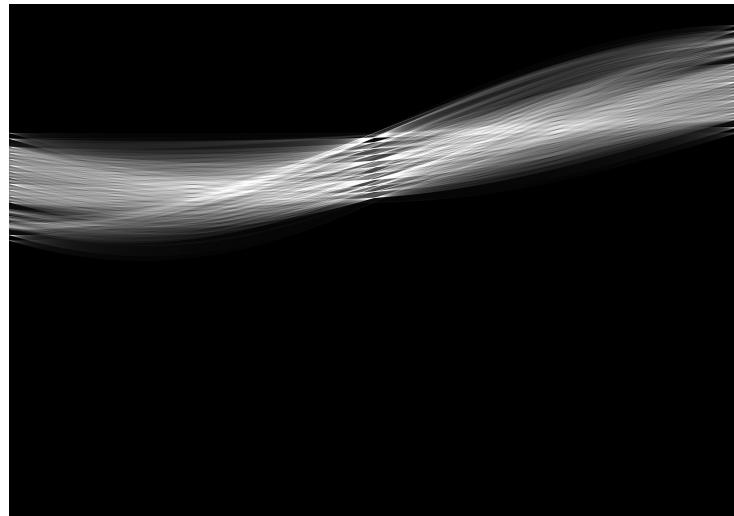
4.2 Traitement

Nous avons développé un script de rotation d'image et le script pour déterminer l'angle automatiquement. Nous utilisons des calculs de gradient pour effectuer des rotations tout en minimisant la perte d'informations. Cependant, nous avons remarqué que certaines images perdaient des informations, surtout celles dont les détails se trouvent près des bords. Pour remédier à ce problème, nous avons dû redimensionner les images en utilisant la valeur maximale entre la hauteur et la largeur. Cette approche nous a permis de préserver toutes les informations essentielles nécessaires pour le traitement de l'image. De plus pour éviter encore plus la perte d'information, nous normalisons les images dans une taille prédéfinie ce qui permet de simplifier l'utilisation de la majorité des filtres et de limiter la perte d'information lors de la rotation.

4.2.1 Transformée de Hough

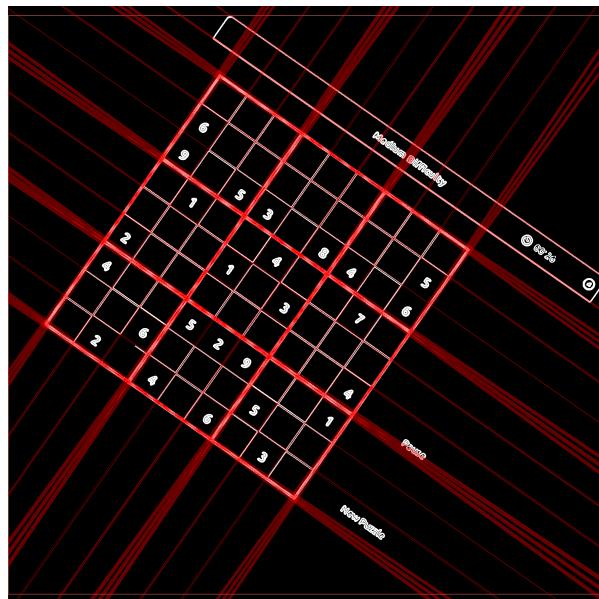
Pour cette deuxième soutenance, nous avons initialement envisagé d'implémenter une méthode simplifiée de la transformée de Hough. Cependant, après des recherches approfondies, nous avons décidé d'opter pour une approche plus standard et précise. Cette décision découle du fait que nous traiterons exclusivement des images redressées manuellement ou alignées avec les axes des abscisses et des ordonnées, facilitant ainsi la détection de droites parallèles à ces axes.

L'accumulateur dans la transformée de Hough est une matrice qui est utilisée pour accumuler les votes lors de la détection de lignes. Chaque point de l'image contribue à plusieurs lignes possibles, et l'accumulateur est utilisé pour identifier les lignes qui ont reçu le plus de votes, indiquant ainsi les lignes les plus susceptibles d'être présentes dans l'image. L'image représentante est la suivante :



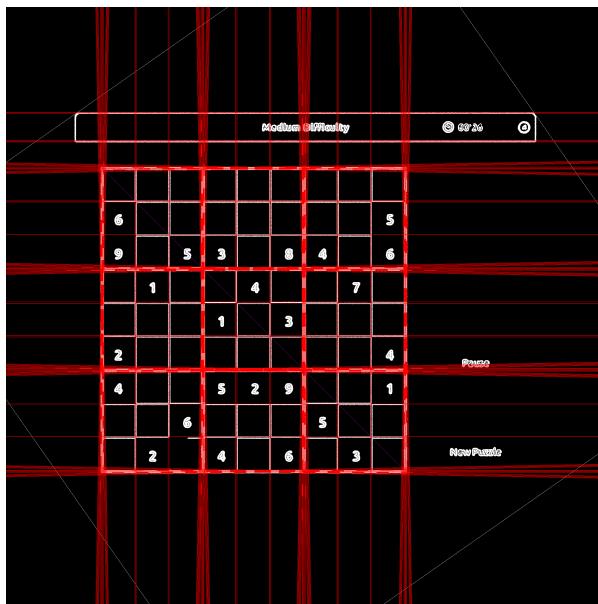
Accumulatuer normalisé

Dans cette nouvelle approche, nous avons intégré une conversion de nos droites dans l'espace de Hough. Cette fonctionnalité sera particulièrement utile pour la rotation automatique des images ou pour une détection plus précise des droites dans l'espace de l'image. Cette mise à jour permettra d'exploiter davantage les avantages de la transformée de Hough et d'améliorer la robustesse de notre solution dans la détection des lignes, renforçant ainsi la qualité globale du traitement d'image.



Transformé de Hough avant rotation

Avec des images présentant les bonnes caractéristiques, la détection des droites devient une tâche aisée. Une fois ces droites mises en évidence, le processus consiste à éliminer les éventuels doublons au sein d'une même droite. Ensuite, en calculant la moyenne des distances entre les différentes droites détectées, nous sommes en mesure de restreindre notre sélection uniquement aux droites appartenant à la grille de Sudoku. Cette étape de post-traitement vise à affiner la détection des droites en éliminant tout bruit ou doublon qui pourrait résulter de l'analyse initiale. En réduisant l'ensemble des droites détectées aux seules droites pertinentes pour la grille de Sudoku, nous optimisons la précision du processus global de détection des lignes, renforçant ainsi la fiabilité.



Transformé de Hough après rotation

En dernière étape, une fois que les droites pertinentes ont été identifiées et que la grille de Sudoku est délimitée, il ne reste plus qu'à récupérer les coordonnées des coins de chaque case pour procéder à leur découpe. Cette découpe permet de segmenter la grille en différentes images, chacune correspondant à une case individuelle du Sudoku. Ces images découpées sont alors prêtes à être traitées par notre réseau de neurones, facilitant ainsi le processus de reconnaissance des chiffres dans chaque case. Cette phase finale de découpe des cases constitue une transition importante entre la détection des lignes de la grille et l'application du réseau de neurones pour la résolution du Sudoku. Elle garantit que chaque chiffre de la grille est isolé dans une image distincte, préparant ainsi le terrain pour une reconnaissance précise des caractères.

5 Détection de la grille

La détection de la grille s'opère à travers plusieurs phases distinctes. Le groupe a décidé d'utiliser sa propre méthode de détection, se distinguant ainsi de l'adoption d'algorithmes préexistants. Cette décision découle de la clarté des objectifs poursuivis, et les membres du groupe ont suivi une voie déterminée par une idée préconçue qui s'est révélée fructueuse. Bien que le succès n'ait pas été immédiat, il s'est concrétisé peu de temps après les premiers essais.

Le processus de traitement des lignes générées par la détection consiste en une classification en deux catégories : les lignes horizontales et les lignes verticales. Cette distinction est possible en analysant les différences significatives entre les coordonnées y pour les lignes verticales et entre les coordonnées x pour les lignes horizontales. Une fois les lignes triées par orientation, une étape cruciale consiste à les organiser dans l'ordre de la grille.

Le tri ultérieur des listes vise à obtenir une séquence ordonnée des lignes, facilitant ainsi le processus de différenciation entre les lignes de la grille. La méthode utilisée pour déterminer si une ligne appartient à la même rangée que la précédente repose sur la différence de distance entre elles. Si cette différence est inférieure à un seuil prédéfini, les lignes sont considérées comme faisant partie de la même rangée. Sinon, une nouvelle rangée est identifiée. Ce processus est répété jusqu'à ce que l'on parvienne à identifier 10 lignes/rangées, puis on répart de la deuxième ligne pour voir si une nouvelle grille peut être retrouvé et ainsi de suite.

Par la suite, le choix de la grille finale se fait en comparant les différences entre les grilles retrouvées. La grille optimale est celle dont les différences entre les lignes sont quasi-égales, indiquant ainsi une régularité dans la disposition des lignes.

Ce processus est appliqué de manière similaire pour les lignes horizontales et verticales. Les coordonnées de la première et de la dernière ligne des deux orientations sont utilisées pour déterminer la diagonale de la grille, facilitant ainsi l'identification du carré de la grille.

5.1 Rotation automatique

L'algorithme de rotation automatique que nous avons développé se révèle être d'une grande efficacité, représentant une réussite majeure dans notre démarche. Son intégration avec notre algorithme de détection de lignes renforce sa performance, offrant une solution complète pour l'ajustement automatique de l'orientation des images. Le processus démarre par la réception d'une liste de points localisés sur l'image à traiter, résultat de notre algorithme de détection de lignes.

À partir de cette liste de points, l'algorithme recherche dans cette liste pour déterminer l'angle de la droite qui présente l'alignement optimal avec la grille de sudoku. Ceci est réalisé grâce à un système combinant des techniques de moyenne et de médiane. En évaluant l'orientation de chaque point par rapport à la grille, le programme parvient à calculer un angle qui, lorsqu'appliqué à l'image, la redresse de manière à ce que la grille soit parfaitement alignée.

La précision de cette approche repose sur la combinaison judicieuse de concepts statistiques tels que la moyenne et la médiane, permettant d'établir un angle représentatif de l'orientation souhaitée. Une fois cet angle déterminé, le programme récupère cette information, et l'appel la fonction de rotation, déjà présente depuis la première soutenance, permet d'ajuster l'image selon l'angle calculé.

Cette stratégie de rotation automatique constitue un ajout significatif à notre ensemble d'outils, améliorant non seulement la présentation visuelle des images, mais également contribuant à une optimisation générale du processus d'OCR. La synergie entre notre algorithme de détection de lignes et celui de rotation automatique démontre notre engagement continu à perfectionner chaque étape du traitement d'images dans le contexte spécifique de la reconnaissance optique des caractères.

5.2 Détection de toutes les cases de la grille

À la suite de l'identification de la grille suivie de l'étape de rotation, le calcul des dimensions des cases s'est avéré relativement aisé. En effet, il a simplement suffi de diviser la diagonale de la grille par neuf. Cette démarche nous a permis d'obtenir les longueurs des diagonales de chaque carré inscrit dans la grille. En ayant connaissance de la largeur de chaque case, il a été possible de déterminer les positions des colonnes en effectuant des décalages appropriés.

5.2.1 Tri des cases

Après la détection de chaque case, elles sont classifiées en deux catégories distinctes : les "cases vides" et les "cases non-vides". Les cases non-vides sont celles qui arborent des numéros compris entre 1 et 9. Cette classification s'opère au moyen d'un filtre de Balance, suivi d'une analyse pixel par pixel de la case, visant à déterminer le pourcentage de pixels blancs. Si ce pourcentage est inférieur à un tiers de la totalité de l'image, nous considérons la case comme étant vide.

Ainsi, les cases identifiées comme vides sont représentées par le chiffre 0 dans la grille de sudoku, tandis que les cases non-vides sont soumises au réseau de neurones afin de récupérer le numéro correspondant. Dans le cadre de notre solution OCR pour la résolution de sudoku, nous utilisons les labels suivants pour représenter l'état des cases de la grille :

Empty (Vide) : Représenté par le chiffre 0, indique qu'une case est actuellement vide et doit être remplie.



Case vide-Empty-0

Full (Pleine) : Représenté par le numéro après traitement par l'IA, indique qu'une case est déjà remplie avec un numéro spécifique après le passage de notre algorithme de résolution.



Case non-vide-Full-6



Case non-vide-Full-9

Ces labels facilitent la compréhension de l'état actuel de chaque case dans la grille de sudoku, distinguant les cases vides de celles qui ont été remplies par notre solution basée sur l'intelligence artificielle.

6 Découpage de la grille

Suite à la détection de chaque case, après le tri entre les cases vides et non vides, elle est découpée, et des ajustements, que nous aborderons, sont effectués pour le traitement par le réseau de neurones.

6.1 Adaptation pour le reseau de neurones

6.1.1 Redimension

La redimension des cases découpées sur la grille est une étape cruciale dans le processus de reconnaissance de chiffres. En utilisant les coordonnées de chaque case de la grille, nous simplifions ce processus en sauvegardant chaque case avec une résolution prédéfinie. Dans notre cas, la reconnaissance des chiffres s'effectue à l'aide d'images de résolution 28 par 28 pixels. Ainsi, les cases découpées sont redimensionnées en respectant ces dimensions avant d'être enregistrées dans un dossier dédié à l'OCR pour un traitement ultérieur par le reste du programme.

Cette approche de redimensionnement uniforme garantit une cohérence dans la taille des images, ce qui est essentiel pour le bon fonctionnement de l'algorithme de reconnaissance. En standardisant la résolution à 28 par 28 pixels, nous facilitons le traitement et l'analyse des chiffres contenus dans chaque case. Ce procédé contribue à améliorer la précision globale du système d'OCR en fournissant des données d'entrée uniformes et bien définies.

Le dossier dédié à l'OCR devient ainsi une ressource organisée, contenant des images normalisées prêtes à être traitées par les étapes suivantes du programme. Cette approche méthodique facilite également la gestion et la maintenance du système, car elle offre une structure claire pour le stockage des données intermédiaires.

6.1.2 Centrage des numéros

Afin d'optimiser la performance du réseau de neurones, chaque numéro de chaque case est centré en détectant la partie de l'image qui présente le plus de blanc. Pour ce faire, on utilise un filtre de flou afin de détecter le tas de pixels blancs. Ensuite, un recadrage supplémentaire est effectué sur la zone ainsi identifiée, garantissant ainsi un meilleur positionnement des caractères pour une reconnaissance plus précise.

7 Reconnaissance de caractères

La reconnaissance de caractères repose sur un réseau de neurones d'apprentissage profond que nous avons choisi de rendre aussi générique que possible. Cela signifie qu'il peut être entraîné pour résoudre des problèmes plus simples que la reconnaissance d'images, ce qui facilite le développement.

7.1 Structure

Un réseau de neurones d'apprentissage profond est composé de couches de neurones, comprenant une couche d'entrée, une couche de sortie et des couches intermédiaires. Il fonctionne en deux étapes : tout d'abord, un algorithme de propagation permet au réseau de faire des prédictions sur un ensemble de données d'entrée. Ensuite, un algorithme de rétropropagation entraîne le réseau à produire des prédictions plus précises.

7.1.1 Propagation

L'algorithme de propagation commence par remplir la couche de neurones d'entrée avec nos valeurs. Ensuite, il propage ces valeurs à travers chaque couche du réseau jusqu'à la couche de sortie en utilisant des poids liant chaque paire de neurones. De plus, des fonctions telles que la fonction sigmoïde sont appliquées sur les neurones pour éviter des valeurs trop importantes.

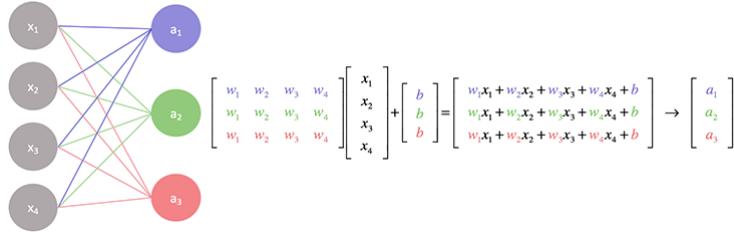


Image de "Neural Networks : representations"-Jeremy Jordan

7.1.2 Rétropropagation

L'algorithme de rétropropagation commence par appliquer l'algorithme de propagation. Ensuite, il compare les résultats obtenus avec les valeurs attendues et calcule les ajustements nécessaires sur les poids entre les neurones pour se rapprocher des valeurs attendues. Ces ajustements sont ensuite appliqués avec un facteur de force souvent appelé "learning rate". L'algorithme est répété sur un grand ensemble de données jusqu'à ce que les ajustements deviennent très petits. Pour améliorer les performances de cet algorithme, nous avons mis en place plusieurs techniques, notamment la moyenne des ajustements sur un lot de données avant leur application, ainsi que l'ajout d'une inertie entre les différentes répétitions de l'algorithme.

7.2 Entraînement sur Image

L'objectif principal du réseau est de reconnaître le chiffre écrit sur une image. Pour faciliter cette tâche, toutes les images sont redimensionnées à la même taille et converties en échelle de gris. La couche d'entrée est composée de neurones associés à chaque pixel de l'image. Par exemple, pour une image de 28 par 28 pixels, nous avons 784 neurones dans la couche d'entrée. La couche de sortie est constituée de 10 neurones, représentant les chiffres de 0 à 9. Pour lire le résultat, nous choisissons le neurone avec la valeur la plus élevée. Après de nombreux tests, nous avons conclu qu'un réseau possédant 3 couches cachées de 40 à 100 neurones chacunes nous donne les meilleurs résultats. Pour plus facilement entraîner sur une grande diversité de données, le CLI du réseau de neurones permet de charger et d'entraîner sur un infinité de bases de données (dans la limite de la mémoire RAM de l'ordinateur). Cette dernière fonctionnalité requiert un format de base de données standard, on a alors choisi de garder le format de la base de données Mnist de Yann Lecun.

7.3 Base de donne d'image

Dans un premier temps, nous avons entraîné le réseau sur la base de données MNIST de Yann LeCun. Un script spécial a été développé pour charger cette base de données. Bien que cette base permette d'obtenir un réseau performant, les images qu'elle contient ne sont pas très représentatives de celles reçues lors de l'utilisation du logiciel. Par conséquent, nous avons ré-entraîné le réseau sur un jeu de données provenant de la partie traitement d'images pour obtenir de meilleurs résultats, cette pratique étant appelée le "fine-tuning".

Les fondements de nos bases de données sont issus d'un script capable de générer des images de 28 par 28 pixels, parfaitement conformes aux sorties de la partie image processing. Ce script se base sur des centaines polices d'écriture libre récupérées sur internet et dans les librairies linux. À ce jour, notre plus vaste base de données englobe environ 800 000 images d'une grande diversité, certaines comportant du bruit, des imperfections ou encore du flou, tandis que d'autres demeurent impeccables ; la taille et la position des chiffres varient aussi. Cette diversité constitue une ressource précieuse, permettant au réseau de maintenir un caractère généraliste et d'éviter une simple mémorisation des images fournies. En exploitant cette variété, notre réseau demeure apte à généraliser ses connaissances et à traiter efficacement des données nouvelles et variées. Pour rendre ces nouvelles images utilisables par le réseau de neurones, il faut, comme dit précédemment, le convertir dans le format de la base de données Mnist. Cette tâche nous a alors demandé d'écrire un sous programme de conversion de format.

8 Résolveur de sudoku

8.1 Solubilité du sudoku

Nous vérifions la solubilité de la grille de sudoku directement dans le programme à l'aide d'une fonction qui, avant de commencer la résolution, vérifie si la case actuelle peut être résolue. En d'autres termes, elle vérifie s'il n'y a pas de doublon sur la ligne, la colonne ou la zone 3x3 concernée. Si la grille est soluble, le programme continue normalement. En revanche, si la grille est insoluble, le programme s'arrête, renvoie un code d'erreur, et affiche un message dans le terminal indiquant que la grille est impossible à résoudre, avant d'arrêter complètement l'exécution du programme.

8.2 Solveur

Pour résoudre le sudoku, nous utilisons une méthode naïve appelée "backtracking". Profitant de la rapidité d'exécution du langage C et de la simplicité de la tâche, qui ne nécessite pas une optimisation complexe en raison du caractère "brute force" de la méthode, nous avons mis en œuvre cette approche. Le principe de résolution est simple : en utilisant un parseur, nous récupérons la grille à partir d'un fichier texte.

Ensuite, le programme parcourt cette grille qui est maintenant stockée dans un tableau à deux dimensions. Pour chaque case, une vérification est effectuée pour déterminer si elle est solvable. Si c'est le cas, le programme attribue une valeur à la case en utilisant la première solution possible. Il peut également ajuster la valeur d'une case déjà remplie si nécessaire. Pour le rendu final, nous envisageons d'optimiser davantage le fonctionnement de cette méthode.

8.3 Sauvegarde des résultats

Actuellement, la grille est sauvegardée de manière simple dans un fichier texte. Cependant, pour la version finale du projet, nous avons prévu une amélioration significative en envisageant de sauvegarder la grille sous forme d'image, conformément à nos intentions initiales. Cette évolution contribuera à offrir une expérience utilisateur plus complète et visuellement attrayante.

9 Interface graphique

9.1 Terminal

Lors de la première soutenance, notre solution OCR ne disposait pas d'un cheminement complet permettant d'utiliser l'ensemble des programmes nécessaires. Toutefois, pour la deuxième soutenance, nous avons mis en place une intégration complète de tous les programmes essentiels à la résolution des grilles de sudoku. Pour cela, la compilation s'effectue en utilisant la commande "make", et l'exécution requiert le lancement du programme nommé "BAME". En cas d'exécution sans argument, un texte informatif s'affiche pour guider l'utilisateur sur la manière d'utiliser le programme.

Après avoir saisi le chemin vers votre image en entrée, vous avez la possibilité de spécifier un chemin vers un réseau de neurones à utiliser. Si aucun chemin n'est fourni, le programme utilisera un réseau préalablement entraîné par notre équipe. Nous allons à présent détailler le fonctionnement des différentes interfaces, expliquant comment utiliser indépendamment les filtres, ainsi que le processus d'entraînement et de test des réseaux de neurones. Ces fonctionnalités offrent une flexibilité accrue aux utilisateurs pour adapter notre solution à leurs besoins spécifiques.

Pour utiliser l'interface, il vous suffit d'exécuter la commande "make interface" pour compiler les composants nécessaires, puis de lancer le programme avec la commande "./interface". Cette séquence simplifiée permettra de mettre en place l'interface et de l'exécuter pour une utilisation immédiate. Assurez-vous que toutes les dépendances et bibliothèques nécessaires sont correctement installées avant d'effectuer ces étapes pour garantir un fonctionnement optimal de l'interface.

9.1.1 Utilisation indépendante des filtres

Pour utiliser les filtres de manière indépendante, procédez comme suit : compilez le programme situé dans le sous-dossier "ImageProcessing" en utilisant la commande "make", puis exécutez l'exécutable nommé "ImageProcess". En lançant ce programme, plusieurs options seront disponibles, que vous pouvez découvrir en utilisant la commande "help". Pour une utilisation de base, il vous suffit de spécifier en argument le chemin vers l'image d'origine et le chemin vers l'image de destination.

Si vous souhaitez observer le fonctionnement des filtres de manière isolée, vous pouvez également préciser le nom du ou des filtres que vous souhaitez visualiser directement dans le terminal, au sein du programme de prétraitement des images. Du point de vue de la programmation, à l'exception de l'initialisation de la bibliothèque SDL, il s'agit simplement de réorganiser les arguments dans le bon ordre au sein d'une boucle parcourant tous les arguments. Cette approche offre à l'utilisateur la possibilité de tester tous les filtres selon ses préférences, offrant une flexibilité totale dans le processus d'exploration des fonctionnalités de prétraitement des images.

9.1.2 Utilisation du réseau de neurones

Pour utiliser exclusivement le réseau de neurones, suivez les étapes suivantes : accédez au sous-dossier "NeuralNetwork" et compilez le programme en utilisant la commande "make". Ensuite, exécutez le programme nommé "mnist". Une fois dans l'interface en ligne de commande (CLI), vous pouvez utiliser la commande "help" pour découvrir l'ensemble des options disponibles.

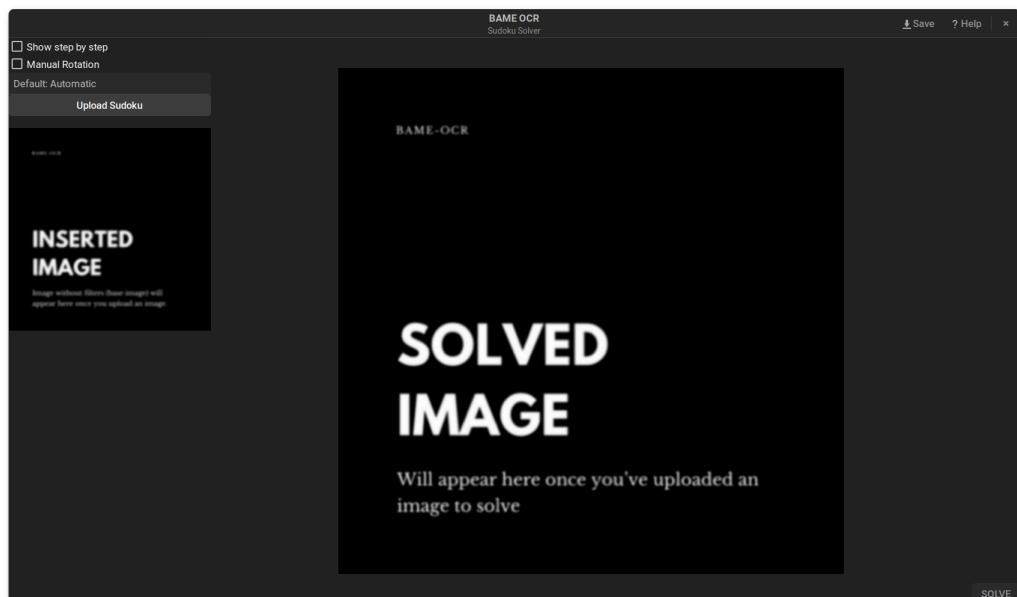
Vous avez la possibilité de créer un nouveau réseau en spécifiant le nombre de neurones et le nombre de couches cachées que vous souhaitez. Ensuite, vous pouvez entraîner votre réseau en utilisant différentes bases de données disponibles au téléchargement dans le dossier "dataset". Vous avez également la possibilité de tester votre réseau et de le sauvegarder si ses performances vous conviennent. Une fois sauvegardé, vous pouvez utiliser ce réseau pour résoudre des sudokus avec notre solution OCR.

Cette approche modulaire offre une flexibilité considérable, permettant aux utilisateurs d'adapter le réseau de neurones en fonction de leurs besoins spécifiques et d'exploiter ses fonctionnalités pour résoudre efficacement les grilles de sudoku.

9.2 GTK

L'un des objectifs primordiaux pour la version finale du projet consistait à développer une application dotée d'une interface graphique (GUI) conviviale en utilisant la puissante bibliothèque GTK. Cette décision stratégique vise à améliorer considérablement l'accessibilité du logiciel, notamment pour les utilisateurs qui ne sont pas familiers avec les interfaces en ligne de commande. En outre, l'ajout d'une interface graphique apportera une dimension esthétique à l'application, créant ainsi une expérience utilisateur plus attrayante et intuitive.

La mise en place d'une interface graphique GTK présente de nombreux avantages. Elle permet aux utilisateurs de manipuler le logiciel de manière visuelle, en interactuant avec des éléments graphiques, des boutons, des menus, et d'autres éléments, facilitant ainsi la prise en main de l'application. Cela ouvre également de nouvelles possibilités pour l'expansion future de fonctionnalités, car une interface graphique offre une plateforme plus flexible pour l'ajout de nouvelles fonctionnalités.

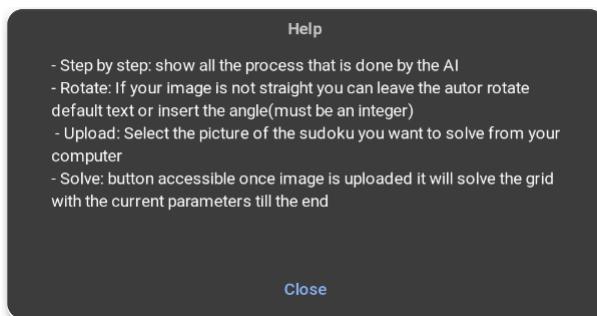


Notre interface a été réalisée avec la librairie GTK et le logiciel d'édition d'interface graphique Glade. La combinaison des 2 nous a permis une conceptualisation rapide et simplifiée de la création de l'interface, avec la position des éléments tels que les images, boutons et champ de texte dans Glade. Ensuite, nous n'avions plus qu'à les lier avec la librairie GTK pour gérer les différentes interactions avec les boutons et les mises à jour des images au fur et à mesure de l'utilisation de l'OCR.

9.2.1 Barre de Titre



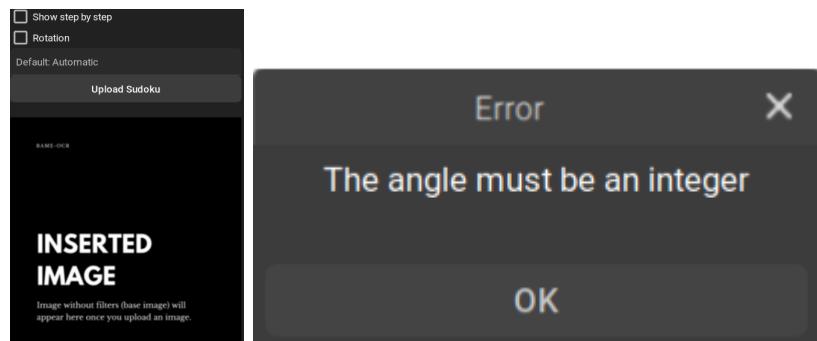
L'interface de l'application se déploie simplement en exécutant le programme principal, disponible dans le dossier "src". À son lancement, l'utilisateur est accueilli par un titre affirmant qu'il se trouve dans l'application graphique. À droite de l'écran, des boutons "Save", "Help", et des boutons système sont accessibles. Le bouton "Save" permet de sauvegarder l'image de la grille résolue, tandis que le bouton "Help" déclenche l'affichage d'une pop-up présentant les instructions détaillées pour l'utilisation de l'OCR.



Popup du bouton d'aide

9.2.2 Rotation et "Step by step"

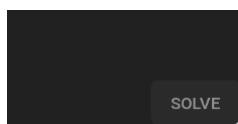
À gauche de l'interface, plusieurs options sont mises à la disposition de l'utilisateur. Le bouton "Upload" facilite le processus en ouvrant une fenêtre où l'utilisateur peut sélectionner la photo de la grille de sudoku à analyser. Cependant, si l'angle est mal indiqué, il est possible de faire apparaître une pop-up pour nous indiquer de corriger l'angle renseigné car ce doit être un entier.



À côté, une case à cocher "Rotation" offre la possibilité de spécifier manuellement un angle de rotation, sinon, celui-ci sera déterminé automatiquement. Une autre option, la case à cocher "Step by Step", permet de visualiser le processus étape par étape, depuis le prétraitement jusqu'à l'écriture des chiffres sur la grille.

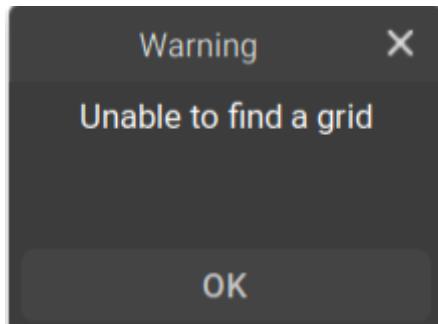
9.2.3 Résolution des sudoku

Finalement en bas à droite de l'interface se trouve le bouton "Solve". Qui exécute l'OCR en fonction des paramètres indiqués précédemment. Si aucune solution n'est trouvée en raison d'une défaillance de l'OCR, un pop-up offre la possibilité de modifier le fichier texte contenant la grille détectée. Cela donne à l'utilisateur la possibilité de corriger manuellement le traitement de l'OCR pour obtenir une solution, qui sera ensuite affichée dans la fenêtre de solution.



Bouton de lancement de programme sur l'image

En cas d'incapacité à détecter la grille ou à clarifier les numéros, et si le Sudoku Solver n'est pas en mesure de résoudre la grille, une fenêtre contextuelle d'erreur s'affichera avec le message suivant : "Unable to find grid", comme illustré dans l'image ci-dessous.



Cette notification claire informe l'utilisateur que le système a rencontré des difficultés lors de la détection de la grille ou de l'interprétation des numéros présents. Elle encourage l'utilisateur à vérifier et ajuster l'image d'entrée pour une meilleure lisibilité, ou à explorer d'autres options pour résoudre le problème. Cette approche transparente et informative vise à améliorer l'expérience utilisateur en fournissant des indications claires sur la nature de l'erreur rencontrée.

9.3 Sauvegarde des résultats

Après l'exécution réussie du programme, la grille de sudoku résolue est présentée au centre de l'application, avec les chiffres générés s'affichant en vert pour indiquer que la solution a été trouvée. Cette représentation visuelle offre à l'utilisateur une manière claire de visualiser le résultat du processus de résolution du sudoku. Cependant, dans le cas où la solution n'est pas trouvée, une fenêtre pop-up informative apparaît, donnant à l'utilisateur la possibilité de modifier le fichier texte contenant la grille initiale, détectée par l'OCR. Cette flexibilité permet à l'utilisateur d'apporter des ajustements manuels, si nécessaire, afin d'aider le programme à parvenir à une résolution.

Finalement, pour conserver la grille de sudoku résolue en tant qu'image, l'application propose un bouton "Save" positionné en haut à droite de l'écran. L'utilisateur peut simplement cliquer sur ce bouton pour sauvegarder l'image affichée à l'écran, fournissant ainsi une copie visuelle de la solution. Cette fonctionnalité offre une façon pratique d'archiver ou de partager les résultats avec d'autres utilisateurs sans avoir besoin de recourir à des captures d'écran externes. En combinant l'affichage clair de la solution avec la possibilité de sauvegarder l'image, l'application assure une expérience utilisateur complète et fonctionnelle pour ceux qui souhaitent résoudre et enregistrer leurs grilles de sudoku.

10 Site internet

Lorsque vous accédez à notre site web, vous êtes accueilli par le titre imposant "BAME-OCR" en rouge vif. En poursuivant votre descente sur la page, vous découvrirez une brève explication de notre solveur de sudoku, BAME-OCR, décrivant succinctement son fonctionnement.

Juste en dessous, vous trouverez une section dédiée aux exemples, mettant en avant des grilles résolues avec et sans rotation, illustrant ainsi la puissance de notre solution. Cette démonstration visuelle vous donne un aperçu immédiat de l'efficacité de BAME-OCR dans la résolution de sudokus variés.

Plus bas sur la page, des captures d'écran de l'interface utilisateur de BAME-OCR sont présentées, offrant un aperçu de son design convivial. Vous pourrez ainsi vous familiariser avec notre interface avant même de télécharger l'application.

En continuant la descente, vous trouverez le lien pour télécharger et d'installer BAME-OCR, assurant une mise en place facile et efficace de notre solveur de sudoku sur votre appareil.

Enfin, à la fin du site, vous rencontrerez les auteurs de BAME-OCR, chacun accompagné de liens de contact vers leurs profils respectifs. Cette section permet de mettre en lumière l'équipe passionnée derrière le développement de cette solution innovante, offrant également une opportunité de vous connecter avec eux.

Explorez notre site pour une plongée captivante dans l'univers de BAME-OCR, découvrez "Notre Solver de Sudoku par IA". Plongez dans les mécanismes complexes de notre solution de résolution de sudoku alimentée par l'intelligence artificielle. Apprenez comment notre algorithme innovant interprète les grilles, résout les puzzles avec précision.

Le site internet n'étant qu'une page vitrine, celui-ci est statique et composé de simple html et css. Pour embellir le site et le rendre adaptatif à tout type d'écran de toute taille - ordinateur, téléphone, télévision ... - le site de BAME-OCR utilise la librairie Bootstrap 5.3.2 , très réputée pour sa gestion des tailles d'écran. Cette librairie est issue du réseau social X (anciennement Twitter).

LIEN : <https://briossant.com/BAME-OCR/>

11 Gestion du Projet

Le groupe BAME-OCR se félicite d'avoir établi une gestion du temps particulièrement efficace tout au long du trimestre, permettant des progrès satisfaisants dans le projet. Malgré la complexité inhérente au projet, l'équipe a réussi à accomplir les tâches requises, même si le programme ne fonctionne pas de manière optimale pour tous les types d'images.

La reconnaissance de la complexité du projet et les défis rencontrés dans l'adaptation du programme à différents types d'images soulignent l'engagement et la persévérance du groupe. Même face à des obstacles, l'équipe a pu fournir des résultats concrets, démontrant ainsi sa capacité à surmonter des défis techniques et à progresser malgré des contraintes potentielles.

Il est important de reconnaître et de célébrer les réalisations accomplies jusqu'à présent, tout en restant conscient des opportunités d'amélioration et d'optimisation du programme pour un fonctionnement plus robuste sur divers types d'images. Cette capacité d'adaptation et d'apprentissage continu est une qualité précieuse dans la réalisation de projets complexes.

11.1 Avancée du projet

Le groupe BAME-OCR célèbre avec satisfaction la réussite totale du projet au cours des huit dernières semaines, témoignant d'une gestion du temps exceptionnellement efficace. Chaque tâche a été menée à bien, avec un pourcentage d'achèvement de 100% pour l'ensemble des éléments du projet.

Les avancées dans le traitement d'images, la résolution du Sudoku, et la reconnaissance de caractères ont atteint leur achèvement maximal, démontrant l'expertise et l'engagement du groupe dans la réalisation de ses objectifs.

Le site web, bien que considéré initialement comme optionnel, a également été développé avec soin et a atteint un niveau de 100% d'achèvement. L'interface a été intégrée de manière à garantir une pleine fonctionnalité du projet, offrant une expérience utilisateur complète et soignée.

La réussite globale du projet ne relève pas seulement de l'efficacité dans la gestion du temps, mais également du dévouement méthodique de chaque membre du groupe. Toutes les composantes, qu'il s'agisse du traitement d'images, du Sudoku Solver, de la reconnaissance de caractères, ou du site web, reflètent l'excellence et la qualité dans leur exécution. Le groupe reste ouvert à d'éventuelles améliorations, mais la satisfaction règne quant à l'atteinte des objectifs fixés initialement.

11.2 Organisation du groupe

La première phase de notre projet a impliqué une division claire en deux grandes parties, avec une équipe se concentrant sur la manipulation et le traitement des images, tandis que l'autre se consacrait au réseau de neurones. Cette approche nous a permis de définir des missions spécifiques, d'établir des objectifs clairs et de travailler de manière plus ciblée. Cette division des tâches a favorisé une collaboration efficace au sein de chaque équipe, avec une compréhension approfondie des domaines respectifs.

Cependant, lors de la phase finale du projet, nous avons constaté que cette approche n'était pas adaptée à toutes les situations, notamment lorsqu'il s'agissait d'assembler différentes parties du projet. C'est alors que nous avons opté pour le pair programming. Lors des tâches impliquant l'assemblage de plusieurs composants, les quatre membres de l'équipe se sont réunis autour d'un écran, échangeant régulièrement le clavier entre eux. Cette approche a permis une collaboration en temps réel, favorisant la résolution immédiate des problèmes et garantissant une cohérence dans l'ensemble du projet.

Pour des tâches plus précises et moins complexes, nous avons choisi de diviser l'équipe en deux, en tenant compte des compétences et des préférences de chacun. Cela a permis une approche plus spécialisée, maximisant l'efficacité de chaque membre dans des domaines spécifiques.

Il est important de souligner que le succès de ce projet n'a pas été le fruit du hasard, mais le résultat d'un travail méthodique de la part de chaque membre de l'équipe. La communication transparente, la collaboration étroite et l'adaptabilité ont été les piliers de notre réussite.

11.2.1 Outils de travail

Nous avons intégré efficacement de nombreuses fonctionnalités de GitHub dans notre processus de travail, notamment la gestion de projet à l'aide d'une liste de tâches réparties en plusieurs catégories, en fonction des priorités du groupe. Cette approche a considérablement simplifié la coordination de nos efforts et nous a permis de suivre de manière transparente l'avancement du projet.

Parallèlement, nous avons établi un système de communication robuste en utilisant Discord comme plateforme principale. Sur Discord, plusieurs canaux dédiés ont été mis en place pour chaque responsable, permettant ainsi un partage organisé de nos recherches et progrès. Cette structure a favorisé une communication claire et efficace au sein du groupe.

Pour le développement de l'interface, nous avons opté pour l'utilisation de l'outil Glade. Cet outil s'est avéré essentiel pour la conception de l'interface utilisateur, offrant une interface graphique conviviale pour la création et la personnalisation des éléments visuels. Grâce à Glade, nous avons pu simplifier le processus de développement de l'interface et assurer une expérience utilisateur optimale.

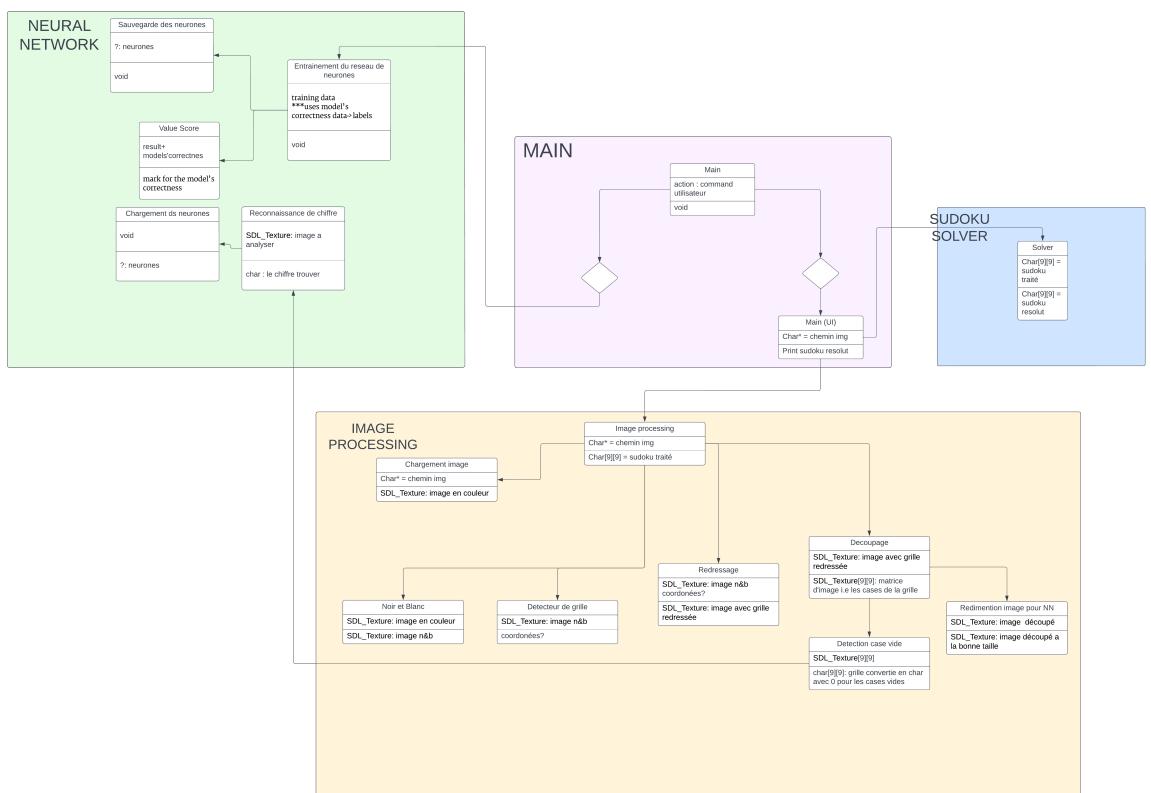
En outre, nous avons continué à partager régulièrement des liens vers les ensembles de données que nous utilisions, facilitant ainsi l'accès aux ressources nécessaires pour notre travail. Cette transparence dans le partage des ressources a renforcé la collaboration au sein du groupe, permettant à chaque membre de travailler de manière autonome tout en restant aligné sur les objectifs du projet.

11.2.2 UML

La réalisation d'un diagramme UML s'est avérée être une décision stratégique cruciale pour notre groupe, offrant une feuille de route claire et structurée dès le début du projet. Ce schéma a joué un rôle essentiel en nous aidant à visualiser l'architecture globale du système, à comprendre les interactions entre ses différentes composantes, et à préciser les étapes nécessaires à la concrétisation du projet. Il a grandement facilité la communication au sein de l'équipe en fournissant un langage visuel commun pour discuter des concepts et des objectifs du projet, tout en permettant une répartition efficace des tâches en identifiant clairement les responsabilités de chaque membre. Ainsi, il a contribué à la coordination harmonieuse de notre équipe, à une meilleure gestion des ressources, et à une compréhension partagée de l'ensemble du projet, favorisant ainsi son avancement en toute fluidité.

Le respect du format de l'UML a perduré tout au long du projet, et il s'est avéré être d'une utilité considérable. Cette approche méthodique a été particulièrement bénéfique étant donné que chaque membre de l'équipe a contribué à toutes les phases du projet, que ce soit dans le traitement d'images, le réseau de neurones, l'interface utilisateur, ou même le site Internet associé. L'utilisation de l'UML a joué un rôle fondamental dans la communication interne de l'équipe, favorisant une compréhension partagée des différents programmes et modules développés par chaque membre.

Cette cohérence dans l'application de l'UML a facilité la collaboration entre les membres de l'équipe, permettant une vision holistique du projet. En somme, l'intégration réussie de l'UML a renforcé la collaboration et a été un élément clé dans la réussite de notre projet multidisciplinaire.



12 Conclusion

Dans son ensemble, le travail accompli par le groupe se révèle extrêmement satisfaisant, et le bilan est résolument positif. Les progrès réalisés sont en parfaite adéquation avec nos attentes initiales pour ce projet ambitieux. Le Sudoku Solver se montre opérationnel, démontrant son efficacité avec les cas idéaux ainsi qu'avec des images de sudoku de haute qualité. En ce qui concerne l'implémentation du XOR, les résultats obtenus sont en parfaite conformité avec nos objectifs préétablis.

Malgré la persévérance dont nous pourrions encore faire preuve pour affiner certains aspects, le projet est considéré comme complet, répondant aux critères et aux éléments requis. L'interface fonctionne de manière satisfaisante, conforme à ce qui avait été spécifié dans le cahier des charges. Bien que quelques ajustements et perfectionnements restent envisageables, regrettablement, le temps imparti ne nous a pas permis de concrétiser ces améliorations. Néanmoins, cette réalisation représente une avancée significative dans notre compréhension et notre application des concepts liés à la détection de grilles de sudoku, réseaux de neurones, traitement d'images et le travail en équipe. L'équipe reste ouvert à de futures améliorations et à des développements potentiels pour renforcer davantage la robustesse et la polyvalence de notre solution.

13 Sources

- <https://zestedesavoir.com/tutoriels/1014/utiliser-la-sdl-en-langage-c/>
- <https://towardsdatascience.com/how-neural-networks-solve-the-xor-problem-59763136bdd7>
- <https://www.youtube.com/watch?v=aircAruvnKk&t>
- <https://www.youtube.com/watch?v=tIeHLnjs5U84>
- <https://www.gtk.org/>
- <https://youtu.be/8H2ODPNxEgA?si=iv0V6ZSBSB2IqKc9>
- <https://www.jeremyjordan.me/intro-to-neural-networks/>