Rapport de Soutenance II

Novembre 2023

OCR Sudoku Solver

par Toujours à l'heure

Djallil Baizid	Dorian Wolff
Simon Arbiser	Rémi Brenaut

Table des matières

Première partie

Introduction

Ce projet a pour but de résoudre une grille de sudoku à partir d'une image. Pour le réaliser, nous avons dû implémenter un algorithme de résolution de sudoku, une reconnaissance d'image (binarisation , détection de grille et un découpage de l'image) et enfin un réseau de neurones (détection de chiffre dans une image).

Pour cela nous allons vous montrez le déroulement de ce projet à travers ses membres, son état actuel ainsi que le fonctionnement des différents aspects techniques de ce projet.

Deuxième partie

Membres de l'équipe

1 Login

Équipe	E-mails
Djallil Baizid	djallil.baizid@epita.fr
Dorian Wolff	dorian.wolff@epita.fr
Simon Arbiser	simon.arbiser@epita.fr
Rémi Brenaut	remi.brenaut@epita.fr

2 Présentation

2.1 Djallil

Ladies and Gentlemen ! Mon nom est **Djallil BAIZID**, et vous vous apprêtez à assister à l'aube d'une nouvelle ère ! Mon but pour ce super projet de la mort qui tue n'est ni la fortune, ni la gloire et encore moins le pouvoir, mais plutôt la satisfaction de mettre au monde ce qui me passe par la tête. Je vais faire de mon mieux pour vous proposer une expérience rocambolesque tout en acquérant des nouvelles compétences ! El Psy Kongroo...

2.2 Dorian

Enchanté, je suis Dorian Wolff. Avec ce projet je souhaite approfondir mes connaissances en C en commençant par l'UI ainsi que la détection de grille. A nous quatre, je suis convaincu que nous pouvons vous apporter un projet qui s'apparente à un produit qualitatif et fonctionnel.

2.3 Simon

Je m'appelle Simon Arbiser, mon but dans ce projet est de faire un réseau de neurones qui fonctionne en C. Ayant déjà touché au réseau de neurones par le passé, je pense être à même de réussir à en coder un uniquement en C.

2.4 Rémi

Salut, je m'appelle Rémi ! Je trouve ce projet très intéressant, car il me permettra d'améliorer grandement mes connaissances en C. J'ai hâte d'en apprendre plus sur la création de réseaux de neurones et sur le traitement d'image. Ce projet est aussi l'occasion de rencontrer de nouvelles personnes.

Troisième partie

Récapitulatif du cahier des charges

3 Le Projet

L'objectif de ce projet est de réaliser un logiciel de type OCR (Optical Character Recognition) qui résout une grille de sudoku. L'application prendra donc en entrée une image représentant une grille de sudoku et affichera en sortie la grille résolue.

Dans sa version définitive, l'application devra proposer une interface graphique permettant de charger une image dans un format standard, de la visualiser, de corriger certains de ses défauts, et enfin d'afficher la grille complètement remplie et résolue. La grille résolue devra également pouvoir être sauvegardée.

L'application devra aussi posséder un aspect apprentissage, qui pourra être séparé de la partie principale, et qui permettra d'entraîner le réseau de neurones, puis de sauvegarder et de recharger le résultat de cet apprentissage.

4 Fonctionnement

Le traitement effectué par l'application sera approximativement le suivant :

— Chargement d'une image ;

— Suppression des couleurs (niveau de gris, puis noir et blanc) ;

— Prétraitement ;

— Détection de la grille ;

— Détection des cases de la grille ;

— Récupération des chiffres présents dans les cases ;

— Reconnaissance de caractères (ici les chiffres) ;

— Reconstruction de la grille ;

— Résolution de la grille ;

— Affichage de la grille résolue ;

— Sauvegarde de la grille résolue.

Quatrième partie

Organisation

5 Répartition des tâches

	Solver	Réseau de Neurones	Détection d'Images	Interface Utilisateur
Djallil			1	1
Dorian			1	1
Simon		1		
Rémi	1			

Tableau de répartition des tâches

6 Calendrier et avancement des tâches

6.1 Première soutenance

Pour illustrer l'accomplissement de nos objectifs pour la première soutenance, voici un comparatif d'avancement des tâches entre les estimations de notre cahier des charges et ce que nous avons réalisé.

Comparatif d'avancement par tâches :

Tâches\Deadlines	Objectifs	Avancée
Sudoku solver	100%	100%
Traitement de l'image	80%	100%
Réseau de neurones	50%	95%
Interface utilisateur	50%	80%

6.2 Deuxième soutenance

Comparatif d'avancement par tâches pour la deuxième soutenance :

Tâches\Deadlines	Objectifs	Avancée
Sudoku solver	100%	100%
Traitement de l'image	100%	100%
Réseau de neurones	100%	100%
Interface utilisateur	100%	100%

Cinquième partie

Projet

7 Sudoku Solver

La première étape pour créer une IA qui résout des sudokus est le solver de sudoku. L'objectif est de concevoir un programme efficace, capable de résoudre des grilles de Sudoku à partir d'un fichier d'entrée spécifique, écrit en langage C, qui prend en compte les contraintes du format de fichier imposé.

7.1 Structure et composition

Ce programme en C, permet de résoudre un sudoku de façon récursive. Il est composé de deux fonctions : *isValid* qui vérifie si la position du chiffre dans la grille est valide, et *solve* qui remplit la grille. Pour faciliter le parcours de la grille de sudoku, on créé deux structures.

La structure *Cell* représente une case de la grille de sudoku et possède deux attributs : *value* qui renvoie la valeur du chiffre présent dans la case, et *fixed* qui marquer les chiffres déjà présents dans la grille d'origine. La structure *SudokuGrid* est un double tableau de Cell.

Ainsi, on peut facilement accéder à n'importe quelle case de la grille de sudoku, savoir quel chiffre y est présent et si ce chiffre était déjà présent dans la grille d'origine.

7.2 Fonctionnement

Le programme prend en entrée le nom d'un fichier représentant une grille de Sudoku au format spécifié. Il lit ce fichier et génère une structure de données en mémoire pour représenter la grille.

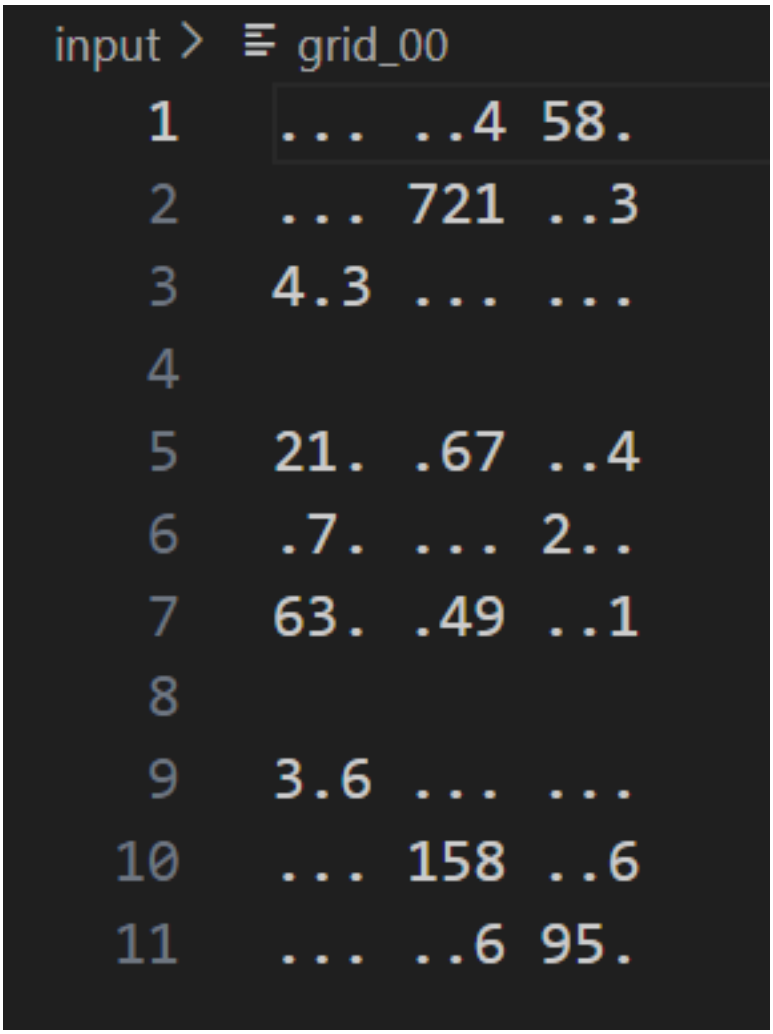


FIGURE 1 – Fichier d'entrée du solver

Avant de commencer la résolution, le programme vérifie la validité de la grille selon les règles du Sudoku. Il s'assure qu'aucun chiffre n'est répété dans les lignes, colonnes ou blocs.

L'algorithme de résolution repose sur une approche de backtracking. Le solver tente différentes combinaisons de chiffres pour chaque case vide, vérifiant la validité à chaque étape, jusqu'à trouver une solution ou déterminer qu'aucune solution n'est possible.

Une fois la grille remplie, le solver génère un fichier de sortie portant le même nom que le fichier d'entrée, mais avec l'ajout de l'extension ".result". Ce fichier contient la grille résolue au même format que le fichier d'origine.

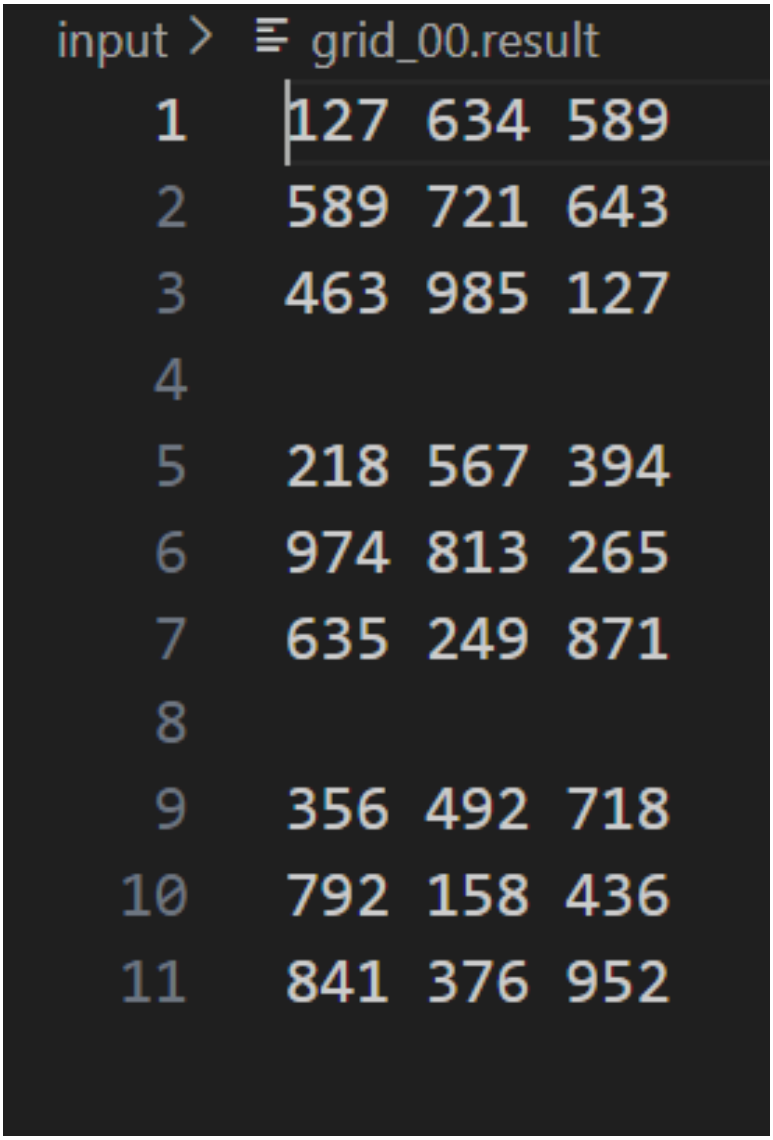


FIGURE 2 – Fichier résultat du solver

Le solver constitue une composante essentielle du projet Sudoku Solver, offrant une solution robuste et efficace pour la résolution de grilles de Sudoku conformes au format spécifié.

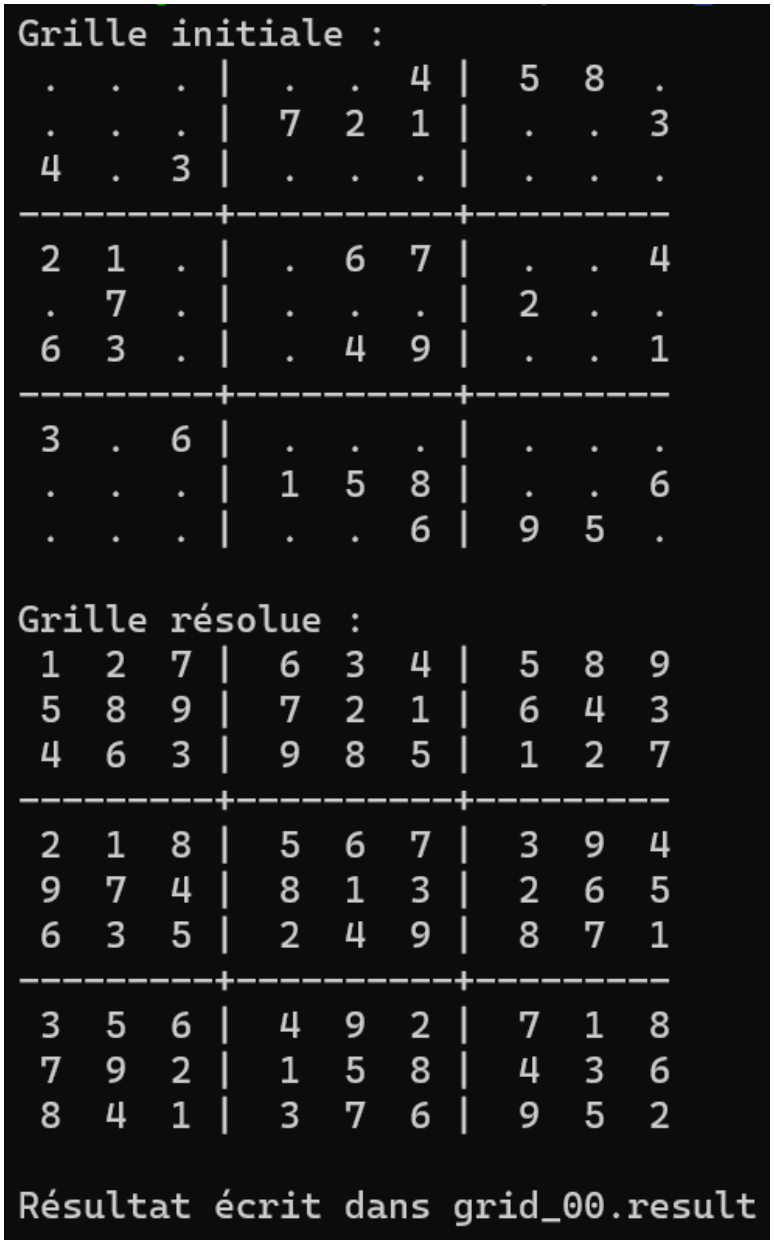


FIGURE 3 – Fonctionnement pas à pas du solver

7.3 Avancement

L'intégralité du solver a été réalisé pour la première soutenance. Cet avancement respecte le cahier des charges et les délais fixés par le groupe.

Pour la deuxième soutenance, l'objectif était d'intégrer le solver au reste du projet. Aucune modification n'a été effectuée depuis la première, le code étant fonctionnel.

7.4 Problèmes rencontrés

Aucun problème majeur n'a été rencontré lors de l'implémentation du solver. Les défis majeurs de cette partie du projet étaient de trouver un algorithme fonctionnel et efficace pour résoudre le sudoku, mais aussi d'apprendre la lecture et l'écriture de fichier en C.

8 Traitement d'image

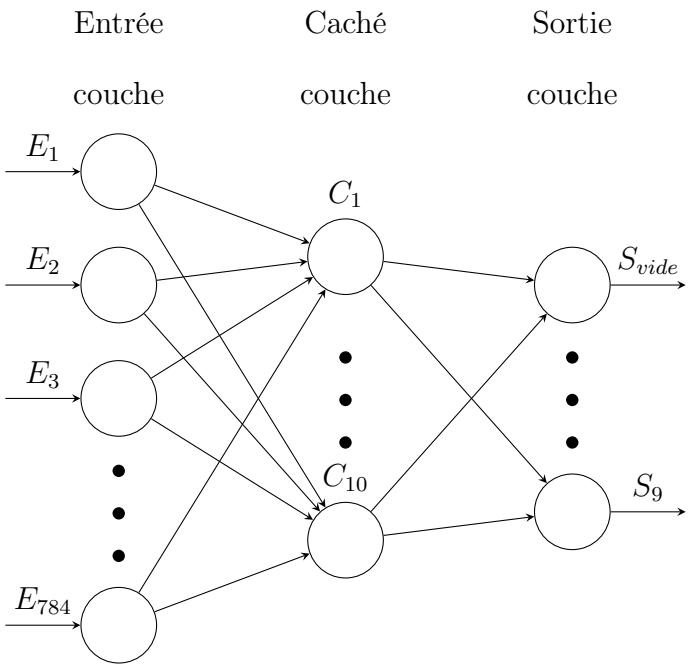
Le traitement d'image consiste en la modification et l'analyse d'images pour en extraire des informations significatives ou les améliorer. La conversion en niveaux de gris (grayscale) est une technique qui transforme une image couleur en une image en nuances de gris, où chaque pixel a une valeur représentant l'intensité lumineuse. Cela simplifie l'image tout en conservant les informations essentielles.

La détection de grille dans un projet de résolution de Sudoku à partir d'une image implique la capacité à identifier et isoler la grille du Sudoku dans l'image initiale. Cela peut nécessiter des techniques de traitement d'image pour détecter les lignes et les contours, et diviser l'image en cases de grille distinctes. Cette étape est cruciale pour extraire chaque cellule du Sudoku individuellement afin de reconnaître les chiffres qui y sont écrits et résoudre le puzzle.

Cette partie a donc pour but d'isoler et de simplifier de nouvelles images des cases présentes sur l'image du sudoku et de les envoyer au réseau de neurones qui s'occupera de les reconnaître.

Cette section est également chargée de la rotation de l'image (manuelle et automatique) afin de correctement reconnaître la grille présente dans l'image.

9 Réseau de neurones



Afin de créer un réseau de neurones en C, nous avons fait le choix de recoder une implémentation de matrice.

Le réseau de neurones pour la reconnaissance de chiffres est composé de trois couches, une couche d'entrée, une couche cachée et une couche de sortie. Les biais et les poids de chaque couche est en implémentation matricielle, La couche cachée et la couche de sortie ont donc une matrice poids et une matrice biais. De plus, ces couches-là ont également une fonction d'activation, respectivement un ReLU et un Softmax.

Il y a donc 784 neurones pour la couche 1 (car les images ont une dimension de 28 par 28), 10 neurones pour la couche cachée et 10 neurones pour la couche de sortie.

9.1 Apprentissage

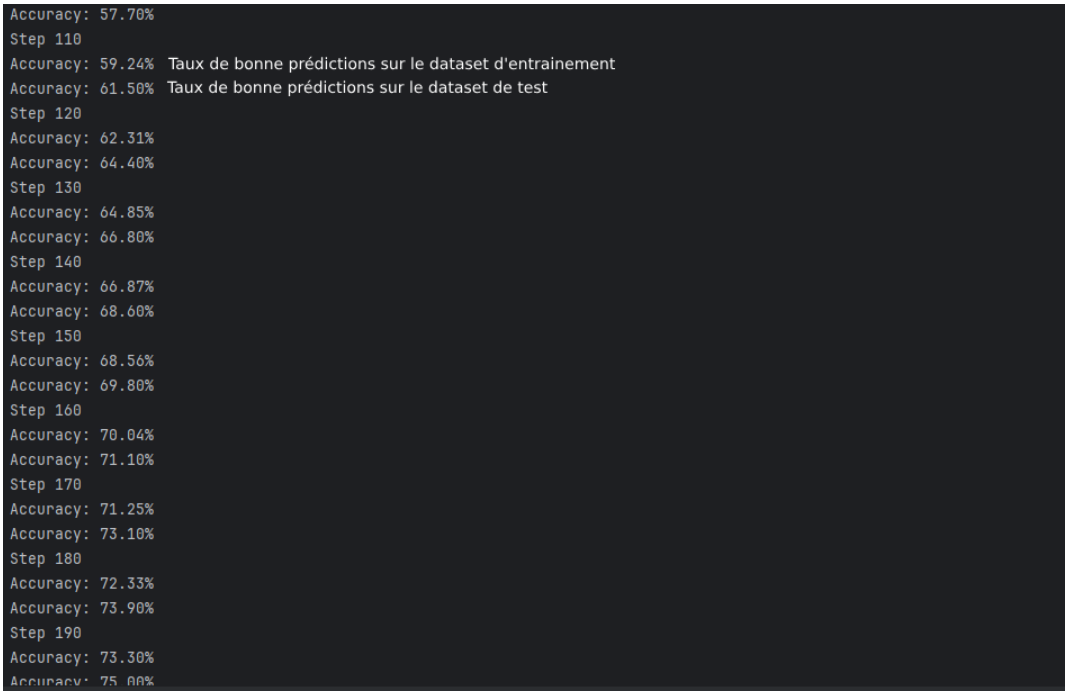


FIGURE 4 – Le réseau en train d'apprendre

Pour faire apprendre le réseau de neurones, nous utilisons la technique de la descente de gradient. La descente de gradient consiste, il y a d'abord, une étape de prédiction du réseau, suivi par un calcul de l'erreur et enfin de mettre à jour les paramètres pour chaque couche.

9.2 Résultat

Le réseau de neurones fonctionne très bien pour les chiffres écrit à la main, avec un taux de bonne prédiction de 85% pour le dataset d'entrainement et de test. Et à un taux de 70% sur les chiffres imprimés pour le dataset d'entrainement et de test.

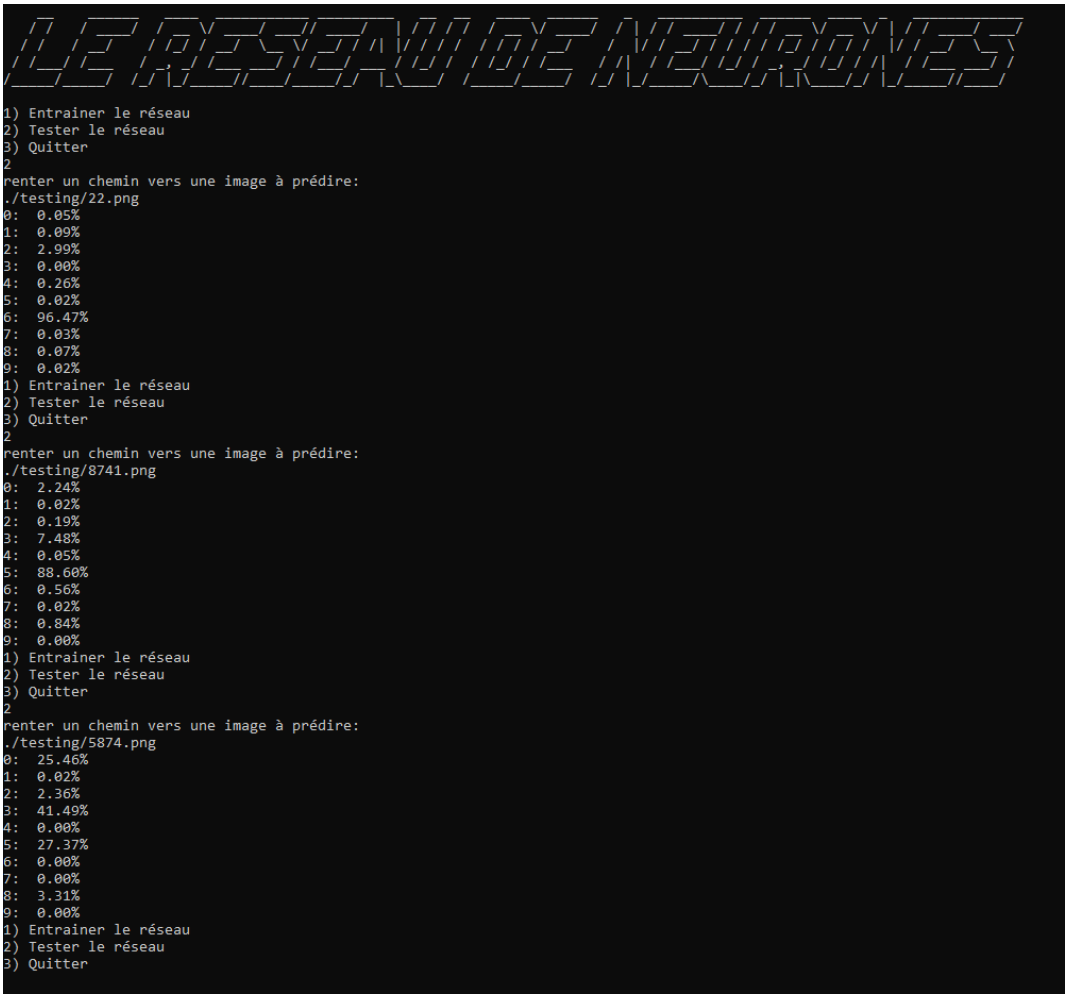


FIGURE 5 – Interface pour entrainer et tester le réseau

10 Interface Utilisateur

L'interface utilisateur (UI) développée en C avec GTK et SDL2 offre une expérience interactive aux utilisateurs.

10.1 Première soutenance

Cette interface comporte plusieurs boutons, chacun attribué à une fonctionnalité spécifique. L'un des boutons permet d'ouvrir un explorateur de fichiers, permettant à l'utilisateur de sélectionner un fichier à traiter. Un autre bouton est dédié au prétraitement de l'image, offrant des fonctionnalités pour manipuler et préparer l'image sélectionnée. Les autres boutons sont pour l'instant configurés comme des remplisseurs, prêts à être associés à des fonctionnalités futures, et ils servent actuellement de réservation pour des améliorations ou des fonctionnalités à venir.

10.2 Deuxième soutenance

L'objectif de la deuxième soutenance était de compléter la version fournie pour la première soutenance.

Ainsi, les boutons non-utilisables lors de la première soutenance ont été affectés à leurs fonctions respectives. Le bouton Preprocess converti l'image chargée en une image en noir et blanc. Un slider a été ajouté pour permettre à l'utilisateur de faire pivoter l'image manuellement. Le bouton Automatic Rotation fait pivoter limage automatiquement pour la mettre droite. Le bouton Neural Network lance la reconnaissance de grille. Il affiche ensuite ces lignes en rouge et leurs points de rencontre en bleu sur l'image. Une case Binarisation permet d'afficher ou non toutes les modifications de prétraitement apportées à l'image. Si elle n'est pas cochée, l'image de base est affichée. Il est toujours possible de faire pivoter l'image manuellement après la reconnaissance de lignes. Enfin, le bouton Solve résout la grille de sudoku chargée et affiche la solution proprement à l'écran. Les chiffres présents sur la grille de base sont en noir et les chiffres ajoutés lors de la résolution sont en rouge. Le bouton Save permet de sauvegarder une copie de la grille résolue (l'image avec les chiffres noirs et rouges). Enfin, nous avons ajouté un bouton Quit permettant à l'utilisateur de quitter l'application.

10.3 Avancement

Pour cette deuxième soutenance, les modifications principales apportées à l'interface utilisateur ont été l'affectation des boutons à leurs fonctions. Aucune modification sur les design ou les graphismes de l'interface utilisateur n'ont été apportés. Les nouveautés apportées sont l'ajout du bouton Save et du bouton Quit.

10.4 Produit final

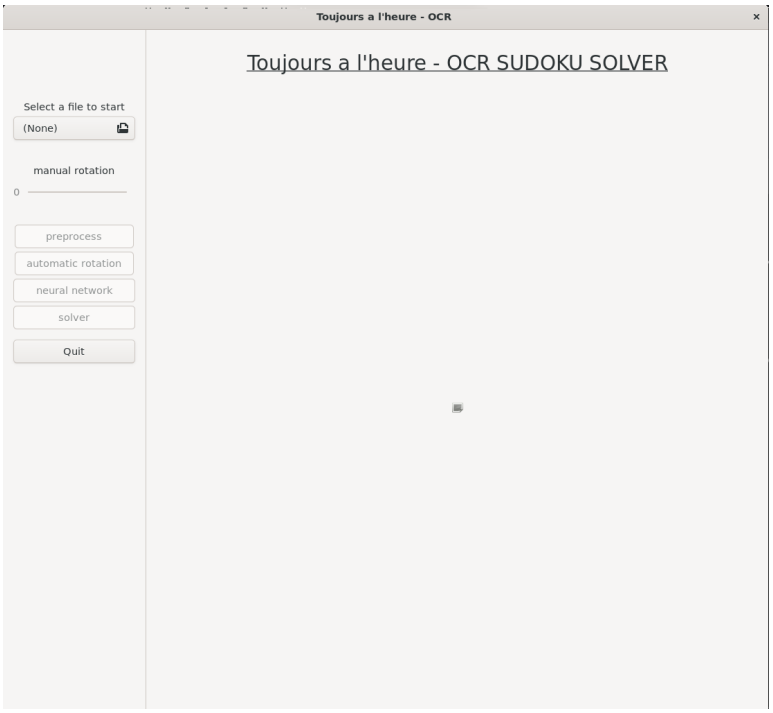


FIGURE 6 – Accueil du l'UI

Seuls les boutons pour sélectionner une image et pour quitter sont accessibles.

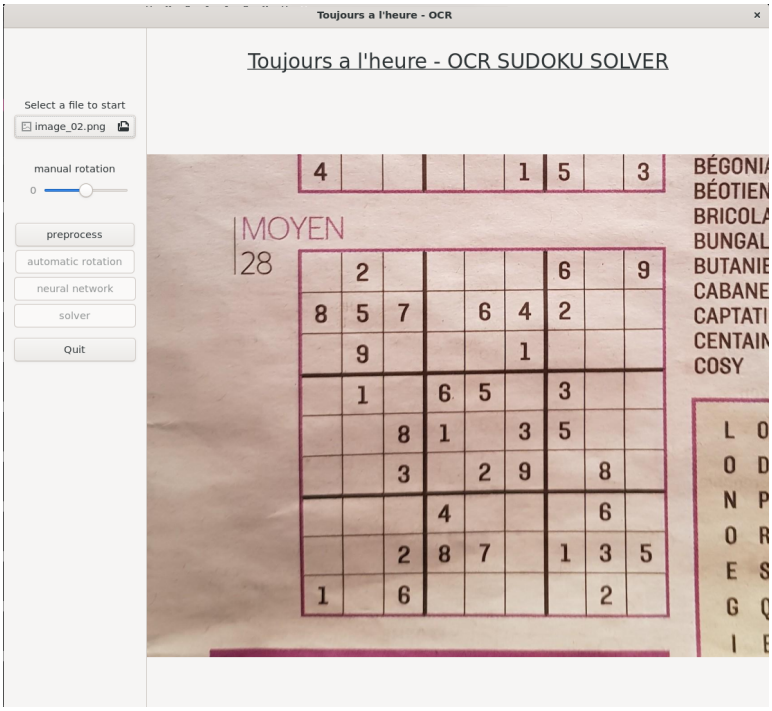


FIGURE 7 – Fonctionnement pas à pas du solver

L'image est chargée en l'état.

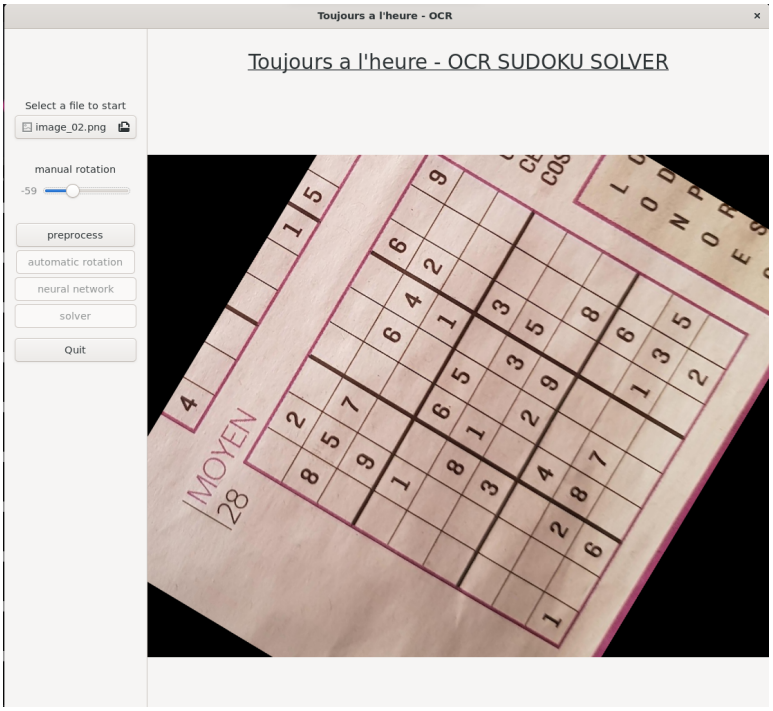


FIGURE 8 – UI après de la rotation de l'image

Le slider de rotation manuelle permet de faire pivoter l'image de angle de -180° à un angle de 180°.

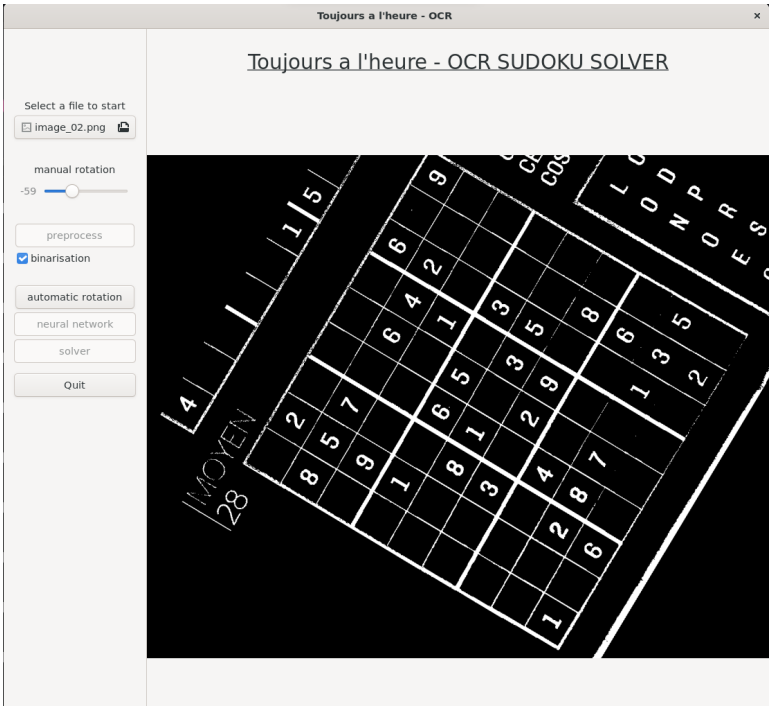


FIGURE 9 – UI après la binairisation

Le bonton Preprocess permet de mettre l'image en noir et blanc.

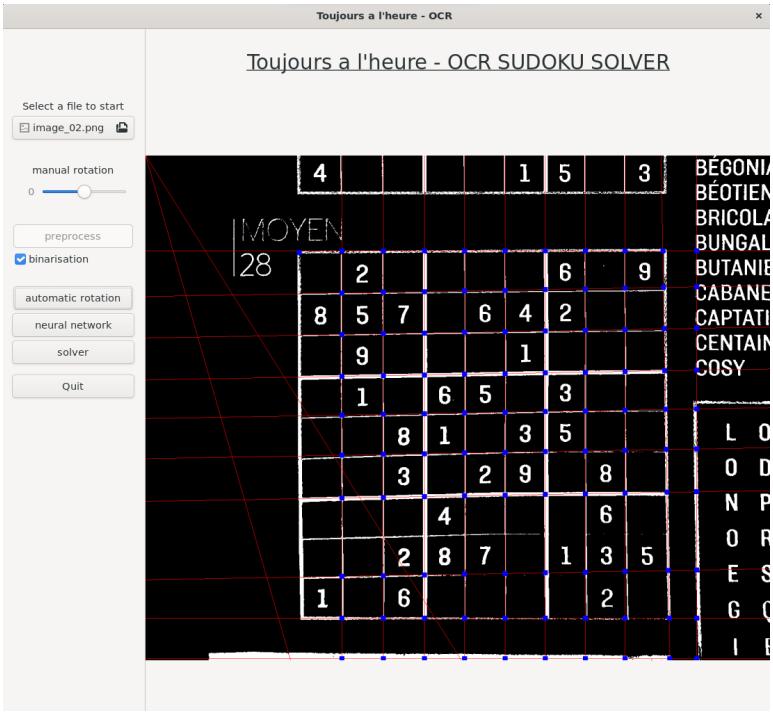


FIGURE 10 – UI après la rotation automatique et la reconnaissance de grille

Le bouton Neural Network détecte la grille de sudoku sur l'image et affiche le résultat.

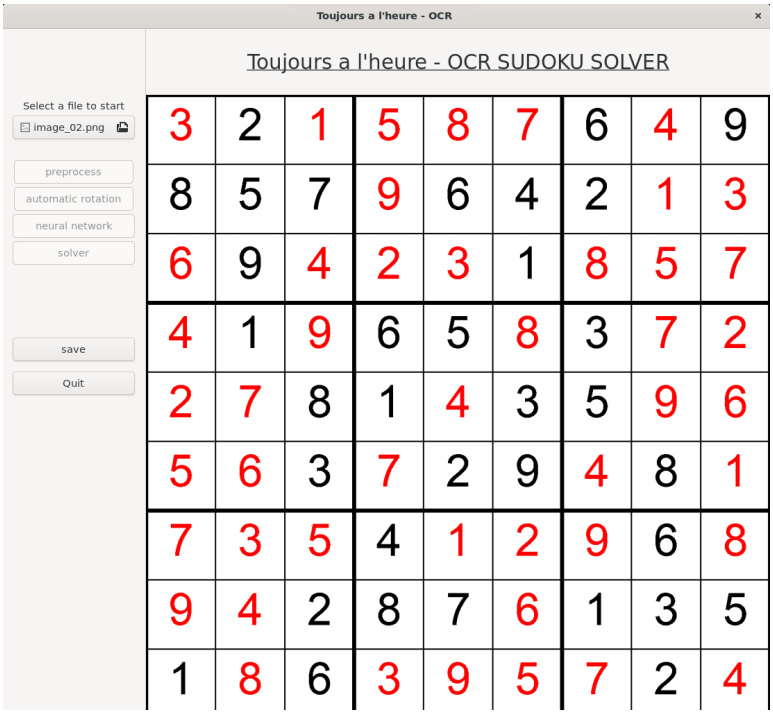


FIGURE 11 – UI après du chargement de l'image

Enfin, le bouton Solve affiche la grille de sudoku résolue proprement en affichant en rouge les chiffres qui ont été ajoutés lors de la résolution.

Sixième partie

Conclusion

Ce projet d'application de résolution d'images de Sudoku représente pour nous une avancée en tant que projet de groupe, en explorant des domaines qui nous étaient jusqu'à présent assez flous. En combinant des techniques de détection de grille, de conversion en niveaux de gris, de rotation automatique d'image et d'analyse de motifs, cette application offre une solution pour résoudre des grilles de Sudoku à partir d'images. En permettant la capture et la transformation d'une image contenant un Sudoku, l'application démontre la capacité à extraire des informations visuelles complexes pour résoudre des casse-tête. Cette solution apporte une nouvelle perspective quant à l'automatisation de la résolution de Sudoku et nous ouvre la voie pour de futur projets dans le domaine du traitement d'image appliqué aux jeux et aux énigmes.

Table des figures