
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 | 3 |
| 2.3 | Maia BUCAMP | 4 |
| 2.4 | Enzo JUHEL | 4 |
| 3 | Répartition des tâches | 4 |
| 4 | Traitement d'images | 5 |
| 4.1 | Prétraitement | 5 |
| 4.1.1 | Nuance des gris | 5 |
| 4.1.2 | Filtre Gaussien | 5 |
| 4.1.3 | Filtre de Canny | 6 |
| 4.1.4 | Seuillage par hystérésis | 7 |
| 4.1.5 | Réduction des bruits | 7 |
| 4.1.6 | Résultats | 8 |
| 4.2 | Traitement | 9 |
| 5 | Réconnaissance de caractères | 9 |
| 5.1 | Structure | 9 |
| 5.1.1 | Propagation | 10 |
| 5.1.2 | Rétropropagation | 10 |
| 5.2 | Problème du XOR | 10 |
| 5.3 | Entraînement sur Image | 10 |
| 5.4 | Base de donne d'image | 11 |
| 6 | Résolveur de sudoku | 11 |
| 6.1 | Solubilité du sudoku | 11 |
| 6.2 | Solveur | 11 |
| 6.3 | Sauvegarde des résultats | 11 |
| 7 | Interface utilisateur | 12 |
| 7.1 | Terminal | 12 |
| 7.1.1 | Utilisation indépendate des filtres | 12 |
| 7.1.2 | Utilisation du réseau de neurones | 12 |
| 7.2 | GTK | 12 |
| 8 | Gestion du Projet | 13 |
| 8.1 | Avancée du projet | 13 |
| 8.2 | Organisation du groupe | 14 |
| 8.2.1 | Outils de travail | 14 |

| | | |
|-----------|-------------------|-----------|
| 8.2.2 | UML | 15 |
| 9 | Conclusion | 16 |
| 10 | Sources | 16 |

1 Introduction

Le groupe BAME-OCR réalise, au cours du semestre 3, un projet de reconnaissance optique de caractères (OCR). Ce document présente l'état d'avancement du projet par le groupe, dans le cadre du temps imparti avant la première soutenance. De plus, ce document évoquera également ce qui sera abordé lors de la prochaine soutenance.

Le travail réalisé par le groupe durant la période allant du 4 septembre au 3 novembre sera détaillé, y compris les difficultés rencontrées et les différentes fonctionnalités mises en place. Les méthodes utilisées seront explicitées dans leurs sections correspondantes.

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 déplaît 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 | Second | | Principale | |

Chaque tâche a été divisée au début du mois de septembre. L'équipe a choisi de répartir le travail en deux parties distinctes, chacune ayant son responsable, noté "Principal" sur le tableau ci-dessus. Ensuite, des rôles secondaires ont été attribués pour accomplir toutes les tâches requises pour ce projet, et mettre en place un suivi secondaire du projet.

4 Traitement d'images

4.1 Prétraitement

Pour que les images puissent être traitées par le réseau de neurones, il faut que celles-ci aient un format spécifique pour qu'il puisse les interpréter le mieux possible. Il est alors important d'appliquer plusieurs filtres, que nous allons vous décrire ci-dessous. En effet, il existe deux types principaux de filtres, les filtres pixel par pixel tel que le filtre noir et blanc et de nuances de gris. Et les filtres de convolution, dont le principe est de modifier un pixel en fonction des pixels qui l'entourent, dans cette catégorie, nous retrouvons le flou gaussien, et la majorité des filtres de Canny.

4.1.1 Nuance des gris

L'objectif de ce filtre est d'atténuer toutes les couleurs de l'image en convertissant tous les pixels en variations de gris. Cela s'effectue en modifiant directement les quantités de rouge, vert, bleu (RGB) de chaque pixel, la formule est alors :

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

4.1.2 Filtre Gaussien

Le filtre Gaussien est un filtre utilisé pour flouter une image afin d'atténuer les bruits qui y sont présents. Il repose sur des filtres de convolution. Nous choisissons une matrice qui va déterminer la perte de précision et de détails dans l'image. 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.

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

4.1.3 Filtre de Canny

Nous avons choisi d'utiliser le filtre de Canny pour mettre en évidence les informations utiles dans les images, en particulier les chiffres et les lignes. Ce filtre est en réalité une combinaison de plusieurs filtres indépendants qui travaillent ensemble pour obtenir le résultat souhaité. Il est composé d'un filtre de gradient d'intensité, d'un filtre d'orientation, d'un algorithme de suppression des maxima locaux, et du seuillage par hystérésis, qui sera expliqué dans la sous-partie suivante. Ici, nous allons détailler uniquement 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, nous utilisons deux matrices de convolution qui sont appliquées à chaque pixel de l'image.

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

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-maxima. 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 consiste à supprimer les pixels qui ont des valeurs en dessous d'un seuil bas ou au-dessus d'un seuil haut. Après avoir passé notre image par le filtre de Canny, nous devons lisser le résultat. Pour commencer ce processus, nous allons déterminer un seuil haut en utilisant la méthode d'Otsu. Pour ce faire, nous commençons par calculer l'histogramme de notre image, puis nous déterminons les poids initiaux. Ensuite, nous parcourons tous les seuils possibles à travers l'histogramme, mettant à jour les poids à chaque étape et calculant la variance de la classe. Le seuil que nous souhaitons utiliser correspond à la valeur qui maximise la variance.

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

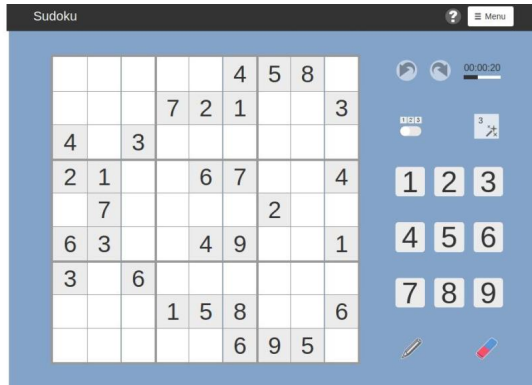
Voici la formule littérale de la méthode d'Otsu. Pour obtenir le seuil bas, nous divisons simplement le seuil haut par 2. Pour poursuivre le seuillage, le programme parcourt tous les pixels de l'image. Si la valeur du pixel est supérieure au seuil haut, nous le "mettons en surbrillance" en attribuant 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.

Une fois le premier passage d'image terminé, nous effectuons un autre passage pour éliminer les pixels marqués avec une valeur intermédiaire. Le programme effectue ceci de manière simple : dès qu'il rencontre un pixel intermédiaire, il vérifie s'il a des voisins maximaux. Si c'est le cas, les composantes du pixel sont mises au maximum, sinon, elles sont mises au minimum. Une fois ces deux passages terminés, le seuillage par hystérésis est achevé.

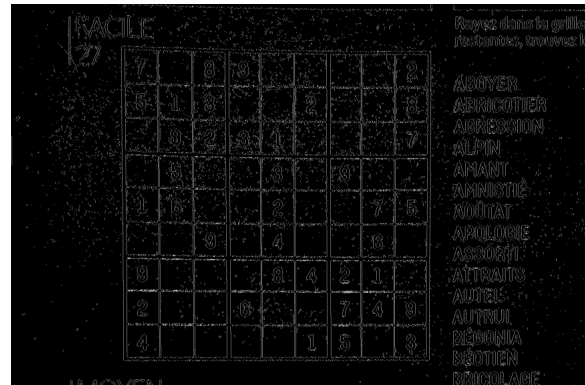
4.1.5 Réduction des bruits

Pour réduire le bruit, nous avons prévu d'utiliser un script capable de détecter les petits regroupements de pixels qui forment du bruit dans l'image traitée. Nous envisageons d'incorporer un seuil pour déterminer la taille des regroupements de pixels à supprimer. Le script est encore à un stade conceptuel, et nous devons résoudre les problèmes et le développer. Nous pensons donc qu'il sera prêt pour la dernière soutenance.

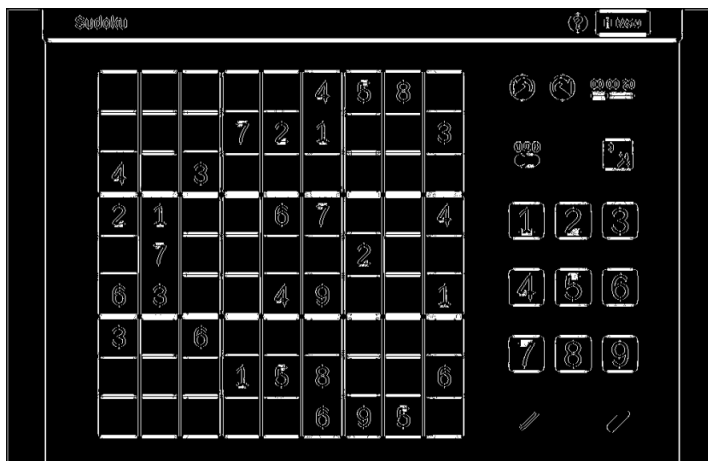
4.1.6 Résultats



*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'intégrer ses résultats dans nos futurs développements.

4.2 Traitement

Nous n'avons pas encore entamé la réflexion sur les scripts nécessaires pour réussir le découpage d'images. Nous envisageons d'utiliser la transformée de Hough pour détecter les lignes dans l'image. Jusqu'à présent, nous avons uniquement développé un script de rotation d'image. 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.

5 Reconnaissance de caractères

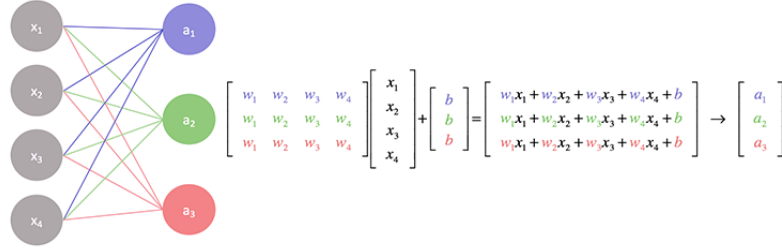
La reconnaissance de caractères repose sur un réseau de neurones d'apprentissage profond que nous avons choisi de rendre aussi générique que possible. Cela signifie qu'il peut être entraîné pour résoudre des problèmes plus simples que la reconnaissance d'images, ce qui facilite le développement.

5.1 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.

5.1.1 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.



5.1.2 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.

5.2 Problème du XOR

Pendant la phase de développement, nous avons utilisé le réseau pour résoudre le problème du XOR. L'objectif était de reproduire la porte logique XOR, avec deux neurones d'entrée et un neurone de sortie. En ce qui concerne les couches intermédiaires, nous avons testé différentes tailles pour évaluer les performances de l'algorithme.

5.3 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.

5.4 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".

6 Résolveur de sudoku

6.1 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.

6.2 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 soluble. 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.

6.3 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.

7 Interface utilisateur

7.1 Terminal

Pour utiliser l'OCR lors de la première soutenance, il vous suffit de compiler le programme avec la commande "Make" et d'exécuter l'exécutable nommé "ocr". Plusieurs options sont disponibles, et vous pouvez les découvrir en utilisant l'argument "-help". Pour une utilisation de base, vous devez spécifier en argument le chemin vers l'image d'origine et l'image de destination.

7.1.1 Utilisation indépendante des filtres

Si vous souhaitez observer le fonctionnement des filtres indépendants, vous pouvez également spécifier le nom du ou des filtres que vous souhaitez visualiser dans le terminal en réalisant également la commande "make" avec l'argument du filtre à tester. Du point de vue de la programmation, en dehors de l'initialisation de la bibliothèque SDL, il s'agit principalement d'une boucle qui parcourt tous les arguments et les compare à l'aide de la fonction "strcmp()".

7.1.2 Utilisation du réseau de neurones

Pour utiliser uniquement le réseau de neurones, vous avez deux options : vous pouvez l'utiliser pour résoudre un XOR en fournissant les deux valeurs en paramètres, ou vous pouvez l'entraîner sur une base de données. Dans le futur, il sera possible d'entraîner le réseau de neurones et de résoudre des images directement depuis l'interface utilisateur, en plus de pouvoir charger et enregistrer des images pour une utilisation ultérieure.

7.2 GTK

L'un des objectifs primordiaux pour la version finale du projet consiste 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 permettra aux utilisateurs de manipuler le logiciel de manière visuelle, en interagissant 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.

8 Gestion du Projet

Le groupe BAME-OCR a réussi à mettre en place une gestion du temps très efficace. En effet, le groupe est satisfait de l'avancée du projet au cours des huit dernières semaines. Vous pouvez retrouver le pourcentage d'avancement de chaque tâche dans le tableau ci-dessous. Ensuite, nous discuterons des modifications et des changements que nous souhaitons réaliser pour la deuxième soutenance.

8.1 Avancée du projet

Le groupe BAME-OCR a réussi à mettre en place une gestion du temps très efficace. En effet, le groupe est satisfait de l'avancée du projet au cours des huit dernières semaines. Vous pouvez retrouver le pourcentage d'avancement de chaque tâche dans le tableau ci-dessous. Ensuite, nous discuterons des modifications et des changements que nous souhaitons réaliser pour la deuxième soutenance.

| Tâche/ Membre | Première Soutenance | Rendu du projet |
|-------------------------------|---------------------|-----------------|
| Traitement d'images | 60% | 100% |
| Solveur de Sudoku | 90% | 100% |
| Reconnaissance des caractères | 80% | 100% |
| Interface | 75% | 100% |
| Site d'internet | 0% | 100% |

Pendant cette période, le domaine du traitement d'images a connu des avancées significatives. Cependant, il est essentiel de ne pas sous-estimer la complexité de la tâche, ce qui signifie que nous avons encore un travail considérable à accomplir dans les semaines à venir. Le Sudoku Solver a été terminé et fonctionne conformément à nos attentes. Néanmoins, il est important de noter que des modifications pourraient être apportées avant le rendu final du projet afin d'optimiser son fonctionnement et d'améliorer la propreté du code. En conséquence, nous pouvons estimer que nous en sommes actuellement à environ 90% d'achèvement pour cette partie du projet.

La reconnaissance des caractères a progressé plus rapidement que le traitement d'images, mais il s'agit également d'une tâche complexe qui demande une attention particulière. Ainsi, il nous reste encore environ 40% de travail à accomplir dans ce domaine pour atteindre nos objectifs.

Quant au site web, bien qu'il soit considéré comme optionnel, l'équipe a exprimé le souhait de le réaliser de manière soignée. Pour la deuxième présentation, nous avons donc l'ambition d'atteindre un achèvement de 100% pour le site web, ce qui signifie que nous prévoyons d'ajouter une interface qui passera de 75% à 100% d'achèvement, garantissant ainsi la pleine fonctionnalité du projet.

8.2 Organisation du groupe

Notre équipe a choisi de diviser notre projet en deux parties majeures, l'une consacrée à la manipulation et au traitement des images, et la seconde axée sur le réseau de neurones. Cette décision a été fondamentale pour assurer une répartition efficace des responsabilités et conférer une structure solide à notre organisation. Grâce à cette approche, nous avons pu attribuer des missions spécifiques à chaque équipe, définir des objectifs clairs, et travailler de manière plus ciblée.

Cependant, il est important de noter que la réussite 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. Chaque semaine, nous nous sommes investis de manière rigoureuse, en particulier lors de la semaine du 30 octobre. À ce moment-là, nous avons consacré quatre journées complètes de travail en équipe au sein de l'établissement EPITA.

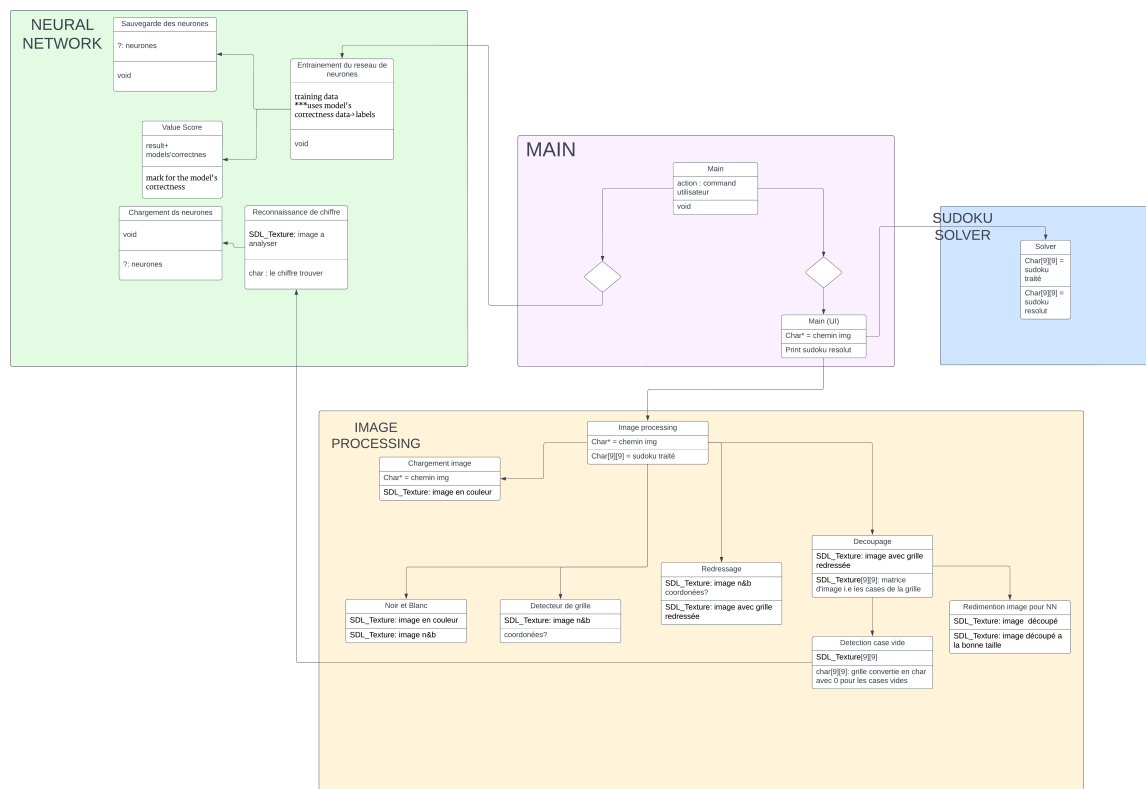
8.2.1 Outils de travail

Nous avons utilisé de nombreuses fonctionnalités de GitHub, telles que la gestion du projet avec une liste de tâches (To-Do list) répartie en plusieurs catégories en fonction des priorités du groupe. Cette approche a grandement facilité la coordination de nos efforts et nous a aidés à suivre l'avancement du projet de manière transparente.

Le groupe a également mis en place un système de communication efficace en restant joignable sur Discord. Sur cette plateforme, plusieurs canaux ont été dédiés à chaque responsable, permettant ainsi de partager toutes nos recherches et avancées de manière structurée. De plus, nous avons régulièrement partagé des liens vers les divers ensembles de données que nous utilisons, facilitant ainsi l'accès aux ressources nécessaires pour notre travail.

8.2.2 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.



9 Conclusion

Dans l'ensemble, le travail réalisé par le groupe est extrêmement satisfaisant, et le bilan est résolument positif. Nous avons accompli des progrès qui correspondent en tous points à nos attentes pour ce projet. Le Sudoku Solver est opérationnel et se rapproche de son achèvement, tout comme le XOR, qui génère les résultats conformes à nos objectifs. Malgré les efforts encore à fournir, le projet se dessine avec clarté. Il ne reste qu'un petit nombre de définitions à finaliser, ainsi que la mise en place de la liaison cruciale entre le traitement d'images et le réseau de neurones. Par ailleurs, l'implémentation de quelques filtres sur l'image demeure à effectuer, et la création du résultat final, incluant la mise en évidence des numéros en rouge, est encore au programme de notre travail à venir.

Ces derniers ajustements et développements ne devraient pas entraver notre élan, étant donné les solides bases que nous avons déjà établies. Notre projet est en bonne voie pour atteindre l'excellence que nous visons.

10 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>