



Rapport de soutenance

Omri Zakaria DAKKA

Mathis PEROT

Fabio SINTONI

Paul FERREIRA

Table des matières

I - Introduction	3
II - Présentation du groupe	4
1. Paul FERREIRA	4
2. Fabio SINTONI	4
3. Mathis PEROT	4
4. Omri Zakaria DAKKA	5
III - Répartition des tâches	6
IV - Avancement du projet	7
V - Avancement individuel	8
1. Paul FERREIRA	8
2. Fabio SINTONI	11
3. Mathis PEROT	13
4. Omri Zakaria DAKKA	14
VI – Conclusion	15

I. Introduction

Cette année encore, l'EPITA a choisi de proposer un projet de type OCR (Optical Character Recognition). Ce dernier a pour objectif de résoudre une grille de Sudoku donnée à partir d'une image. Pour y parvenir, nous avons décidé de décomposer cet ensemble en quatre grandes parties qui sont : un programme de traitement de l'image et de découpage des caractères, un réseau de neurones permettant de reconnaître les chiffres dans la grille, un programme de résolution de la grille et enfin un logiciel combinant le tout, nous permettant ainsi le fonctionnement de notre OCR.

À la suite de l'annonce de ce projet, nous avons organisé un groupe de 4 personnes constitué de : Omri Zakaria DAKKA, Mathis PEROT, Fabio SINTONI et Paul FERREIRA. De plus, nous appartenons tous à la même classe, ce qui facilite la communication au sein du groupe.

Dans ce rapport de soutenance, nous allons tout d'abord vous présenter chaque membre du groupe, nous poursuivrons ensuite cette présentation avec la répartition individuelle de nos tâches, puis celle de notre avancement sur le projet. Nous détaillerons par la suite les travaux réalisés par chacun et les éventuels problèmes rencontrés lors de la réalisation de ceux-ci. Enfin nous finirons par une conclusion.

II. Présentation du groupe

1. Paul FERREIRA

Je suis issu d'une terminale générale et d'une première année à l'EPITA. Cette année encore, je suis de nouveau chef de projet ; Une tâche que je prends à cœur malgré sa difficulté résultante du fait de toujours devoir chercher à organiser les réunions de groupe de la façon la plus optimale pour faire de ce projet un véritable succès. D'autre part, je cherche particulièrement à m'améliorer par rapport à l'année dernière où mon manque de prise de responsabilités a entraîné un retard conséquent sur le projet qui avait rendu mon jeu injouable.

2. Fabio SINTONI

J'ai également eu mon baccalauréat général avec les spécialités mathématiques et physique-chimie. L'an passé, à EPITA, j'ai pu réaliser avec mon groupe un jeu vidéo en 2D en C# qui m'a permis de beaucoup progresser en programmation et notamment en programmation orientée objet. Cette année, je souhaite également m'améliorer et participer du mieux que possible à ce projet en réalisant un maximum de tâches. Le traitement d'image en C paraît difficile car je n'en ai jamais fait auparavant personnellement, mais je compte réaliser au mieux la mission confiée par mon chef de projet.

3. Mathis PEROT

Je viens également d'une terminale générale avec les spécialités mathématique et NSI, en plus de l'option Mathématiques expertes suivie de la première année de classe préparatoire à EPITA où j'ai pu développer avec mon équipe un jeu en 2D de type boss rush en C# avec l'aide de l'outil Unity. Je suis cette année dans l'objectif d'apprendre un maximum sur le traitement d'image, la programmation en C d'un logiciel et celle d'un réseau de neurone grâce à ce projet.

4. Omri Zakaria DAKKA

Lors de mon parcours scolaire, je suis passé par une terminale générale avec spécialité mathématique avant d'intégrer les classes préparatoires à l'EPITA qui m'ont permis de faire mes premiers pas dans le monde de l'informatique et de la programmation. Mon rôle au sein de ce projet est d'établir un réseau de neurones qui sera capable de reconnaître les chiffres dans une grille de Sudoku.

III. Répartition des tâches

Nous avons convenu que deux des quatre parties de ce projet seraient à finir en premier à savoir le traitement de l'image et son découpage ainsi que le réseau de neurones. Nous avons donc créé deux binômes pour s'occuper de ces tâches majeures : deux personnes se sont donc totalement focalisées sur le traitement de l'image étant donné que c'est la partie demandant le plus de travail pour la première soutenance. Une autre personne s'est focalisée sur la réalisation du réseau de neurones avec une autre en soutien. Le logiciel et le solveur étant des tâches plus rapides à réaliser, ont été mis en place par la même personne.

Tâches	Omri Zakaria DAKKA	Mathis PEROT	Fabio SINTONI	Paul FERREIRA
Logiciel				X
Réseau de neurones	X			x
Traitement de l'image		X	X	
Résolveur de Sudoku				X

Légende :

X : tâche principale

x : tâche secondaire

IV. Avancement du projet

Ci-dessous le tableau présentant l'avancement général du projet avec ce que nous avons prévu au départ et ce qui a été vraiment réalisé :

Tâches	Avancement Prévu	Avancement réel
Chargement d'une image et suppression des couleurs.	100%	100 %
Rotation manuelle de l'image.	100 %	100 %
Détection de la grille et de la position des cases.	100 %	100%
Découpage de l'image.	100 %	80%
Résolution du sudoku.	100 %	100 %
Preuve de concept du réseau de neurones	100 %	100 %

V. Avancement individuel

Dans cette partie chacun des membres du groupe va présenter ce qu'il a réalisé sur le projet.

1. Paul FERREIRA

J'ai commencé à travailler sur le projet en m'attaquant au logiciel. Pour cela, j'ai décidé d'utiliser la bibliothèque GTK ainsi que le logiciel Glade. La librairie GTK m'a permis de développer assez facilement un logiciel fonctionnel et facile d'utilisation grâce au système de signaux. Glade de son côté m'a permis de développer une interface utilisateur en plaçant les éléments visuels et en convertissant le tout dans un fichier au format xml. Ce format xml est par la suite converti par GTK via sa fonction builder qui permet ensuite de récupérer les éléments et d'interagir avec eux.

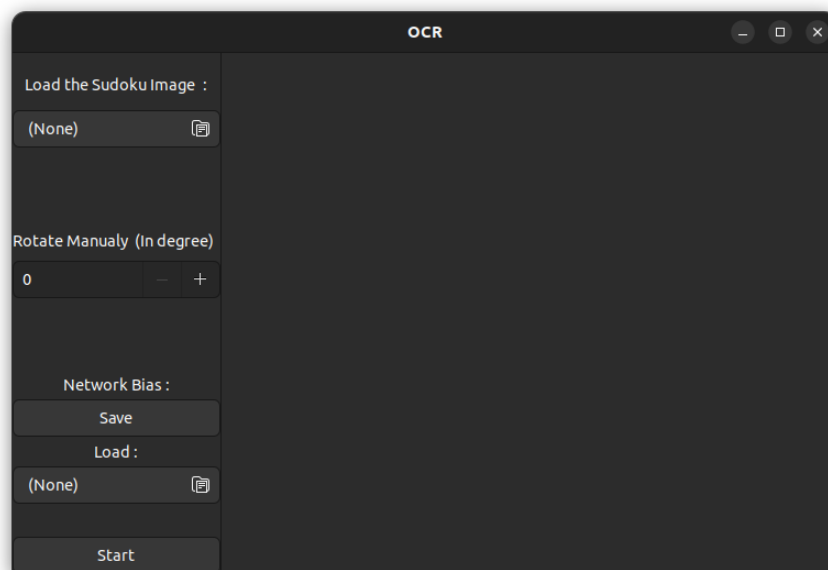


Figure 1 – Logiciel

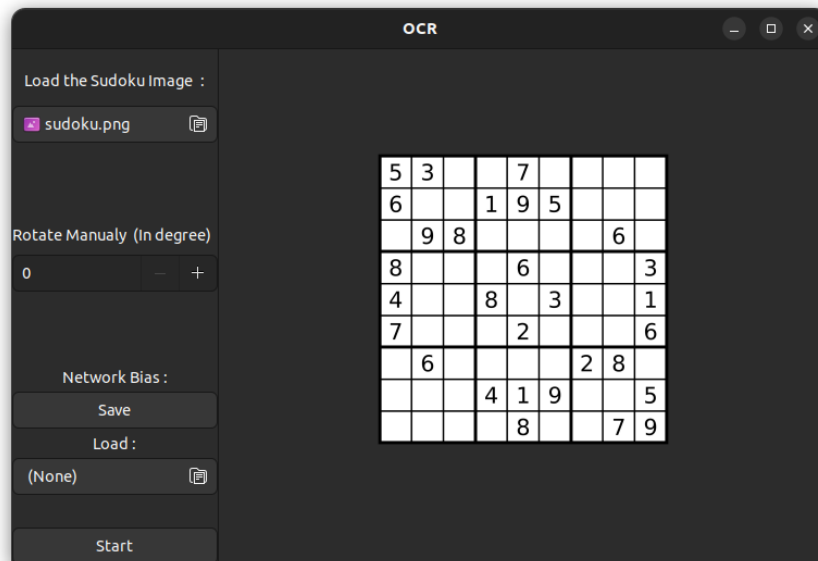


Figure 2 – Logiciel après chargement d'une image

J'ai ensuite travaillé sur la rotation de l'image au sein du logiciel. Pour cela, j'ai utilisé une formule assez simple de conversion de pixels, qui, bien qu'elle crée du grain fonctionne assez bien.

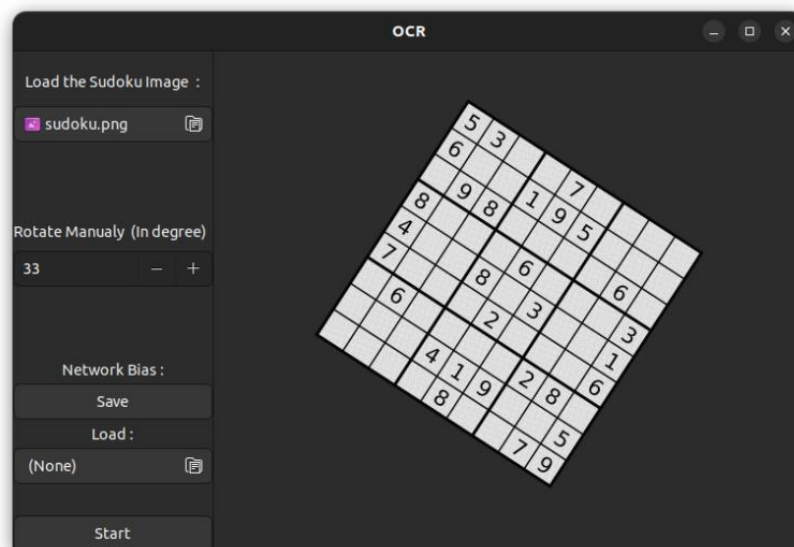
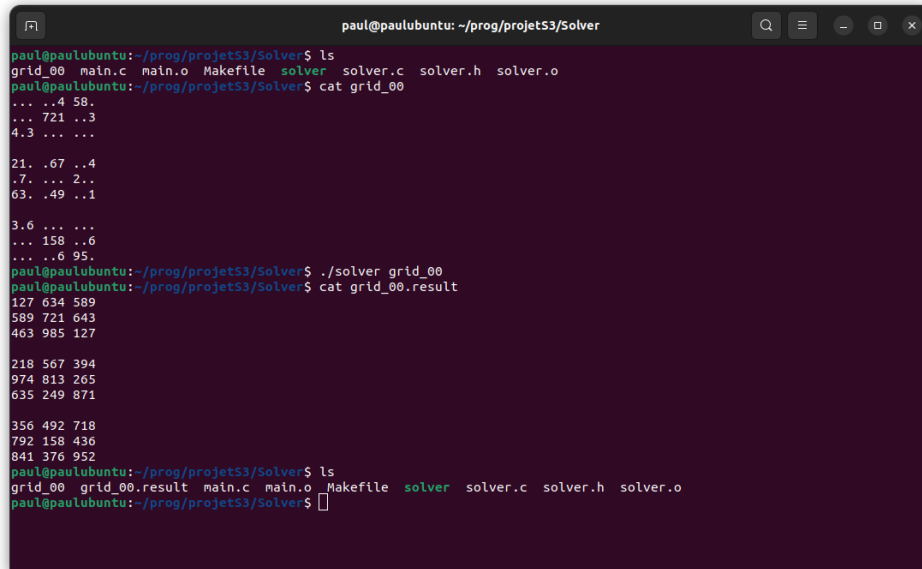


Figure 3 – Image après une rotation de 33°

J'ai ensuite travaillé sur la résolution de la grille. Pour ceci, j'ai utilisé une fonction récursive qui place les nombres de 1 à 9 et qui teste si la grille est solvable en testant l'axe X et Y du point où on se trouve et sur la zone de 3x3 cases dans laquelle se trouve le point et on continue jusqu'à qu'une solution soit trouvée si elle existe.



```
paul@paulubuntu: ~/prog/projetS3/solver
paul@paulubuntu:~/prog/projetS3/solver$ ls
grid_00  main.c  main.o  Makefile  solver  solver.c  solver.h  solver.o
paul@paulubuntu:~/prog/projetS3/solver$ cat grid_00
... ..4 58.
... 721 ..3
4.3 ... ..

21. .67 ..4
..7. ... 2..
63. .49 ..1

3.6 ... ...
... 158 ..6
... ..6 95.
paul@paulubuntu:~/prog/projetS3/solver$ ./solver grid_00
paul@paulubuntu:~/prog/projetS3/solver$ cat grid_00.result
127 634 589
589 721 643
463 985 127

218 567 394
974 813 265
635 249 871

356 492 718
792 158 436
841 376 952
paul@paulubuntu:~/prog/projetS3/solver$ ls
grid_00  grid_00.result  main.c  main.o  Makefile  solver  solver.c  solver.h  solver.o
paul@paulubuntu:~/prog/projetS3/solver$
```

Figure 4 – Exemple de résolution de grille

J'ai fini par travailler sur les éléments devant être commencés pour cette première soutenance. J'ai donc créé des images pour entrainer le réseau de neurone. J'ai rajouté les boutons sur le logiciel pour sauvegarder et importer les poids de ce dernier. J'ai ensuite imaginé comment nous allons donner les résultats sous forme de grille. Pour cela nous prendront une image de grille déjà créée et des trous que nous remplirons avec des images de chiffres en fonction de ce que notre programme a trouvé.

2. Fabio SINTONI

Je me suis occupé du traitement de l'image avec Mathis. Pour cela, nous avons utilisé la librairie SDL en C. J'ai codé un programme permettant d'augmenter les contrastes d'une image puis de la mettre en noir et blanc pour pouvoir mieux la traiter et reconnaître la grille.

Voici deux exemples d'images obtenus après cette transformation :

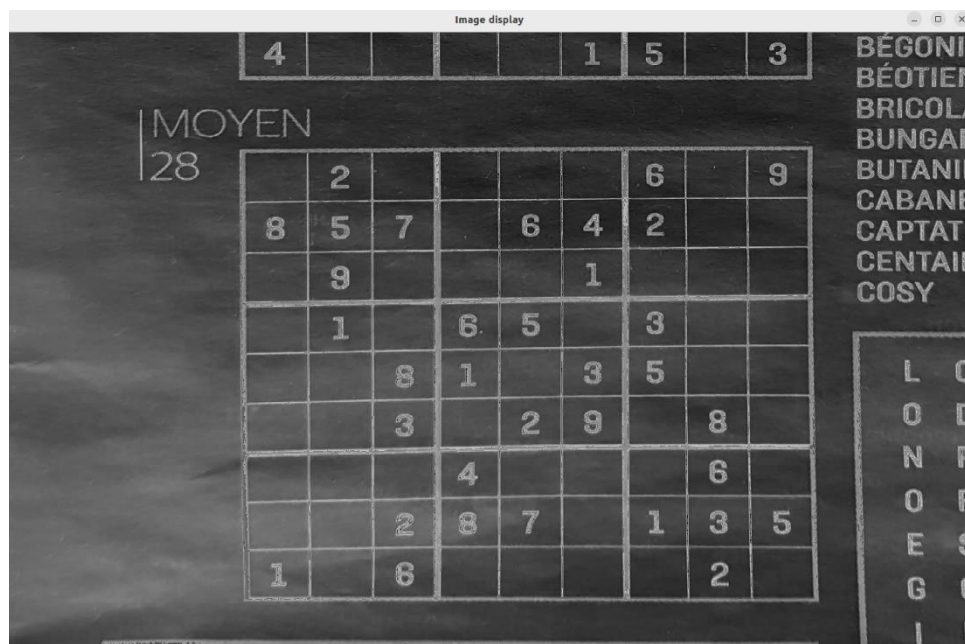


Figure 5 – 1^{er} Exemple d'image après une transformation en noir et blanc.

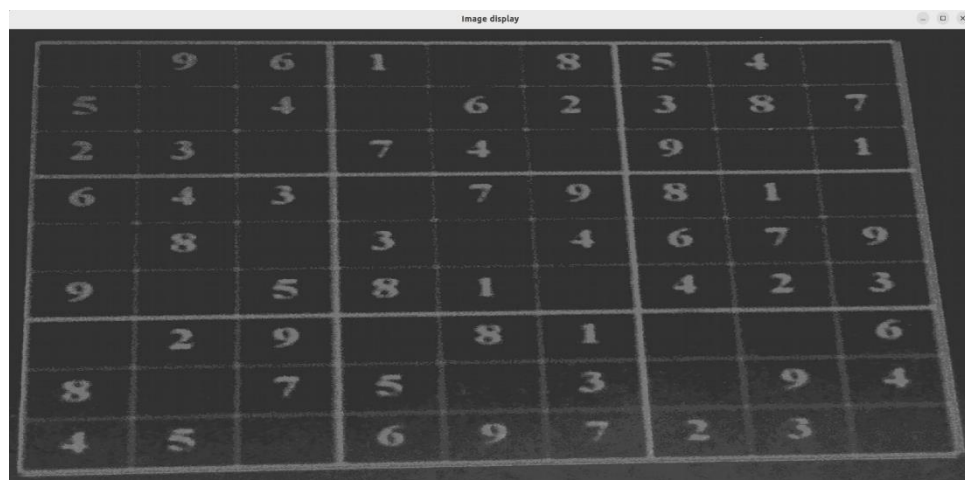


Figure 6 – 2^{ème} Exemple d'image après une transformation en noir et blanc.

Par la suite, nous avons trouvé un algorithme du nom de “transformée de Hough” permettant de reconnaître des formes dans une image. Ainsi, j’ai participé au code de cet algorithme en C pour pouvoir détecter les différentes lignes présentes et ainsi découper la grille de sudoku en différentes cases.

3. Mathis PEROT

J'ai travaillé avec Fabio tout au long de ce temps sur le traitement de l'image, lui comprenant bien mieux la librairie SDL que je ne le faisais, il s'est occupé de la majorité du code en rapport avec cette librairie, étant de mon côté en train d'essayer de trouver des algorithmes pour réussir à trouver le quadrillage. Nous avons donc commencé par partir sur la piste des algorithmes de détection de Contours utilisant les filtres de Sobel, Canny et Prewitt.



Figure 7 – Exemple du filtre de Sobel

Cependant, ces filtres nous semblaient très difficiles à implémenter et après quelques jours à bloquer dessus nous avons décidé de chercher un autre moyen de détecter des carrés : nous avons décidé de détecter les lignes. Pour cela, nous avons trouvé Hough Transform, un algorithme qui va prendre plusieurs points et en créer 2 fonctions sinusoïdale, et détecter les endroits où le plus de fonctions se recoupe afin de savoir où les lignes sont.

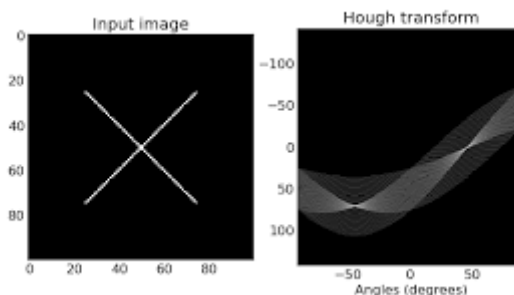
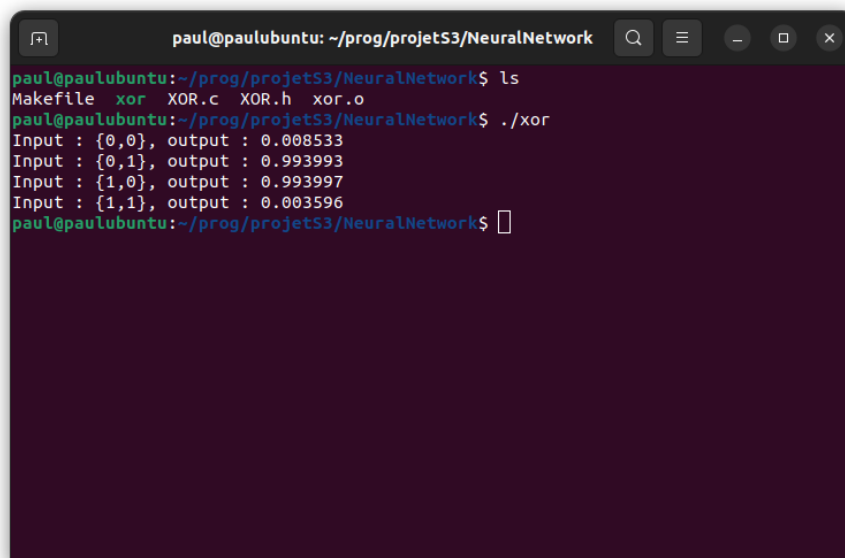


Figure 8 – Exemple d'un Hough Transform

Le Hough Transform ensuite nécessite quelques calculs pour comprendre où sont les lignes, mais la matrice obtenue est bien plus facile à manipulée par un ordinateur que la matrice originale.

4. Omri Zakaria DAKKA

Pour reconnaître les chiffres de la grille, le réseau de neurones a avant tout besoin de s'entraîner. C'est dans cet état d'esprit que nous lui avons appris la fonction XOR (ou exclusif) qui sert à exprimer une condition d'exclusivité qui peut se décrire comme « Soit l'une ou l'autre des valeurs, mais pas les deux ». Ainsi pour que la condition soit vraie, il faut qu'exclusivement une seule des valeurs 0 ou 1 soit active. La résolution d'un problème d'une telle complexité reflète parfaitement l'utilité de la mise en place d'un système de réseau de neurones interconnectées entre elles. Entraîner notre réseau de neurones sur la résolution de la condition du ou exclusif revient ainsi à dire que ce dernier est bien fonctionnel comme nous pouvons le constater sur les tests affichés plus haut.



```
paul@paulubuntu: ~/prog/projetS3/NeuralNetwork
paul@paulubuntu:~/prog/projetS3/NeuralNetwork$ ls
Makefile  xor  XOR.c  XOR.h  xor.o
paul@paulubuntu:~/prog/projetS3/NeuralNetwork$ ./xor
Input : {0,0}, output : 0.008533
Input : {0,1}, output : 0.993993
Input : {1,0}, output : 0.993997
Input : {1,1}, output : 0.003596
paul@paulubuntu:~/prog/projetS3/NeuralNetwork$
```

Figure 9 – Sortie du réseau de neurone après apprentissage de la fonction XOR.

VI. Conclusion

En conclusion, grâce à l'avancement conséquent sur chacune des parties vis-à-vis du planning qui nous a été attribué, le projet commence lentement à prendre forme et bien qu'il ne soit pas encore complètement achevé, notre groupe reste optimiste et très confiant quant à la finalisation de ce dernier avant la prochaine soutenance. D'autre part, nous espérons également réussir à transmettre nos compétences ainsi que le plaisir que nous avons pris à coder ensemble durant la réalisation de projet.