

RAPPORT DE SOUTENANCE 1

SUDOKU SOLVER

Mathieu Neumar
Andrea Matera
Titouan Locatelli
Daniel Miguel

Promotion 2025

Table des matières

1 Introduction

2 - Le Projet

2.1 - Présentation du groupe

2.1.1 - MATD

2.1.2 - Mathieu NEUMAR

2.1.3 - Andrea MATERA

2.1.4 - Titouan LOCATELLI

2.1.5 - Daniel MIGUEL

2.2 - Répartition des tâches

3 - Traitement de l'image

4 - Détection de la grille et des cases

5 - Réseaux de neurones

6 - Sudoku Solver

7 - Conclusion

1 - Introduction

Ce projet d'OCR entre dans le cadre de notre projet S3 à EPITA. Il doit être réalisé en groupe de 4 personnes et se déroule de Septembre à Décembre. L'objectif de l'OCR sera de récupérer une grille de sudoku non résolu, d'améliorer la qualité de l'image pour la rendre plus facile à utiliser. On va ensuite la segmenter et la passer dans le réseau de neurones pour obtenir les informations de la grille. Pour finir, notre solveur résolvera ce sudoku et le renverra dans un fichier texte. Le tout accompagné d'une interface graphique permettant une utilisation plus simple

2 - Le projet

2.1 - Présentation du groupe

2.1.1 - MATD

MATD est un groupe composé de Andrea Matera, Daniel Miguel, Mathieu Neumar et Titouan Locatelli. On a repris le même groupe de projet de l'année dernière, étant donné que nous nous étions bien entendus, à l'exception de Titouan qui lui, a décidé de repasser l'OCR pour faire cette fois le réseau de neurones.

2.1.2 - Mathieu NEUMAR

Comme la majorité des élèves à Epita je suis passionné par l'informatique, tout ce qui touche aux nouvelles technologies et à la cybersécurité. J'ai déjà réalisé quelques projets durant mes années de lycée (sous Python) ainsi que le projet S2 l'année précédente. J'ai eu la chance de découvrir de nombreux domaines étant jeune ce qui me permet d'être assez flexible dans le développement du jeu. Je serais également le chef de projet donc je ferais en sorte de garder mon équipe motivée avec un planning bien construit et réalisable. Pour ce projet je m'occuperais notamment du solver ainsi que de l'interface du projet. Étant chef de projet, je me sens responsable d'assister les autres membres dans leurs tâches.

2.1.3 - Andrea MATERA

Je m'appelle Andrea Matera. Je suis né à Turin, en Italie, où j'ai passé pratiquement mon enfance et 18 ans de ma vie. Je suis issu d'une famille 100% italienne ce qui rend le fait que je parle et j'habite en France un peu une anomalie. J'ai eu la chance, comme ma sœur, d'intégrer une école française à l'étranger du réseau AEFE. J'ai participé au full cursus, de la maternelle au lycée, en passant donc 15 années à apprendre le français contre la volonté de ma patrie. Cette décision de la part de mes parents m'a permis d'ouvrir plusieurs portes qu'avec une école publique italienne je n'aurais pas pu ouvrir. Je me retiens chanceux aujourd'hui

de pouvoir accéder à des établissements français sans avoir de problèmes de langues. Je vante aussi d'un très bon niveau en Anglais, un bon niveau en Espagnol et bien sûr un niveau maternel en Italien.

Ma passion pour l'informatique est née dès mon enfance. Je dois cela à ma sœur, qui m'a introduit dès mes 5 ans au monde des jeux sur PC avec des jeux tels que Spore ou Ghost Master. J'avais compris d'avoir découvert un monde qui me fascinait, dans lequel je pouvais m'amuser et me relaxer quand j'avais du temps libre. C'est un monde qui ne me lassait jamais, en découvrant chaque jour un jeu nouveau. mais tout en étant inconscient de une chose, la plus importante: la machine sur laquelle je jouais. Pendant toutes ces années j'ignorais la technologie qui me permettait de faire tourner les jeux vidéos auxquels je me suis autant passionné, mais finalement un jour de collège, je me rends compte que sans l'informatique et sans cette magnifique technologie je n'aurais jamais pu m'entretenir autant pendant toute mon enfance. Je décide alors de vouloir découvrir ce qui se cachait derrière en ayant un but bien précis, intégrer une école d'informatique. Je choisis donc la filière S spé maths en Terminale pour approfondir mes études dans les matières scientifiques car malheureusement, bien étant dans un établissement français, celui-là n'offrait pas de spécialisation en SI. Je passe donc mon Bac, un VRAI Bac en 2019 et je fais mes valises pour Lyon, car j'ai à peine été accepté dans l'université Lyon 1. Mais après plusieurs mois dans cette faculté, je me rends compte que ce type d'enseignement n'est pas fait pour moi. Je décide de me mettre à la recherche d'une autre école qui offre le parcours informatique et qui s'approchait du type d'enseignement que je recherchais. Je me rappelle donc d'un pote du Lycée Français Jean Giono de Turin qui, lui aussi ayant les mêmes passions et avait intégré EPITA la même année. Je passe donc un l'examen Advance Parallèle et je me retrouve aujourd'hui sur les bancs de l'EPITA à lancer le projet OCR avec une équipe que j'estime beaucoup.

2.1.4 - Titouan LOCATELLI

Je suis Titouan Locatelli, je suis né à Montpellier, mais je n'ai jamais vécu en France. J'ai suivi mes parents chercheurs au Costa Rica, Indonésie puis au Pérou. Cela a donc permis de m'ouvrir l'esprit depuis jeune grâce à une personnalité curieuse de nature. J'ai fais une fac de maths arrivant en France après mon bac. J'avais des facilités en maths et le côté scientifique aurait bien satisfait ma curiosité. Mais les maths restent théoriques et cela devient vite ennuyant, l'informatique était un super recours pour ne pas perdre non plus les maths et sciences totalement, mais tout en intégrant des matières telles que la programmation beaucoup plus pratiques que théoriques donnant plus de sens à mes études. Je refais l'OCR cette année par choix, programmation était validée le semestre dernier. Mais cette année j'ai préféré de me charger du réseau de neurones, c'est le sujet qui m'intéressait le plus dans l'OCR que je n'avais pas eu l'opportunité de travailler l'année dernière. J'ai hâte de faire un projet performant avec une équipe performante.

2.1.5 - Daniel MIGUEL

Je m'appelle Daniel Miguel, je suis né à Saint-Cloud en région parisienne. J'ai toujours été attiré par les sciences et l'informatique. J'ai choisi de faire mes études à Epita car encore une fois car j'aime l'informatique et les sciences et je voulais en apprendre plus sur le fonctionnement des ordinateurs et des programmes informatiques, j'ai toujours été curieux de nature car j'aime bien savoir comment les choses marchent. Ce projet OCR est pour moi l'occasion justement de savoir comment marche la reconnaissance d'image, quelque chose qui relevait de la magie pour moi depuis petit. De plus, accompagné de mes camarades avec qui j'avais fait mon projet de S2, ainsi que d'un très bon ami, l'ambiance dans le groupe ne peut être que bonne ce qui me motive d'autant plus à essayer de comprendre et appliquer des algorithmes complexes dans un cas concret.

2.2 - Répartition des tâches

	Mathieu	Andrea	Titouan	Daniel
Pré traitement de l'image		X		
Segmentation				X
Réseaux de neurones			X	
Solver	X			
Interface	X			

3 - Traitement de l'image

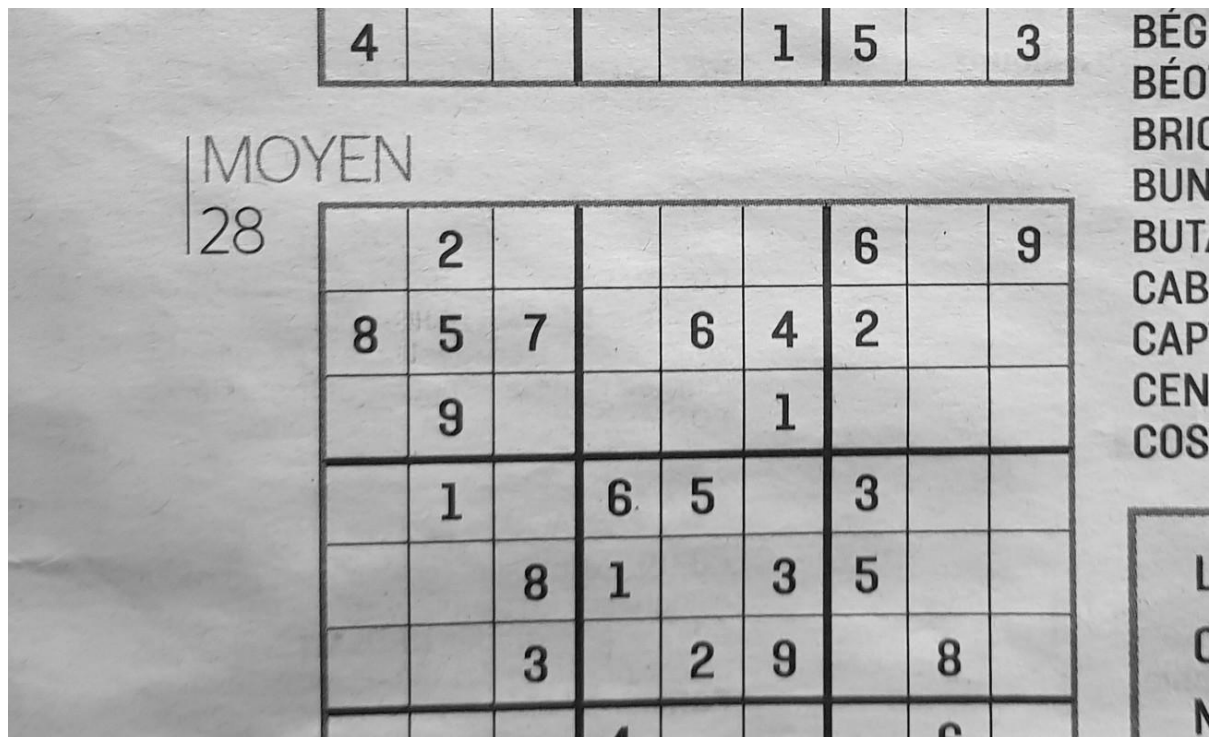
Le prétraitement de l'image peut se diviser en deux importantes parties, binarisation et rotation de l'image. Deux parties qui au début semblent assez simples à réaliser mais qui, une fois vues en détail, le problème est plus gros que ce que l'on pense. Sous estimer le prétraitement a été une de mes plus grosses erreurs au début de ce projet, mais bien évidemment on apprend de nos erreurs et je ne vais plus rien sous estimer du projet d'OCR.

BINARISATION

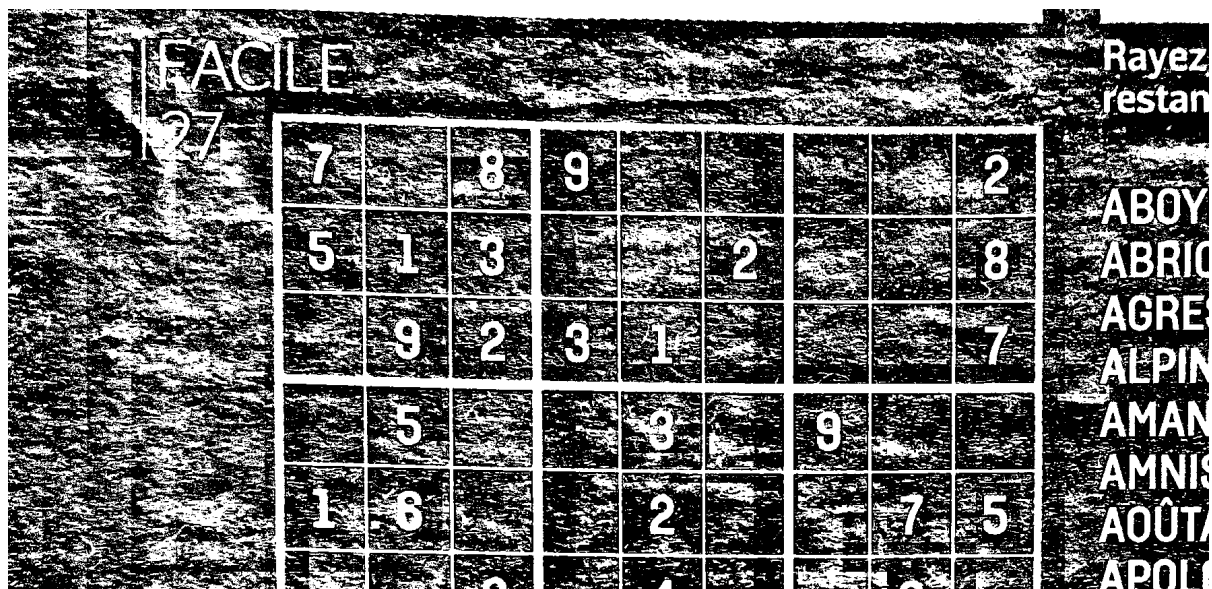
La binarisation consiste à prendre une image vierge, c'est-à-dire en couleur et la rendre lisible pour un réseau de neurones qui va reconnaître et résoudre la grille. la binarisation se base sur deux étapes, la mise en noir et blanc de l'image et le renfort des contrastes en éliminant éventuellement les "bruits" ou "obstacles" de l'image.



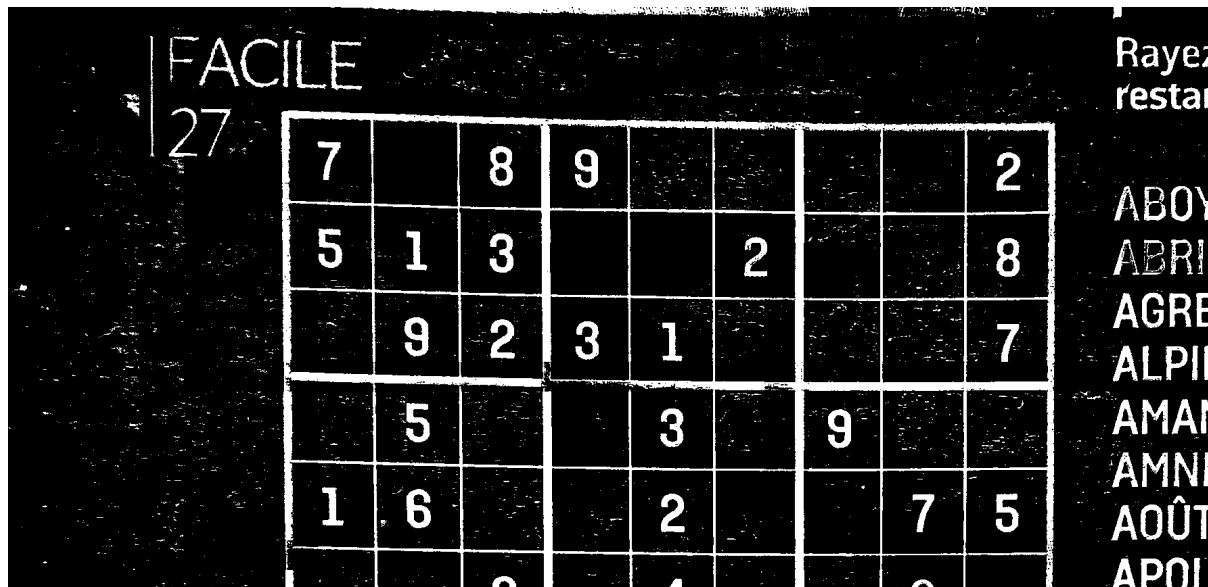
Prenons celle-ci comme image de départ. Nous allons appeler notre première étape le “grayscale”. Ceci consiste à appliquer à chaque pixel de l’image une couleur moyenne par un simple calcul qui est : $0.2126 * R + 0.7152 * G + 0.0722 * B$, avec RGB étant respectivement le Red le Green et le Blue du pixel en question. Une fois avoir parcouru toute l’image et avoir effectuée le changement de couleur sur chaque pixel, on devrait aboutir à une image en noir et blanc ressemblante à celle ci-dessous.



Les couleurs ont été supprimées et le renfort des contrastes maintenant va être plus facile à effectuer sur cette image. Pour ceci il faudrait agir de la manière suivante, ou au moins je pensais. Il faut prendre la moyenne de couleur de l'image simplement en additionnant toutes les couleurs moyennes du pixel (voir calcul au-dessus) pour enfin diviser par le nombre de pixel. Ensuite on parcourt à nouveau l'image en noir et blanc et à chaque pixel si la couleur moyenne est supérieure à la moyenne des moyennes on va dire, on rends le pixel totalement noir et inversement en blanc. Mais cette technique m'a tout de suite posé un problème avec les ombres.



La bonne méthode pour avancer s'est révélée d'être celle de diviser l'image en par exemple 32 x 32 petites sous-images et d'effectuer la moyenne des moyennes sur tous ces carrés de l'image. Mais il a fallu rajouter un petit détail très important, qui va permettre de combattre contre ces ombres. Le **global threshold** ou seuil global en français va nous permettre de gérer les ombres. Il va nous suffir de soustraire ce threshold à la valeur moyenne des moyennes et magiquement les ombres ne posent plus de problème.



L'image est beaucoup plus lisible que celle d'avant.

ROTATION

La rotation d'une image s'effectue par une simple rotation du pixel, c'est à dire que nous prenons le pixel de départ et grâce à une formule nous pouvons trouver la nouvelle position de ce pixel par rotation d'un angle connu. Mais le point de rotation doit être le centre de l'image, ce qui signifie qu'il faut convertir les coordonnées de l'image de base en prenant en compte que le milieu a pour coordonnées (0,0) et non plus le coin haut gauche. Ceci complique la tâche. Mais après avoir "centré" les coordonnées et appliqué aux coordonnées d'un point la formule de rotation avec l'angle en radians donné, nous avons tout les ingrédients pour tourner une image.



image de base

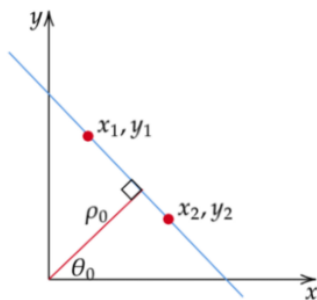


image après rotation de 10°

4 - Détection de la grille et des cases

Nos objectifs pour cette soutenance était en partie de repérer la grille dans l'image. En effet, sans cette étape on ne peut pas traiter notre grille afin d'extraire les caractères de cette dernière, et donc on ne peut pas reconnaître ces derniers.

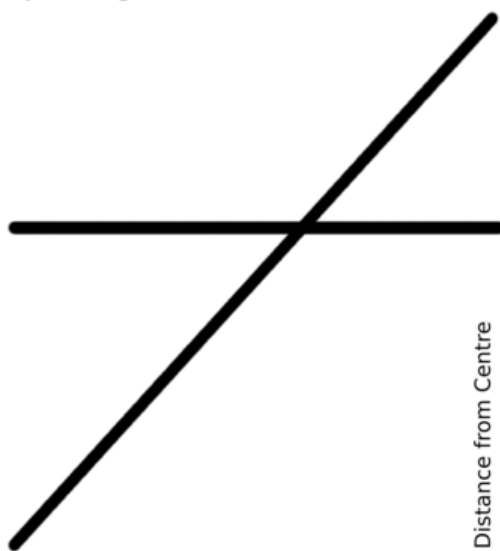
La méthode que nous avons décidé de mettre en place est celle du transformé de Hough. Cette méthode revient à créer une matrice avec pour chaque coordonnées (Tête, Rho) de cette matrice, le nombre de pixel blanc dans l'image présent sur la droite d'équations polaire de paramètres (Tête, Rho).



$$\rho = x_1 \cos(\theta) + y_1 \sin(\theta)$$

On utilisera les droites par lesquelles le plus de points passe afin de détecter la grille. Elles sont représentées par les intersections des courbes dans le tableau. (Comme ci-dessous)

Input Image



Rendering of Transform Results



Une fois les lignes du tableau trouvées il nous suffit de parcourir notre image suivant les lignes si nous sommes dessus afin de découper les case de notre Sudoku et les envoyer dans le réseau de neurones. (Sur l'image ci-dessous, seul les droites passant par les lignes épaisses de l'image sont dessinées)

	9	6	1		8	5	4	
5		4		6	2	3	8	7
2	3		7	4		9		1
6	4	3		7	9	8	1	
	8		3		4	6	7	9
9		5	8	1		4	2	3
	2	9		8	1			6
8		7	5		3		9	4
4	5		6	9	7	2	3	

On sauvegarde le caractère trouvé avec le réseau de neurones dans une chaîne de caractères, on fait un test au préalable afin de savoir si le fragment de l'image découpée est vide ou non, dans ce cas là on sauvegarde 0 dans la chaîne de caractères. A la fin on sauvegarde cette chaîne de caractères dans un fichier afin qu'elle soit utilisée dans la résolution du Sudoku.

5 - Réseaux de neurones

Pour cette soutenance j'ai dû créer un réseau de neurones XOR. Le réseau est assez simple en soi. Les inputs et outputs sont de 0 et des 1.

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

J'ai décidé de faire la structure suivante:

- 2 neurones sur la couche d'input: qui correspondent à x_1 et x_2 dans le tableau ci-dessus.
- 4 neurones sur la première couche cachée
- 4 neurones sur la deuxième couche cachée
- 1 neurone sur la couche d'output, la dernière couche.

Bien que ce réseau comporte peu de neurones et donc les connexions entre neurones, je pense qu'une grande partie du travail pour l'OCR final a été fait. C'est-à-dire la recherche et la mise en place de structures telles que neurones et couches de neurones.

La recherche a commencé évidemment par des vidéos de 3Blue1Brown comme pour tous les groupes je pense. Ces vidéos m'ont beaucoup aidé à comprendre le concept d'un réseau de neurones. Elle n'approfondit cependant pas sur les maths, mais m'a aidé à comprendre les éléments d'un réseau de neurones et les procédés par lesquels un réseau de neurones apprend, tel que le forward propagation et back propagation.

Je comprenais désormais le concept, les coder était une autre histoire. Je me suis donc mis à chercher des templates de codes de réseaux de neurones déjà existants. J'ai trouvé un code assez complet par Mayur Bhole. J'ai donc pris sa structure de neurones et ai adapté le réseau à un réseau XOR.

Les mathématiques derrière le réseau me sont encore un peu obscures mais le réseau marche. Je peux entraîner le réseau tant de fois que désiré, mais déjà au bout de 1000 à 2000 itérations d'entraînement les Outputs après un test sont corrects.

Pour la prochaine soutenance je vais adapter le réseau pour la tâche plus large de déterminer un chiffre à partir d'une image. La première couche du réseau va passer de 2 inputs à hauteur*largeur de l'image (entre 400 et 1000+) inputs. Les deux prochaines layers comprendrons a peu pres $\sqrt{\text{hauteur} \times \text{largeur}}$ (car mes recherches internet me l'ont suggéré, je n'ai pas fait assez de théorie mathématique en réseau de neurones pour dire autrement). Si besoin, j'ajouterais une couche cachée de neurones dans le réseau. La couche de output aura naturellement qu'un neurone (le chiffre).


6 - Sudoku solver

Pour cette soutenance on a dû réaliser le sudoku solver , que nous avons déjà fait l'année précédente en C#, mais en version plus optimisée.

Dans un premier temps, on récupère la matrice du fichier renvoyé par le réseau de neurones. On effectue ensuite la résolution de ce sudoku s'il est solvable et enfin, on crée un nouveau fichier .result avec la matrice résolue.


Concernant la partie de la résolution, on s'est basé sur un algorithme de backtracking afin de résoudre le sudoku le plus efficacement possible. Pour faire simple, on parcourt le sudoku à la recherche d'un 0 (case vide).

On va ensuite lui assigner une valeur et tester s'il elle est cohérente avec le carré, la colonne et la ligne liée à la case. On passe ensuite à la case vide suivante jusqu'à la fin de la matrice. Néanmoins, si les 9 chiffres ne passent pas, on va alors revenir en arrière et modifier la dernière case modifiée et ainsi de suite jusqu'à ce que la case qui posait problème soit résolue.




0	0	0	0	0	4	5	8	0
0	0	0	7	2	1	0	0	3
4	0	3	0	0	0	0	0	0
2	1	0	0	6	7	0	0	4
0	7	0	0	0	0	2	0	0
6	3	0	0	4	9	0	0	1
3	0	6	0	0	0	0	0	0
0	0	0	1	5	8	0	0	6
0	0	0	0	0	6	9	5	0

**parcours de la grid
jusqu'à trouver une case vide**




1	0	0	0	0	4	5	8	0
0	0	0	7	2	1	0	0	3
4	0	3	0	0	0	0	0	0
2	1	0	0	6	7	0	0	4
0	7	0	0	0	0	2	0	0
6	3	0	0	4	9	0	0	1
3	0	6	0	0	0	0	0	0
0	0	0	1	5	8	0	0	6
0	0	0	0	0	6	9	5	0

Incrémenter de 1



1	0	0	0	0	4	5	8	0
0	0	0	7	2	1	0	0	3
4	0	3	0	0	0	0	0	0
2	1	0	0	6	7	0	0	4
0	7	0	0	0	0	2	0	0
6	3	0	0	4	9	0	0	1
3	0	6	0	0	0	0	0	0
0	0	0	1	5	8	0	0	6
0	0	0	0	0	6	9	5	0

Vérifier si la case est safe



1	2	0	0	0	4	5	8	0
0	0	0	7	2	1	0	0	3
4	9	3	0	0	0	0	0	0
2	1	0	0	6	7	0	0	4
0	7	0	0	0	0	2	0	0
6	3	7	0	4	9	0	0	1
3	0	6	0	0	0	0	0	0
0	0	0	1	5	8	0	0	6
0	0	0	0	0	6	9	5	0

**Backtracking si la case
n'est pas safe**

1	2	7	6	3	4	5	8	9
5	8	9	7	2	1	6	4	3
4	6	3	9	8	5	1	2	7
2	1	8	5	6	7	3	9	4
9	7	4	8	1	3	2	6	5
6	3	5	2	4	9	8	7	1
3	5	6	4	9	2	7	1	8
7	9	2	1	5	8	4	3	6
8	4	1	3	7	6	9	5	2

résolu

D'ici la prochaine soutenance, on verra s'il est possible d'optimiser le sudoku à un stade supérieur .

7 - Conclusion

Nous vous remercions pour la lecture de notre premier rapport de soutenance et espérons qu'il est à la hauteur des attentes du premier rendu. Il est vrai que nous appréhendions un peu le début du projet étant donné que l'on venait de débiter les cours sur le C et que nous ne connaissions rien au principe de l'OCR. Néanmoins, nous sommes fiers de vous présenter notre première version de notre sudoku solver.