

LAPI – Lecture automatique des plaques d'immatriculation

Initiation à la recherche

Université Sorbonne Paris Nord

Institut Galilée

Superviseur principal : Prof. Guénaël CABANES

Présenté par : Mohand Lounis BENSEKHRI
Master 1 informatique
N° étudiant : 11710457
lounisbensekhri@gmail.com

Soumission : Avril 2022

ABSTRAC

Dans cet article vous trouverez ma recherche que j'ai effectuée durant le second semestre de mon master informatique au sein de la matière « Initiation à la recherche ».

Ma recherche a porté sur la détection de plaque d'immatriculation dans une image de voiture avec le package « OpenCV » de Python et par la suite la lecture de celle-ci avec deux technologies de reconnaissance optique de caractères qui sont « EasyOCR » et « Tesseract » que j'expliquerai plus en détail dans cet article, et enfin faire des statistiques sur le taux de réussite de lecture pour ces deux technologies en utilisant un jeu de données de cent (100) images de voitures et grâce à un outil d'alignement biologique « BioPython » dans le cas où la plaque n'est lue qu'en partie pouvoir faire des digrammes avec la librairie « Matplotlib » en fonction de la réussite de la lecture partielle ou complète de celle-ci. Ma recherche consiste notamment à étudier le temps d'exécution que prend chaque technologie pour déterminer la plus rapide et cela en utilisant la bibliothèque « time ».

Enfin dans la section « Discussion » je ferai un bilan général sur la détection de plaque d'immatriculation, le taux de réussite de reconnaissance de celle-ci, le temps que prend chaque technologie de détection pour déterminer quelle technologie de reconnaissance est meilleure.

TABLE DES MATIERES

Abstrac	3
Figures	5
Tables	6
Introduction	7
Etat de l'art	8
Proposition	10
Expériences et résultats	12
Expériences	12
Résultats	13
Discussion	14
Conclusion et perspectives	15
Conclusion	15
Perspectives	15
Bibliographie	16

FIGURES

Figure 1: Étapes 2, 3 et 4 : La plaque d'immatriculation est normalisée en termes de luminosité et de contraste, puis les caractères sont segmentés pour être prêts pour l'OCR. -----	8
Figure 2: Exemple de détection de contours (Filtre de Sobel). -----	9
Figure 3: Vector - Jenoptik (Caméra qui fournit des images ANPR) -----	9
Figure 4: Exemple d'application du programme de Jenoptik-----	9
Figure 5: Exécution du script initial sur un exemple de voiture. -----	10
Figure 6: Résultat après découpage de la plaque d'immatriculation. -----	10
Figure 7: Utilisation de EasyOCR-----	11
Figure 8: Utilisation de Tesseract -----	11
Figure 9: Exemple d'alignement de la seq. ACGT a d'autres séquences-----	12
Figure 10: Logo BioPython-----	12
Figure 11: Résultats de l'expérience (en partie) + temps d'exécution- EasyOCR -----	13
Figure 12: Résultats de l'expérience (en partie) + temps d'exécution- Tesseract -----	13

TABLES

Tableau 1: Tesseract VS EasyOCR -----	11
Tableau 2: Graphique de taux de réussite de lecture des plaques d'immatriculation avec EasyOCR et Tesseract -----	13
Tableau 3: Diagramme de taux de réussite de lecture des plaques d'immatriculation avec EasyOCR et Tesseract -----	13

INTRODUCTION

Que signifie LAPI ?

LAPI est l'abréviation de « Lecture automatique des plaques d'immatriculation », en anglais cela s'appelle ANPR qui est l'abréviation de « Automatic Number Plate Recognition » est une technologie destinée à identifier les plaques d'immatriculation de véhicules via l'utilisation de techniques de reconnaissance optique de caractères. Cette technologie fournit ensuite ces informations à une étape suivante au traitement informatique au cours de laquelle les informations peuvent être interprétées, stockées ou comparées. Le LAPI est donc utilisé pour capturer et identifier avec précision n'importe quelle plaque d'immatriculation à l'aide de séquences vidéo ou de photo provenant de caméras. Le système LAPI est utilisé dans divers secteurs comme pour l'application de la loi : Les forces de l'ordre l'utilisent pour vérifier si un véhicule est immatriculé ou pour identifier les véhicules liés aux infractions au code de la route, pour l'analyse de temps de parcours pour vérifier quels véhicules sont en excès de vitesse, on l'utilise aussi pour la gestion du stationnement des voitures, pour l'enregistrement aux postes de péages autoroutier...

Comment fonctionne LAPI ?

Tout d'abord une caméra capture des images contenant une plaque d'immatriculation, ensuite la plaque est détectée à l'aide de processus d'apprentissage automatique, et enfin avec un logiciel de reconnaissance optique de caractères (OCR) on transforme le texte de la plaque d'immatriculation découpé en un numéro de plaque d'immatriculation brut, utilisable pour autre application.

Durant ma recherche j'ai trouvé plusieurs scripts qui permettent de mettre des contours pour la plaque d'immatriculation sur une image initiale d'une voiture dans laquelle la plaque est visible, ceci en utilisant la librairie « OpenCV » de Python et d'ailleurs la plupart des programmes que j'ai trouvés sont implémentés en Python.

Ma contribution sur les scripts que j'ai trouvés est dans un premier temps d'identifier le meilleur script pour détourer la plaque d'immatriculation ensuite ce que j'ai fait est d'ajouter la fonctionnalité principale qui est la conversion de cette plaque en texte brut avec les deux technologies de reconnaissance optique de caractères de langage Python qui sont « EasyOCR » et « Tesseract », et enfin j'ai fait des tests pour déterminer quelle technologie est plus rapide et/ou la meilleure pour cette conversion.

Cet article est constitué de plusieurs sections qui sont « Etat de l'art » dans cette section je présenterai les principaux travaux qui portent sur le sujet du LAPI, « Proposition » qui représente le cœur de l'article en effet je parlerai développement et proposition, « Expérience et résultats » dans cette section je montrerai l'ensemble des expériences et résultats que j'ai obtenus, « Discussion » consiste à discuter et faire des remarques sur les résultats, « Conclusion et perspectives » dans laquelle je ferai un bilan général de la technologie du LAPI, et enfin y aura une « Bibliographie » qui renferme la liste des articles et techniques citées dans cet article.

ETAT DE L'ART

Dans cette section je vais présenter l'ensemble des travaux et articles qui portent sur le sujet de la reconnaissance automatisée des plaques minéralogiques.

En général il existe sept algorithmes principaux dont le logiciel a besoin pour identifier une plaque d'immatriculation :

1. Localisation de la plaque : responsable de trouver et d'isoler la plaque sur l'image.
2. Orientation et dimensionnement de la plaque : compensation de l'inclinaison de la plaque et ajustement des dimensions à la taille requise.
3. Normalisation : ajustement de la luminosité et du contraste de l'image.
4. Segmentation des caractères : recherche des caractères individuels sur la plaque
5. Reconnaissance optique des caractères.
6. Analyse syntaxique/géométrique : vérification des caractères et des positions par rapport aux règles spécifiques du pays.
7. Le calcul de la moyenne de la valeur reconnue sur plusieurs champs /images afin de produire un résultat plus fiable ou plus sûr d'autant plus que chaque image peut contenir un reflet lumineux, être partiellement obscur ou possède d'autres effets d'obscurcissement.



Figure 1: Étapes 2, 3 et 4 : La plaque d'immatriculation est normalisée en termes de luminosité et de contraste, puis les caractères sont segmentés pour être prêts pour l'OCR.

La complexité de chacune de ces sous-sections du programme détermine la précision du système. Au cours de la troisième (03) phase (normalisation), certains systèmes utilisent des techniques de détection des bords pour augmenter la différence d'image entre les lettres et le support de la plaque. Un filtre numérique peut aussi être utilisé pour réduire le « bruit » visuel de l'image et c'est d'ailleurs ce que j'ai utilisé dans mon approche.

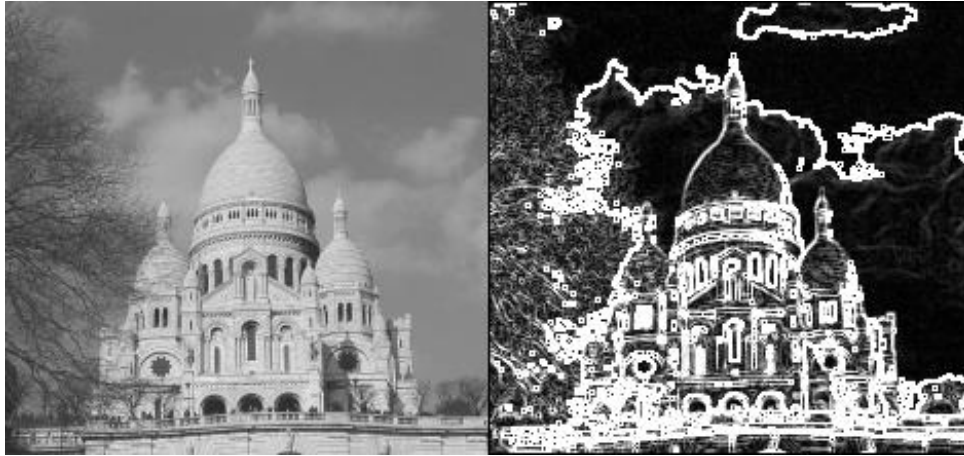


Figure 2: Exemple de détection de contours (Filtre de Sobel).

Comme déjà expliquer précédemment dans l'introduction le système de reconnaissance de plaque d'immatriculation est beaucoup sollicité dans le secteur de la sécurité routière en effet les autorités de police et de sécurité du monde entier utilise le LAPI, et parmi les logiciels existant sur le marché nous trouvons le logiciel d'apprentissage en profondeur du groupe allemand d'opto-électronique « Jenoptikⁱⁱ » qui rend l'ANRP particulièrement puissant en étant capable d'extraire plus d'informations des images enregistrées. Le logiciel différencie de manière fiable les catégories de véhicules telles que les voitures particulières les camions, les bus et les motos. Il peut également reconnaître tous types de plaques d'immatriculation du monde. Le logiciel s'intègre aux caméras ANPR, de sorte qu'aucun équipement ou installation supplémentaire n'est requis, permet de donner de bien meilleurs résultats que les technologies visuelles conventionnelles.



Figure 3: Vector - Jenoptik (Caméra qui fournit des images ANPR)



Figure 4: Exemple d'application du programme de Jenoptik

PROPOSITION

Comme nous venons de voir la détection de plaque d'immatriculation dans une image de voiture passe par plusieurs étapes qu'on peut découper en deux principales sections qui sont : la première est l'identification et le contour de la plaque d'immatriculation dans une image de voiture et la seconde consiste à convertir cette plaque en un texte brut qui est la plaque.

Durant ma recherche j'ai trouvé des scripts prêts qui font la première étape c'est-à-dire identifier et détourner la plaque. Et cela avec le langage de programmation Python et plus précisément avec la librairie « OpenCV ».

OpenCVⁱⁱⁱ (Open Computer Vision) est une librairie graphique libre de python développer par Intel, spécialisée dans le traitement d'images. Elle propose plusieurs opérations de traitement de bas niveau des images comme le filtrage (ce que d'ailleurs est utilisé dans le script que j'ai trouvé).

Ma première tâche était de trier ces programmes est de trouver le meilleur d'entre eux.



Figure 5: Exécution du script initial sur un exemple de voiture.



Figure 6: Résultat après découpage de la plaque d'immatriculation.

Ensuite j'ai commencé à chercher pour trouver les technologies qui permettent de convertir une image, dans notre cas une plaque d'immatriculation en un texte brut et j'ai trouvé que pour cela on utilisait la reconnaissance optique de caractères (OCR : Optical Character Recognition) et pour effectuer cette OCR, il existe plusieurs librairies déjà prêtes en Python comme « EasyOCR^{iv} » et « Tesseract^v ».

EasyOCR est un module Python qui permet d'extraire du texte d'une image.

Tesseract est un logiciel de reconnaissance optique de caractères sous licence Apache. Conçu par les ingénieurs de Hewlett Packard de 1985 à 1995.

Le résultat à étape n'est malheureusement pas garanti toujours à 100% comme nous le verrons dans la prochaine section, en effet la technologie d'OCR repose sur plusieurs aspects et conditions qui doivent être présente sur l'image depuis laquelle nous souhaitons identifier le texte, et parmi ces conditions nous trouvons par exemple la clarté de l'image, sa résolution, l'angle de capture de l'image...

Ce que j'ai étudié durant ma recherche est justement l'efficacité entre ces deux technologies d'OCR. On remarque parfois des erreurs la lecture d'un caractère ou deux par rapport à une autre, d'autre part on remarque que l'une arrive à lire l'intégralité de la plaque sans aucun souci tandis que l'autre n'arrive pas du tout.

En théorie il y a certaines différences déjà constatées par rapport à Tesseract et EasyOCR que nous pouvons voir dans la figure suivante^{vi}.

	Actual Value	Tesseract Prediction	EasyOCR Prediction
Number Plate High Quality	HR26DK8337	HR26DK8337	HR26DK8337
Number Plate Low Quality	MH14GN9239	Spaces	9239
Handwritten Low Quality	AMIT ASHISH	Spaces	AdIT ASHISH
Handwritten High Quality	LAKSHMINIVAS TOURIST HOME	LAKSHMINIVAS TOURIST HOME	LAKSHMINIVAS TOURIST HOME
Image with text High Quality	Albert Einstein	Albert Einstein	A ber t Einsteim
Image with text Low Quality	Kotak Mahindra Bank	Kotak Mahindra Bank	Kotak Mahindra Bank
Reciept High Quality	Order #19866	Order #19866	Order #19866
Reciept Low Quality	Amoxicillin 500mg	Spaces	Amoxicillin 500mg

Tableau 1: Tesseract VS EasyOCR

La remarque que nous pouvons d'ores et déjà constater est que EasyOCR est meilleure en matière de détection de texte que nous étudierons mieux dans la prochaine section. Dans la figure suivante nous trouverons l'exécution des deux techniques d'OCR sur notre exemple. Ma première contribution était de compléter le LAPI en y ajoutant une technique de conversion d'image en texte :

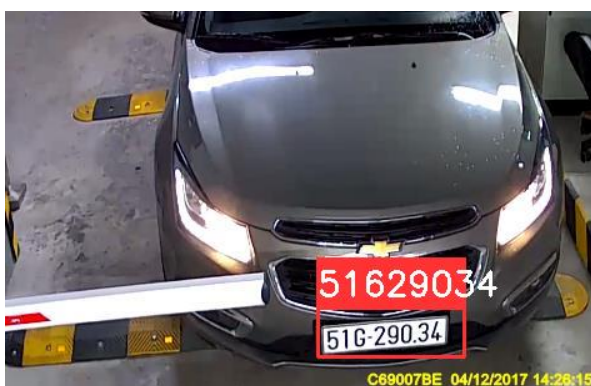


Figure 7: Utilisation de EasyOCR

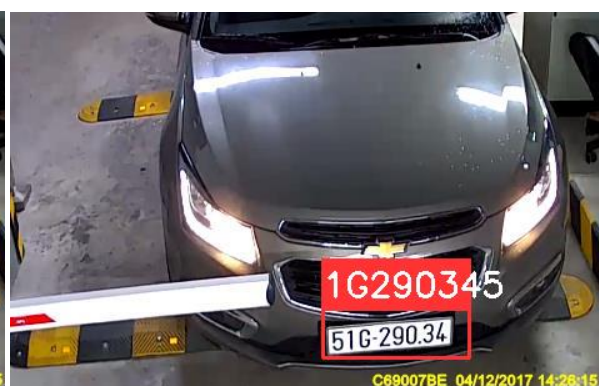


Figure 8: Utilisation de Tesseract

On remarque que dans les deux technologies il y a des erreurs comme pour le 6 et G de la plaque d'immatriculation mais qu'avec EasyOCR on a un meilleur résultat. Chose qui reste un peu floue c'est qu'avec Tesseract on réussit à lire le G contrairement à EasyOCR.

EXPERIENCES ET RESULTATS

Expériences

Dans cette section de l'article je vais décrire l'ensemble des expériences et tests que j'ai effectués sur mon projet de LAPI. J'ai commencé par créer un jeu de données (dataset) de cent (100) images de voitures ainsi que le texte contenu dans les plaques d'immatriculation de ces véhicules que j'ai mis sous forme de liste dans un fichier « json ».

L'expérience que j'ai réalisée est de tester pour chaque image si elle est traitée par le logiciel et dans le cas où s'est traitée comme dans l'exemple de la précédente section, voir le taux de réussite de lecture de la plaque, et pour cela j'ai utilisé la librairie « Biopython » qui est utilisée généralement dans la biologie, qui permet dans ce domaine-là d'aligner une certaine séquence génétique sur une autre pour vérifier si elles sont identiques ou non et vérifier le taux de compatibilité, dans notre cas vérifier ce taux entre la plaque lue par le logiciel et la plaque originale stocker dans le fichier « json » pour chaque voiture.

Biopython^{vii} est un ensemble d'outils librement disponibles pour le calcul biologique écrit en Python par une équipe internationale de développeurs. Ce qui nous intéresse dans ces outils pour notre projet est la partie Alignement^{viii} de séquences qui permet de donner le score obtenu par alignement de deux séquences.



Figure 10: Logo BioPython

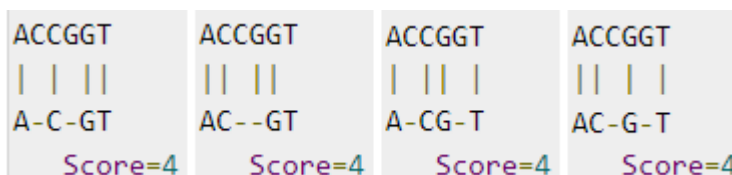


Figure 9: Exemple d'alignement de la seq. ACGT à d'autres séquences

J'ai modélisé ensuite ces résultats en diagramme et graphique pour une meilleure visualisation de la donnée et cela en utilisant la bibliothèque « Matplotlib »

Matplotlib^{ix} est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques.

Le second facteur que j'ai étudié durant cette expérience est le temps d'exécution de chaque technologie d'OCR et cela en utilisant la bibliothèque « Time »

Time^x est un module de python qui permet de manipuler le temps dans un programme.

Résultats

Dans cette section je présente l'ensemble des résultats obtenus lors d'exécutions des deux scripts que j'ai implémentées avec « EasyOCR » et « Tesseract ».

```

PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL

Voici la plaque d'immatriculation lue : 51G29034
Voici la vraie plaque d'immatriculation : 51G29034
Taux de réussite de lecture: 87 %

Traitement de la voiture : car097.jpg
Using CPU. Note: This module is much faster with a GPU.
Voici la plaque d'immatriculation lue : 61F5988
Voici la vraie plaque d'immatriculation : 51F59881
Taux de réussite de lecture: 50 %

Traitement de la voiture : car098.jpg
Using CPU. Note: This module is much faster with a GPU.
Voici la plaque d'immatriculation lue : COVIDIG
Voici la vraie plaque d'immatriculation : COVID19
Taux de réussite de lecture: 71 %

Traitement de la voiture : car099.jpg
Using CPU. Note: This module is much faster with a GPU.
Voici la plaque d'immatriculation lue : 61A72078
Voici la vraie plaque d'immatriculation : 51A72078
Taux de réussite de lecture: 87 %

Traitement de la voiture : car100.jpg
Using CPU. Note: This module is much faster with a GPU.
Voici la plaque d'immatriculation lue : 3144MOS
Voici la vraie plaque d'immatriculation : B144MOS
Taux de réussite de lecture: 85 %

Le temps total écoulé depuis le début du programme est: 290.6055784225464
PS C:\Users\HP\OneDrive\Bureau\LAPI>

```

Figure 11: Résultats de l'expérience (en partie) + temps d'exécution- EasyOCR

```

PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL

Voici la vraie plaque d'immatriculation : 51G29034
Taux de réussite de lecture: 0 %

Traitement de la voiture : car097.jpg
Erreur de lecture de la plaque :(

Voici la vraie plaque d'immatriculation : 51F59881
Taux de réussite de lecture: 0 %

Traitement de la voiture : car098.jpg
Erreur de lecture de la plaque :(

Voici la vraie plaque d'immatriculation : COVID19
Taux de réussite de lecture: 0 %

Traitement de la voiture : car099.jpg
Erreur de lecture de la plaque :(

Voici la vraie plaque d'immatriculation : 51A72078
Taux de réussite de lecture: 0 %

Traitement de la voiture : car100.jpg
La plaque d'immatriculation est : 3144MOS
Voici la plaque d'immatriculation lue : 3144MOS
Voici la vraie plaque d'immatriculation : B144MOS
Taux de réussite de lecture: 85 %

Le temps total écoulé depuis le début du programme est: 27.901565551757812
PS C:\Users\HP\OneDrive\Bureau\LAPI>

```

Figure 12: Résultats de l'expérience (en partie) + temps d'exécution- Tesseract

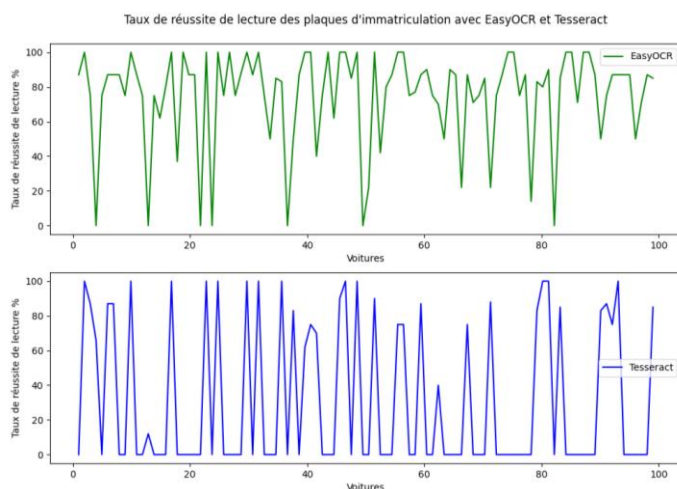


Tableau 2: Graphique de taux de réussite de lecture des plaques d'immatriculation avec EasyOCR et Tesseract

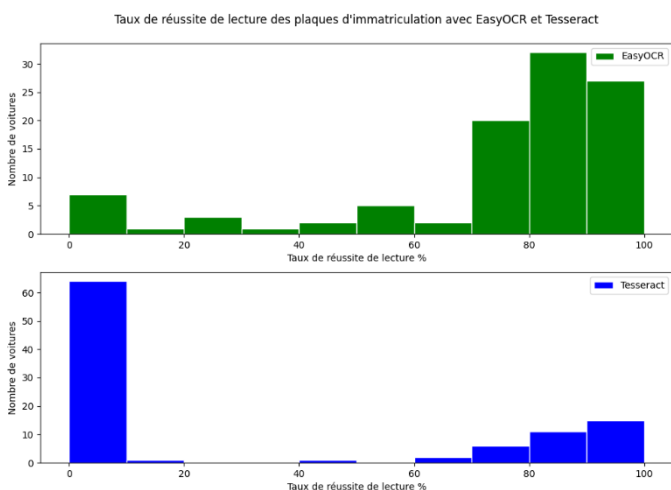


Tableau 3: Diagramme de taux de réussite de lecture des plaques d'immatriculation avec EasyOCR et Tesseract

DISCUSSION

Dans cette section je présente les résultats obtenus et je les commente.

Dans les figures 11 et 12 qui sont des captures d'écran d'exécution des deux scripts implémentées l'un avec « EasyOCR » et l'autre avec « Tesseract » nous remarquons que pour chaque image (véhicule) j'affiche la plaque lue par le logiciel, la plaque réelle du véhicule ainsi que le taux de réussite de la lecture de la plaque d'immatriculation en pourcentage.

On remarquera notamment sur l'exécution de « EasyOCR » la phrase 'Using CPU. Note : this module is much faster with a GPU' qui se traduit par 'Utilisation du CPU. Note : ce module est beaucoup plus rapide avec un GPU' mon ordinateur hélas n'a pas de GPU donc le résultat du temps donné par « EasyOCR » est nettement plus important que celui de « Tesseract », il est fort possible qu'avec un ordinateur qui contient ce processeur graphique que le temps soit inférieur à celui que je montre dans la figure 11. De toute manière dans mon cas l'utilisation de « Tesseract » reste plus rapide.

Dans le Tableau 2 qui représente un graphique de taux de réussite de lecture de plaque d'immatriculation en fonction des voitures avec les deux technologies, on remarque bien qu'il y a plus de réussite au niveau de EasyOCR que dans Tesseract ce que nous pouvons remarquer aussi c'est que généralement EasyOCR permet d'offrir un certain résultat même si juste en partie correct contrairement à Tesseract qui de son côté pour la plupart des résultats soit donne une réponse correcte à 100 % soit c'est lecture impossible (0 %).

Dans le Tableau 3 qui représente un diagramme de taux de réussite de lecture de plaque d'immatriculation en fonction du nombre de voitures avec les deux technologies, on remarque encore une fois qu'il a énormément de voitures qui ont été bien traité avec un taux de réussite de détection allant de 70 % à 100 % avec EasyOCR, tandis que pour avec Tesseract il y a plus de 60 plaques d'immatriculation non lues.

Bien évidemment cela ne dit pas que Tesseract ne sert à rien en présence de EasyOCR mais juste que chaque technologie permet de lire à un certain niveau de précision et de degré d'erreur.

On a constaté avec ces résultats que pour mon projet de LAPI, qu'il est préférable d'utiliser EasyOCR à Tesseract car les résultats obtenus sont favorables plus pour EasyOCR.

CONCLUSION ET PERSPECTIVES

Conclusion

Durant tout cet article j'ai présenté ce que c'est que LAPI « Lecture automatique des plaques d'immatriculation », ses différentes fonctionnalités, ces cas d'utilisation et les programmes qui existent déjà dans ce domaine.

Ma recherche au sein de ce sujet a consisté justement à implémenter ce système et ainsi étudier ces différentes technologies, j'ai eu l'occasion durant cette recherche de côtoyer le langage de programmation Python et les différentes librairies même celle dont le rapport est infime à la programmation en général comme BioPython.

La reconnaissance des plaques d'immatriculation passe par deux étapes majeures l'identification de la plaque et la conversion de celle-ci en texte, j'ai eu l'opportunité de faire des tests et des expériences et visualiser ces données sur les deux technologies d'OCR : « EasyOCR » et « Tesseract » qui nous ont donné chacune des résultats pour un jeu de données de cent (100) véhicules et au final après analyse des données on peut conclure cet article par dire qu'en matière de reconnaissance optique de caractères des plaques d'immatriculation « EasyOCR » est meilleure que « Tesseract ».

Perspectives

La perspective que je peux avoir pour le futur est d'améliorer mes travaux et cela en rendant le logiciel beaucoup plus précis pour qu'il n'ait que des réussites lors d'identifications des plaques d'immatriculation et ensuite la conversion de celle-ci en texte brut.

Cela dépendra aussi de l'utilisation du logiciel, pour lequel nous pourrions ajouter une base de données pour traitement de véhicule si c'est pour un usage de gestion de flux de véhicules par exemple ou pour la détection de véhicules utilisés à des fins criminelles.

On pourrait intégrer un système de vidéo directement pour que le logiciel puisse lire toutes les plaques se trouvant dans cette vidéo, donc il devrait faire la différence entre une plaque de publicité par exemple et celle d'un véhicule et cela reste un très bon défi pour l'avenir !

BIBLIOGRAPHIE

Voici l'ensemble des références que j'ai utilisées pour la rédaction de cet article :

ⁱ https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_reconnaissance_automatique_des_plaques_min%C3%A9ralogiques

ⁱⁱ <https://www.jenoptik.com/products/civil-security/anpr>

ⁱⁱⁱ <https://opencv.org/>

^{iv} <https://github.com/JaidedAI/EasyOCR>

^v <https://github.com/tesseract-ocr/tessdoc>

^{vi} <https://medium.com/mlearning-ai/tesseract-vs-keras-ocr-vs-easyocr-ec8500b9455b>

^{vii} <https://biopython.org/>

^{viii} https://www.tutorialspoint.com/biopython/biopython_sequence_alignments.htm

^{ix} <https://matplotlib.org/>

^x <https://docs.python.org/fr/3/library/time.html>