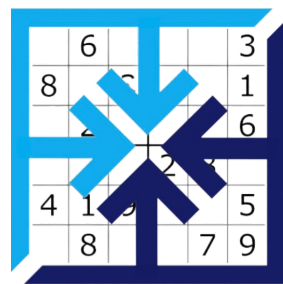


Rapport de soutenance

OCR Sudoku

Marie BOUET
Théo Esprimont
Maxime CAMBOU

10 Novembre 2022 - S3



OCR
SUDOKU SOLVER



Table des matières

1	Introduction	3
1.1	Qu'est ce qu'un OCR?	3
1.2	Notre équipe	3
1.2.1	Maxime CAMBOU	4
1.2.2	Théo ESPRIMONT	4
1.2.3	Marie BOUET	4
1.3	Le projet	5
1.4	Répartition des tâches	5
2	Avancement du projet	6
2.1	Traitement d'image	6
2.1.1	Chargement de l'image	6
2.1.2	Transformation en niveau de gris	7
2.1.3	Application d'un flou	8
2.1.4	Image en noir et blanc	8
2.1.5	La rotation manuelle de l'image	9
2.1.6	Détection de la grille	10
2.2	Le réseau de neurones	11
2.2.1	Le XOR (ou exclusif)	11
2.2.2	L'apprentissage	12
2.3	Le solveur	13
3	Conclusion	15
3.1	Expérience globale	15
3.2	Expériences personnelles	15
3.2.1	Maxime CAMBOU	15
3.2.2	Théo ESPRIMONT	15
3.2.3	Marie BOUET	16
4	Bibliographie et sitographie	17
4.1	Vue d'ensemble du projet	17
4.2	Solveur	17
4.3	Réseau de neurone	17
4.4	Traitement d'image	17

1 Introduction

1.1 Qu'est ce qu'un OCR ?

Les technologies informatiques sont en constante évolution d'année d'année en année. L'une d'entre elle a été la reconnaissance optique de caractère, autrement dit OCR (Optical Character Recognition), qui est au coeur de notre projet pour ce troisième semestre à EPITA.

Cette technologie est présente dans de nombreux domaines divers et variés de la vie quotidienne, autant pour les professionnels que pour les particuliers. Elle peut servir pour scanner des plaques d'immatriculation, numériser des documents conséquents comme des livres, dans le domaine bancaire pour le traitement des chèques, ou encore transformer une écriture manuscrite en temps réel sur un ordinateur, le tout sans aucune intervention humaine.

Nous retrouvons ce système de reconnaissance de caractères (OCR) chez Google. Par exemple, avec leur application "Google Traduction" permettant de scanner un texte manuscrit ou non, à l'aide de l'appareil photo d'un mobile ou d'une webcam, afin de le traduire instantanément dans le langage souhaité par l'utilisateur.



Aperçu de l'utilisation d'un OCR avec Google traduction

Vous l'aurez compris avec ce que nous venons de dire que l'OCR nous simplifie de nombreuses tâches qui peuvent paraître longues sur la durée. Ainsi, nous allons maintenant vous expliquer comment nous l'avons implémenté dans notre projet, de ce semestre, tout au long de ce rapport.

1.2 Notre équipe

Notre équipe pour ce projet se compose de trois étudiants en deuxième année de classe préparatoire intégrée à EPITA. Notre équipe se compose donc de Maxime Cambou, Théo Esprimont et Marie Bouet.

1.2.1 Maxime CAMBOU

Nous voici lancés dans un tout nouveau projet. En effet, nous avons tous participé au premier projet de grande envergure l'année dernière en SUP et ce nouveau projet est donc la suite logique des événements.

De mon côté, j'ai énormément apprécié le projet jeu vidéo de l'année dernière et c'est donc avec de l'enthousiasme que j'aborde cette nouvelle expérience. Le fait d'utiliser un langage de programmation différent rend le tout un peu plus compliqué, mais cela fait partie du jeu !

Ce projet aura également pour but de fortifier considérablement nos connaissances en C, et donc d'étendre encore plus notre savoir mais aussi d'accentuer notre efficacité lorsque l'on travaille en groupe.

1.2.2 Théo ESPRIMONT

Ayant redoublé ma deuxième année de prépa, j'ai déjà effectué ce projet. Cependant, j'ai voulu le refaire avec un nouveau groupe pour m'améliorer. En effet, l'année dernière, je n'avais pas les mêmes connaissances dans le langage de programmation C qu'aujourd'hui.

J'attends de ce projet qu'il m'aide à me donner une nouvelle vision du travail de groupe car je suis avec de nouvelles personnes que je ne connais pas. J'espère aussi pouvoir encore m'améliorer un peu plus dans le langage C.

1.2.3 Marie BOUET

Ayant déjà participé l'année dernière à un projet de groupe et ayant vraiment apprécié travailler en groupe, j'avais vraiment hâte de commencer ce nouveau projet. De surcroît, étant une grande fan de sudoku, j'ai été particulièrement captivée par ce projet.

J'espère que ce projet m'aidera à progresser dans le langage de programmation C et me donnera, encore, une bonne expérience de travail de groupe et une vision différente de celui que j'ai pu expérimenter l'année passée avec le projet jeu vidéo.

1.3 Le projet

Nous avons comme projet de troisième semestre de créer un logiciel de reconnaissance de caractères (OCR) afin de pouvoir résoudre un sudoku, le tout automatiquement. En effet, notre logiciel doit prendre une image de sudoku quelconque donnée par l'utilisateur, peu importe sa luminosité, sa netteté et son cadrage, et lui renvoyer la solution de cette grille en une fraction de seconde. L'utilisateur n'en est pas conscient, mais il y a évidemment plusieurs étapes intermédiaires cachées avant d'avoir le rendu final. Chaque étape sera détaillée et illustrée dans les différentes parties ci-dessous.

1.4 Répartition des tâches

Pour être le plus efficace possible, nous avons, dès le départ, établi un tableau avec la répartition des tâches. Pour cette répartition des tâches, elle a été faite au bon vouloir de chacun. Au départ, Maxime s'est occupé de créer le GITHUB pour le projet, Marie a créé un serveur Discord pour pouvoir discuter plus facilement entre nous. En réalité, ce tableau n'a pas été totalement respecté puisque chacun de nous a rencontré des problèmes de son côté, nous devons donc de ce fait nous entraider en permanence.

De plus, certaines parties ont été commencées par certains d'entre nous, puis terminées par leur suppléant : notre projet a avant tout été, pour le moment, un travail de groupe, solidaire entre nous. Le tableau n'est donc qu'approximativement ce que chaque personne du groupe a pu faire. Néanmoins, tout le monde avait toujours du travail à faire puisqu'il a été réparti de façon optimale, car certaines tâches étaient bien plus longues et conséquentes que d'autres.

Voici le tableau récapitulatif des tâches requises pour la première soutenance avec en correspondance la personne du groupe responsable de cette tâche ainsi que son éventuel assistant.

	Maxime	Théo	Marie
Solveur			Responsable
Chargement de l'image			Responsable
Suppression des couleurs			Responsable
Prétraitement	Responsable		
Détection de la grille	Responsable		Suppléant
Détection de la position des cases	Responsable		
Découpage de l'image			Responsable
XOR		Responsable	

2 Avancement du projet

2.1 Traitement d'image

Une des parties les plus importantes de ce projet était d'importer l'image de notre grille à résoudre puis de la transformer afin que notre reconnaissance de caractères (réseau de neurones) puisse fonctionner correctement. Pour ce faire, nous avons travaillé en 4 étapes distinctes : l'importation de la grille, la transformation en niveau de gris, une application d'un flou Gaussien et pour terminer une transformation en noir et blanc.

Le traitement de l'image a été rendu possible grâce à l'utilisation de la librairie "Simple Direct-Media Layer", SDL. Cette librairie est une librairie de développement software multiplateforme. Elle fournit entre autres de nombreux outils afin de manipuler facilement des images. C'est pour cela que nous avons choisi d'utiliser cette librairie et non une autre.



Logo de la librairie SDL

De surcoût, ayant déjà eu un travaux pratiques utilisant cette librairie, elle ne nous était donc pas inconnue et nous avions déjà certaines connaissances.

2.1.1 Chargement de l'image

Avant de commencer à transformer notre image, nous avons dû l'importer dans notre logiciel afin de pouvoir y appliquer nos différentes transformations. Pour ce faire, nous avons utilisé la librairie SDL qui permet de travailler sur les images. Nous commençons donc par télécharger l'image puis nous appliquons à une sortie vidéo.



Image du sudoku importée dans la sortie vidéo

2.1.2 Transformation en niveau de gris

Afin de transformer notre image en niveau de gris, nous avons appliqué une fonction à chaque pixel la composant. Un pixel est composé de trois couleurs : le rouge, le vert et le bleu. Nous faisons donc une moyenne des couleurs du pixel à l'aide d'une formule mathématique, puis nous appliquons cette moyenne à la place des trois couleurs du pixel.

Pour que nous ayons entièrement l'image en niveau de gris, on applique donc cette fonction à l'ensemble des pixels qui composent notre photo initiale.



Image du sudoku après la transformation en niveau de gris

2.1.3 Application d'un flou

L'application d'un flou va nous permettre d'améliorer la netteté de notre image en sortie. Pour ce faire, le flou va agir en deux étapes.

Premièrement, il va recréer une image où chaque pixel va être transformé en la moyenne des pixels l'entourant.

Puis, nous allons pour chaque pixel de notre image de départ lui ôter la valeur du pixel de l'image créé. Cela va permettre d'augmenter le contraste entre les caractères ou les lignes que nous allons vouloir trouver par la suite.



Image du sudoku après l'application d'un flou

2.1.4 Image en noir et blanc

Afin de terminer de supprimer les couleurs de l'image, nous allons appliquer une fonction qui va regarder chaque pixel. Si le pixel est clair, il sera transformé en blanc. Dans le cas contraire, il sera transformé en noir.

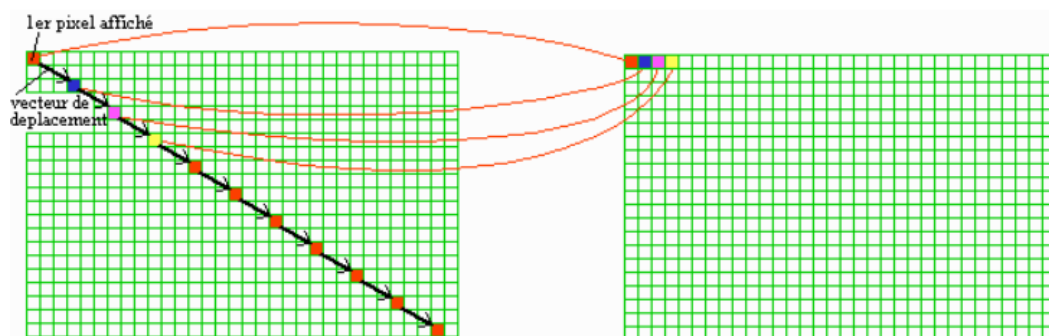


Image du sudoku après l'application du noir et blanc

2.1.5 La rotation manuelle de l'image

L'image que donne l'utilisateur à notre programme n'est évidemment pas parfaite. Il est donc important de la traiter dans de bonnes conditions. Pour l'image traitée que nous allons renvoyer, il est important de la faire pivoter, elle aussi.

Pour ce faire, dans un premier temps pour cette première soutenance, la rotation se fait manuellement. C'est donc à l'utilisateur d'entrer lui-même la valeur de l'angle en degré qu'il souhaite appliquer à l'image lors de sa rotation. Pour être plus précis sur la méthode que nous employons, nous commençons par créer une nouvelle surface, qui sera la base de l'image de rendu. Ensuite, nous parcourons chaque pixel de notre image initiale à l'aide d'un vecteur, en les ajoutant un à un sur la surface créée précédemment avec l'angle voulu, par l'utilisateur, en utilisant un cosinus ou un sinus. L'image ci-dessous montre le phénomène décrit.



Phénomène de roattoin avec RotoZoom de la librairie SDL

Pour simplifier cette tâche, nous avons utilisé une fonction liée à la librairie SDL, souvent utilisée pour effectuer des zooms sur une image : Rotozoom. Dans notre cas, nous nous limitons à la rotation.

2.1.6 Détection de la grille

Pour détecter la grille, nous avons utilisé la transformée de Hough, une technique de reconnaissance de forme. Cette technique permet de repérer les lignes sur une image en noir et blanc.

On enregistre donc les points qui sont traversés par le plus de droites. Ceci permet, alors, de mettre en évidence les lignes présentes dans les images. Cela nous permettra alors de récupérer plus facilement les cases du sudoku.

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

Image avant le tracé des droites

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

Image après le tracé des droites

2.2 Le réseau de neurones

Le réseau de neurones se décompose en 2 parties distinctes. La première partie est l'opération XOR (appelée plus communément "ou exclusif") et la seconde partie correspond à l'apprentissage. Ce réseau de neurones du ou exclusif est composé de 5 neurones : 2 neurones d'entrée, 2 neurones dans la couche cachée et 1 neurone de sortie.

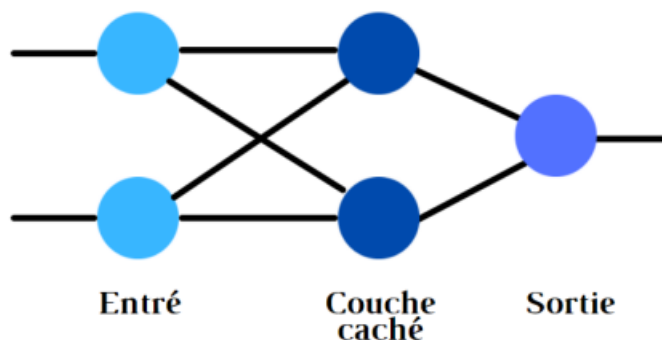


Schéma simplifié d'un réseau de neurone

2.2.1 Le XOR (ou exclusif)

Pour pouvoir utiliser l'opération décrite ci-dessus, il faut appeler l'exécutable fichier.exe avec deux arguments. Ces derniers doivent soit prendre la valeur 0, soit la valeur 1, et si ce n'est pas le cas, le programme retourne alors

un message d'erreur. Une fois la fonction appelée avec les bons paramètres, le réseau va alors pouvoir faire son travail.

Pour commencer, le réseau va lire un fichier texte (fichier.txt) pour charger les valeurs des différents neurones. Dans ce fichier se trouve les poids des différentes liaisons ainsi que les valeurs des différents neurones. Une fois les valeurs de notre réseau chargé, nous allons pouvoir commencer les calculs.

Pour ces calculs, nous avons une fonction "sigmoid" qui est la fonction d'activation, elle nous permet de modifier la valeur d'entrée suivant les poids des liaisons des neurones de la précédente couche vers notre neurone. La valeur du neurone sur lequel nous voulons faire l'opération est le résultat de nos précédents neurones.

Nous allons donc effectuer cette opération sur nos deux neurones de la couche cachée ainsi que sur notre neurone de sortie. Une fois ces trois calculs effectués, nous avons comme résultat de nos derniers calculs un nombre à virgule compris entre zéro et un. Qui sera proche de un pour le calcul zéro XOR un ou un XOR zéro, et qui sera proche de zéro pour le calcul zéro XOR zéro ou un XOR un. En résumé :

X	Y	$S = X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité du ou exclusif

Cependant, pour que notre réseau soit le plus performant possible, il faut l'entraîner.

2.2.2 L'apprentissage

Pour pouvoir appeler l'apprentissage, il faut passer en paramètre « 1 » au moment de l'appel de la fonction. Une fois cette condition remplie, le programme va pouvoir détecter que l'on veut apprendre.

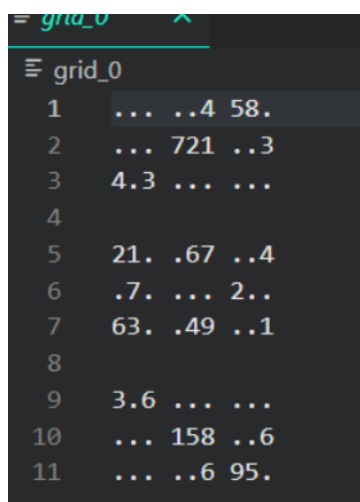
L'apprentissage commence alors par les mêmes étapes que la sous-partie décrite précédemment, c'est-à-dire : lire le fichier où se trouvent les valeurs puis les lire, faire les calculs pour les 3 neurones et enfin, récupérer les valeurs. Une fois cette partie faite, nous allons maintenant pouvoir apprendre.

Pour apprendre, nous utilisons la méthode de la propagation arrière. Cette méthode consiste à calculer l'erreur par rapport à un résultat attendu, puis le gradient d'erreur, puis avec un degré d'apprentissage, nous allons donc pouvoir mettre à jour nos différentes valeurs des poids des différentes liaisons ainsi que les valeurs des différents neurones.

Nous allons alors entraîner notre réseau avec les 4 valeurs connues et cela tant que la somme des erreurs est trop grande. Une fois que le réseau est assez entraîné, il va pouvoir sauvegarder les nouvelles valeurs dans un fichier texte.

2.3 Le solveur

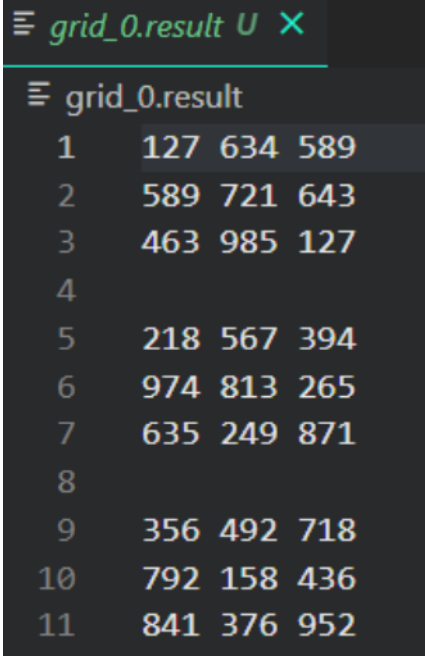
Le solveur est la dernière partie de notre programme à être appelée, c'est celui ci même qui va prendre en entrée un fichier texte comportant les chiffres déjà présents sur la grille du sudoku repéré par le biais du réseau de neurones ainsi que des points aux cases ne contenant pas de chiffres.



```
grid_0
1  ... ..4 58.
2  ... 721 ..3
3  4.3 ... ..
4
5  21. .67 ..4
6  .7. ... 2..
7  63. .49 ..1
8
9  3.6 ... ..
10 ... 158 ..6
11 ... ..6 95.
```

Sudoku avant la résolution

Avec ce fichier entré, le programme va pouvoir résoudre cette grille grâce à un algorithme de retour sur trace. En effet, cet algorithme permet de tester systématiquement l'ensemble des numéros qui peuvent potentiellement être mis dans la case. Ce procédé consiste à sélectionner une case vide de la grille, et pour chaque chiffre possible, à tester récursivement si c'est une solution viable ou non, c'est-à-dire savoir si la solution peut fonctionner ou non. Si aucune solution n'est trouvée, la méthode abandonne et revient sur les affectations qui auraient été faites précédemment.



```
≡ grid_0.result U X
≡ grid_0.result
1 127 634 589
2 589 721 643
3 463 985 127
4
5 218 567 394
6 974 813 265
7 635 249 871
8
9 356 492 718
10 792 158 436
11 841 376 952
```

1	127	634	589
2	589	721	643
3	463	985	127
4			
5	218	567	394
6	974	813	265
7	635	249	871
8			
9	356	492	718
10	792	158	436
11	841	376	952

Sudoku après la résolution

Le nom de cette méthode, "retour sur trace", vient de ce principe de retour lorsque que la solution n'est pas viable. En d'autres termes, le retour sur traces est un parcours profondeur de l'arbre décisionnel lié au problème qu'est la résolution de la grille de sudoku.

3 Conclusion

3.1 Expérience globale

Pour cette soutenance, nous sommes assez fiers de ce que nous avons pu produire. En effet, nous avons bien avancé dans le projet. De plus, voir le projet commencer à prendre forme est assez réjouissant et satisfaisant pour chacun d'entre nous

Pour la partie traitement de l'image, il nous reste encore pas mal de petits problèmes à régler et certaines choses à améliorer. Cependant, nous avons déjà bien avancé et nous n'avons pas pris de retard sur cette partie selon ce qui était prévu.

Nous sommes plutôt fiers du fait que nous avons un réseau de neurones fonctionnel et cela pas uniquement en XOR, c'est-à-dire le ou exclusif.

Nous sommes en conclusion plutôt fiers de ce que nous avons produit et nous souhaitons continuer sur la même voie.

3.2 Expériences personnelles

3.2.1 Maxime CAMBOU

En cette fin de première soutenance, grâce à notre organisation de travail, nous avons pu grandement avancer sur le projet et ainsi remplir les critères attendus. Notre groupe est efficace et propice au travail, ce qui améliore notre productivité.

Pour ce qui est de mon ressenti personnel, j'ai déjà appris énormément de nouvelles notions. C'est très intéressant et, comme prévu, renforce nos connaissances que ce soit en C, mais également en programmation en général. Github nous permet d'avoir une bonne structure de travail et m'aide à me concentrer uniquement sur mes tâches.

Il nous faut juste garder ce rythme de travail afin de ne pas prendre de retard et donc de mener à bien notre projet.

3.2.2 Théo ESPRIMONT

Après la première partie de ce projet, je suis plutôt content de moi même et du travail que moi ainsi que mon groupe avons réussi à produire. Ayant fait le réseau de neurones, je suis assez fier d'avoir réussi à le faire assez rapidement et qu'il soit déjà fonctionnel.

De surcroît, j'apprécie vraiment le fait de refaire le projet avec mes connaissances acquises l'année dernière.

Enfin, j'ai hâte de voir comment le projet va continuer d'avancer mais je pense que cela va continuer en bonne voie à l'aide d'une très bonne entente dans le groupe ainsi que l'envie de chacun de voir ce projet mené à bien.

3.2.3 Marie BOUET

Pour cette première partie du projet, je suis satisfaite de ce que nous avons pu produire en assez peu de temps. En effet, entre le moment où nous avons commencé le projet et cette première soutenance, il ne s'est pas passé beaucoup de temps. Je suis contente aussi car il règne une très bonne ambiance de travail dans le groupe et nous sommes assez productif malgré le fait que nous n'avons pas les mêmes horaires de travail.

De plus, je suis satisfaite de moi même car j'ai vu une vraie progression dans sa maîtrise du langage de programmation C. De surcroît, j'ai pu mieux comprendre le fonctionnement du logiciel GITHUB. En effet, ne l'ayant pas utilisé l'année dernière lors du projet jeu vidéo, ce site m'était alors inconnu au début de ce nouveau projet, je ne dirai pas que je le connais vraiment mais maintenant, je sais l'utiliser.

J'espère que nous continuerons sur cette même lancée par la suite et que le travail de groupe ainsi que l'ambiance en son sein resteront agréables et dans une bonne dynamique de travail.

4 Bibliographie et sitographie

4.1 Vue d'ensemble du projet

<https://liorsinai.github.io/coding/2021/08/10/sudoku-reader-part1-intro.html>

<https://www.codeproject.com/Articles/238114/Realtime-Webcam-Sudoku-Solver>

4.2 Solveur

<https://www.youtube.com/watch?v=9aMUyoYDI-0>

<https://en.wikipedia.org/wiki/Backtracking>

4.3 Réseau de neurone

<https://towardsdatascience.com/simple-neural-network-implementation-in-c-663f51447547>

http://www8.umoncton.ca/umcm-cormie_gabriel/SystemesIntelligents/GIND5439_Chapitre6.pdf

<http://neuralnetworksanddeeplearning.com/>

4.4 Traitement d'image

https://fr.wikipedia.org/wiki/Niveau_de_gris

<https://docs.gimp.org/2.8/fr/plugin-gauss.html>