

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة عين تموشنت بلحاج بوشعيب  
Université -Ain Temouchent- Belhadj Bouchaib  
Faculté des Sciences et de Technologie  
Département de Mathématiques et Informatique



Projet de Fin d'Etudes  
Pour l'obtention du diplôme de Master en : Réseau et Ingénierie des  
Données  
Domaine : Mathématiques et Informatique  
Filière : Informatique  
Spécialité : Réseau et Ingénierie des Données

Thème

**Étude comparative entre des techniques de reconnaissance de caractères en vue de son application à l'identification des plaques d'immatriculation**

Présenté par :

- 1) M ERROUANE Walid
- 2) Melle BENAOUA HASCAR feriel

Devant le jury composé de :

<b>Dr BOUCHAKOUR ERRAHMANI</b> Hichem	.....	UAT.B.B Temouchent)	(Ain Président
<b>Dr BENZERBADJ Ali</b>	.....	UAT.B.B Temouchent)	(Ain Examineur
<b>Dr MERAD BOUDIA Djalal</b>	.....	UAT.B.B Temouchent)	(Ain Encadreur

Année Universitaire 2022/2023

There are currently many license plate recognition systems in the world, and these systems are based on two main axes:

Finding out the position of the license plate and then identify its numbers and letters, to do this, there are several approved methods to extract and recognize letters and numbers, we propose through this work to design a system in which we have chosen five detection methods. on license plates and for this we created a voting system instead of improving one of the methods, so we decided to complete the voting system.

Thanks to it, we form a long-awaited license plate, the results obtained and applied to 160 images were promising so that the percentage of recognition of a painting was always a percentage of ration greater than one individual ratio for each method.

Il existe actuellement de nombreuses méthodes de reconnaissance de plaques d'immatriculation et ces systèmes sont basés sur deux axes principaux :

Découvrir la position de la plaque d'immatriculation puis l'identification de ses chiffres et ses lettres, nous proposons à travers ce travail de concevoir un système dans lequel nous avons choisi cinq méthodes de détection sur les plaques d'immatriculation, afin d'améliorer les performances globales des 5 méthodes, nous avons conçu un système de votes où chaque méthode pourra voter du caractère qu'elle a détecté.

Les résultats obtenus et appliqués à 160 images étaient prometteurs, le système de votes a réussi à obtenir un taux d'erreur inférieure aux taux d'erreur individuel de chaque méthode de détection.

يوجد حاليا العديد من أنظمة للتعرف على لوحات الترخيص في العالم، وهذه الأنظمة تركز على محورين أساسيين هما اكتشاف موضع لوحة الترخيص والتعرف بعد ذلك على أرقامها وحروفها، للقيام بهذه العملية هناك عدة طرق معتمدة لاستخراج الحروف والأرقام والتعرف عليها، نقترح من خلال هذا العمل تصميم نظام الذي اخترنا فيه خمسة طرق للكشف على لوحات ترخيص ومن أجل ذلك قمنا بإنشاء نظام تصويت بدلا من تحسين إحدى الطرق فارتأينا بإنجاز نظام التصويت ومن خلاله نشكل لوحة ترخيص مرتقبة.

النتائج المتحصل عليها والمطبقة على 170 صورة فكانت واعدة بحيث كانت نسبة تعرف على لوحة كانت دائما نسبة أعلى من نسبة فردية لكل طريقة.

# Sommaire

<b>Introduction générale</b> .....	1
<b>Chapitre I: Généralités sur le traitement d'images</b> .....	4
I.1 Introduction.....	5
I.2 Définition de l'image.....	5
I.3 Images animées .....	5
I.4 Images en 3D .....	6
I.5 Image en couleur .....	7
I.6 Images binaires (noire et blanc) .....	8
I.7 Représentation d'image .....	8
I.7.1 Image vectorielle.....	8
I.7.2 Image matricielle.....	9
I.8 Caractéristiques d'une image.....	9
I.8.1 Taille.....	10
I.8.2 Bruit.....	10
I.8.3 Luminance .....	11
I.8.4 Résolution.....	11
I.8.5 Contraste.....	11
I.8.6 Histogramme.....	12
I.8.7 La profondeur de couleur.....	13
I.9 Acquisition d'une image .....	13
I.10 Les format d'image .....	13
I.10.1 Format JPEG.....	13
I.10.2 Format PNG.....	14
I.10.3 Format GIF .....	14
I.10.4 Format BMP .....	14
I.10.5 Format TIFF.....	14
I.10.6 Format SVG.....	14
I.11 Traitement d'images .....	14
I.12 Les principales étapes de traitement d'image.....	15
I.13 Les techniques de traitement d'images .....	15
I.13.1 Filtrage d'images.....	15
I.13.2 Segmentation d'images: .....	15
I.13.3 Reconnaissance d'objets:.....	15

I.13.4	Restauration d'images: .....	15
I.14	L'application du traitement d'images dans la vie quotidienne .....	16
I.14.1	Authentication par empreintes digitales .....	16
I.14.2	Reconnaissance de visages .....	16
I.14.3	Surveillance de la circulation.....	16
I.14.4	Diagnostic médical.....	16
I.14.5	Réalité augmentée .....	16
I.14.6	Contrôle qualité dans l'industrie.....	16
I.15	Conclusion .....	16
Chapitre II:	La reconnaissance optique de caractères (OCR) .....	17
II.1	Introduction sur L'OCR.....	18
II.1	Deep learning .....	19
II.1.1	Neurone naturel.....	19
II.1.2	Première tentative de modélisation d'un neurone .....	19
II.1.3	La règle de Hebb .....	20
II.1.4	Perceptron.....	21
II.1.5	Réseau de perceptron multi couche.....	25
II.2	Phase de détection du texte.....	26
II.2.1	Introduction sur la détection du texte .....	26
II.2.2	Les challenges que rencontrent les algorithmes de détection de texte .....	27
II.2.3	Les méthodes de détection du texte.....	29
II.3	Phase de reconnaissance du texte.....	31
II.3.1	Introduction à la reconnaissance de texte .....	31
II.3.2	CTC .....	32
II.3.3	Sequence2sequence .....	32
II.3.4	RARE .....	32
II.3.5	Attention algorithm.....	33
II.3.6	Transformers .....	33
II.4	Conclusion .....	33
Chapite III:	Études comparatives entre les principaux modèles de reconnaissance optique de caractères	35
III.1	Présentation des modèles utilisés .....	36
III.1.1	CRNN (EasyOcr et keras-ocr) .....	36
III.1.2	LTSM (Pytesseract).....	37
III.1.3	Transformers(TrOCR) .....	38
III.1.4	Reciprocal Feature Learning(PadleOCR).....	40

III.2	Présentation de l'algorithme de votes.....	40
III.3	Études comparatives.....	42
III.3.1	Résultats des tests des 5 modèles sur le premier data sets.....	43
III.3.2	Résultats des tests des 5 modèles sur le deuxième data sets.....	47
III.3.3	Résultats des tests sur l'algorithme de votes .....	49
III.4	Conclusion .....	51
	Conclusion générale et perspectives.....	53
	Bibliographie.....	55



## *Figures et tableaux*

Figure 1 : Image animées.....	5
Figure 2: Images en 3D.....	6
Figure 3: Image en niveau de gris .....	7
Figure 4: Image en couleur .....	7
Figure 5: Image noire et blanc .....	8
Figure 6: Image vectorielle .....	9
Figure 7: Image matricielle.....	9
Figure 8 : Exemple de calcul de la taille d'image .....	10
Figure 9: Le bruit numérique en noir et blanc .....	10
Figure 10: Exemple de résolution d'image.....	11
Figure 11: Le contraste d'image en gris .....	12
Figure 12: Exemple d'image de histogramme .....	12
Figure 13: Exemple de la profondeur de couleur .....	13
Figure 14: Conversion d'un "A" en matrice binaire .....	18
Figure 15 : neurone naturel [5].....	19
Figure 16 : neurone artificiel.....	20
Figure 17: nuage de point .....	21
Figure 18: prédiction de l'animal qui figure sur la photo .....	22
Figure 19 : graphe de la fonction Sigmoidale.....	23
Figure 20: fonction de coût.....	24
Figure 21: nuage de points non linéairement séparables .....	25
Figure 22: schéma représentant un MLP [17].....	25
Figure 23 : classification non linéaire .....	26
Figure 24 : détection d'une plaque d'immatriculation sur une voiture .....	27
Figure 25: Algorithme de détection qui ne prend pas en compte les symboles.....	28
Figure 26 : quartier de Shibuya , Tokyo , Japon.....	29
Figure 27: Diagramme représentant les frames après le changement de ratio [19] .....	30
Figure 28: Transformation d'une image contenant un texte non régulier en texte régulier .....	33
Figure 29: démonstration d'une opération de filtrage sur Microsoft Excel .....	36
Figure 30: Réseau de types RNN [30].....	37
Figure 31: architecture d'un CRNN [30] .....	37
Figure 32 : architecture d'un LSTM .....	38
Figure 33 : traduction d'une phrase dans une architecture Encoder-Décodeur [39] .....	39
Figure 34: traduction d'une phrase dans une architecture Encoder-Décodeur en utilisant le mécanisme d'attention.....	39
Figure 35: architecture du modèle de reconnaissance de paddle [41] .....	40
Figure 36: schéma montrant les entrées et les sorties de la fonction d'élection.....	41
Figure 37 : Plaque d'immatriculation .....	41
Figure 38: résultats de la prédiction avec leurs tailles respectives.....	41
Figure 39: Valeurs finales retournées par l'algorithme de votes ainsi que des informations relatives à la plaque détectée.....	42

Figure 40: Explication de N,S,I,D .....	43
Figure 41: Histogramme des taux S,D,I sur le premier Data Sets .....	46
Figure 42: Graphique circulaire représentant le CER de chaque modèle sur le data sets 1 .....	46
Figure 43 : Histogramme représentant les taux S,D,I de chaque modèle sur le data sets 2 .....	49
Figure 44: Graphique circulaire représentant le CER de chaque modèle sur le data sets 2 .....	49
Tableau 1: Méthodes utilisé pour la détection de texte .....	29
Tableau 2: Tableau contenant les résultats des testes du premier data sets sur les 5 modèle ..	45
Tableau 3 : résultats des modèles sur le deuxième datasets .....	48
Tableau 4 : résultats de l'algorithme de votes sur le premier datasets.....	51
Tableau 5: Récapitulatif des résultats.....	51
Tableau 6: catégorie selon la performance .....	52

# *Introduction générale*

Dans le monde où nous vivons aujourd'hui, nous sommes entourés de texte en tous genres. Avec l'évolution de l'informatique, le besoin de trouver un procédé permettant la numérisation de textes imprimés ou écrits à la main est vite fait ressenti. Car le format numérique est le format le plus avantageux pour les raisons suivantes : il est facile à stocker car il ne requiert pas un espace de stockage important, il est facilement partageable avec d'autres personnes via un réseau interne ou via internet, sur un texte en format numérique il est possible d'appliquer des techniques d'analyse automatisées pour faire de la fouille de données « Data Mining ». Le procédé qui fait la conversion d'un texte contenu dans une image en caractère lisible par la machine est appelé la reconnaissance optique de caractères, en anglais « Optical Character Recognition » abrégé en « OCR ».

L'OCR trouve son application dans des domaines divers et variés : les voitures intelligentes utilisent des algorithmes d'OCR pour la reconnaissance des panneaux de signalisation. Aussi, il existe des solutions de reconnaissance de plaque d'immatriculation, certains endroits disposent d'un portail intelligent qui s'ouvre uniquement aux voitures dont la plaque d'immatriculation est référencée sur sa base de données, l'OCR a permis à certains de contourner les systèmes anti-Bots, généralement ce genre de système demande à un utilisateur souhaitant accéder à un site web, de remplir un formulaire selon le texte qui figure sur une image.

Généralement, l'OCR se divise en trois parties. La première partie est le traitement de l'image d'entrée, dans cette première étape l'image subit des transformations (Grey Scaling, Thresholding...) pour obtenir de meilleurs résultats pour les étapes qui suivent. Pour faire la reconnaissance d'un texte, il faut d'abord détecter où il se trouve. C'est à ça que sert la deuxième partie. Une fois que le texte est détecté, il est possible de faire la reconnaissance de celui-ci. Les solutions d'OCR qui font ces trois tâches à la fois sont appelées « End-to-End ».

La majorité des algorithmes de détection et de reconnaissance de texte sont basés sur des modèles de Deep Learning car ils ont prouvé qu'ils étaient les meilleurs à ce domaine. Toutefois, pour obtenir des résultats corrects avec ce genre de méthode, la démarche à suivre n'est pas à la portée de tout le monde. Pour sa phase d'apprentissage, un modèle de Deep Learning a besoin d'une grande quantité de photos (des millions). À ces photos est associée un autre fichier qui contient le contenu textuel de l'image qu'on appelle « Label ». Bien qu'un modèle puisse bien performer sur ses données d'apprentissage, il peut toutefois échouer sur des données qu'il n'a pas encore vues et ce problème s'appelle « Overfitting ». Un autre challenge que rencontrent les concepteurs de modèles de Deep Learning est le coût matériel pour la phase d'apprentissage, pour la conception de MT-NLG le CEO de Nvidia a déclaré qu'il a fallu la puissance de calcul de milliers de cartes graphiques et cela pendant des semaines.

C'est dans ce contexte que nous proposons une solution permettant d'améliorer les performances des modèles de reconnaissance optique basée sur le Deep Learning sans pour autant refaire l'apprentissage ou de modifier l'architecture des modèles. Nous savons très bien que les méthodes sélectionnées ont été optimisées et testées en utilisant des machines puissantes et des bases de données de grande taille et diversifiées, alors l'amélioration de ces méthodes nécessite sûrement des moyens plus avancés et un temps d'étude et d'analyse considérable. Pour cela, nous avons pensé à combiner les résultats de ces méthodes pour en déduire un résultat final. Nous allons utiliser des modèles pré-entraînés pour faire la reconnaissance de caractères, les résultats de chacun de ses modèles seront les inputs d'un système de vote que nous avons conçu. La valeur retournée sera issue d'un vote entre les modèles de reconnaissance. Notre solution sera focalisée sur les plaques d'immatriculation.

Notre mémoire se divise en trois chapitres qui sont organisés de la manière suivante : un premier chapitre, dans lequel nous présentons le traitement d'images, qui est la première phase de la reconnaissance optique de caractères. Le deuxième chapitre est un état d'art de la détection et de la reconnaissance de caractères. Le dernier chapitre est consacré à une étude comparative entre 5 modèles de Deep Learning pour l'OCR. Ensuite, nous présentons l'algorithme que nous avons conçu qui permet d'obtenir de meilleurs résultats à partir des 5 modèles. Enfin, nous achevons ce mémoire par une conclusion générale ou nous revenons sur les résultats obtenus et où nous ouvrons une discussion sur les principales perspectives.

## *Chapitre I: Généralités sur le traitement d'images*

### I.1 Introduction

Le traitement d'images est un domaine de l'informatique qui a connu un développement considérable ces dernières années grâce à l'avancement des technologies numériques. Il consiste en l'application de techniques et d'algorithmes pour analyser et manipuler des images numériques afin d'améliorer leur qualité, de les modifier ou d'en extraire des informations utiles. Il est utilisé dans de nombreux domaines, tels que la photographie, la médecine, l'industrie, etc.

Dans ce chapitre, nous allons aborder les notions de base nécessaires à la compréhension des techniques de traitement d'images. (1)

### I.2 Définition de l'image

Une image est obtenue par transformation d'une scène réelle par un capteur, donc c'est un signal 2D  $(x, y)$ , qui représente souvent une réalité 3D  $(x, y, z)$ . Pour un ordinateur, une image est un ensemble de pixels. Un pixel, c'est le plus petit élément d'une image d'un point de vue mathématique, une image est une matrice de nombres représentant un signal, plusieurs outils permettent de manipuler ce signal.

### I.3 Images animées

Images animées ce sont des images qui contiennent des séquences d'images qui sont affichées à une vitesse déterminée pour créer une illusion de mouvement. (2)

Les formats courants pour les images animées sont GIF, APNG et WEBP.

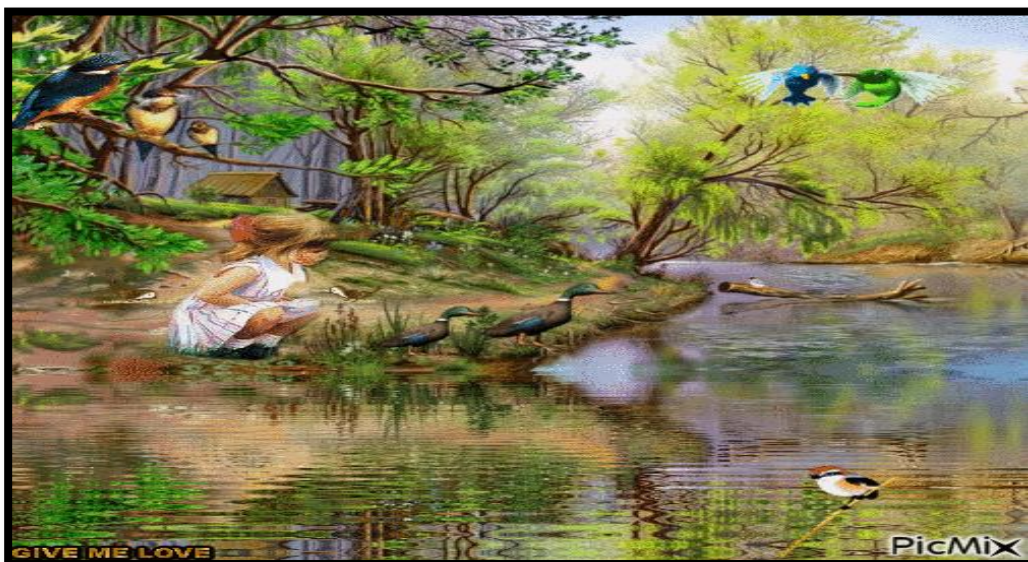


Figure 1 : Image animées

## I.4 Images en 3D

Ce sont des images créées à partir de modèles 3D, qui peuvent être visualisées en utilisant des logiciels de visualisation 3D ou des lunettes 3D spéciales. (2)

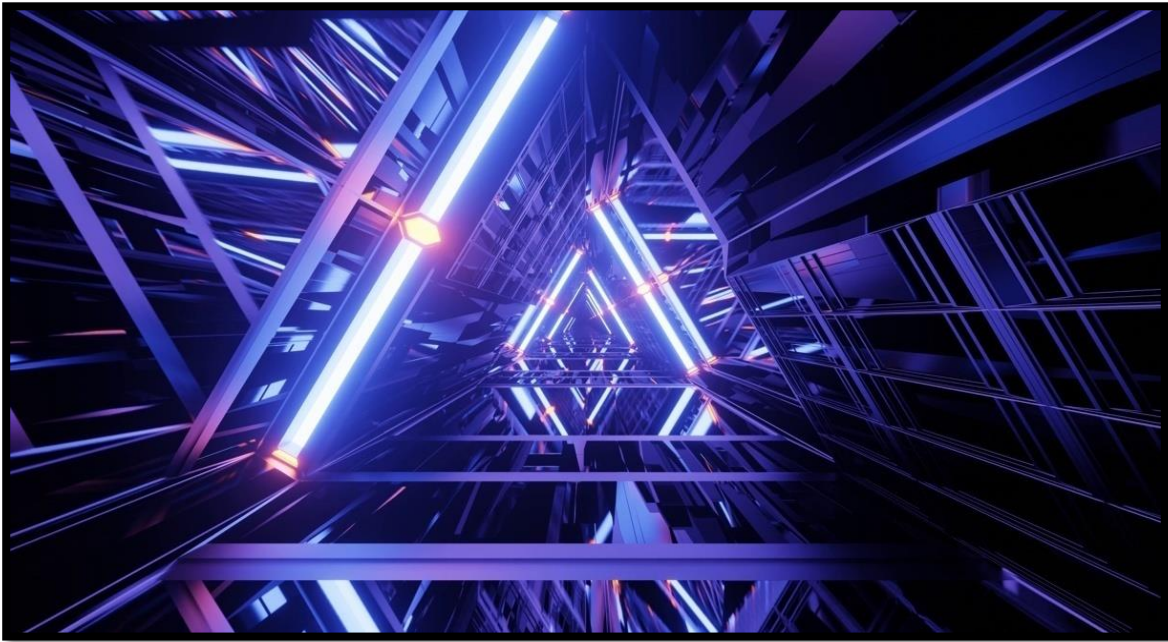


Figure 2: Images en 3D

## 3. Types d'images

Il existe trois principaux types d'images :

### 3.1 Image en niveau de gris

Ce sont des images qui utilisent uniquement des nuances de gris pour représenter des images, plutôt que des couleurs. Chaque pixel est un niveau de gris, allant de 0 (noir) à 255 (blanc). Cet intervalle de valeur signifie que chaque pixel est codé sur huit bits (un octet). 256 niveaux de gris suffisent pour la reconnaissance de la plupart des objets d'une scène. (1)

Elles sont souvent utilisées pour les documents tels que les livres, les journaux et les magazines.





Figure 3: Image en niveau de gris

### I.5 Image en couleur

Ce sont des images qui utilisent des couleurs pour représenter des images. Chaque pixel possède une couleur décrite par la quantité de rouge (R), vert (G) et bleu (B). Chacune de ces trois composantes est codée sur l'intervalle  $[0, 255]$ , ce qui donne  $256^3=16\ 777\ 216$  couleurs possibles. Il faut 24 bits pour coder un pixel. (1)

Les formats courants pour les images en couleur sont JPEG, PNG et TIFF.



Figure 4: Image en couleur

## I.6 Images binaires (noire et blanc)

Image binaire est une image qui ne contient que deux couleurs, généralement noir et blanc. Chaque pixel de l'image est soit noir (représenté par la valeur 0) ou blanc (représenté par la valeur 1). (1)

Il existe plusieurs formats de fichiers pour les images binaires, tels que BMP, TIFF, PNG et GIF. Dans ces formats, les pixels de l'image sont stockés dans une matrice qui peut être lue et manipulée à l'aide de logiciels de traitement d'images.



Figure 5: Image noire et blanc

## I.7 Représentation d'image

L'image est un ensemble structure d'informations représente par les paramètres suivants :

### I.7.1 Image vectorielle

Image vectorielle en informatique (image en mode trait), est une image numérique qui est créée en utilisant des formes géométriques individuels telles que des arcs de cercle, segments de droite, des polygones..., définis chacun par divers attributs de couleur, de position, de forme.... Par exemple, une image vectorielle d'un cylindre est définie par des attributs de types : hauteur, rayon, position du centre.

Les images vectorielles ont plusieurs avantages. Tout d'abord, elles peuvent être agrandies sans perte de qualité ou de résolution. Cela les rend particulièrement adaptées aux logos, aux graphiques et aux illustrations qui doivent être utilisés dans différents formats et tailles. De plus, les images vectorielles ont des tailles de fichier plus petites ce qui les rend plus faciles à stocker, à transférer et à éditer. [1]

Les formats de fichiers courants pour les images vectorielles sont SVG, PDF et AI.

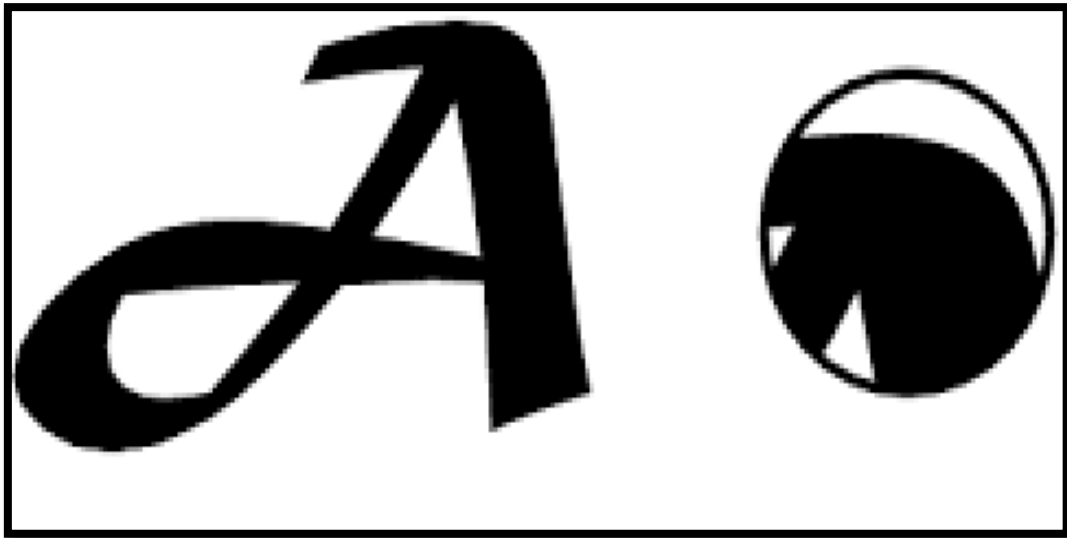


Figure 6: Image vectorielle

### I.7.2 Image matricielle

Image matricielle (image bitmap ou images raster), est une représentation numérique d'une image sous forme de matrice de pixels ou d'un assemblage de points ou de pixels. Chaque pixel est représenté par une valeur numérique qui indique sa luminosité et sa couleur.

Les formats d'image matricielle courants incluent JPEG, PNG, GIF et BMP.

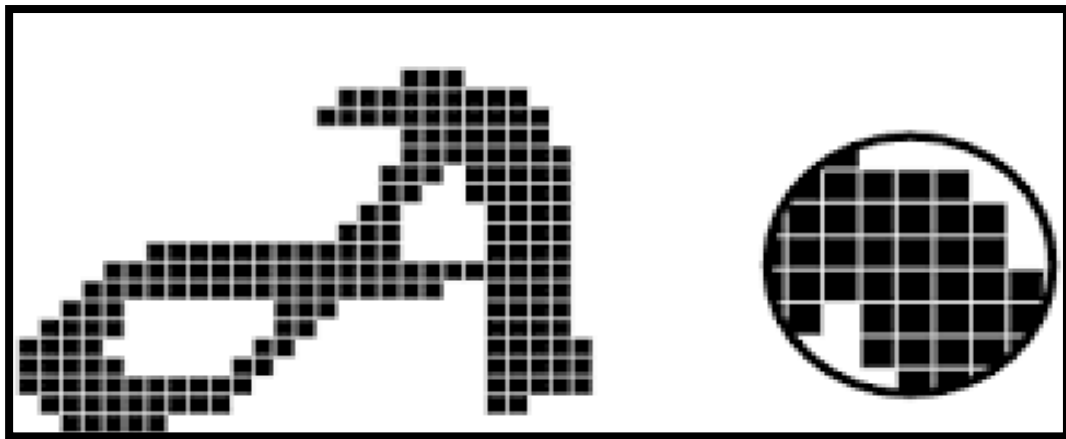


Figure 7: Image matricielle

### I.8 Caractéristiques d'une image

Les caractéristiques d'une image dépendent du type d'image et de l'application dans laquelle elle est utilisée. Cependant, voici quelques-unes des caractéristiques générales qui peuvent s'appliquer à la plupart des images :

### I.8.1 Taille

La taille de l'image est généralement mesurée en kilooctets (Ko) ou mégaoctets (Mo), mais dans une image numérique c'est la taille de nombre de pixels constituant l'image, c'est à dire sa dimension informatique (le nombre de colonnes de l'image que multiplie son nombre de lignes). [2]

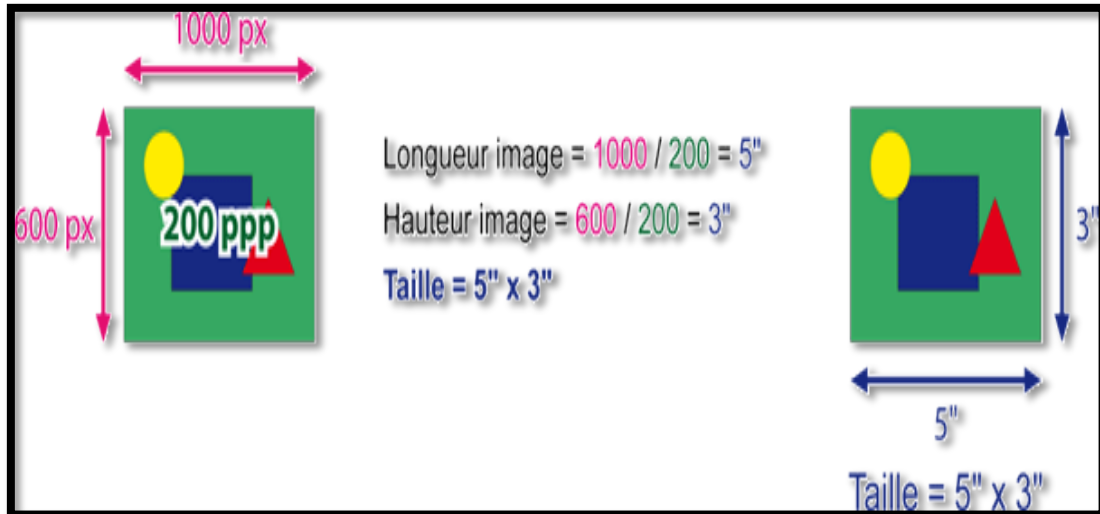


Figure 8 : Exemple de calcul de la taille d'image

### I.8.2 Bruit

Le bruit dans une image est l'ensemble des pixels qui parasitent la photo, c'est un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, c'est à dire consiste en la dégradation progressive de la qualité de l'image au fur et à mesure que l'on augmente la sensibilité du capteur, il est dû aux limites en faible lumière du capteur et de l'électronique de l'appareil photo et donc entraîne une perte de la netteté dans les détails de l'image [2]



Figure 9: Le bruit numérique en noir et blanc

### I.8.3 Luminance

La luminance (ou brillance) est le degré de luminosité des points de l'image autrement dit c'est le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface pour un observateur lointain.

### I.8.4 Résolution

C'est le nombre de pixels par unité de mesure (pouce ou centimètre) dans une image affichable horizontalement ou verticalement sur un moniteur. On dit la résolution est élevée alors l'image est détaillée et de qualité supérieure. [2]

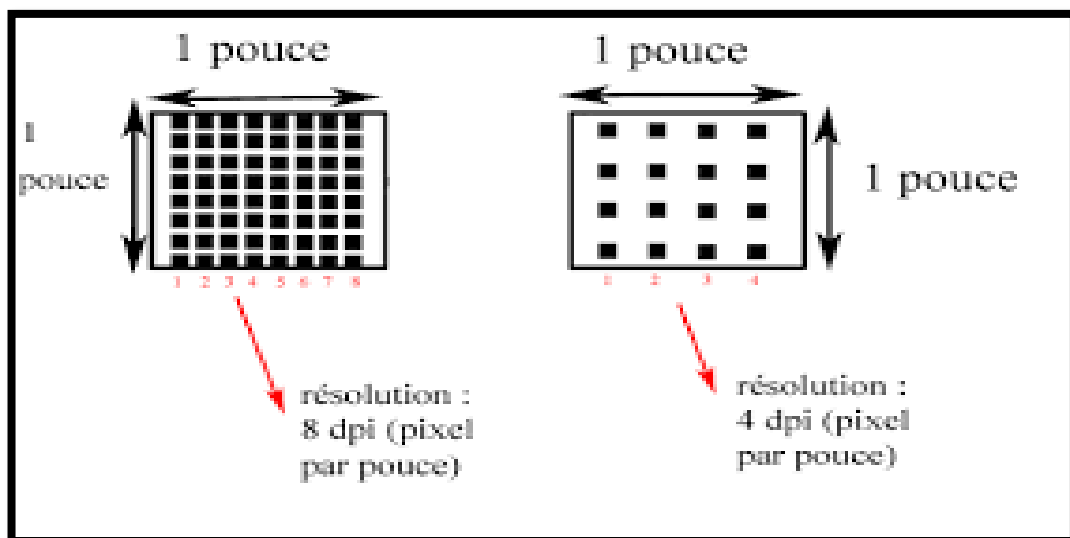


Figure 10: Exemple de résolution d'image

### I.8.5 Contraste

Le contraste est une mesure de la différence entre les valeurs les plus sombres et les plus claires d'une image ou d'un objet. En d'autres termes, c'est la différence de luminosité entre les parties les plus lumineuses et les plus sombres d'une image.

Le réglage du contraste est un élément clé de la retouche d'images pour améliorer leur qualité visuelle ou pour répondre à des besoins spécifiques. [2]

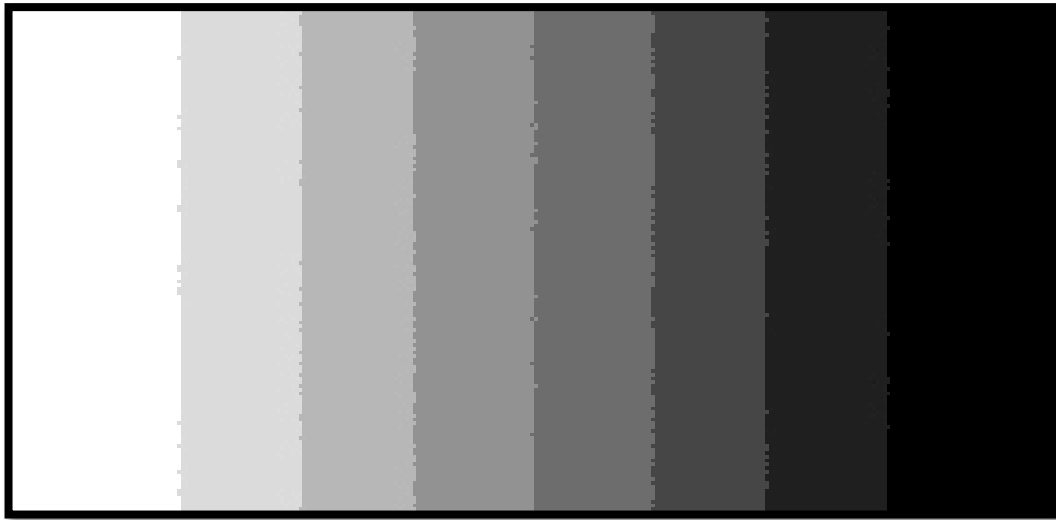


Figure 11: Le contraste d'image en gris

### I.8.6 Histogramme

L'histogramme est un graphique qui représente la répartition des valeurs d'un ensemble de données en groupes, appelés classes. L'histogramme de niveaux des gris d'une image est une fonction qui permet de représenter la distribution des couleurs de l'image, en indiquant pour chaque valeur entre le noir (0) et le blanc (255), combien il y a de pixels de cette valeur dans l'image ; l'axe horizontal représente les valeurs des classes donc c'est le niveau de gris (de 0 à 255) et l'axe vertical représente la fréquence ou le nombre de données qui se trouvent dans chaque classe donc c'est le nombre de pixels.

Les pixels clairs paraissent à droite de l'histogramme et sombres à gauche de l'histogramme, et les pixels gris au centre de l'histogramme. [2]

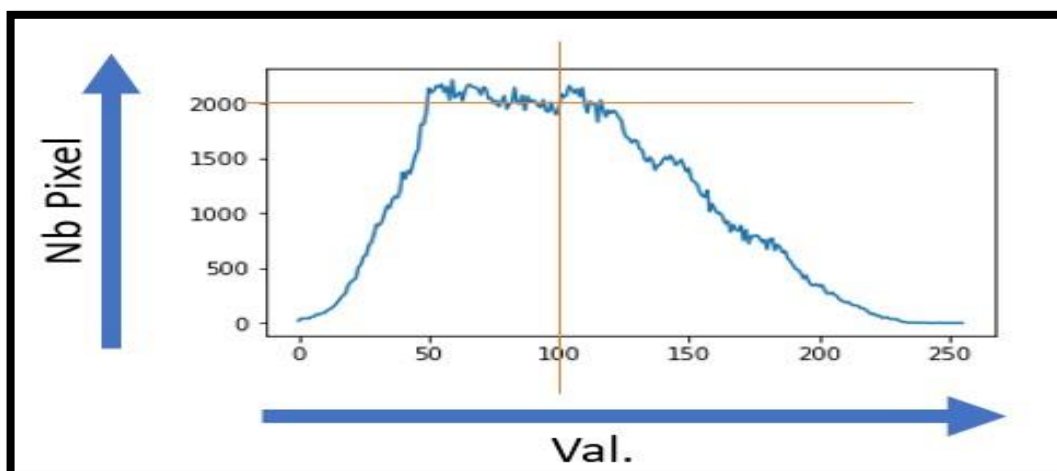


Figure 12: Exemple d'image de histogramme

### I.8.7 La profondeur de couleur

La profondeur de couleur, également connue sous le nom de profondeur de bits, se réfère à la quantité de bits utilisée pour stocker la couleur de chaque pixel dans une image numérique. [2]

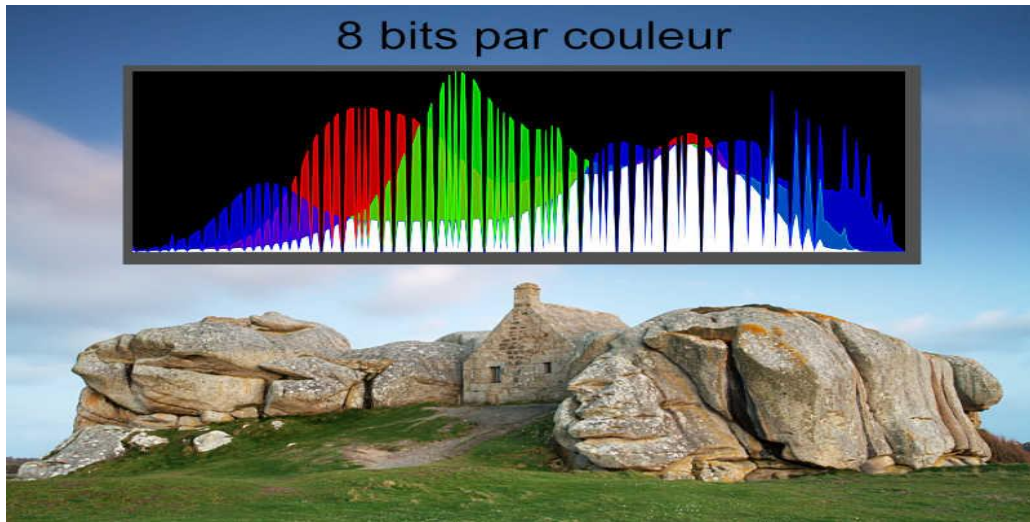


Figure 13: Exemple de la profondeur de couleur

### I.9 Acquisition d'une image

L'acquisition d'une image fait référence au processus de capturer une représentation visuelle d'un objet, elle constitue un des opérations essentielles de toute chaîne de conception et de production d'images. [2]

Elle permet de transformer l'objet à reconnaître d'une forme réelle ou dite physique en une forme numérique comprise par les ordinateurs, cela peut être réalisé par classement en deux catégories principales :

- Les caméras numériques
- Les scanners.

### I.10 Les formats d'image

Il existe de nombreux formats d'image différents, chacun ayant ses propres avantages et inconvénients en termes de taille de fichier, qualité d'image, transparence, compression, support de couleurs, compatibilité logicielle, etc. Voici une brève présentation des formats d'image les plus courants : [2]

#### I.10.1 Format JPEG

Le format JPEG (Joint Photographic Experts Group) est le format le plus couramment utilisé pour les images photographiques sur le web et dans les applications mobiles. Il permet une compression avec perte de qualité, ce qui signifie que la qualité de l'image est légèrement réduite pour réduire la taille de fichier.

### I.10.2 Format PNG

Le format PNG (Portable Network Graphics) est largement utilisé pour les images en ligne qui nécessitent une transparence et une qualité d'image élevée, comme les logos et les icônes. Il utilise une compression sans perte de qualité, ce qui signifie que la qualité de l'image n'est pas réduite lorsqu'elle est sauvegardée en tant que fichier PNG.

### I.10.3 Format GIF

Le format GIF (Graphics Interchange Format) est utilisé pour les images animées et les graphiques simples avec peu de couleurs. Il utilise une compression sans perte de qualité, mais il est limité à 256 couleurs et ne convient donc pas pour les images photographiques ou les illustrations complexes.

### I.10.4 Format BMP

Le format BMP (Bitmap) est utilisé pour les images sans compression de qualité maximale. Il est souvent utilisé pour les images de haute résolution pour l'impression, mais sa taille de fichier élevée rend son utilisation limitée sur le web.

### I.10.5 Format TIFF

Le format TIFF (Tagged Image File Format) est utilisé pour les images haute résolution pour l'impression professionnelle et les échanges de fichiers d'images haute qualité. Il utilise une compression sans perte de qualité et peut stocker des images avec des couleurs profondes et une haute résolution.

### I.10.6 Format SVG

Le format SVG (Scalable Vector Graphics) est utilisé pour les graphiques vectoriels, tels que les logos et les icônes. Il utilise une compression sans perte de qualité et permet aux images d'être redimensionnées sans perte de qualité.

## I.11 Traitement d'images

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui traite de l'étude des images numériques et de leurs transformations afin d'améliorer, d'analyser, de modifier, d'interpréter ou d'extraire certaines informations. Il peut s'agir de techniques permettant de créer des effets visuels (par exemple, en appliquant des filtres ou des transformations géométriques), d'améliorer la qualité de l'image (par exemple, en réduisant le bruit ou en améliorant la netteté), de reconnaître des objets ou des formes dans une image (par exemple, en utilisant des motifs techniques de reconnaissance), ou pour extraire des informations utiles de l'image (par exemple, en détectant les bords ou en segmentant l'image en différentes zones). Quelques exemples des types d'informations que vous pouvez obtenir à partir d'une image numérique :

- ✓ Le taux d'acuité moyen (précis ou flou).
- ✓ Le taux d'uniformité des couleurs.



✓ La luminance moyenne.

✓ La couleur prédominante.

✓ Le contraste moyen. [1]

## I.12 Les principales étapes de traitement d'image

Le traitement d'image consiste en un ensemble d'opérations effectuées sur une image numérique afin d'en améliorer la qualité, la lisibilité, la visualisation ou l'analyse. Les principales étapes de traitement d'image sont les suivantes : [2]

- Acquisition
- Prétraitements des images.
- Segmentation
- Extraction des caractéristiques
- Classification
- Post-traitement
- Visualisation

## I.13 Les techniques de traitement d'images

Il existe de nombreuses techniques de traitement d'images, telles que :

### I.13.1 Filtrage d'images

Les filtres peuvent être utilisés pour réduire le bruit dans une image, améliorer la netteté, ou encore détecter des contours ou des motifs dans l'image. [1]

### I.13.2 Segmentation d'images:

La segmentation d'images est une technique de traitement d'images qui permet de diviser une image en régions homogènes. [1]

### I.13.3 Reconnaissance d'objets:

Cette technique est souvent utilisée dans des applications telles que la surveillance vidéo, la reconnaissance de caractères et la reconnaissance faciale. [1]

### I.13.4 Restauration d'images:

Cette technique peut être utilisée pour éliminer les rayures, les taches ou les artefacts numériques présents dans une image.

### I.14 L'application du traitement d'images dans la vie quotidienne

L'application du traitement d'images dans la vie quotidienne est de plus en plus courante et diversifiée. Elle est utilisée dans plusieurs domaines tels que la sécurité, la médecine, la recherche scientifique, l'industrie, l'automobile, l'agriculture, etc.

Parmi les applications les plus courantes, on peut citer :

#### I.14.1 Authentification par empreintes digitales

Le traitement d'images est utilisé pour capturer l'image de l'empreinte digitale d'une personne et la comparer avec les empreintes digitales stockées dans une base de données pour vérifier l'identité de la personne.

#### I.14.2 Reconnaissance de visages

Cette technique est utilisée pour identifier les personnes à partir de leurs images faciales. Elle est utilisée dans les applications de sécurité pour la surveillance des personnes et la détection des criminels.

#### I.14.3 Surveillance de la circulation

Le traitement d'images est utilisé pour surveiller la circulation routière en utilisant des caméras de sécurité pour détecter les infractions et les accidents.

#### I.14.4 Diagnostic médical

Le traitement d'images est utilisé pour l'analyse des images médicales, telles que les radiographies, les IRM et les scanners, pour diagnostiquer des maladies et des troubles.

#### I.14.5 Réalité augmentée

Cette technologie utilise le traitement d'images pour superposer des images virtuelles sur le monde réel. Elle est utilisée dans les jeux vidéo, les publicités, l'éducation, etc.

#### I.14.6 Contrôle qualité dans l'industrie

Le traitement d'images est utilisé pour inspecter les produits industriels pour détecter les défauts de fabrication, les irrégularités et les variations de la qualité.

### I.15 Conclusion

Dans ce chapitre, nous exposons la notion de traitement d'images ainsi que les termes de base qui représentent l'image et ses différentes caractéristiques puis, nous avons abordé les étapes et les techniques de traitement d'image.

Dans ce qui va suivre, nous allons présenter les méthodes de détection et de reconnaissance des images.

*Chapitre II: La reconnaissance optique de caractères (OCR)*

### II.1 Introduction sur L'OCR

La reconnaissance optique de caractères en anglais « Optical Character Recognition » OCR, est le fait de transformer des textes figurant sur des images en caractère numérique qui peuvent être traité par la machine, il existe principalement deux types d'algorithmes d'OCR :

- Les algorithmes de reconnaissance simple sont une méthode rudimentaire, l'image à reconnaître est convertis en une matrice binaire ou chaque élément de la matrice représente pixel qui prend soit 1 ou 0 car les couleurs de l'image sont soit blanc 0 ou noir 1, l'algorithme dispose d'une base de donnée qui contient chaque alphabet écrite dans différent police, par la suite il fait une comparaison pour déterminer quelle est le caractère qui a le plus de similitude,



Figure 14: Conversion d'un "A" en matrice binaire

- Les algorithmes de reconnaissance intelligent ont pour objectif d'identifier les caractères de la même façon que les humains le font ce qui est d'ailleurs l'un des objectifs de l'intelligence artificielle qui est d'essayer de reproduire l'intelligence humaine à travers des réseaux de neurone artificiel, cette méthode est bien plus performante que les algorithmes de reconnaissance simple

- La compétition annuelle ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (3) teste des algorithmes de reconnaissance d'Object sur un data sets de 1.2 millions d'images pour les catégoriser sur plus de 5000 catégorie, avant 2012 le meilleur score était un taux d'erreur de 26.2%, cependant en 2012, un algorithme appelé AlexNet basé sur le Deep Learning a obtenu pour la première fois un taux d'erreur record de 17.0% (4), depuis L'OCR est dominé par des algorithmes de reconnaissance intelligent basée sur le Deep Learning.

-L'OCR trouve de nombreux domaines d'applications par exemple dans les entreprises beaucoup de documents circule sous format papier tel que les factures, les demandes d'achats, demande de maintenance, les CV... Ses documents seront ensuite scannés, après le scan c'est là qu'intervient l'OCR. Les documents scannés seront transformés en caractères lisibles par la machine. Ensuite ils pourront être utilisés pour l'analyse de données. Un autre exemple est celui des portails intelligent, une caméra filme la voiture qui se positionne devant l'entrée de façon à avoir une vue sur sa plaque d'immatriculation, d'abord l'algorithme détecte où se trouve la plaque. Puis il effectue la reconnaissance. Le résultat est comparé avec la base des matricules

autorisés à passer à travers le portail. Si la plaque qui a été reconnue s'y trouve alors le portail s'ouvre automatiquement sinon le portail reste fermé et un agent est dépêché sur place.

-Dans ce chapitre nous allons faire une introduction sur les bases du Deep Learning, sur quoi il a été inspiré et comment ils fonctionnent, nous allons ensuite faire un état de l'art sur les algorithmes OCR basés sur le deep Learning.

### II.1 Deep learning

Le Deep learning est basé sur le fait d'imiter le fonctionnement d'un neurone naturel du cerveau humain.

#### II.1.1 Neurone naturel

-En biologie les neurones sont des cellules excitables qui sont interconnectés entre eux, leur rôle est la transmission d'information via des signaux électriques. Chaque neurone est composé de plusieurs dendrites, d'un soma et d'un axone, les dendrites constituent la principale structure réceptrice des neurones (5),

Les signaux électriques qui reçoivent le neurone passent par les synapses, ces signaux peuvent prendre deux formes ils sont soit Inhibiteur ou Excitateur, l'activation du neurone se fait lorsque la somme des signaux atteint un certain seuil, quand le neurone s'active ce dernier produit un signal électrique qui est véhiculé à travers l'axone, enfin le signal sera transmis à d'autres neurones qui reproduiront la même opération.

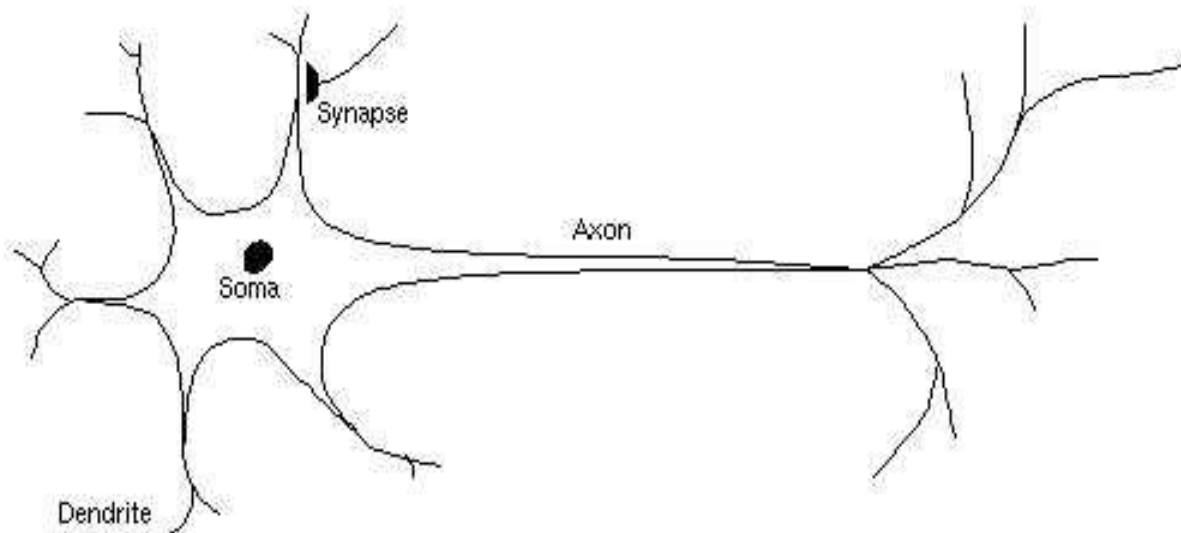


Figure 15 : neurone naturel (5)

#### II.1.2 Première tentative de modélisation d'un neurone

-L'être humain c'est toujours inspiré de la nature pour ses inventions, de très nombreux exemples pourraient être cités le plus évident serait celui de l'avion, les frères Wright ont étudié les mouvements des ailes d'un oiseau et ils ont en conclu que la forme et la structure des ailes jouent un rôle important pour l'envol de l'oiseau et sa stabilité, Pour le Deep Learning et l'intelligence artificielle, en générale c'est pareil, en 1943 Warren McCulloch et Walter Pitts

(6) ont publié un article dans lesquelles ils ont pu mettre au points un algorithme de neurone artificielle en s'inspirant des neurones organiques W. McCulloch et W.Pitts ont tenté de faire une modélisation du neurone, le neurone sera représenté par une fonction qui a pour entrées des signaux  $x$  et elle retourne une sortie  $y$ , La fonction se définit de la façon suivante : la fonction a un nombre  $n$  ( $x_1, \dots, x_n$ ) entrées et possède une seule sortie  $o$ , les poids ( $w_1, \dots, w_n$ ) et un biais  $\theta$  (7)

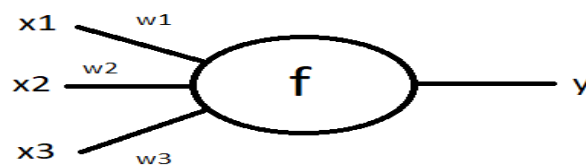


Figure 16 : neurone artificiel

$$o = f(z) = \begin{cases} 1 & \text{si } \sum_{i=1}^n w_i x_i \\ 0 & \text{sinon} \end{cases} \quad 1$$

### II.1.3 La règle de Hebb

-Les neurones artificielle que nous connaissons aujourd'hui sont capable d'auto apprentissage ce qui n'est pas le cas du neurone artificiel évoqué précédemment, cette capacité d'apprentissage repose sur la **théorie de Hebb** qui a été démontré en 1949 par Donald Hebb (8), la théorie part du principe suivant « des neurones qui s'excitent ensemble se lient entre eux. » (9), lorsque deux neurones s'excitent entre eux cela renforce leurs **plasticité synaptique**, cela expliquera comment les neurones biologiques arrivent à apprendre de nouvelles choses, Pour ce qui est d'intelligence artificiel la règle de Hebb est utilisé pour savoir comment ajuster les coefficients synaptiques entre les réseaux de neurone artificiel, si deux neurones du réseau s'activent simultanément alors leurs coefficients synaptiques synaptique augmentent et ils baissent lorsque les neurones s'activent séparément. Cette théorie tente d'expliquer comment le cerveau humain arrive à établir sa mémoire à travers l'apprentissage par l'association (10)

## II.1.4 Perceptron

### II.1.4.1 Historique sur le perceptron

-En 1957 Frank Rosenblatt développe un neurone artificiel qui a la capacité d'apprendre et il lui donne le nom de **Perceptron**, le principe de fonctionnement du perceptron est assez similaire à celui du neurone artificiel vue précédemment, le perceptron est capable de reproduire certaine fonctionnalité du neurone naturel (11), le perceptron s'active lorsque la somme de ses entrées atteint un certain seuil, mais l'avantage du perceptron est son algorithme d'apprentissage, Thorsten Joachims par sa citation estime que la base de l'intelligence artificiel a été avec le perceptron "The foundations for all of this artificial intelligence were laid at Cornell." (12), l'apprentissage du perceptron consiste à passer des données ( $y$ ) à la fonction  $f$ , les poids ( $w_1 \dots, w_n$ ) devront s'auto-ajuster pour que la sortie  $o$  soit la plus correct possible, ceci vise à reproduire le théorème de Hebb afin d'obtenir le même résultats que sure un neurone naturel c'est-à-dire l'apprentissage par l'association.

-Le perceptron est l'unité de base du réseau de neurone artificiel, c'est un classificateur linéaire capable de tracer une ligne entre deux classes qui sont linéairement séparable, la séparation se fait à l'aide d'une droite qui s'appelle frontière de décision, cette droite devra s'auto s'ajuster au cours des itérations pour obtenir la meilleure classification possible, l'équation de la frontière de décision est déductible à partir du modèle linéaire suivant :

$$z(x_1, x_2) = w_1x_1 + w_2x_2 + b \quad 2$$

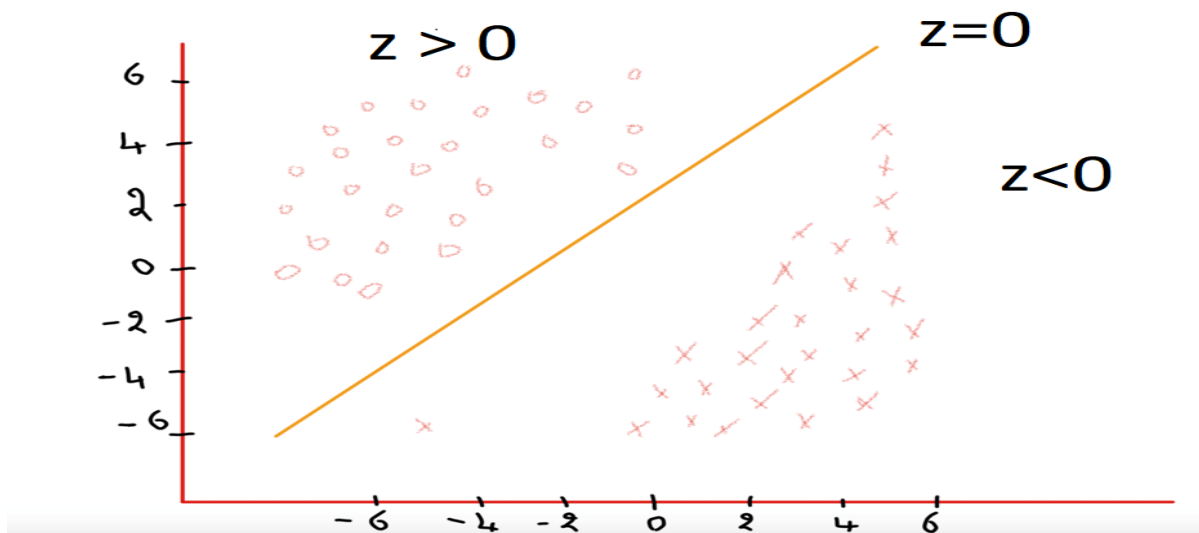


Figure 17: nuage de point

- $x_1$  et  $x_2$  sont les variables,  $w_1$   $w_2$  sont les poids et  $b$  est un biais, l'équation de la droite équivaux à la formule suivante :

$$Z(x_1, x_2) = 0$$

3

-à partir de là il est possible de faire une prédiction sur une nouvelle entrée à partir des variables  $x_1$  et  $x_2$ , la classification linéaire est assez utilisé dans la vision par ordinateur, la figure suivante contient un exemple assez utilisé pour le tester le perceptron, à partir d'une image le réseau de neurone doit être capable de déterminer si l'animal qui y figure est un chat ou un chien.

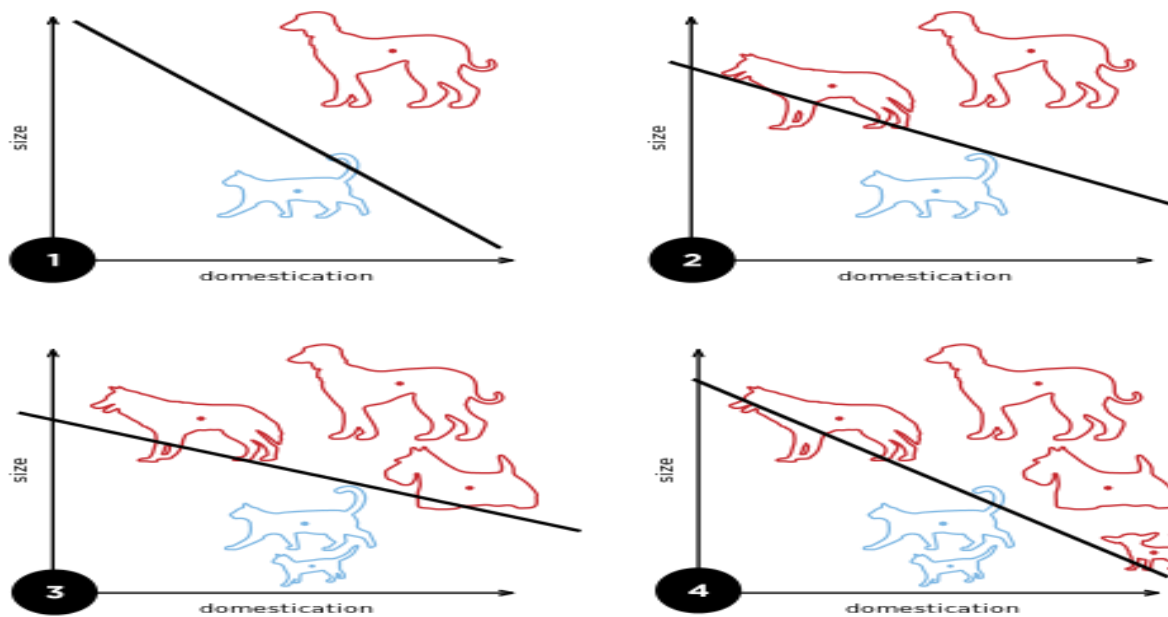


Figure 18: prédiction de l'animal qui figure sur la photo

#### II.1.4.2 Probabilité de la prédiction

-Il est possible de déterminer la probabilité d'une prédiction à partir de la fonction Sigmoidale, cette fonction à l'expression suivante (13):

$$A(z) = \frac{1}{1 + e^{-z}} \quad 4$$



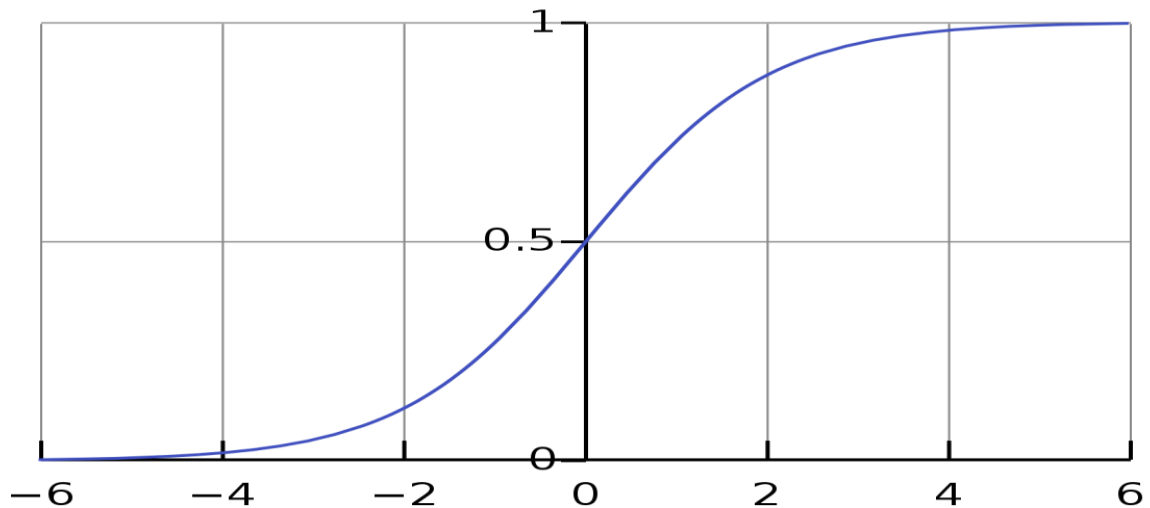


Figure 19 : graphe de la fonction Sigmoide

-Cette fonction retourne la probabilité d'un individu à appartenir à la classe 1, maintenant grâce à la loi de Bernoulli il est possible déduire la probabilité pour qu'un individu appartient à l'autre classe :

Tel que  $y$  un individu de  $Y$  :

$$Y \ p(Y = y) = A(z)^y * (1 - A(z))^{1-y} \quad 5$$

$$p(Y = 0) = (1 - a(z)) \quad 6$$

$$p(y = 1) = a(z) \quad 7$$

#### II.1.4.3 Evaluations des erreurs

-L'optimisation de l'auto apprentissage requiert que le modèle doit être capable de détecter ses propres erreurs, il est donc nécessaire d'établir une expression qui est capable de quantifier les erreurs d'un modèle au cours de la phase d'apprentissage, cette expression est appelé **fonction coût** (14) se traduit par la formule mathématique suivante :

$$L = \frac{-1}{m} \sum_{i=1}^m y_i \log a_i + (1 - y_i) + \log(1 - a_i)$$

8

Où m est le nombre de données, le  $y_i$  est donnée sur l'itération i,  $a_i$  est la sortie sur l'itération i

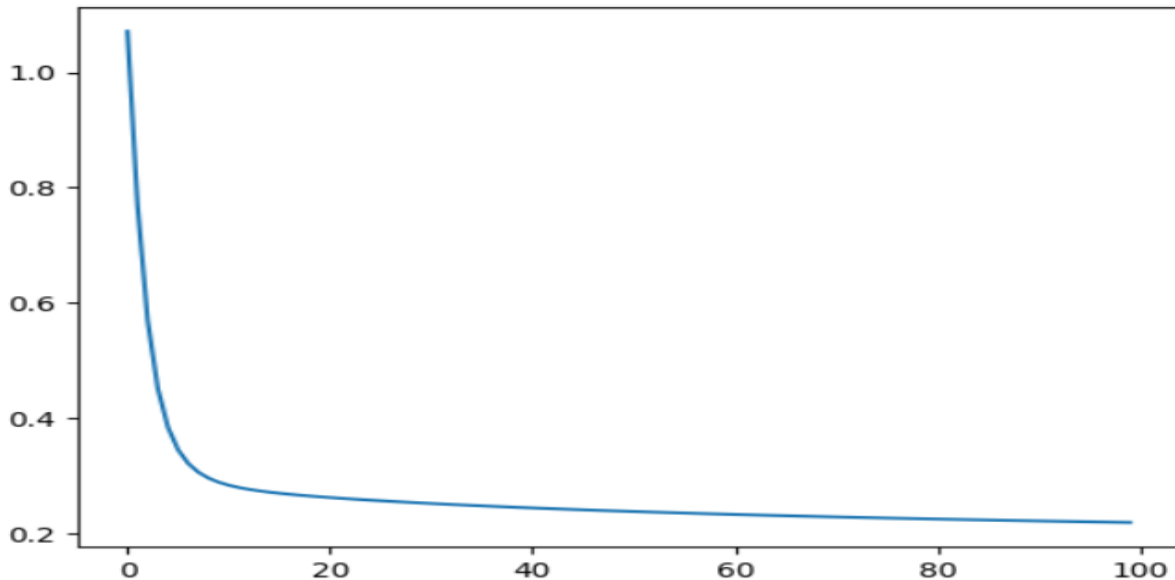


Figure 20: fonction de coût

#### II.1.4.4 Optimisation de la fonction coût

L'objectif avec la fonction coût vas être de faire une minimisation de cette dernière en fonctions des différents paramètres W, pour se faire il faudra calculer le dérivé de la fonction coût si  $\frac{\partial L}{\partial W} > 0$  cela signifie que la fonction coût augmente quand W augmente, alors il faudra diminué W pour que la fonction puisse diminuer, si  $\frac{\delta L}{\delta W} < 0$  la fonction coût descend alors il faudra augmenter W pour que la fonction continue de diminuer, la formule suivante permet de faire cela :

$$w_{t+1} = w_t - \alpha \frac{\delta L}{\delta w_t} \quad 9$$

Tel que :

$w_{t+1}$ : W quant à l'instant  $t = t+1$

$w_t$  : w à l'instant t

$\alpha$  : apprentissage positif

### II.1.5 Réseau de perceptron multi couche

-Le perceptron est un classificateur binaire, c'est-à-dire qu'il n'est pas capable d'auto apprentissage sur des patterns non séparables linéairement (15), la spécification linéaire du modèle empêche celui-ci de d'appréhender toute la complexité du problème.

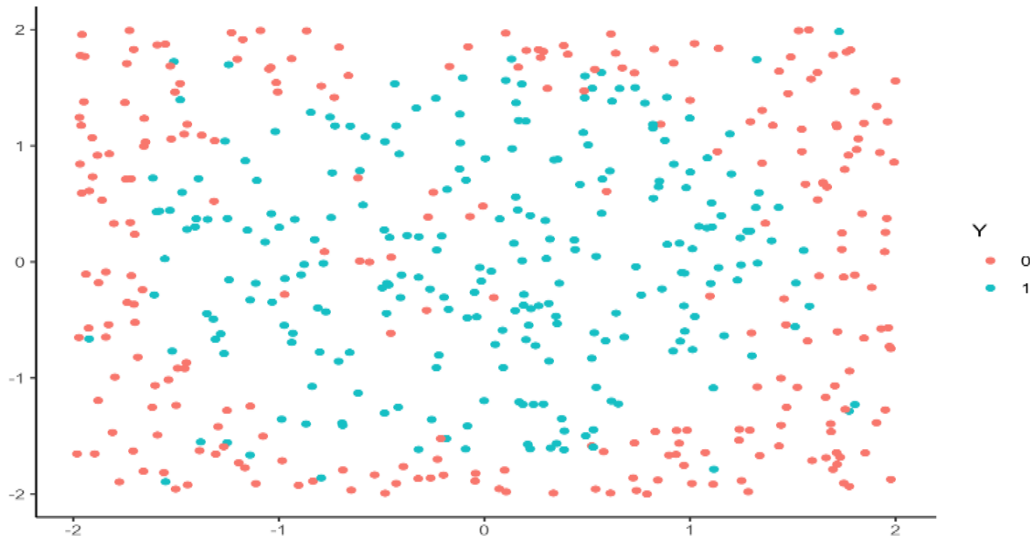


Figure 21: nuage de points non linéairement séparables

#### II.1.5.1 Architecture du MLP

-Pour un réseau de ce type de problème, il faut utiliser un réseau de perceptron contenant plusieurs couches (MLP) au lieu d'un seul perceptron. Plus on rajoute des couches plus le réseau sera puissant mais il nécessite plus de temps à l'apprentissage (16). La première sera appelée couche d'entrée c'est là où les données sont injectées dans le réseau. Ensuite viendront les couches cachées qui sont des couches intermédiaires entre la couche d'entrée et la couche de sortie où les données seront transformées et il peut y'en avoir plusieurs. Enfin, vient la couche de sortie où se trouveront les sorties finales du modèle.

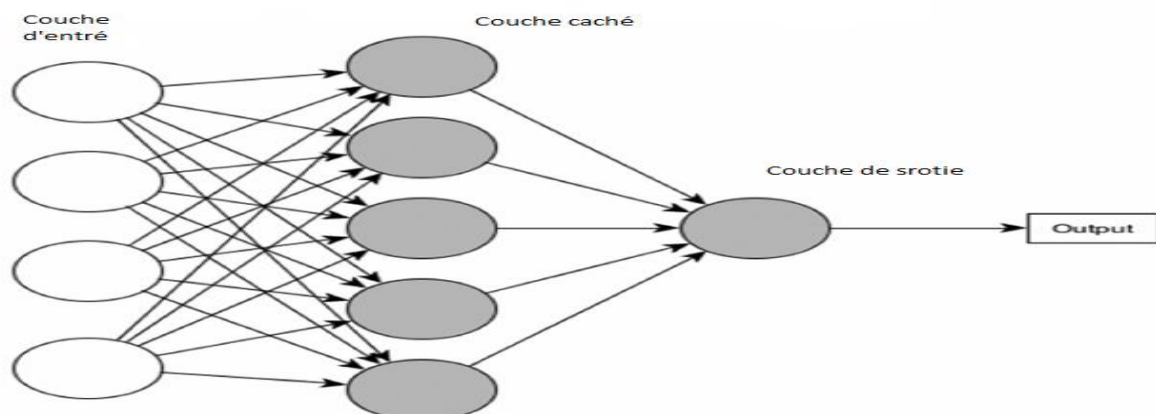


Figure 22: schéma représentant un MLP (16)

#### II.1.5.2 Forward propagation

-L'étape où les données de l'entrée passent à travers le réseau pour obtenir une sortie est appelé forward propagation, les données transitent aux couches cachées et chaque fois qu'une donnée

entre dans un perceptron elle active une fonction d'activation, la forward propagation est conçue pour que les données ne tournent pas en rond car cela empêche d'avoir des données de sortie.

### II.1.5.3 Fonction d'activation

-La fonction d'activation est appliquée à l'input du perceptron elle modifie l'entrée d'une façon non linéaire afin qu'elle puisse changer la représentation d'une donnée, par exemple 'amine jouer de la guitare le dimanche' une autre perspective de cette phrase est 'amine aime la musique'.

### II.1.5.4 Backward propagation

-La forward propagation se propage de gauche à droite, tandis que la backward propagation se propage de droite à gauche, son rôle est de modifier les poids des perceptrons afin de minimiser la fonction de coût ainsi les résultats seront optimisés.

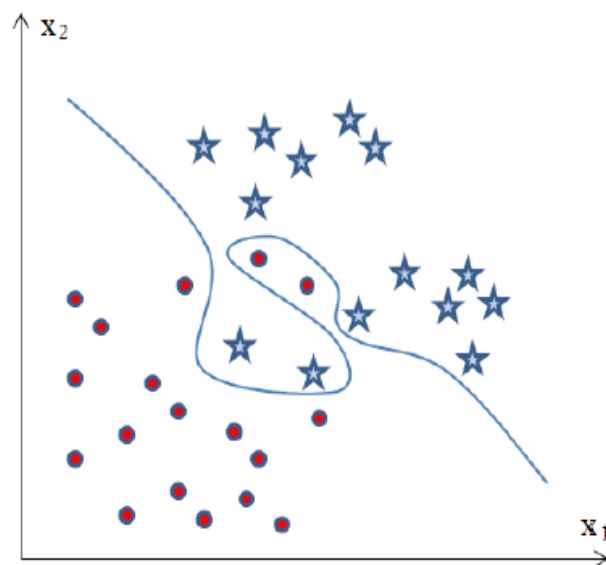


Figure 23 : classification non linéaire

## II.2 Phase de détection du texte

### II.2.1 Introduction sur la détection du texte

-La majorité des algorithmes de reconnaissance de caractères contient un module de détection du texte, il s'agit de pouvoir trouver la position du texte sur une image ou sur une vidéo, par la suite quand la région où se trouve le texte est détecté l'algorithme de reconnaissance entre en jeu, les algorithmes de détection de texte se divisent en plusieurs catégories chacun d'entre eux répond à une problématique particulière.



Figure 24 : détection d'une plaque d'immatriculation sur une voiture

### II.2.2 Les challenges que rencontrent les algorithmes de détection de texte

-L'objectif d'un algorithme de détection est de déterminer si oui ou non il y'a du texte sur une image ou sur une vidéo, et s'i il y'en a alors où se trouve-t-il ? Il n'est pas toujours facile de dire où se trouve le texte. Au début de l'OCR les algorithmes étaient seulement capables de détecter du texte sur un document imprimé (17), mais sur un document imprimé les condition de détection sont idéales car le texte est écrit sur des lignes droites et la visibilité est relativement bonne.

-Pour une image prise du monde réelle la complexité de l'environnement sur laquelle est prise l'image pose de nombreux challenges, ils sont répertoriés sur le tableau suivant :

Catégorie	Sous-catégorie
Contenu du texte	Variant de l'aspect
	Text courbé
	Variation des polices d'écriture
	Possibilité que l'environnement contient des textes en plusieurs langues
Obtention de l'image	Texte flou ou dégradé
	Distorsion de la perspective
Environnements	Complexité de l'environnement
	Éclairage inégal

Table 1 : Liste des difficulté rencontré lors de la détection (18)

### II.2.2.1 Complexité de l'environnement

-Dans les environnements urbain le texte est partout, l'algorithme pourrait prendre pour texte quelque chose qui ne l'est pas, par exemple sur les plaque d'immatriculation il est assez courant de trouver des drapeaux des symboles représentant la marque du fabricant.

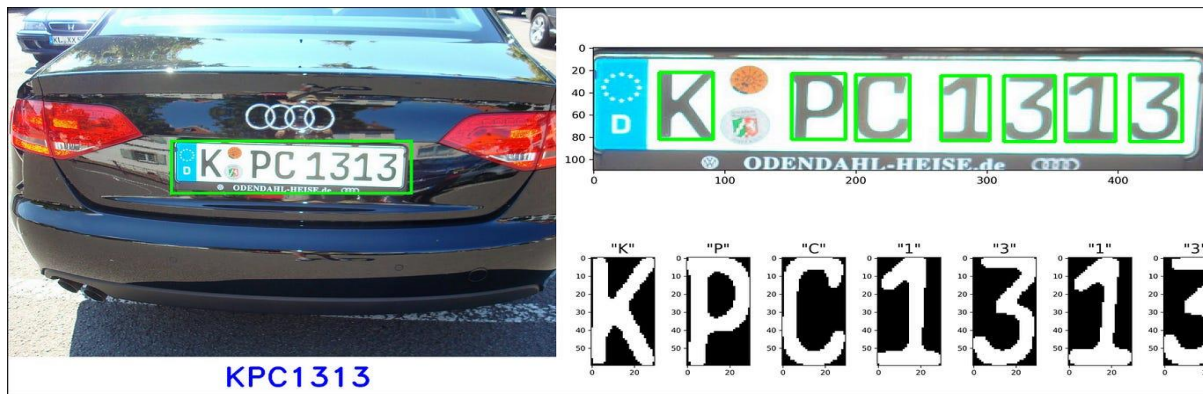


Figure 25: Algorithme de détection qui ne prend pas en compte les symboles

### II.2.2.2 Texte flou et dégradé

-Les conditions de travail des caméras sont très variable et pas toujours optimaux de plus les caméras disposent d'une fonctionnalité d'auto focus, en cas de mauvaise utilisation de celle si le texte pourra se retrouver flouté, pour ce qui en est de la dégradation les images conservent rarement leurs qualité d'origine, des algorithmes de compression peuvent dégrader la qualité de l'image pour gain d'espace ou pour un transfert plus rapide sur le réseau afin de compenser un faible débit.

### II.2.2.3 Eclairage inégale

-Dans un milieu rurale ou urbain, il se peut que la lumière ne soit pas répartie équitablement sur toute la surface du texte à détecter, alors la partie mal éclairée du texte ne pourra pas être détectée correctement ce qui implique que la reconnaissance du texte sera erronée.

### II.2.2.4 Police d'écriture

-Sur les affiches publicitaires le style d'écriture est fantaisiste, parfois des phrases sont plus garantes que d'autres, des phrases peuvent être écrite à la verticale et des caractères peuvent se chevaucher.

### II.2.2.5 Environnements contenant des textes en plusieurs langages

-La mondialisation fait que nous pourrons retrouver des affiches publicitaires en anglais à Tokyo, des noms de restaurent français a New York ou des marques de voitures japonaises à Paris, l'anglais et langages latin et des dizaines de lettres, et le japonais à peu près 50 000 caractères (19)



Figure 26 : quartier de Shibuya , Tokyo , japon

### II.2.3 Les méthodes de détection du texte

-Dans cette partie les différentes méthodes de détection de texte seront répertoriées selon leurs capacités à détecter la forme que prend le texte (horizontal, texte, sans forme précise, texte courbé) et selon la méthode qu'il utilise pour détecter le texte qu'elle soit basée sur la régression ou sur la segmentation.

	Détection de texte horizontal	Détection de texte courbé	Détection de texte sans forme précise
Détection de texte basé sur la régression	CTPN  TextBox	MOST  EAST  Textbox++	CTD  LOMO  PCR  ContourNet
Détection de texte basée sur la segmentation	PAN  MSR  DB  PSENet  SegLink++		

Tableau 1: Méthodes utilisé pour la détection de texte

II.2.3.1 Méthode de détection basé sur la régression

-Pour les méthodes de détection basées sur la régression, l'image est catégorisée selon deux critères, la partie de l'image contenant du texte et tous ce qui reste est appelé background et il peut être ignoré, pour rappel les premiers algorithmes de détection de texte étaient des algorithmes de détection d'objets qui ont subi quelque modification, par exemple le **CPTN** est un algorithme dérivé du Fast-RCNN et le **TextBoxes** du SSD algorithme.

-TextBox++ est capable de détecter un texte quel que soit son ongle de courbure, pour se faire il doit modifier le ratio de la photo pour les paramètres suivant 1, 2, 3, 5 ½, 1/3, et 1/5 cela permet à l'algorithme de mieux saisir les caractéristiques du texte.

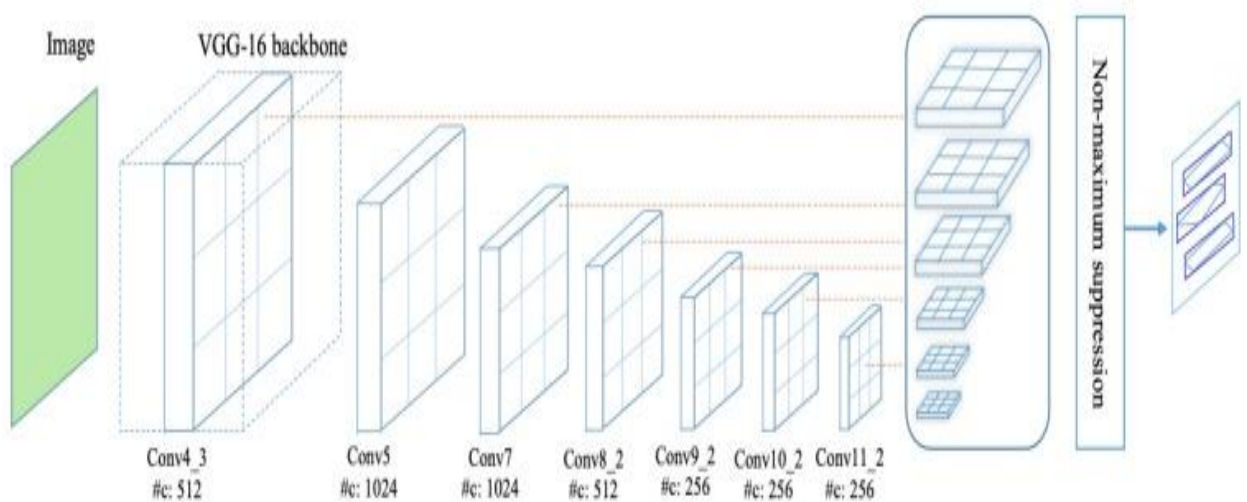


Figure 27: Diagramme représentant les frames après le changement de ratio (18)

-Pour ce qui est de la détection de texte qui n'a pas de forme précise, l'algorithme Contournet modélise un contour de points pour représenter une boîte qui contourne le texte. Après un RPN-adaptative génère une proposition pour la région du texte, après un module appelé 'Local Orthogonal Texture-aware Module' apprend les features verticale et horizontale du modèle et les représente avec des points de contours, un autre module appelé 'Re-Scoring algorithm' filtre les prédictions et assure que le texte est représenté par un groupe de points de haute qualité qui le contourne.



### II.2.3.2 Méthodes de détection basée sur la segmentation

-Les méthodes de détection de texte basées sur la segmentation sont bien plus performantes que ceux basées sur la régression. En ce qui concerne les textes courbés, leurs modèle d'apprentissage sont plus complexe et donc leurs performances en terme de vitesse d'exécution et en capacité de reconnaissance sont moindre. Les algorithmes de régression font une classification des pixels puis ils font une comparaison sur chaque pixel pour trouver une correspondance, après l'algorithme génère un arbre de probabilité sur la région de la photo où se trouve le texte, finalement une courbe entourant les phrases du texte est générée.

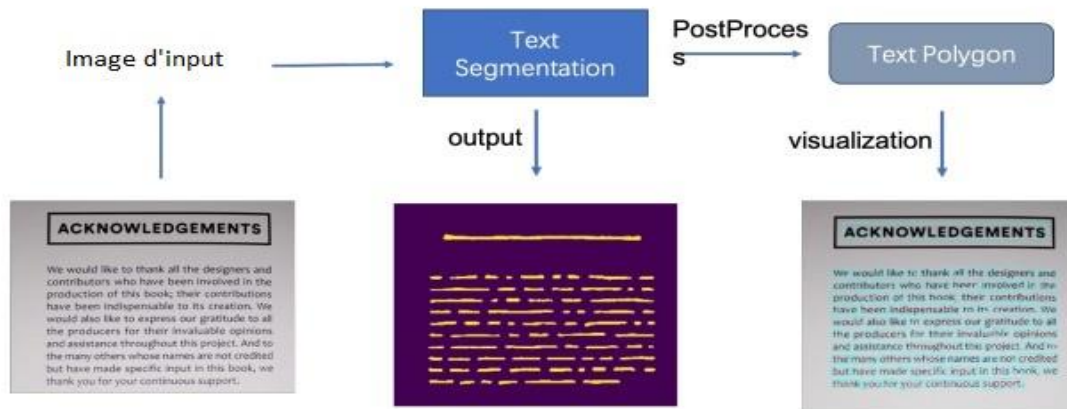


Figure 14 : schéma d'un algorithme de segmentation de texte

-Pixelink utilise la segmentation pour la détection de texte. L'objet segmenté est une région de l'image contenant du texte, les pixels d'une même phrase seront reliés entre eux pour segmenter le texte, et la boîte de contours du texte est extraite sans la régression de position. Le résultat final sera meilleur que celui basé sur la régression, toute fois le default avec cette méthode est que les régions d'image qui contient le texte peuvent se chevaucher entre eux. Pour régler ce problème, l'algorithme Yue et al propose de séparer le texte et de retenir la position de la bordure du texte pour mieux distinguer les différentes régions qui contient du texte.

## II.3 Phase de reconnaissance du texte

### II.3.1 Introduction à la reconnaissance de texte

- La deuxième phase de l'OCR est la reconnaissance de texte. Après la région où se trouve le texte a été établie par l'algorithme de détection, vient la reconnaissance du texte. Il s'agit de transformer un ensemble de pixels en une suite de caractères. Avec la transformation vient généralement une valeur de confiance de l'algorithme, selon le texte que l'algorithme aura à reconnaître on peut établir deux catégories de reconnaissance :

- Reconnaissance de texte régulier : il s'agit des textes qui sont écrit sur une ligne droite, et qui ont une police d'écriture uniforme. Se sont généralement des documents imprimés et qui ont été scannés.
- Reconnaissance de texte irrégulier : pour les textes qui ne suivent pas une ligne horizontale tel que l'exemple que nous avons évoqué sur la partie de détection de texte,

ou des textes qui peuvent subir un mauvais éclairage, des textes floues, et d'autres avec des polices d'écritures totalement différentes sur une même zone.

-Généralement les algorithmes de reconnaissance passent par deux étapes avant de pouvoir procéder à la reconnaissance. Ses étapes servent à avoir de meilleurs résultats. La première étape est le prétraitement de l'image à laquelle nous nous sommes appuyés dans le premier chapitre. La deuxième étape est la segmentation d'images qui est similaire à cela que nous avons évoquée sur la partie de détection de texte.

### II.3.2 CTC

-L'algorithmes le plus récurrent qui utilise CTC est le CRNN (20) (Convolutional Recurrent Neural). Le CRNN réalise l'extraction des « features » par d'autres algorithmes tels que MobileNet, VGG. La particularité de la reconnaissance du texte fait qu'il y a beaucoup d'informations de contexte dans les entrées, mais le module appelé CNN (convolution kernel feature of the convolutional neural networks) et il est difficile pour lui d'analyser les informations de contextes, pour résoudre ce problème le CTC utilise un module appelé LSTM (Long Short-Term Memory) qui est plus adapté pour l'extraction des informations de contextes.

### II.3.3 Sequence2sequence

-Pour ce genre d'algorithme, il y'a d'abord le codeur qui code les informations d'entrée sur des vecteurs. L'étape suivante est le décodage, pour décoder chaque sortie de l'algorithme à  $t$ , la sortie à l'instant  $t-1$  est utilisée comme entrée du décodage cette opération est répétée jusqu'à la fin des caractères d'entrée, l'encodeur est un réseau RNN (Recurrent neural network) qui retourne un vecteur et un « état caché » (hidden state) ensuite cet état caché est utilisé pour la prochaine itération pour que l'algorithme puisse établir la sémantique de la phrase, pour ce qui est de la phase de décodage, le décodeur est lui-même un autre réseau RNN, il reçoit les vecteurs du codeur et il les transforme en mots.

### II.3.4 RARE

- L'algorithme **RARE** transforme les textes irréguliers en texte régulier, pour se faire il dispose d'un module appelé STN (Spatial Transformer Network), il permet de faire la rectification des textes non réguliers en textes réguliers, par exemple si un texte est courbé alors il va le mettre sur une ligne horizontale.

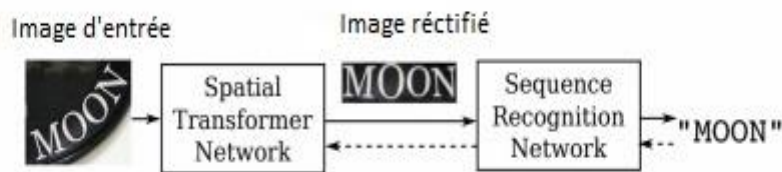


Figure 28: Transformation d'une image contenant un texte non régulier en texte régulier

-Les concepteurs de **RARE** l'ont testé sur le dataset **CUTE80** et **SVTP**, qui sont des datasets spécialisé dans les images à textes irréguliers et il ont obtenu un taux de réussite supérieur à 5% par rapport aux **CRNN** (21).

### II.3.5 Attention algorithme

-Ce genre d'algorithme recherche la corrélation entre des mots et son but est de trouvé le sens syntaxique de la phrase, pour se faire l'algorithme par du principe que certains **mots** sont plus important que d'autres pour parvenir à son but, et que pour ses **mots** là, l'algorithme devrait leurs offrir plus de poids, pour le décodage à chaque itération le contexte le plus approprié est choisi pour la prochaine itération, cette manière de faire permet d'obtenir de meilleurs résultats, R2AM est le premier algorithme à avoir introduit cette technique pour la reconnaissance optique de caractères (22).

### II.3.6 Transformers

-Les transformer ont vu le jour en 2017 après la publication de l'équipe de recherche de Google, par rapport aux CNN qui ont des limitations avec le mécanisme d'attention, pour ce qui en est des Transformers ils n'ont pas cette limitation, la façon dont sont utilisé les transformer dans un algorithme de OCR peut varier, l'architecture du transformer est composé d'encoder et de décoder qui sont monté les uns sur les autres pour que l'entré d'un encoder soit la sortie du précédent.

-NRTR utilise la structure complète d'un transformer pour faire l'encodage et le décodage de l'entrée, et il utilise un CNN pour l'extraction de features (23).

- SRACN utilise les Transformers en tant que décoder afin de remplacer les LSTM (24)

## II.4 Conclusion

-Durant ce chapitre nous avons fait l'état d'art de la reconnaissance optique de caractères, nous avons expliqué pourquoi ce domaine de l'intelligence artificielle était dominé par des modèle basé sur le Deep Learning, cependant on peut remarquer que les articles publiés sur le Deep Learning ont était publié il y'a pas mal de temps, alors que les modèles de Deep Learning ont commencé à gagner de la popularité que très récemment, quelle est l'explication a cella ?

-Comme nous l'avons précédemment expliqué dans la problématique les modèles de Deep Learning ont besoin d'une grande quantité de données (images pour l'OCR) pour la phase d'entraînement, or ces données n'étaient pas forcément disponibles avant l'apparition d'internet et des réseaux sociaux. Un autre point qui explique l'émergence des modèles de Deep Learning est l'augmentation de la puissance de calculs des ordinateurs car ils sont très demandant en puissance de calcul.

***Chapite III: Études comparatives entre les  
principaux modèles de reconnaissance optique  
de caractères***



jour il ferrera soleil, à partir de ses récurrences les RNN pourront faire des prédictions futures et grâce à ce qui est appelé « feed back loop ». Cella

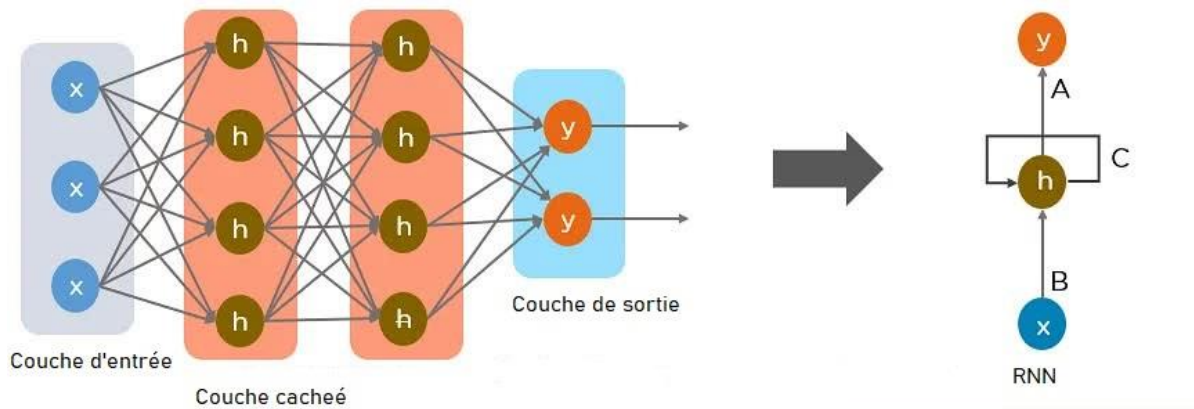


Figure 30: Réseau de types RNN (29)

Maintenant que nous avons brièvement introduit les CNN et les RNN, il nous est possible de passer aux CRNN, il est composé de 3 modules (20). Le premier est un CNN, le deuxième est un RNN, et le troisième est une couche de transcription. Le but du premier module est l'extraction des caractéristique d'une image, et le deuxième module est utilisé pour générer la séquence de caractères détectés. Le dernier module prédit la valeur de sortie à partir des données retournées par les couches précédentes.

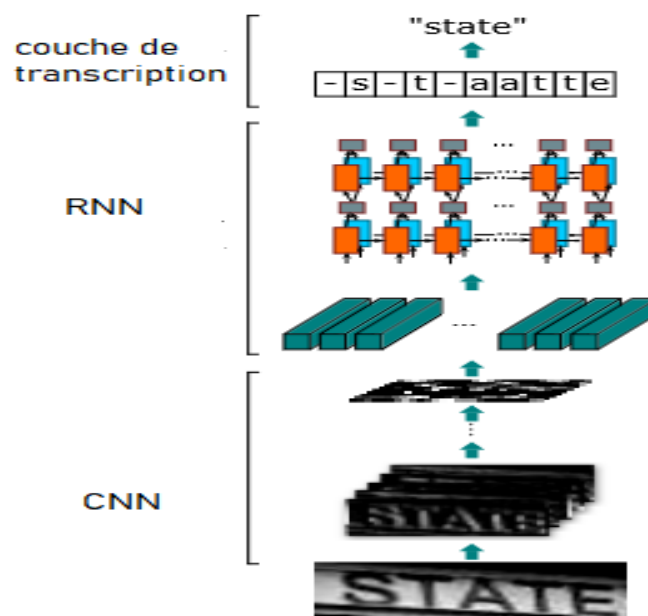


Figure 31: architecture d'un CRNN (29)

### III.1.2 LSTM (Pytesseract)

Pytesseract est une implémentation de tesseract sur python, Tesseract a été développée en 1994 par une équipe de chercheurs des laboratoires « Hewlett-Packard », Tesseract est portable

sur une multitude de système d'exploitation (30). En 2006, il a été racheté par Google sous format open source (31) et dans les années qui suivent, il était considéré comme l'algorithme d'OCR le plus performant, il est capable de détecter plus de 116 langues (32).

Le modèle de reconnaissance de Tesseract est un réseau récurrent à mémoire court et long en anglais « **Long short-term memory** » (LSTM). Les LSTM sont la solution proposée pour le problème de disparition de gradient des RNN (33), l'architecture d'un LSTM se présente ainsi :

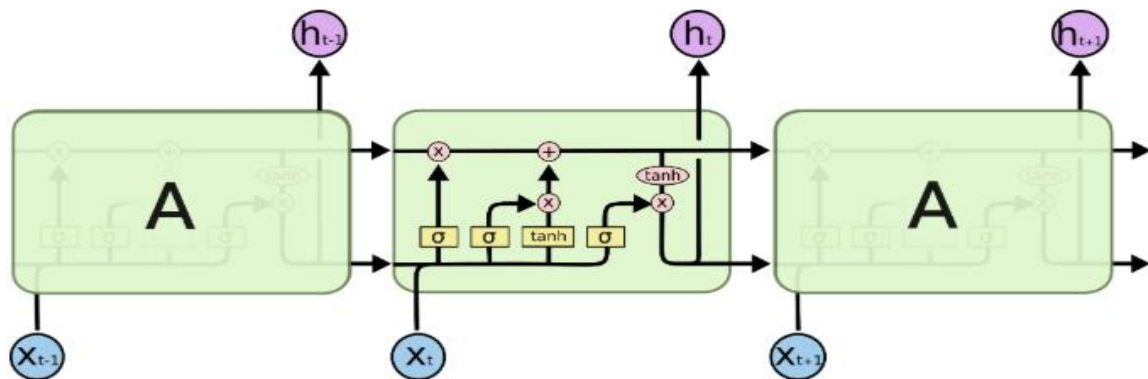


Figure 32 : architecture d'un LSTM

- **Forget Gate** : C'est le composant qui permet de réguler le flux de données, il décide quelle donnée ne sont plus pertinente et devons être oublier.
- **Input Gate** : Ce composant permet lui aussi de réguler le flux de données en déterminant si de nouvelle information devons être ajouté.
- **Output Gate** : la cellule vient de déterminer quelle était les informations qu'elle devait oublier, et elle a déterminer quelle sont les informations qu'elle devait ajouter, il ne reste plus qu'à générer les informations de sortie.

### III.1.3 Transformers(TrOCR)

En 2017 le monde du Deep Learning a fait une grande avancée de par la publication par une équipe de Google d'un nouveau modèle de Deep Learning appelé **Transformers** (34). Dans cette publication les auteurs ont démontré que leurs modèle était plus performant pour faire la traductions sur le data sets WMT2014, jusque-là pour ce qui concernait le **traitement automatique de la langue naturelle (NLP)** ont utilisé des CNN et des RNN (35). Depuis la publication de cette article, les transformers commencent à impacter d'autres domaines tel que le détection d'objets notamment avec DETR (36). Pour ce qui nous concerne TrOCR est un modèle pour la reconnaissance optique de caractère basé sur un modèle de Transformers (37), La traduction est un excellent exercice pour tester les modèles de **NLP** sur la figure suivante figure un exemple de traduction de phrase du français vers l'anglais.



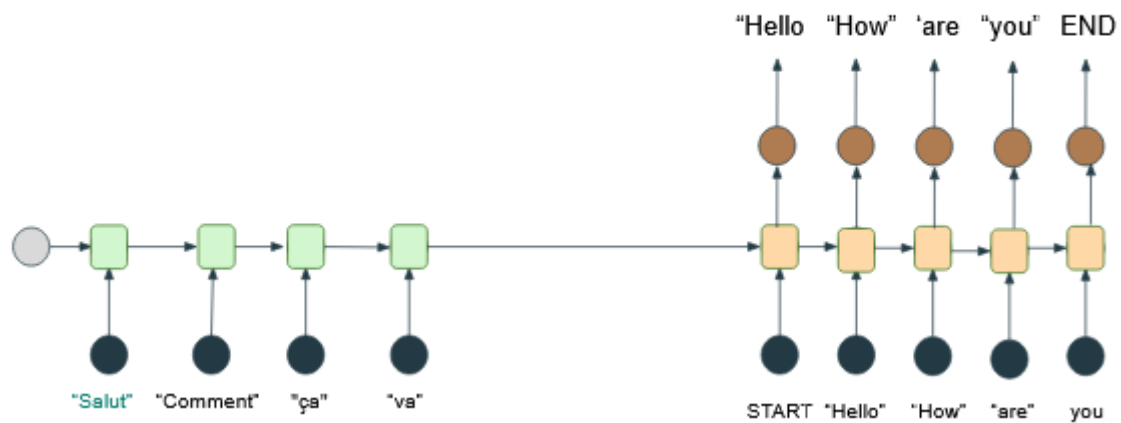


Figure 33 : traduction d'une phrase dans une architecture Encoder-Décodeur (38)

L'encoder qui est donné à chaque mot une représentation vectorielle qu'on appelle token, en plus des tokens qui représentent les mots de la phrase il y'en a d'autres tel que « START » et « end » qui eux mentionnent le début et la fin d'une phrase, ensuite le décodeur fera une traduction de chaque, la sortie de chaque cellule est l'entrée de la suivante, cela permettra au décodeur à avoir le contexte d'une phrase.

Quand est venu le temps de publier leurs travaux sur les Transformers, l'équipe de chercheurs a choisi comme nom de leur article « Attention is all you need » (34). Les Transformers utilisent le mécanisme d'attention pour faire leurs prédictions, c'est-à-dire que le décodeur ne se contente pas seulement des informations fournis par l'encodeur car des pertes d'information peuvent se produire à cause du phénomène de disparition des gradients, il regarde en arrière vers chaque token.

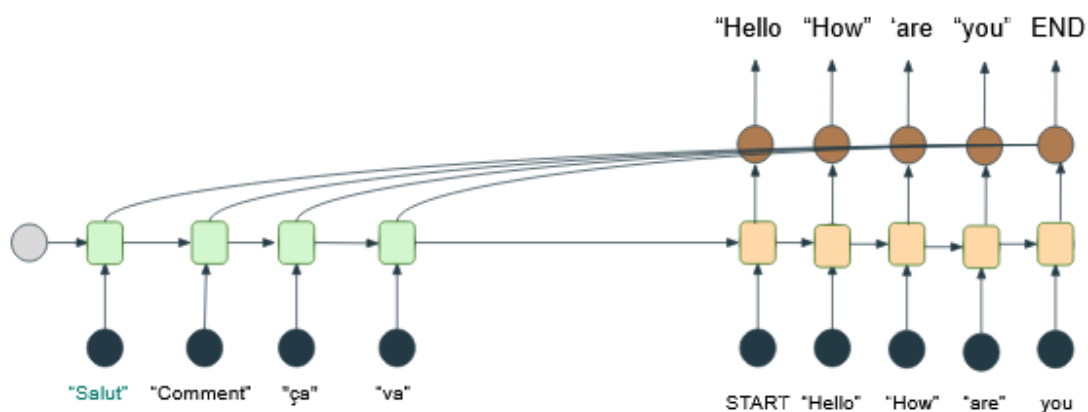


Figure 34: traduction d'une phrase dans une architecture Encoder-Decoder en utilisant le mécanisme d'attention

### III.1.4 Reciprocal Feature Learning(PadleOCR)

PadleOcr supporte plus de 80 langues et peut être déployé sur des serveurs et des appareils embarqués et même sur des objets connectés (39). Le module de reconnaissance de Padle est basé un modèle appelé « Reciprocal Feature Learning ».

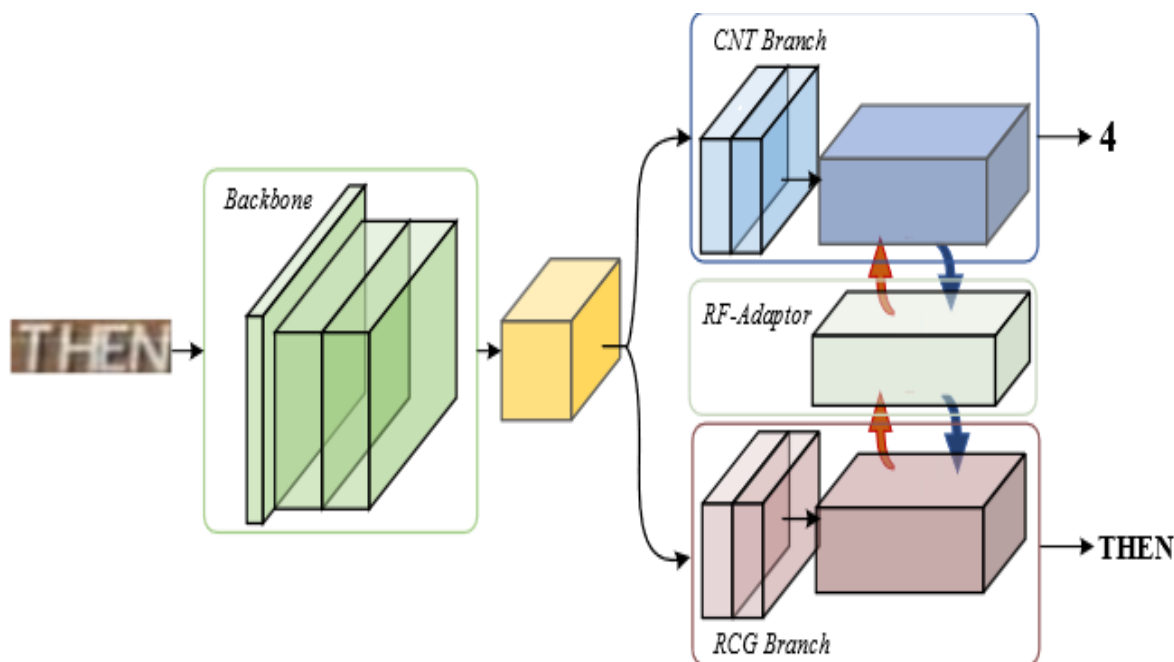


Figure 35: architecture du model de reconnaissance de paddle (40)

La première partie appelée « **Backbone** » a pour rôle l'extraction de « features ». Les concepteurs du modèle ont choisi d'utiliser un réseau **Resnet**. Resnet est un diminutive de « **Residual Network** » (41) qui est un type de réseau qu'on appelle « **Very Deep Neural Network** ». Ce sont des réseaux de Deep Learning qui permettent de mettre en place un très grand nombre de couches, mais pour rappel plus le nombre de couche est grand plus grand sera le problème de disparition des gradients. Pour parer à ce problème le réseau utilise des « **Residual block** », il permet à l'information de circuler directement depuis les couches inférieures vers les couches extérieures (42), la partie appelé « CNT Branch » compte le nombre de caractères reconnu, et enfin la partie « **RCG** » qui est responsable de la reconnaissance des caractères est un réseau de type RNN.

### III.2 Présentation de l'algorithme de votes

Pour tenter de répondre à la problématique que nous avons évoqué, on a conçu un algorithme qui prend en entrée les résultats de chaque modèle que nous avons présenté dans la partie précédente, et à partir de ces entrées il retourne un résultat finale issue d'un vote.

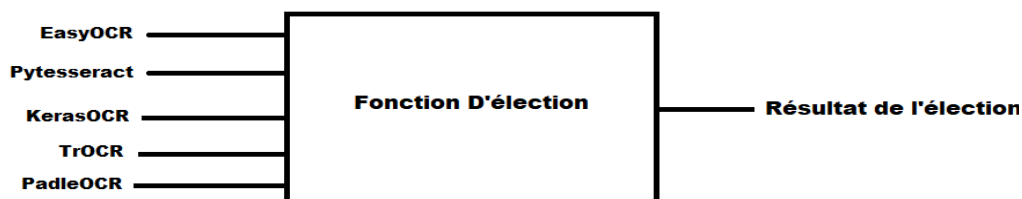


Figure 36: schéma m'entrant les entrée et les sortie de la fonction d'élection

Maintenant nous allons faire une démonstration de l'algorithme de votes, pour se faire nous allons prendre, comme exemple, la plaque d'immatriculation suivante :



Figure 37 : Plaque d'immatriculation

Les prédictions des modèles sur cette image sont répertoriées sur la figure suivante :

```

-----Easyocr-----
1705911442
Taille de la chaine : 10
-----pytess-----

Taille de la chaine: 0
-----keras-----
1705911442
Taille de la chaine: 10
-----padle-----
1705911442
Taille de la chaine: 10
-----TR-----
1705911442
Taille de la chaine: 10
-----
  
```

Figure 38: résultats de la prédictions avec leurs tailles respective

Les modèles qui n'arrivèrent pas à détecter le texte qui figurent sur l'image seront ignorée (Taille de la chaine 0). Par la suite, l'algorithme fait un nettoyage des caractères détectés et supprime les caractères indésirables tel que @ \*\$%, Ensuite, tous les caractères restant seront transformés en majuscule. Une fois ses étapes préliminaires terminées, l'algorithme crée un tableau contenant tous les caractères détectés par les 5 modèles mais sans répétition. Pour notre exemple, le tableau sera le suivant : ['1', '7', '0', '5', '9', '4', '2']. Ensuite, un vote d'existence se fera sur chaque élément du tableau. Par exemple, pour le caractères « 1 » il sera détecté au

positions [0,5,6] de la chaîne 1. Les autres chaînes devront voter pour le caractère 1 aux positions évoqués précédemment. Si le caractère 1 existe dans les autres chaînes à la position 0 alors ils devront voter positivement selon leurs coefficients respectifs. Sinon, ils devront voter négativement, donc pour le caractère « 1 » aux positions « 0,1,5 » le résultat sera le dictionnaire suivant {0 :13,5 :13,6 :13}. Cette structure de données est appelée un dictionnaire, pour chaque clé il y'a une valeur paire. Au final, lorsque toutes les valeurs du tableau d'occurrence seront testées et votées sur toutes les positions, l'algorithme fera un autre dictionnaire qui contient chaque caractère sur ses position réciproque {'1' : [0, 5, 6], '7' : [1], '0' : [2], '5' : [3], '9' : [4], '4' : [7, 8], '2' : [9]}, et à partir de ce dictionnaire l'algorithme fera la reconstruction de la plaque d'immatriculation

-En plus de proposer un système de votes performant notre application est capable de :

- Détecte si une plaque d'immatriculation est algérienne ou si elle est de l'union européenne, si la plaque est détectée comme étant algérienne alors il retournera quelque information contenue sur la figure suivante :

```
Le résultat du vote ['1', '7', '0', '5', '9', '1', '1', '4', '4', '2']
2014
Véhicule privée
{'year': '2014', 'pays': 'algerie', 'wilaya': 'Tipaza', 'vehiculeType': 'Véhicule privée'}
```

Figure 39: Valeurs finale retourné par l'algorithme de votes ainsi que des informations relatives à la plaque détectés

- D'enregistré les résultats de chaque model et de l'algorithme de votes sur un seul fichier.
- De détecter si une image est floue, si elle l'est l'application retourne un message indiquant que les résultats ne seront pas optimaux.
- De faire un nettoyage des résultats retourné par les model, ce nettoyage aura pour but de supprimer les caractères jugés indésirables, qu'il est impossible de trouvé sur une plaque d'immatriculation.

### III.3 Études comparatives

-Dans cette partie nous allons dans un premier temps présenté la méthodologie utilisé pour mesuré la performance des modèles de chaque algorithme, ensuite nous allons présenter les résultats obtenus et les analyser.

Pour mesurer les performances de nos modèles, il est important de choisir une méthodologie rigoureuse. Pour se faire, nous avons une formule spécialement conçu pour mesurer les performances des modèle OCR, cette formule ce nome « **Character Error Rate** »(CER) (43)

$$CER = \frac{S + D + I}{N}$$

- S= Nombre de substitutions
- D= Nombre suppression (Deletions)
- I= Nombre d'insertion
- N= Nombre de caractère réel

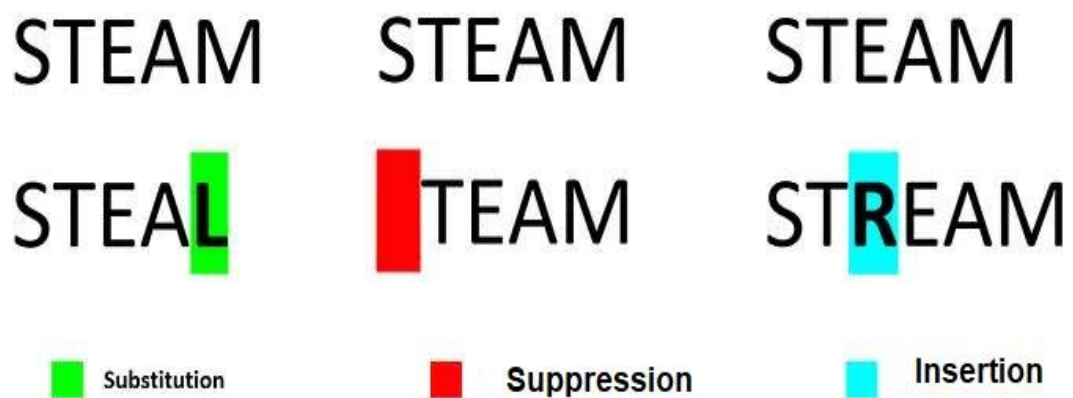


Figure 40: Explication de N,S,I,D









-Par la suite selon les résultats qu'obtiendront les modèles, il sera possible de les classier selon 3 catégorie (44) :

- **Excellent** : CER 1-2%, avec un taux de reconnaissance compris entre 98 et 99%.
- **Moyen** : CER 2-10%, avec un taux de reconnaissance compris entre 90 et 98%.
- **Faible** : CER >10%, avec un taux de reconnaissance inférieure à 90%.

Nous nous sommes basé sur la classification de ses modèles, pour l'attribution des coefficients à chacun d'entre eux. Pour les modèles qui auraient un CER compris entre 1 et 2 %, ils auront un coefficient de 4 et 5, les modèles dont le CER serait moyen auront un coefficient de 3, et les modèles dont le CER est faibles auront un coefficient de 1.

### III.3.1 Résultats des tests des 5 modèles sur le premier data sets

#### III.3.1.1 Tableau contenant les résultats des dix premières images

	EASY-OCR					PYTESSERACT				KERS-OCR				PADAL				TR			
	N	S	D	I	CER	S	D	I	CER	S	D	I	CER	S	D	I	CER	S	D	I	CER
	10	0	0	0	0%	0	10	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	11	0	0	0	0%	0	0	0	0%	0	1	0	10%	0	0	0	0%	0	0	0	0%
	10	0	10	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	7	1	6	0	100%	0	7	0	100%	0	1	0	10%	0	0	0	0%	0	0	0	0%
	11	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	11	1	10	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	7	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	10	0	0	1	10%	0	10	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%



	10	0	9	0	90%	0	10	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%
	11	1	10	0	100%	0	11	0	100%	0	0	0	0%	0	0	0	0%	0	0	0	0%

Tableau 2: Tableau contenant les résultats des testes du premier data sets sur les 5 modèle

III.3.1.2 Calcul du CER de chaque modèle en se basant sur les résultats du data sets 1

$$CER(EASYOCR) = \frac{58 + 197 + 13}{958} = 27\% \quad 11$$

$$CER(PYTESSERACT) = \frac{25 + 412 + 27}{985} = 48\% \quad 12$$

$$CER(KERAS) = \frac{31 + 27 + 4}{985} = 6\% \quad 13$$

$$CER(PADDEL) = \frac{2 + 0 + 1}{985} = 0.3\% \quad 14$$

$$CER(TROCR) = \frac{1 + 4 + 1}{985} = 0.6\% \quad 15$$

III.3.1.3 Histogramme représentant les taux d'insertion de substitutions et le taux de suppression pour chaque modèle

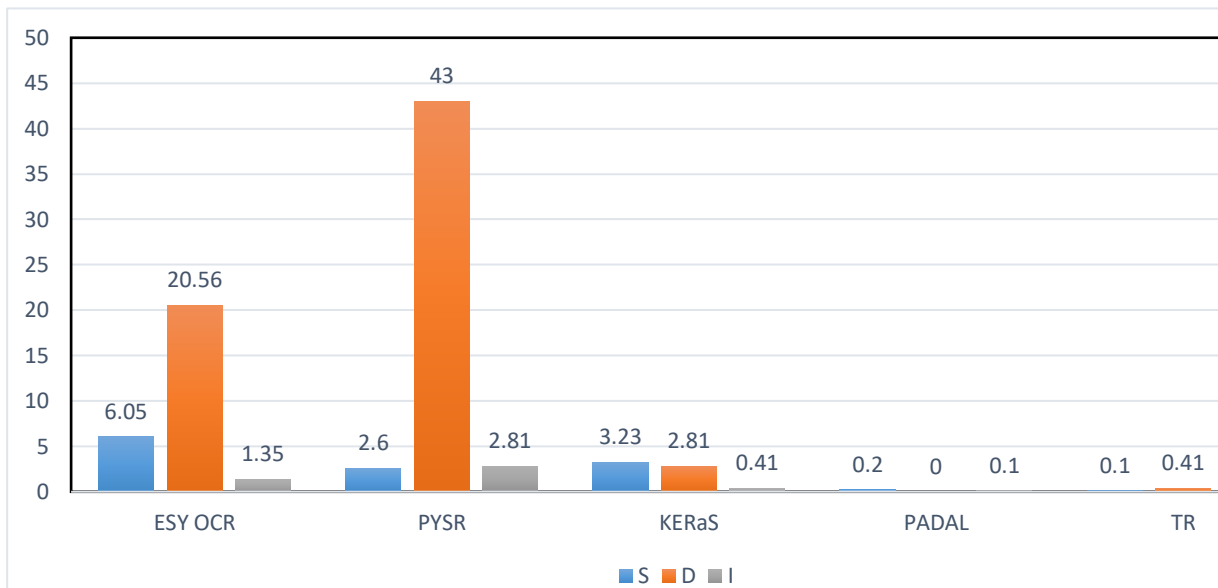


Figure 41: Histogramme des taux S,D,I sur le premier Data Sets

III.3.1.4 Graphique circulaire représentant le CER de chaque modèle

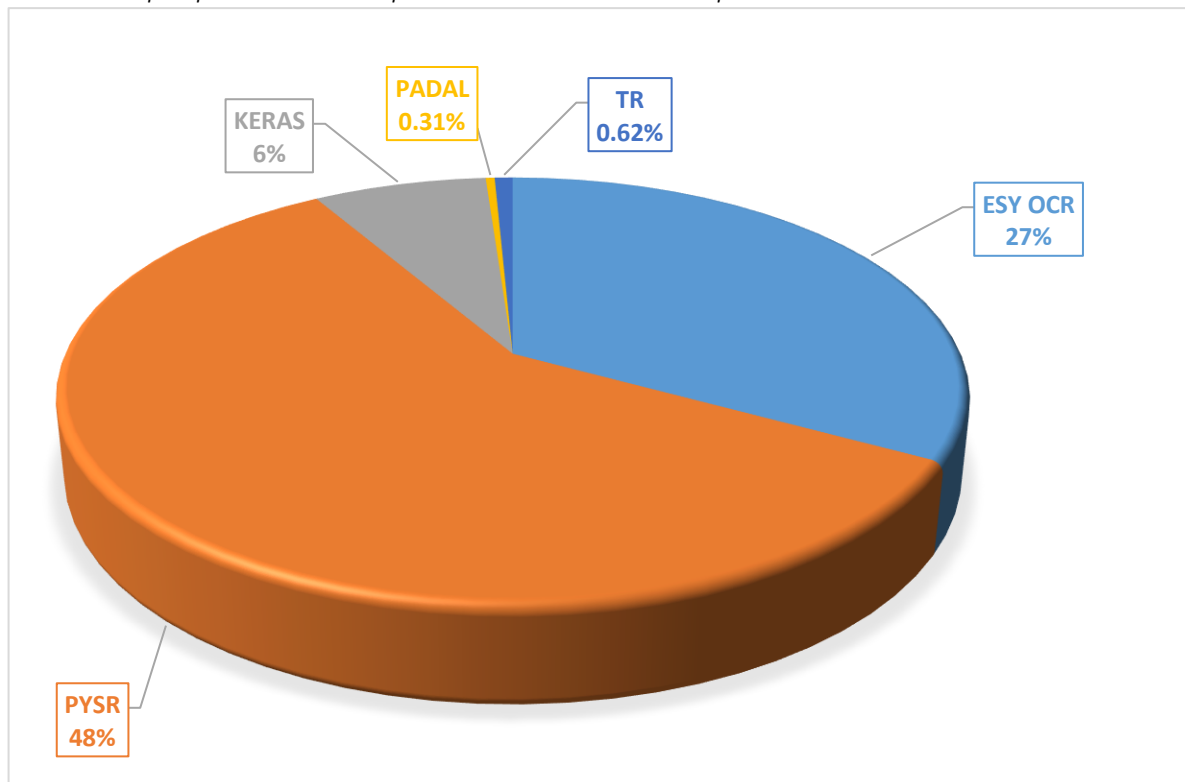



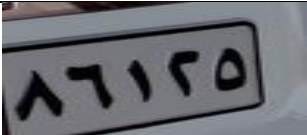




Figure 42: Graphique circulaire représentant le CER de chaque modèle sur le data sets 1



III.3.2 Résultats des tests des 5 modèles sur le deuxième data sets

III.3.2.1 Tableau contenant les résultats des dix premières images

	ESY OCR					PYSR				PADAL			
	N	S	D	I	CER	S	D	I	CER	S	D	I	CER
	5	0	0	0	0%	0	5	0	100%	1	0	0	20%
	5	1	3	0	80%	2	1	0	60%	0	5	0	100%
	5	0	0	0	100%	0	5	0	100%	0	0	0	0%
	5	0	0	0	0%	0	5	0	100%	0	5	0	100%
	4	0	0	0	0%	0	4	0	100%	0	0	2	50%
	4	0	1	0	25%	0	4	0	100%	0	0	2	50%

	6	0	0	0	100%	4	1	0	83%	0	1	0	16%
	5	0	1	0	20%	0	5	0	100%	1	3	0	66%
	4	0	0	0	0%	0	4	0	100%	0	4	0	100%
	4	0	2	0	50%	0	4	0	100%	0	2	0	50%

Tableau 3 : résultats des modèles sur le deuxième datasets

III.3.2.2 Calcul du CER de chaque modèle en se basant sur les résultats du data sets 2

$$CER(EASYOCR) = \frac{19 + 55 + 2}{274} = 27\% \quad 16$$

$$CER(PYTESSERACT) = \frac{37 + 226 + 0}{274} = 95\% \quad 17$$

$$CER(padle) = \frac{24 + 97 + 20}{274} = 51.4\% \quad 18$$

III.3.2.3 Histogramme représentant le taux d'insertion de substitutions et de suppression de chaque modèle

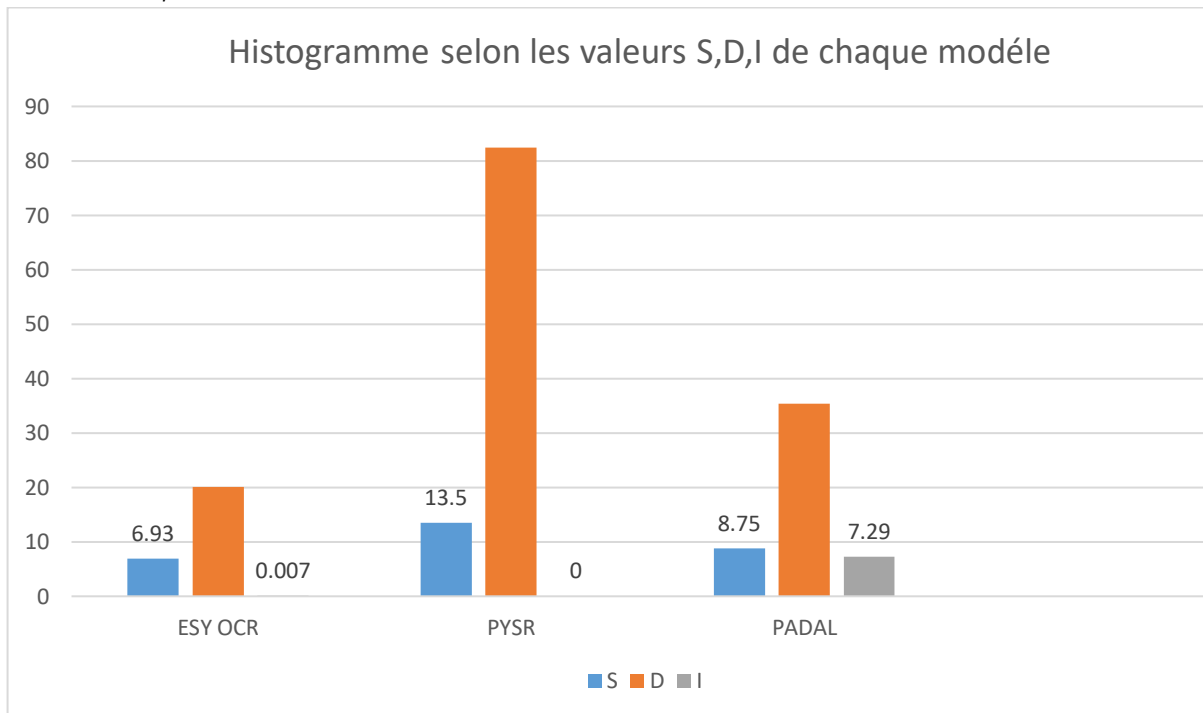


Figure 43 : Histogramme représentant les taux S,D,I de chaque modèle sur le data sets 2

III.3.2.4 Graphique circulaire représentant le CER de chaque modèle

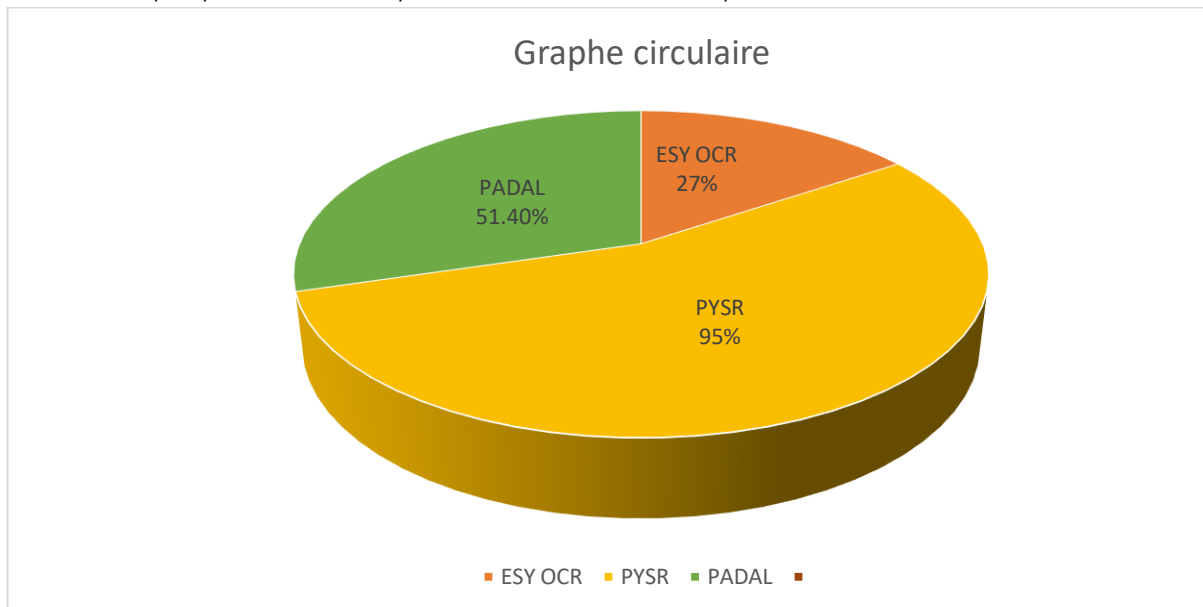


Figure 44: Graphique circulaire représentant le CER de chaque modèle sur le data sets 2

III.3.3 Résultats des tests sur l'algorithme de votes

III.3.3.1 Tableau contenant les résultats de l'algorithme de votes

Images	N	S	D	I	CER
	10	0	0	0	0%
	11	0	0	0	0%
	10	0	0	0	0%
	7	0	0	0	0%
	11	0	0	0	0%
	11	0	0	0	0%
	7	0	0	0	0%
	10	0	0	0	0%
	10	0	0	0	0%
	10	0	0	0	0%


	11	0	1	1	18%
---	----	---	---	---	-----

Tableau 4 : résultats de l'algorithme de votes sur le premier datasets

III.3.3.2 CER de l'algorithmes de votes

$$CER(Votes) = \frac{1 + 1 + 0}{985} = 0.2\%$$

19

III.4 Conclusion

Le tableau suivant fait un réduplicatif des résultats obtenus par les modèles sur les deux data sets :

		CER					
	Nombre d' image	EASY-OCR	Pytesseract	KERAS-OCR	PADDAL	TR-OCR	Algorithme de votes
datas set 1	100	27%	48%	6%	0,31%	0,62%	<b>0.2%</b>
datas set 2	60	27%	95%		51,04%		

Tableau 5: Récapitulatif des résultats

A partir des résultats obtenus, il est possible de catégoriser les modèles et l'algorithme selon leurs résultats, sur le tableau suivant les algorithmes sont catégorisés selon leurs score CER :

Catégorie du modèle	Nom du modèle	CER obtenu
Excellent(0-2%)	Algorithme de votes	0.20%
	Paddal	0.31%
	TR-OCR	0.62%
Moyen (2-10%)	KERAS-OCR	6%
Faible (CER>10%)	EASY-OCR	27%
	Pytesseract	48%

Tableau 6: catégorie selon la performance

Celui qui a réussi à obtenir le taux le plus faible est l'algorithme de votes que nous avons conçu avec un CER de 0.20% ce qui 0.1% de moins de Paddal qui lui a obtenu un taux de 0.31%, donc cela prouve que notre système de vote est performant, et qu'il est possible d'obtenir de meilleurs résultats à partir de plusieurs modèles sans pour autant modifier leurs architectures, hyper paramètre, ou bien de refaire leurs apprentissages.

Toutefois nous sommes conscient que le nombre d'image que contient notre data sets (160 images) peut être considéré comme insuffisant. Mais, dans son contenu il reste assez varié et les images dont il est composé sont tiré de situations réel ou les conditions de la reconnaissance de caractères n'est pas toujours optimales.

## *Conclusion générale et perspectives*

Dans ce mémoire, notre travail s'est porté sur la déduction d'un résultat obtenu par la combinaison des résultats de 5 modèles de Deep Learning pour la reconnaissance de plaque d'immatriculation.

Pour essayer d'améliorer un modèle, il faut d'abord mesurer ses performances. Pour se faire, nous avons présenté dans le chapitre 3 la méthodologie qu'on a utilisée pour l'évaluation des performances. Nous avons conçu un système de votes où chaque modèle vote pour ce qu'il a détecté, à chaque modèle est associé un coefficient de vote. Le choix de ces coefficients est fait selon la performance individuel de chaque modèle. Il est possible de catégoriser les performances d'un modèle en 3 catégorie : si le taux d'erreur est entre 1 et 2% le modèle est jugé comme étant excellent, entre 2 et 10 % il est moyen, et si le taux est supérieur à 10% sa performance est jugée comme étant faible.

Les résultats de ce modeste travail ont permis d'établir qu'un système de vote pouvait améliorer les performances de modèles de reconnaissance optique de caractères, le système de vote a obtenu un taux d'erreur de seulement 0.2% ce qui est 0.1% de moins que le modèle Paddal et 0.4% de moins de TrOcr, toutefois cela ne constituent que les bases d'un travail future qui devra être plus approfondi.

Les perspectives futures seraient, en premier lieu, l'élargissement du data sets du test. Pour évaluer les performances de nos modèles et du système de votes, nous avons utilisé un data sets de 160 images seulement, ce qui peut être considéré comme peu. Une autre perspective est l'amélioration du système d'identification, actuellement l'application est capable de reconnaître les plaques algériennes, celle de l'union européenne, et celle des pays du golfe arabe, on pourrait élargir cette identification à d'autres pays. Aussi, notre application traite seulement les images, une autre amélioration future consisterait à ajouter une fonctionnalité permettant de traiter les flux vidéo.



## *Bibliographie*

1. Tutoriel sur le traitement d'images. [En ligne] 12 juin 2019. [<https://www6.inrae.fr/pfl-cepia/Axe-images/Tutoriel>].
2. Djema Chahrazed, Boulahia Siham. «*Local features vs global features pour la reconnaissance des plaques d'immatriculations*», *Thèse de master*; Tizi-Ouzou : Université mouloud Mammeri d, tizi-Ouzou., 2016-2017.
3. *Imagenet large scale visual recognition challenge*. Russakovsky, Olga, et al. s.l. : Springer, 2015, *International journal of computer vision*, Vol. 115, pp. 211–252.
4. *Imagenet classification with deep convolutional neural networks*. Krizhevsky, Alex, Sutskever, Ilya et Hinton, Geoffrey E. s.l. : AcM New York, NY, USA, 2017, *Communications of the ACM*, Vol. 60, pp. 84–90.
5. Marieb, Elaine et Hoehn, Katja. *Anatomie et physiologie humaines: Livre+ eText+ plateforme numérique MonLab-Licence étudiant 60 mois*. s.l. : Pearson Education France, 2014.
6. *The First computational theory of mind and brain: a close look at mcculloch and pitts's "logical calculus of ideas immanent in nervous activity"*. Piccinini, Gualtiero. s.l. : Springer, 2004, *Synthese*, Vol. 141, pp. 175–215.
7. *Perceptron learning in engineering design*. Adeli, Hojjat et Yeh, C. s.l. : Wiley Online Library, 1989, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 4, pp. 247–256.
8. Hebb, D.O. *The Organization of Behavior*. s.l. : New York: Wiley & Sons, 1949.
9. *Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity*. Löwel, Siegrid et Singer, Wolf. s.l. : American Association for the Advancement of Science, 1992, *Science*, Vol. 255, pp. 209–212.
10. *The synaptic theory of memory: a historical survey and reconciliation of recent opposition*. Langille, Jesse J. et Brown, Richard E. s.l. : Frontiers Media SA, 2018, *Frontiers in systems neuroscience*, Vol. 12, p. 52.
11. *Linear summation of excitatory inputs by CA1 pyramidal neurons*. Cash, Sydney et Yuste, Rafael. s.l. : Elsevier, 1999, *Neuron*, Vol. 22, pp. 383–394.
12. Lefkowitz, Melanie. Professor's perceptron paved the way for AI – 60 years too soon. [En ligne] CORNELL CHRONICLE, 25 septembre 2019. [Citation : 4 Mai 2023.] <https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon>.
13. *The influence of the sigmoid function parameters on the speed of backpropagation learning*. Han, Jun et Moraga, Claudio. 1995. From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7–9, 1995 Proceedings 3. pp. 195–201.
14. Bierlaire, Michel. *Enseignement des mathématiques*. s.l. : Press polytechniques et universitaires romandes, 2006.

15. Sejnowski, Terrence J. *The deep learning revolution*. s.l. : MIT press, 2018.
16. Haykin, Simon. *Neural networks: a comprehensive foundation*. s.l. : Prentice Hall PTR, 1998.
17. Vinciarelli, A. *A Survey on Off-line Word Recognition*,”. 2002.
18. *Text detection and recognition in imagery: A survey*. Ye, Qixiang et Doermann, David. s.l. : IEEE, 2014, IEEE transactions on pattern analysis and machine intelligence, Vol. 37, pp. 1480–1500.
19. Chenduo, Li Gefei and Zhao. *Hanyu dazidian lunwenji ("Collected Essays on the Hanyu da zidian")*. 1990.
20. Shi, B., Bai, X. and Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*. 2016, pp. 2298-2304.
21. *Aster: An attentional scene text recognizer with flexible rectification*. Shi, Baoguang, et al. s.l. : IEEE, 2018, IEEE transactions on pattern analysis and machine intelligence, Vol. 41, pp. 2035–2048.
22. *Recursive recurrent nets with attention modeling for ocr in the wild*. Lee, Chen-Yu et Osindero, Simon. 2016. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2231–2239.
23. *NRTR: A no-recurrence sequence-to-sequence model for scene text recognition*. Sheng, Fenfen, Chen, Zhineng et Xu, Bo. 2019. 2019 International conference on document analysis and recognition (ICDAR). pp. 781–786.
24. *A holistic representation guided attention network for scene text recognition*. Yang, Lu, et al. s.l. : Elsevier, 2020, Neurocomputing, Vol. 414, pp. 67–75.
25. Jaided AI. aboutus. [En ligne] 2020. [Citation : 5 5 2023.] <https://www.jaided.ai/aboutus/>.
26. JaidedAI. EasyOCR. [En ligne] Github, 2020. [Citation : 6 5 2023.] <https://github.com/JaidedAI/EasyOCR>.
27. *A convolutional neural network VLSI for image recognition using merged/mixed analog-digital architecture*. Korekado, Keisuke, et al. 2003. Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003, Oxford, UK, September 2003. Proceedings, Part II 7. pp. 169–176.
28. Valueva, M.V., et al. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*. 2020, pp. 177: 232–243.
29. Yin W, Kann K, Yu M, Schütze H. Comparative study of CNN and RNN for natural language processing DELETE. *arXiv preprint arXiv*. 7 fevrier 2017.
30. *Tesseract: an open-source optical character recognition engine*. Kay, Anthony. s.l. : Belltown Media Houston, TX, 2007, Linux Journal, Vol. 2007, p. 2.

31. Luc Vincent, Uber Tech Lead. Google code. [En ligne] Google, aout 2006. [Citation : 6 Mai 2023.] <http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html>.
32. Schmidt, Julia. TESSERACT manual page. [En ligne] Github, 15 mars 2018. [Citation : 6 mai 2023.] <https://github.com/tesseract-ocr/tesseract>.
33. *Long short-term memory*. Hochreiter, Sepp et Schmidhuber, Jürgen. s.l. : MIT press, 1997, Neural computation, Vol. 9, pp. 1735–1780.
34. *Attention is all you need*. Vaswani, Ashish, et al. 2017, Advances in neural information processing systems, Vol. 30.
35. *Comparative study of CNN and RNN for natural language processing*. Yin, Wenpeng, et al. 2017, arXiv preprint arXiv:1702.01923.
36. *End-to-end object detection with transformers*. Carion, Nicolas, et al. 2020. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 213–229.
37. *Trocr: Transformer-based optical character recognition with pre-trained models*. Li, Minghao, et al. 2021, arXiv preprint arXiv:2109.10282.
38. *Sequence to sequence learning with neural networks*. Sutskever, Ilya, Vinyals, Oriol et Le, Quoc V. 2014, Advances in neural information processing systems, Vol. 27.
39. YanhuiDua. PaddleOCR. [En ligne] Github, 2012. [Citation : 8 Mai 2023.] <https://github.com/PaddlePaddle/PaddleOCR#LICENSE>.
40. *Reciprocal feature learning via explicit and implicit tasks in scene text recognition*. Jiang, Hui, et al. 2021. Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I. pp. 287–303.
41. *Resnet in resnet: Generalizing residual architectures*. Targ, Sasha, Almeida, Diogo et Lyman, Kevin. 2016, arXiv preprint arXiv:1603.08029.
42. *Demystifying resnet*. Li, Sihan, et al. 2016, arXiv preprint arXiv:1611.01186.
43. *Improving Character Error Rate is Not Equal to Having Clean Speech: Speech Enhancement for ASR Systems with Black-Box Acoustic Models*. Sawata, Ryosuke, Kashiwagi, Yosuke et Takahashi, Shusuke. 2022. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 991–995.
44. *How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs*. Holley, Rose. s.l. : D-Lib Magazine, 2009, D-Lib Magazine, Vol. 15.