

# Rapport de Projet

L'équipe *CharacTech*

**DUFOUR Guillaume**  
**BONDARENKO Lucas**  
**OUHAIMMED Nassima**  
**MONROC Leonardo**

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation de l'équipe</b>	<b>3</b>
2.1	DUFOUR Guillaume ( <b>chef de projet</b> ) . . . . .	3
2.2	OUHAIMMED Nassima . . . . .	3
2.3	MONROC Leonardo . . . . .	4
2.4	BONDARENKO Lucas . . . . .	5
<b>3</b>	<b>Répartition des charges</b>	<b>6</b>
<b>4</b>	<b>État d'avancement du projet</b>	<b>7</b>
<b>5</b>	<b>Aspects techniques</b>	<b>9</b>
5.1	Bibliothèques et langages utilisées . . . . .	9
5.2	Prétraitement de l'image . . . . .	9
5.3	Découpage de la grille . . . . .	10
5.4	Résolution des mots mêlés . . . . .	11
5.5	Le réseau de neurones . . . . .	11
5.6	Pourquoi ces choix . . . . .	14
<b>6</b>	<b>Conclusion</b>	<b>15</b>

# 1 Introduction

Dans un monde où le numérique évolue de plus en plus chaque jour, la capacité à traiter et à analyser des données via une image devient essentielle. C'est pourquoi le projet que nous allons présenter est une intelligence artificielle (IA) capable de résoudre des mots mêlés à partir d'une image donnée. Par mot mêlé, on entend par là une activité divertissante qui consiste à trouver une liste de mots dans une grille remplie de lettres. L'objectif principal de ce projet est de concevoir un algorithme qui permettra d'identifier les mots cachés dans la grille de lettres, mais aussi qui le fait de manière rapide et efficace. En utilisant des techniques avancées telles que l'intelligence artificielle, on vise à offrir une expérience fluide permettant aux utilisateurs de pouvoir envoyer une photo de mot mêlé à résoudre à l'IA afin qu'ils puissent avoir une correction de leur partie ou encore découvrir où se cachent les derniers mots de leurs grilles. Enfin, nous explorerons les défis techniques liés à la création d'une telle IA, notamment en ce qui concerne l'optimisation des algorithmes de recherche et les outils employés. Dans ce rapport, vous retrouverez les détails de notre travail, la présentation des membres de l'équipe, la répartition des tâches au sein de notre groupe, ainsi que l'état d'avancement du projet jusqu'à présent. En définitive, ce projet représente l'opportunité de pouvoir nous surpasser en fusionnant l'intelligence artificielle avec une activité passionnante que représentent les mots mêlés.

## **2 Présentation de l'équipe**

### **2.1 DUFOUR Guillaume (chef de projet)**

Guillaume DUFOUR, passionné par l'informatique depuis le lycée grâce aux cours de Numérique et Sciences Informatiques, a décidé de suivre une formation spécialisée dans cette voie. Après avoir obtenu son baccalauréat, il a contribué à la création d'un jeu multijoueur de tir à la troisième personne, un projet de groupe qui lui a permis de renforcer ses compétences en conception de jeux. Après la fin de ses études, il souhaiterait mettre en place sa propre activité, dans le domaine de l'informatique. En participant à ce projet, il souhaite non seulement se professionnaliser davantage mais aussi acquérir une expérience précieuse en conception de jeux en C. Dans ce projet, il s'est occupé du formatage de l'image d'origine.

### **2.2 OUHAIMMED Nassima**

Je m'appelle Nassima OUHAIMMED, et je suis actuellement en deuxième année à l'EPITA. Passionnée par l'informatique depuis le collège, plus précisément depuis la classe de troisième, c'est à ce moment-là que j'ai pris conscience de mon aspiration à en faire mon métier. Pour poursuivre dans cette voie, j'ai choisi les spécialités NSI (Numérique et Sciences Informatiques) et Maths au lycée, ce qui m'a naturellement menée vers cette école d'ingénieurs en informatique. J'ai réussi ma première année à l'EPITA, notamment grâce au projet de jeu vidéo que nous avons réalisé. Cette année, nous recommençons sur de nouvelles bases, avec de nouveaux projets et de nouveaux groupes. J'ai l'honneur de faire partie du groupe CharacTech et de travailler sur un pro-

jet d'OCR (reconnaissance optique de caractères), qui consiste à créer une IA capable de résoudre des mots mêlés. Ce projet est une opportunité pour moi d'acquérir des compétences avancées en programmation d'IA, tout en améliorant mes connaissances en langage C. Il représente également un défi stimulant qui me pousse à me surpasser chaque jour. L'expérience ainsi acquise dans le développement d'applications basées sur l'IA est précieuse et très recherchée dans le secteur technologique actuel.

### **2.3 MONROC Leonardo**

Je m'appelle Leonardo Monroc, étudiant en deuxième année à l'école d'ingénieurs en informatique EPITA. Passionné par l'informatique depuis mon enfance, je me suis toujours demandé comment fonctionnaient les intelligences artificielles qui m'entourent, comme Google Lens ou les filtres des réseaux sociaux, qui me semblaient presque magiques. Aujourd'hui, il est temps pour moi de comprendre leur fonctionnement afin de mener à bien notre projet. Cela me permettra d'apprendre à maîtriser une technologie centrale dans le monde de demain. Je suis enthousiaste à l'idée de réaliser à nouveau cet exercice de projet collaboratif avec une nouvelle équipe.

## 2.4 BONDARENKO Lucas

Je m'appelle Lucas Bondarenko, étudiant en deuxième année de prépa informatique à l'école EPITA. J'ai rejoint cette école car j'ai toujours eu un esprit logique et un intérêt pour l'informatique, renforcés lors de mes cours de NSI en première. Mon ambition professionnelle est de gérer des équipes sur des projets informatiques, et je suis donc ravi qu'EPITA propose des projets comme celui-ci, qui me permet de me projeter dans mon futur métier. Ce projet, qui consiste à créer une application pour résoudre des mots mêlés en langage C, couvre des domaines variés tels que l'IA, le traitement d'images, et la conception d'applications et d'interfaces, des compétences clés dans notre société. Cela nous permet d'acquérir des connaissances indispensables pour notre future carrière.

### 3 Répartition des charges

Tâche	Responsable	Co-Responsable
Le découpage de l'image	Guillaume	Lucas
Le réseau de neurones	Leonardo	Nassima
Le prétraitement de l'image	Lucas	Guillaume
La résolution d'une grille de mots cachés (le Solver)	Nassima	Leonardo
L'affichage de la grille résolue	Nassima	Leonardo
Une interface graphique	Guillaume	Lucas

Cette répartition des tâches a été décidée lors de notre première réunion après la constitution de notre groupe. Étant quatre membres, nous avons opté pour un travail en binômes. Ainsi, Leonardo et Nassima s'occupent du réseau de neurones et du solver, c'est-à-dire de l'OCR, tandis que Guillaume et Lucas se concentrent sur le traitement et le découpage de l'image, en prenant en charge toutes les étapes de manipulation de l'image avant son envoi aux réseaux de neurones.

## 4 État d'avancement du projet

Étapes	Avancement Actuel	Avancement Prévu
Le découpage de l'image	50%	50%
Le réseau de neurones	50%	50%
Le prétraitement de l'image	70%	70%
La résolution d'une grille de mots cachés (le Solver)	100%	100%
L'affichage de la grille résolue	0%	0%
Une interface graphique	0%	0%

### AVANCEMENT DU PROJET

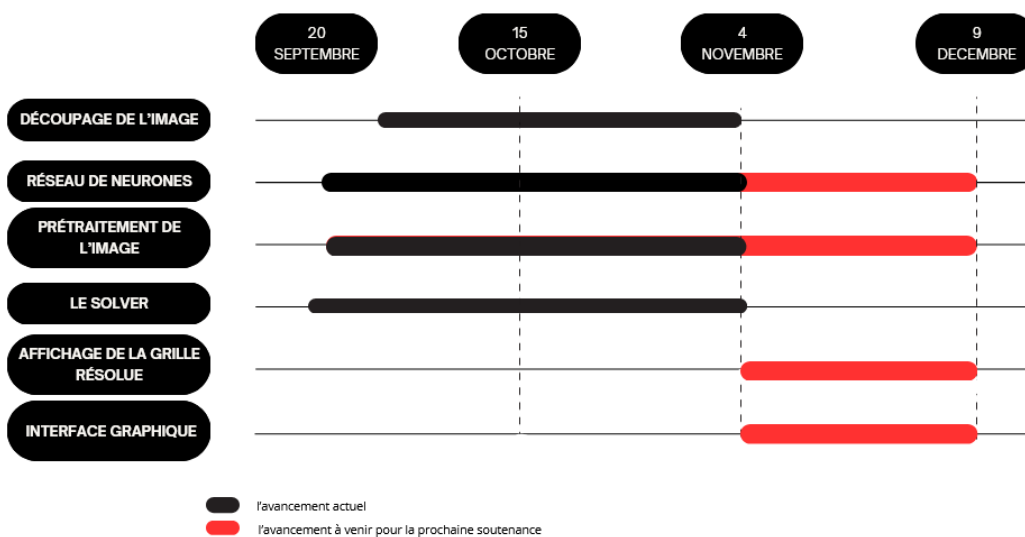


FIGURE 1 – Diagramme de Gantt



D'après les informations du tableau, nous constatons que le projet progresse très bien, conformément aux objectifs fixés. Par exemple, la résolution d'une grille de mots cachés, autrement dit le Solver, a atteint un taux d'avancement de 100%, ce qui signifie que cette tâche est complétée comme prévu. En revanche, l'affichage de la grille résolue et la création d'une interface graphique n'ont pas encore commencé, car ils sont planifiés pour la prochaine soutenance. Ces tâches affichent donc un avancement de 0%.

En analysant le graphique fourni en complément du tableau pour mieux visualiser notre progression par rapport aux prévisions jusqu'à la soutenance finale, nous pouvons observer que certaines phases, comme le découpage de l'image et le développement du réseau de neurones, affichent un avancement de 50%, en phase avec nos attentes. D'autres phases, telles que le prétraitement de l'image, ont avancé plus rapidement que prévu, atteignant ainsi un avancement de 70%.

Cette progression, sans aucun retard dans les tâches du projet, s'explique par la bonne organisation de notre groupe et par le fait que nous avons commencé à travailler sur le projet dès la constitution de notre équipe.

## 5 Aspects techniques

### 5.1 Bibliothèques et langages utilisées

**Langage C** : L'application sera codée uniquement en C.

**SDL2** : Permet de manipuler les images à notre guise.

**La bibliothèque standard** : Permet de pouvoir afficher, renvoyer les erreurs dans le terminal mais aussi de lire dans les fichiers.

**Makefile** : Permet de compiler et exécuter nos fichiers.

### 5.2 Prétraitement de l'image

La première partie du projet a consisté à préparer une image pour faciliter la détection des éléments pertinents. Voici les étapes principales du prétraitement :

- **Conversion en Niveaux de Gris** : Nous avons commencé par convertir l'image en niveaux de gris afin de simplifier la détection des éléments. Cette conversion a été réalisée en calculant une moyenne pondérée des composantes rouge, verte et bleue de chaque pixel.

- **Rotation de l'image** : Une application a été développée pour permettre à l'utilisateur de charger une image et de la faire pivoter selon un angle spécifié. Pour ce faire, nous avons commencé par initialiser la bibliothèque SDL et SDL\_image, puis charger l'image fournie par l'utilisateur. L'image est ensuite convertie au format `SDL_PIXELFORMAT_RGBA8888` afin de faciliter les manipulations précises lors de la rotation. Une nouvelle surface est créée

pour contenir l'image tournée, et un `render` est utilisé pour appliquer la rotation avec un pivot au centre de l'image. Enfin, l'image tournée est sauvegardée dans un fichier PNG avec un suffixe `"_turned"` pour la différencier de l'originale. Cette partie du projet a nécessité une collaboration importante pour vérifier la logique de rotation et s'assurer de la qualité de la sauvegarde des fichiers.

### 5.3 Découpage de la grille

La seconde partie du projet a été dédiée à la détection et au découpage d'une grille de lettres présente dans l'image, puis à l'extraction des caractères individuels. Voici un résumé des étapes principales :

- **Détection de la Grille** : Nous avons implémenté une fonction pour parcourir les pixels de l'image afin de détecter les lignes et colonnes de la grille. Cette détection repose sur la présence de pixels sombres qui indiquent les bords des cellules.
- **Extraction des Lettres** : Une fois la grille détectée, nous avons extrait chaque lettre et l'avons enregistrée sous forme d'image individuelle. Cette partie a nécessité une collaboration active pour affiner la logique de détection des bordures et garantir que chaque lettre soit correctement isolée.
- **Détection de la Liste de Mots** : En plus de la grille de lettres, le programme devait également identifier une liste de mots située à droite de la grille et extraire chaque mot.

## 5.4 Résolution des mots mêlés

Dans le fichier C nommé **solver**, nous avons implémenté un algorithme pour résoudre une grille de mots cachés. Celui-ci commence par lire un fichier contenant la grille de lettres, grâce à la bibliothèque standard `jstplib`, et le transforme en un tableau de lettres. Une fois le tableau créé, l'algorithme vérifie chaque lettre pour voir si elle correspond à la première lettre du mot recherché.

Si une correspondance est trouvée, l'algorithme procède à vérifier si la position de la dernière lettre du mot correspond bien à l'emplacement attendu, en ajoutant la longueur du mot à la position de la première lettre. Si ces conditions sont remplies, l'algorithme vérifie ensuite que toutes les lettres entre la première et la dernière position correspondent bien aux lettres du mot recherché. Si ce n'est pas le cas, l'algorithme continue à chercher dans le tableau jusqu'à épuisement des lettres disponibles.

Lorsqu'il trouve le mot, le programme renvoie les positions des lettres dans le tableau sous la forme de couples  $(x0, y0)$  et  $(x1, y1)$ . Le couple  $(x0, y0)$  indique la position de la première lettre du mot, tandis que  $(x1, y1)$  donne la position de la dernière lettre. C'est ainsi que le **solver** identifie et localise les mots dans la grille.

## 5.5 Le réseau de neurones

Pour réaliser le projet, il faut créer un réseau de neurones capable de reconnaître une lettre à partir d'une image. Pour l'instant, un réseau de neurones capables de prédire le résultat de la porte logique NXOR à partir de deux booléens a été créé. Un réseau de neurones est organisé en plusieurs couche, la couche

d'entrée, la couche de sortie et les couches cachée. Puisque les calculs n'ont pas besoin d'être très complexe pour réaliser un NXOR, on utilise qu'une seule couche cachée. La couche d'entrée est constituée de deux neurones qui correspondent à nos deux valeurs d'entrée. La couche de sortie n'est constituée que d'un seul neurone puisque nous voulons que le réseau de neurones nous renvoi un booléen (0 ou 1). Chaque neurone possède une liste de poids, un pour chaque neurone de la couche précédente, et un biais. Un neurone renvoie aux neurones de la couche suivante la somme de toutes les valeurs renvoyées par les neurones de la couche précédente multipliées par les poids du neurone correspondant plus le biais. On applique ensuite la fonction Sigmoidale sur la valeur de retour du neurone qui agit comme une fonction d'activation, c'est à dire que la fonction va "trancher la poire en deux" pour obtenir une meilleure répartition des valeurs et donc des résultats plus pertinents. Ceci n'est pas le cas pour les neurones de la couche d'entrée qui renvoient simplement la valeur qu'on leur a donné. C'est avec toute ces opérations que les valeurs entrées vont de la couche d'entrée, à la couche cachée, à la couche de sortie censée nous renvoyer le résultat souhaité. Pour bien organiser toutes ces données, j'utilise le "type struct" qui me permet de regrouper tous les attributs d'un réseau de neurone avec des pointeurs, ce qui ressemble un peu aux objets en C#. Avec ce "type struct" je regroupe les poids de la couche cachée, les poids de la couche de sortie, les biais de la couche cachée, le biais de la couche de sortie ainsi que le nombre de neurones dans chaque couche, ce qui me permet de connaître la taille des différentes listes citées ci-dessus. Pour calibrer tous ces paramètres, initialement choisis de manière aléatoire, on utilise la rétropropagation qui consiste à régler les paramètres en fonction de la différence entre les valeurs obtenues et les valeurs

souhaitées. Plus précisément, le programme va calculer la valeur de retour de la couche de sortie et va faire la différence avec la valeur souhaitée pour avoir une valeur d'erreur, cette valeur est multipliée par la fonction dérivé de sigmoïde qui permet d'ajuster le résultat à la hausse ou à la baisse. Ce produit est ensuite multiplié par un nombre que l'on choisit qui définit à quel point on modifie les valeurs du réseau de neurone, puis on modifie chaque poids et chaque biais de la couche de sortie en le soustrayant par la valeur d'erreur calculée précédemment. Puis l'on répète l'opération avec les valeurs de retour de la couche cachée pour modifier les valeurs des poids et des biais de la couche cachée. On répète la rétropropagation plusieurs milliers de fois pour ajuster petit à petit le réseau de neurone pour que celui-ci puisse prédire l'opération logique NXOR.

## 5.6 Pourquoi ces choix

Nous avons choisi le langage C pour sa capacité à manipuler directement la mémoire, ce qui permet une optimisation accrue de l'application. Cela facilite également la gestion des images grâce à la bibliothèque SDL2. En effet, nous utilisons SDL2 pour la manipulation des images, notamment pour le prétraitement et le découpage des grilles. Autrement dit, SDL2 s'avère très efficace pour le rendu graphique, ce qui améliore l'expérience utilisateur et permet de produire une image résolue du mot croisé de manière précise.

Par ailleurs, nous avons développé un fichier C nommé `solver`, qui permet de lire une grille de mots contenue dans un fichier et de renvoyer la position des mots dans cette grille. Enfin, grâce aux capacités de manipulation mémoire offertes par le C, nous avons pu concevoir notre propre réseau de neurones pour reconnaître les caractères, en appliquant des concepts d'apprentissage automatique.

En somme, en suivant cette approche technique avec le langage C et la bibliothèque SDL2, toute personne souhaitant repartir de zéro pourrait reproduire notre application pour résoudre des mots mêlés.

## 6 Conclusion

Pour conclure, la création d’une application OCR pour résoudre des mots mêlés a représenté un véritable défi technique pour notre groupe, nous permettant d’acquérir des compétences variées en programmation, en traitement d’images et en réseaux neuronaux. En utilisant le langage C et la bibliothèque SDL2, nous avons pu développer une première version de l’application, capable de traiter des images de mots mêlés et d’en extraire les informations nécessaires. L’interface graphique reste à réaliser pour la prochaine soutenance, avec les autres tâches encore inachevées.

Nous avons débuté par le prétraitement de l’image, une étape cruciale impliquant la manipulation des pixels pour segmenter la grille en cellules distinctes. Ce découpage précis des cellules a facilité la reconnaissance des mots, une étape essentielle pour notre application. En parallèle, nous avons développé un algorithme de résolution de mots mêlés, le Solver, capable de lire une grille de lettres et de retourner la position du mot recherché.

L’intégration d’un réseau de neurones, conçu autour du problème du XOR, nous a permis de comprendre les bases de la reconnaissance de motifs et d’explorer les concepts de l’apprentissage automatique, à appliquer pour la reconnaissance de caractères. Bien que ce réseau soit encore en phase exploratoire, il nous offre une base solide pour la suite.

En somme, ce projet nous a permis d’acquérir des compétences précieuses en traitement d’images et en réseaux neuronaux. Nous avons atteint nos objectifs initiaux et sommes confiants quant à notre progression pour la soutenance finale, convaincus que les connaissances acquises ici nous serviront dans l’avenir.