EPITA

RAPPORT DE SOUTENANCE INTERMÉDIAIRE

- SEEINC -

”OCR”

---

11 mars 2020

Etienne Lazarz

Yael Magnier

Tristan Hillion

François Soulier

Contents

1 Introduction

2 Les Origines Du Groupes

2.0.1 Yaël Magnier

Curieux et avide de connaissances, j'aime expérimenter de nouvelles choses et ainsi de me lancer dans des projets plutôt stimulants. Faisant preuve d'une ouverture d'esprit remarquable, j'ai eu la chance de pouvoir découvrir le monde (notamment par des voyages, j'ai visité 36 pays dont 5 où j'ai pu vivre). Écrivain à mes heures perdues (des fanfictions en particulier), l'expression de ma créativité s'illustre dans le développement informatique. Pour moi, programmer est un moyen de me challenger et de travailler en équipe.

2.0.2 Etienne Lazarz

Avant tout attiré par les projets, c'est dans cet objectif que j'ai intégré l'EPITA en septembre 2018. En effet, la plupart des choses que j'entreprends se font sous la forme de projet (voyage à l'étranger, projet solidaire). Mais c'est en arrivant à l'EPITA que mon ambition s'est révélé. C'est donc tout naturellement que je me suis proposé en tant que chef de projet. J'aime aider et guider les gens dans leur travail. Je n'ai pas de compétence particulière en programmation. J'ai donc tout à apprendre !

2.0.3 Tristan Hillion

Un groupe se forme sur des affinités et le nôtre en a eu assez pour décider de travailler ensemble sur ce projet éprouvant. J'espère y développer des compétences sur des concepts qui me sont inconnus pour le moment. Cette nouvelle équipe me change beaucoup du projet de premier semestre dans sa manière de travailler et cela dans une bonne dynamique qui permet à chacun de mettre ses points forts au service du groupe. C'est aussi l'autonomie que je voit à travers ce projet dans la mesure où peu d'information nous est donné sur la marche à suivre dans l'obscurité de la segmentation et des réseaux neuronaux.

2.0.4 François Soulier

Pris de passion pour l'informatique dès la classe de première scientifique, je décide de rentrer à l'EPITA après le baccalauréat. Après avoir réussi avec briot un projet en première année, j'ai très vite pris goût aux travaux de groupes et au projets communs. Connaissant déjà mes collègues de projet, le groupe s'est formé naturellement, ainsi que la répartition des tâches, qui s'est faite dès le début du projet. Jusqu'à présent je n'ai eu que l'occasion d'aider mes collègues dans la résolution de certains problèmes, mais je me suis occupé seul de ma partie, ce qui s'est avéré ne pas être un avantage lorsque l'on veut effectuer des mises au point en groupe. Il ne me reste plus qu'à me plonger dans des parties communes et complexes, comme le réseau neuronal, qui est en cela un parfait exemple.

3 Avancement du Projet

3.1 Répartition des tâches

Nom	Personne en Charge	Suppléant
Pré-traitement		
Binarisation	François	
-chargement de l'image		
-suppression des couleurs		
Segmentation	Etienne	Tristan
-detection de blocs		
-detection de lignes		
-detection de mots		
-detection de caractères		
Réseau Neuronal	Yael	François
-Chargement des poids des neurones		
-XOR		
-reconaissance des caractères		
-sauvegarde des poids des neurones		
Reconstruction du texte		
-manipulation des fichiers	Tristan	Etienne
-Bonus et autres		
-Interface utilisateurs	Tristan	Etienne
-LateX	SeeInC	
-intégration d'un correcteur orthographique		

3.2 Avancement

3.2.1 Binarisation

Cette section du projet concerne la binarisation de l'image (suppression des couleurs puis transformation de l'image en noir et blanc) et a été prise en charge par François. La méthode utilisée dans cette partie est la méthode d'Otsu, qui est une méthode de seuillage de l'image. La valeur d'entrée que prend cette méthode est une image à niveaux de gris, et il en ressort une image binarisée. On considère de ce cadre que l'image d'entrée comporte deux classes de pixels : les pixels dont l'intensité est supérieure au seuil, et ceux dont l'intensité est inférieure ou égale au seuil.

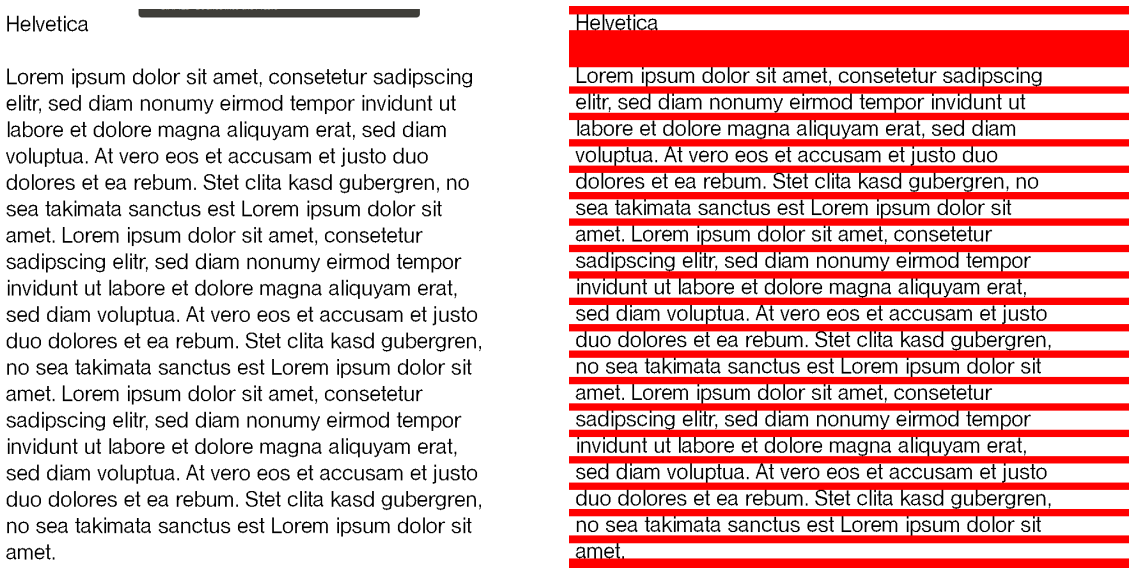
La méthode d'Otsu utilise un histogramme montrant le nombre de pixels par intensité. Ensuite pour chaque intensité entre 0 et 255 (car l'image est en niveaux de gris), elle détermine la probabilité de tomber sur chacune des intensités. De cela se calculent les variances (mesure de dispersion de valeurs) intra-classes de manière à ce qu'elles soient minimales, et que l'on puisse en déduire le seuil optimal. Cela revient à maximiser la

variance inter-classe, et permet alors de faciliter le calcul du seuil. Afin d'optimiser la reconnaissance de caractère par la suite, si le seuil est trop faible, alors ce dernier est augmenté pour renforcer les contrastes.

L'algorithme principal peut se diviser en deux parties. La première est de déterminer le seuil optimal. Pour cela on itère sur toutes les valeurs de 0 à 255, on calcul la variance inter-classe, et on prend comme seuil la valeur pour laquelle la variance inter-classe est la plus élevée. Ensuite, avec cette valeur de seuillage, vient la deuxième partie de l'algoritme. Celle-ci consiste à changer en noir tous les pixels dont l'intensité est inférieure ou égale au seuil, et en blanc tous ceux dont l'intensité est supérieure à ce même seuil.

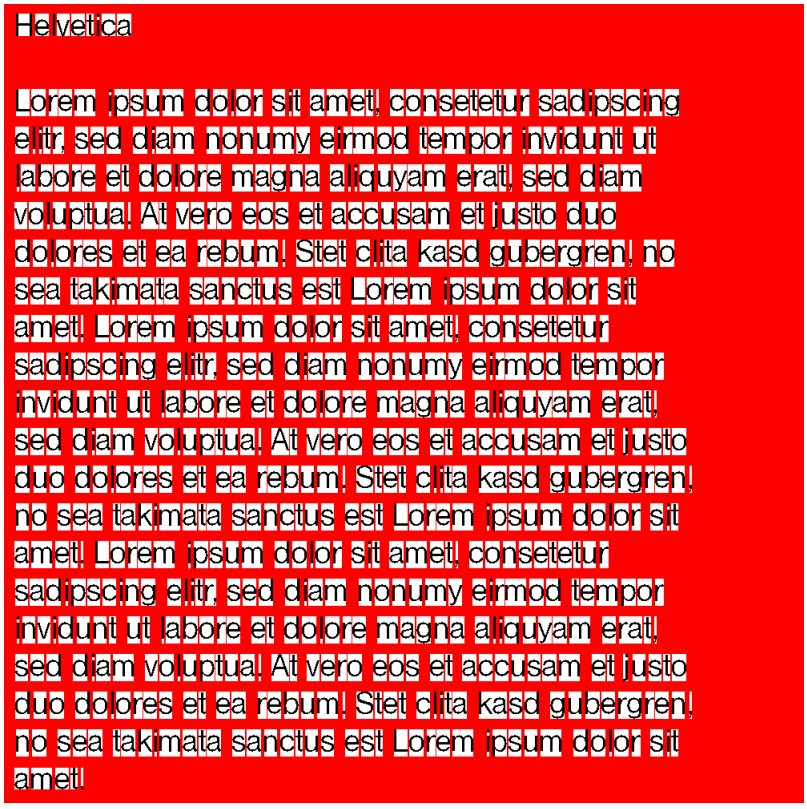
3.2.2 Segmentation

Pour cette partie c'est Etienne et Tristan qui se sont chargés de travailler sur la segmentation. Nous travaillons à partir des résultats du prétraitement ainsi que de ceux de la binarisation. La première étape dans notre cas fût la séparation du texte en ligne de caractères. C'est en soi assez simple, avec un parcours des pixels de l'image: si on arrive au bout d'une ligne sans avoir rencontré de pixel noir, alors on trace une ligne rouge.



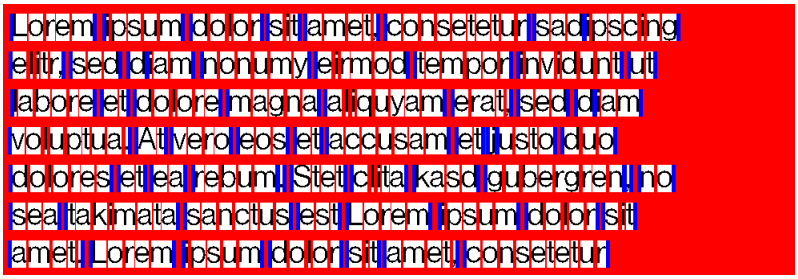
Deuxième étape: tentée de faire une première séparation des caractères. Dans l'idée, ma première méthode ressemblait beaucoup à celle du traçage horizontale mais ces fois-ci, verticalement et entre deux bandes rouges. "Je trace un trait rouge si entre deux bandes rouges je n'ai pas rencontré de pixel noir"

On obtient donc ce résultat:



Cependant, le résultat était très coûteux et peu utilisable (seulement pour une police d'écriture du type Helvetica donc avec des caractères bien espacés. J'ai donc utilisé une deuxième méthode qui consiste à tracer l'histogramme des pixels noirs d'une ligne de caractère. Avec cet histogramme on peut directement voir les colonnes à 0 pixel noir. De plus pour la suite, pourra plus facilement détecter les mots et éventuellement des liaisons entre deux lettres dans une police d'écriture "attachée".

Dans ce profil de phrase Telugu, on repère instantanément les mots. C'est donc en poussant cette méthode que nous sommes parvenus à réaliser une première détection des mots:

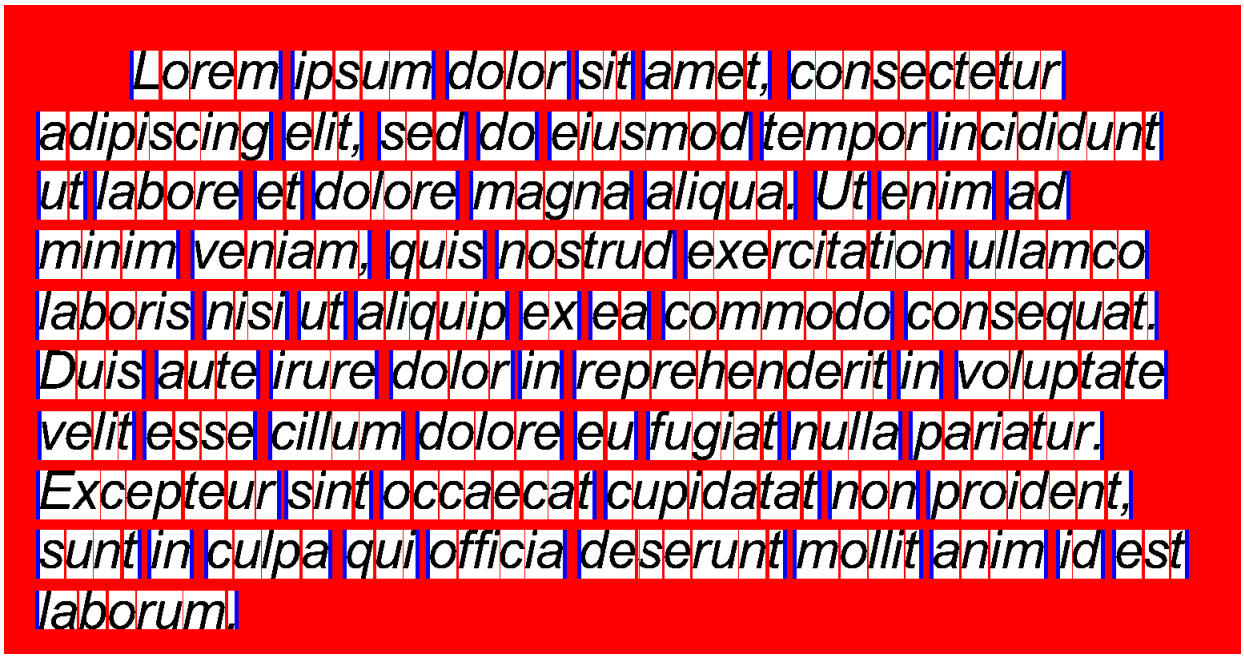


En Bleu sont symbolisés les marqueurs de mots. Chaque mot est censé se situer entre deux marqueurs bleus. On remarque la présence de quelques erreurs sur cette reconnaissance.

Enfin, parlons des limites de la segmentation actuelle:

-tout d'abord comme abordé ci-dessus, la reconnaissance n'est pas encore parfaite, cela vient entre autres des moyennes de pixels que nous calculons qui ne sont pas encore suffisamment précisent.

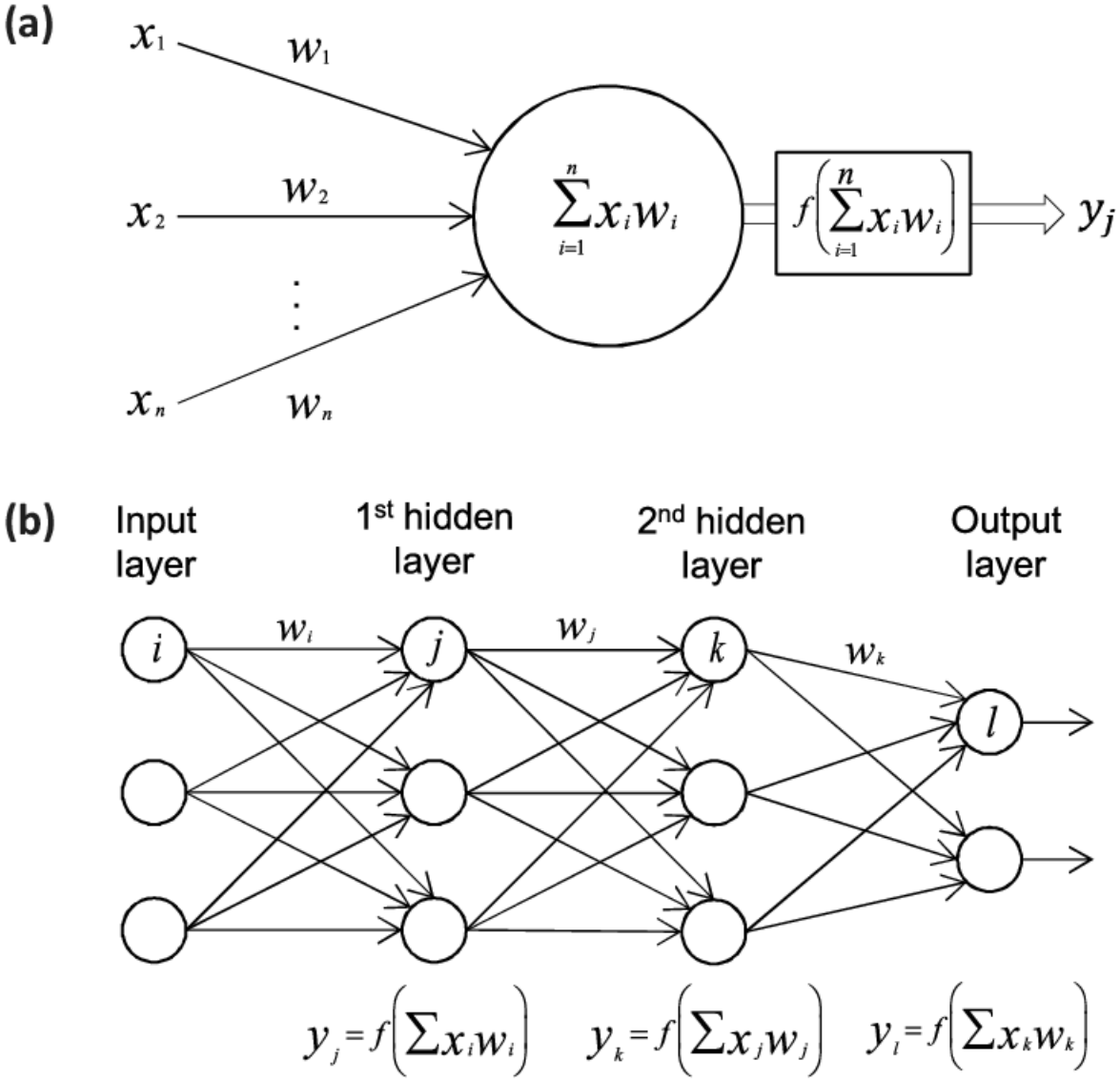
-Ensuite, les polices d'écritures italiques et attachées ne sont pas encore bien segmentées. Nous avons commencé l'"Overlapping segmentation" qui est censé résoudre ce problème. Mais pour le moment elle n'est pas fonctionnelle.



3.2.3 Réseau Neuronal

Maintenant que nous savons ou est située chaque caractères, il nous faut déterminer quel caractère est affiché. Pour cela, nous utilisation un réseau neuronal. Cette partie, faîte par Yael, ce charge de cela.
Dans un premier temps, il nous a fallu décider de quel type de réseau neuronal utiliser et quelle structure avoir. Nous pour cela décidé de créer une base de réseau utilisant la méthode du "feedforward" pour le calcul et de "backpropagation" pour pouvoir entraîner le réseau.

Comment fonctionne un réseau neuronal ?



Un réseau neuronal est un ensemble de neurones inter-connectés permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage.
Pour cela, nous faisons un produit matriciel entre la couche précédente et les poids correspondants, puis, on ajoute un biais et utilise une fonction sigmoïdale pour borner les valeurs par 0 et 1.
Dans notre programme final, nous aurons en entrée un tableau a une dimension des valeurs des pixels du caractère et une sortie par lettre possible. La lettre lue sera ainsi celle avec la plus haute valeur, mais cela nous permettra aussi de récupérer les autres lettres probables en cas de doute afin d'implémenter plus tard un correcteur orthographique.

Initialement, nous avions décidé de ne pouvoir choisir que de la taille des couches du réseau, plutôt que d'avoir un nombre de couches variable, c'est pourquoi nous avons décidé d'en réécrire un permettant de créer et d'entraîner un réseau de la taille de notre choix.
Pour l'instant, nous avons un problème de calcul des poids dans notre réseau. Le réseau neuronal n'es donc pas encore entièrement fonctionnel. Nous avons par contre déjà implémenté le système de chargement et de sauvegarde des poids dans un format lisible par l'utilisateur et par l'ordinateur.

3.3 Bilan de l'avancement des tâches principales

3.4 Les Ressources

3.4.1 Logiciels

Les logiciels qui ont été utilisé par l'équipe jusqu'à maintenant sont :

- Gitlab : Logiciel de partage du code de manière sécurisée.
- Overleaf : Rédaction et rendu écrit en LaTeX
- Trello : Site internet permettant simplifiant l'organisation au sein de l'équipe
- Discord: Service de communication permettant l'organisation de l'équipe, le partage de document et la communication orale.
- Vim : Editeur de texte indispensable au projet

- Vscode : Environement pratique pour concevoir un projet

3.4.2 Lien vers des ressources en ligne

http://www.saedsayad.com/artificial_neural_network.html
https://www.researchgate.net/publication/50366348_SEGMENTATION_OF_OVERLAPPING_TEXT_L
https://ensiwiki.ensimag.fr/index.php?title=Projet_image_2016_:_Reconnaissance_de_documents

4 Conclusion

Avec l'avancement des différents blocs qui formeront notre projet OCR, nous apporteront donc pour la prochaine soutenance une application correspondant aux attentes du cahier des charges avec une interface utilisateur, une binarisation et une segmentation perfectionné ainsi qu'un réseau neuronal entrainé. Le tout pour reconstituer dans cette interface utilisateur le texte donné en image.