REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA11/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle (IA)**

Utilisation de méthodes de Deep learning pour l'extraction de texte dans les images

---

Par :
MAAMOULI ABDELOUAHEB

Soutenu le jj/mm/aaaa devant le jury composé de :

NOM PRENOM	Grade	Président
NOM PRENOM	Grade	Rapporteur
NOM PRENOM	Grade	Examinateur

Année universitaire 2020-2021

Remerciements

Tout d'abord, je veux remercier Allah pour tout ce qu'il m'a donné et pour toutes les forces qu'il m'a données. Je voudrais également remercier mes parents pour m'encourager et mes frères aussi Raouf, Ali et Amine.

Je tiens à remercier mon encadrant de thèse DR.Tibarmacine Ahmed de m'avoir guidé dans cette recherche que j'ai donnée et de m'avoir montré la bonne direction pour pouvoir accomplir la tâche de mon travail.

Abstract

Due to the text diversity, scene complexity, and distortion factors, text detection is a challenging task. The traditional image processing method relies heavily on manmade features, which is not universally applicable to all text scenarios. In order to solve this problem, the deep learning technology is applied to learn and extract features of texts adaptively. In this work, a novel learning method based on Convolution Neural Network (CNN) is proposed to detect texts in different scenarios. Experiments on benchmark comparison are conducted upon four popular datasets, i.e., ICDAR2015, ICDAR2013, MSRA-TD500, and ICDAR2017-MLT. The results indicate that the proposed model significantly outperforms state-of the- art methods in terms of detection efficiency while maintaining high accuracy; for example, the proposed method achieves an precision more than 80% at 130 fps with 416 p resolution on the ICDAR2013, ICDAR2017 and MSRA-TD500 datasets.

Key words- Machine learning, deep learning, Computer vision, Text detection, Convolutional neural networks.

Résumé

En raison de la diversité du texte, de la complexité de la scène et des facteurs de distorsion, la détection de texte est une tâche difficile. La méthode de traitement d'image traditionnelle repose fortement sur des caractéristiques artificielles, ce qui n'est pas universellement applicable à tous les scénarios de texte. Afin de résoudre ce problème, la technologie d'apprentissage en profondeur est appliquée pour apprendre et extraire les caractéristiques des textes de manière adaptative. Dans ce travail, une nouvelle méthode d'apprentissage basée sur Convolution Neural Network (CNN) est proposée pour détecter des textes dans différents scénarios. Des expériences de comparaison de référence sont menées sur quatre ensembles de données populaires, à savoir ICDAR2015, ICDAR2013, MSRA-TD500 et ICDAR2017-MLT. Les résultats indiquent que le modèle proposé surpasse de manière significative les méthodes de pointe en termes d'efficacité de détection tout en maintenant une grande précision ; par exemple, la méthode proposée atteint une précision plus que 80% à 130 fps avec une résolution de 416 p sur l'ensemble de données ICDAR2013, ICDAR2017 et MSRA-TD500.

Mots clés- apprentissage automatique, apprentissage profond, vision par ordinateur, Détection de texte, Réseaux de neurones convolutifs.

Table des matières

CHAPITRE 01 : MACHINE LEARNING

CHAPITRE 02 : VISION PAR ORDINATEUR

CHAPITRE03 : CONCEPTION ET IMPLEMENTATION

Liste des figues

Liste des tableaux

Introduction Générale

Le texte est un outil essentiel de communication et joue un rôle important dans notre vie quotidienne. Il peut être intégré dans des documents ou des scènes comme moyen de transmettre d'informations. L'extraction du texte peut être considérée comme un élément de base principal pour une variété d'applications basées sur la vision par ordinateur, telles que la robotique, l'automatisation industrielle, la recherche d'images, la traduction instantanée, l'assistance automobile et l'analyse de vidéos sportives.

L'extraction de texte est un problème difficile, en particulier lorsque les images de texte sont prises dans un environnement sans contrainte « unconstrained environnement ». Généralement, la détection du texte peut être classée en deux catégories principales:

- Texte structuré : texte dans un document dactylographié, ou l'arrière-plan, la police et la densité du texte sont standard et le texte est organisé dans des lignés
- Texte non structuré (texte de scène): il s'agit de textes à des endroits aléatoires dans une scène naturelle. Ce texte peut être clairsemé, il n'est pas structuré en des lignes appropriées et ni écrit en police standard. L'arrière-plan dans ce type de texte est complexe et le texte peut être dans des endroits aléatoires dans l'image.

Les principaux défis présents dans l'extraction des textes de l'image sont :

- Diversité de texte: le texte peut être en différents couleurs, polices, orientations et langues.
- Complexité de la scène : les éléments constituant la scène peuvent être similaires au texte, tels que: les signes, les briques et les symboles.
- Facteurs de distorsion: L'image peut être subir à divers facteurs des distorsions tels que : le flou de mouvement, la résolution insuffisante de la caméra, l'angle de capture et l'occlusion partielle.

De nombreuses techniques antérieures ont résolu le problème de détection du texte structuré. Cependant, ces techniques n'ont pas correctement fonctionné pour le texte d'image de scène naturelle, qui est clairsemée et a des attributs différents de ceux des données structurées.

Dans ce travail, nous nous concentrerons sur le texte non structuré qui est un problème plus complexe à résoudre.

Au cours de la dernière décennie, de nombreuses méthodes ont été développées pour gérer les tâches mentionnées ci-dessus [19-21] , dans lesquelles les régions de texte sont

détectées avec une précision encourageante. En général, ces méthodes suivent une stratégie en deux étapes basée sur le CNN profond.

Le processus de détection est le suivant : premièrement, les propositions de région sont générées via une méthode de recherche sélective ou un réseau de propositions régionales ; ensuite, la régression des cadres de délimitation est utilisée pour les propositions. Malgré une excellente précision, l'efficacité de calcul de ces méthodes dans les applications en temps réel est loin d'être satisfaisante. Pour obtenir une détection de texte de haute précision et en temps réel dans des environnements complexes, nous proposons un détecteur de texte en une étape, basé sur l'architecture YOLO (You Only Look Once)[19-21].

Le présent mémoire est organisé en trois chapitres dont les thèmes sont donnés ci-dessous: dans un premier temps, nous introduisons la notion de Machine Learning « ML » et de nouveau paradigme Deep Learning « DL ». Le deuxième chapitre est consacré pour explorer le domaine interdisciplinaire: vision par ordinateur et ses différentes théories. Nous introduisons également dans ce chapitre les travaux connexes à notre problématique de recherche. Dans le troisième chapitre nous présentons la conception générale de notre proposition de détection de texte dans les images de scènes, les bases de données de référence utilisé pour évaluer notre modèle, les expérimentations réalisées et les résultats obtenus.

Chapitre 01 :

Machine Learning

1.1 Introduction

Machine Learning est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité «apprendre »à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, cela concerne la conception, l'analyse, le développement et l'implémentation de telles méthodes.

1.2 Différents types d'apprentissage

1.2.1 Apprentissage supervisé

L'apprentissage supervisé commence généralement par un ensemble de données clairement définies et comprend la manière dont ces données sont classées. Le but de l'apprentissage supervisé est de découvrir des modèles dans les données et de les appliquer au processus d'analyse. Ces données comprennent des caractéristiques associées à l'étiquette qui définit sa signification. Par exemple, vous pouvez créer une application d'apprentissage automatique qui peut marquer des millions d'animaux sur la base d'images et de descriptions écrites[1 ,2].

En apprentissage supervisé, les exemples d'apprentissage sont donnés sous forme de paires $P(X,Y) \in R^d \times R^m$ appelés exemples étiquetés, où x est appelé vecteur d'entrée, d est la dimension des entrées, y est le vecteur cible (étiquette), et m sa dimension [4]. L'ensemble d'apprentissage $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ contient n exemples dont le but est d'apprendre une fonction fDtelle que $fD(x) \simeq y$ [4].

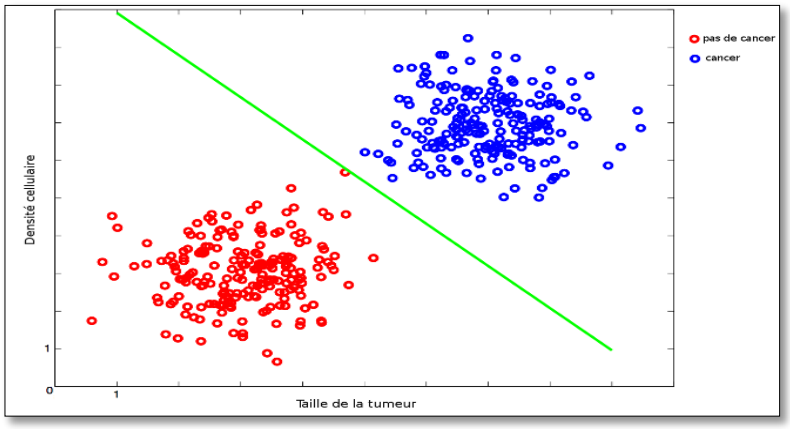

Figure1. 1 Apprentissage supervisé : classification.

1.2.2 Apprentissage non supervisé

L'apprentissage non supervisé est utilisé lorsque le problème nécessite une quantité massive de données non étiquetées. Par exemple, les applications de réseaux sociaux, telles que Twitter, Instagram et Snapchat, exploitent toutes de très grandes quantités de données non étiquetées. Pour comprendre le sens de ces données, il est nécessaire d'utiliser des algorithmes qui classifient les données en fonction des tendances ou des clusters qu'ils décèlent. L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine. Il est utilisé avec la technologie de détection de spam envoyé par e-mail. Les e-mails normaux et les spams comportent un nombre de variables beaucoup trop élevé pour qu'un analyste puisse étiqueter les e-mails indésirables envoyés en masse. En revanche, les discriminants d'apprentissage automatique, basés sur la mise en cluster et l'association, sont appliqués pour identifier les courriers électroniques non désirés[1,2].

La classification non supervisée consiste à attribuer à chaque $x_i \in D$une classe $f_D(x_i) \in \{1,2,\ldots.,c\}$.Le nombre de classes c peut, selon l'algorithme f, être soit fixé par l'utilisateur, soit déterminé automatiquement. Le but est de regrouper les exemples de telle sorte que tous les exemples partageant le même label de classe soient similaires (selon des critères dépendant de de f)[4].

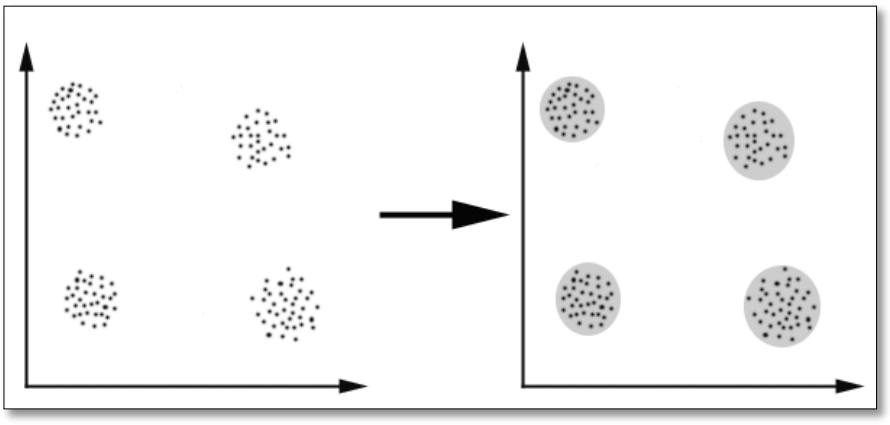

Figure1. 2 Apprentissage non supervisé : K-means

1.2.3 Apprentissage semi-supervisé

Ces algorithmes fournissent une technique qui utilise la puissance de l'apprentissage supervisé et non supervisé. Dans les deux premiers types, soit des étiquettes de sortie sont fournies pour tous les cas, soit aucune étiquette n'est fournie. Il peut arriver que certaines observations soient marquées, mais en raison du coût élevé du marquage et du manque d'expertise humaine qualifiée, la plupart des observations ne sont pas marquées. Dans ce cas, l'algorithme semi-supervisé est le plus approprié pour la construction de modèles.

L'apprentissage semi-supervisé peut être utilisé pour traiter des problèmes de classification, de régression et de prédiction [2].

Il s'applique à toute tâche où l'ensemble d'apprentissage est de la forme $D = \{(x_1, y_1), \cdots, (x_l, y_l), x_{l+1}, .., x_n\}$c'est-à-dire contient l paires d'exemples étiquetés, et $u = n - l$ exemples non étiquetés [4].

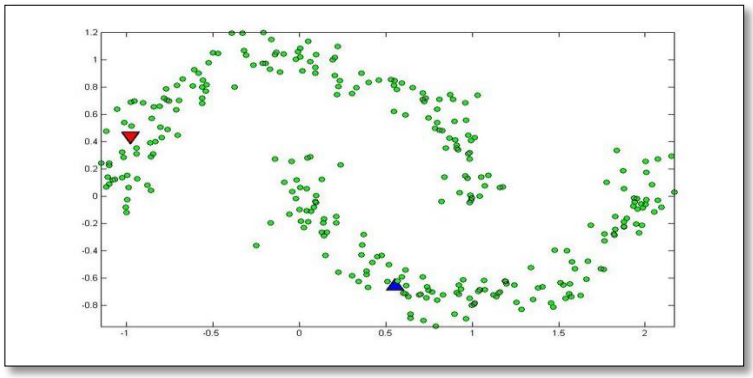

Figure1. 3 Apprentissage semi-supervisé : des modèles de mélange

1.2.4 Apprentissage par renforcement

L'apprentissage par renforcement est un sous-domaine de l'apprentissage automatique inspiré de la psychologie comportementale, traitant de certains concepts sur la façon dont les agents logiciels doivent agir dans l'environnement pour maximiser les récompenses cumulatives. Il a été étudié et utilisé dans de nombreuses théories telles que la théorie des jeux, la cybernétique, la recherche opérationnelle, la théorie de l'information, l'intelligence en essaim, la statique et les algorithmes génétiques [1-3]. On concentre sur la recherche de différents algorithmes d'apprentissage automatique qui aideront le système à classer les données et leur permettront de prendre des décisions dans des situations complexes.

1.3 Généralisation

1.3.1 Sur-apprentissage

Le sur-apprentissage est souvent utilisé comme terme générique pour décrire toute baisse de performance indésirable d'un modèle d'apprentissage automatique. Ici, nous nous concentrons sur le sur-ajustement adaptatif, qui est un sur-ajustement causé par la réutilisation des ensembles de tests. Alors que d'autres phénomènes sous l'égide du sur-apprentissage sont également des aspects importants d'un apprentissage machine fiable (par exemple, des baisses de performances dues à des changements de distribution) [5].

Formellement, soit $f: X \rightarrow Y$ un modèle entraîné qui mappe les exemples $x \in X$ aux valeurs de sortie $y \in Y$ (par exemple, les étiquettes de classe ou les cibles de régression).

L'approche standard pour mesurer les performances d'un tel modèle entraîné consiste à définir une fonction de perte $L: Y \times Y \rightarrow R$ et à tirer des échantillons $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ à partir d'une distribution de données D que nous utilisons ensuite pour évaluer la perte de test $L_S(f) = \sum_{i=1}^{n} L(f(x_i), y_i)$ .Tant que le modèle f ne dépend pas de l'ensemble de test S, les résultats de concentration standard montrent que$L_S(f)$ est une bonne approximation de la véritable performance donnée par la perte de population $L_D(f) = E_D[L(f(x_i), y_i)]$ [5].

Cependant, les praticiens de l'apprentissage automatique sapent souvent l'hypothèse selon laquelle f ne dépend pas de l'ensemble de tests en sélectionnant des modèles et en ajustant les hyper paramètres en fonction de la perte de test. Surtout lorsque les concepteurs d'algorithmes évaluent un grand nombre de modèles différents sur le même ensemble de test, le classificateur final peut ne fonctionner correctement que sur les exemples spécifiques de l'ensemble de test. L'échec à généraliser à l'ensemble de la distribution de données D se manifeste par un grand écart d'adaptabilité $L_S(f) - L_D(f)$et conduit à des estimations de performances trop optimistes [5].

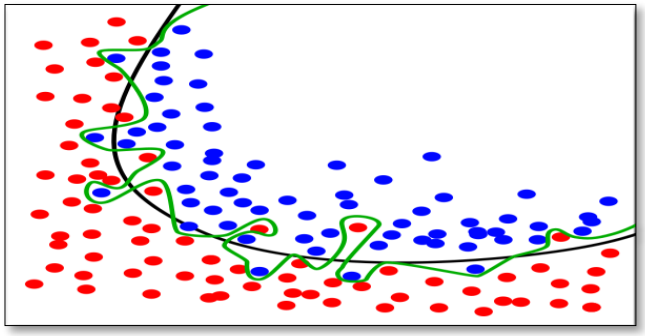

Figure1. 4 .Exemple sur-apprentissage d'un classificateur

1.3.2 Régularisation

La régularisation est le processus d'ajout d'informations afin de résoudre un problème mal posé ou d'éviter le sur-apprentissage [2]. On l'utilise en mathématiques, statistiques, finance, informatique, notamment en apprentissage automatique et problèmes inverses.

Pour les réseaux de neurones, les méthodes de régularisation les plus populaires sont le Dropout (les poids – paramètres du réseau de neurones – sont remplacés par zéro de manière aléatoire pendant l 'entrainement), l'Early Stopping (l'apprentissage s'arrête plus tôt pour favoriser les modèles simples) ou la régularisation euclidienne évoquée plus haut.

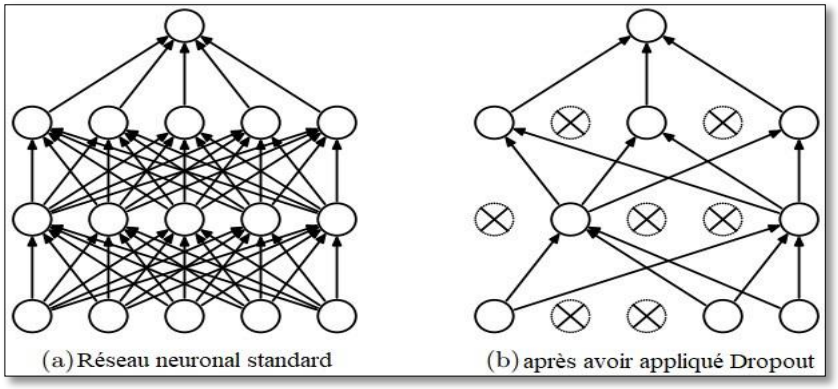

Figure1. 5 Exemple de régularisation: Droupout

1.3.3 Malédiction de la dimensionnalité

On Le fléau de la dimension ou malédiction de la dimension est un terme inventé par Richard Bellman en 1961 pour spécifier divers phénomènes qui se produisent lorsque les gens essaient d'analyser ou d'organiser des données dans un espace de grande dimension. Cela ne se produit pas dans un espace plus petit [9].

Il implique plusieurs domaines, notamment l'apprentissage automatique, l'exploration de données, les bases de données, l'analyse numérique et l'échantillonnage. L'idée générale est qu'à mesure que la dimensionnalité augmente, le volume de l'espace augmente rapidement, rendant les données "isolées" et dispersées. Ceci est problématique pour les méthodes qui nécessitent de grandes quantités de données pour être efficaces, ce qui les rend inefficaces [9].

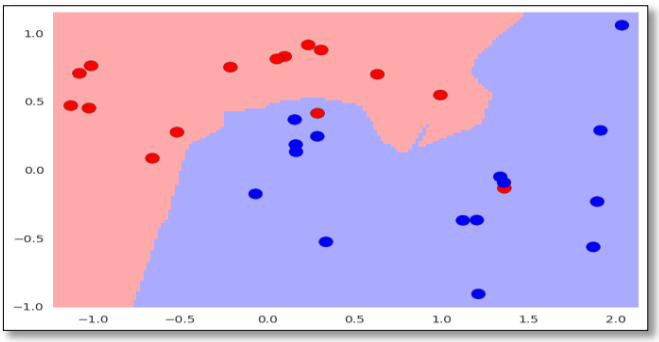

Figure1. 6 Modèle 5-NN appliqué à un jeu de données

1.4 Différents types de modèles

1.4.1 Modèles paramétriques

En apprentissage automatique, un modèle paramétrique est tout modèle qui capture toutes les informations sur ses prédictions dans un ensemble fini de paramètres. Parfois, le modèle doit être entraîné pour sélectionner ses paramètres, comme dans le cas des réseaux de

neurones, réseau neuronal artificiel, Régression vectorielle de support, Régression des processus gaussiens [6].

Les modèles paramétriques peuvent également être statistiquement inefficaces. S'il y a moins d'exemples d'apprentissage, alors le problème est sur-paramétré et on risque le sur-apprentissage : le modèle pourrait apprendre des paramètres taillés "sur mesure" pour les données d'entraînement, mais qui mèneront à une mauvaise généralisation. La conséquence de cette observation est qu'en général, un modèle avec un grand nombre de paramètres est statistiquement inefficace [9].

1.4.2 Modelés non paramétriques

Les modèles non paramétriques sont des modèles statistiques qui ne se conforment pas souvent à une distribution normale, car ils reposent sur des données continues plutôt que sur des valeurs discrètes. Les statistiques non paramétriques traitent souvent de nombres ordinaux ou de données dont la valeur n'est pas aussi fixe qu'un nombre discret comme modèle Black–Scholes Merton, modèle Heston et modèle Merton [6].

1.5 Algorithmes d'apprentissage

1.5.1 Machines de Boltzmann restreintes

La machine de Boltzmann restreinte est un type de réseau de neurones artificiels pour l'apprentissage non supervisé. Elle est couramment utilisée pour avoir une estimation de la distribution probabiliste d'un jeu de données. Elle a initialement été inventée sous le nom d'Harmonium en 1986 par Paul Smolenski [11]

Dans sa forme la plus simple, une machine de Boltzmann est composée d'une couche de neurones qui reçoit l'entrée, ainsi que d'une couche de neurones cachée. Si on suppose que les neurones d'une même couche sont indépendants entre eux, on appelle cette configuration une machine de Boltzmann restreinte (RBM).

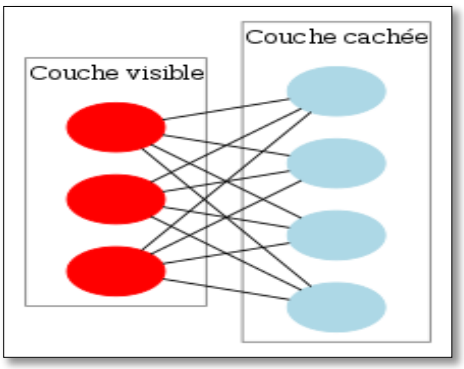

Figure1. 7 Machine de Boltzmann restreinte

On définit une énergie d'activation pour une Machine de Boltzmann Restreinte de la manière suivante:

$$\mathrm{E} = -(\sum_{i,j} w_{ij}\, x_i h_j + \sum_i b_i\, x_i + \sum_j c_j\, h_j)$$

Où :

- w_{ij} est le poids entre le neurone i et j
- x_i est l'état, $\in \{0,1\}$, du neurone visible i
- h_j est l'état du neurone caché j
- b_i , c_j sont respectivement les biais des neurones x_i , h_j

Et la probabilité conjointe d'avoir une configuration (x_i , h_j) [12]:

$$P(x_i , h_j) = \exp\left(-E(x_i , h_j)\right)/Z$$

Où:

- E est la fonction d'énergie
- Z est une fonction de normalisation, qui fait en sorte que la somme de toutes les probabilités fasse 1.

1.5.2 K-plus proches voisins

K-plus ou K-Nearest Neighbors proche voisin est une série de techniques qui peuvent être utilisées pour la classification ou la régression. En tant qu'algorithme d'apprentissage non paramétrique, le K-plus proche voisin n'est pas limité à un ensemble de paramètres. Cependant, leur complexité dépendra de la taille de la base d'apprentissage.

L'algorithme KNN peut rivaliser avec le modèle le plus précis car il peut faire des prédictions extrêmement précises. Par conséquent, il peut être utilisé pour des applications qui nécessitent une grande précision mais n'ont pas besoin de générer des modèles lisibles par l'homme, car KNN ne génère pas de modèle intelligible. K-plus proche voisin a été utilisé dans de nombreux domaines et de nombreuses applications, telles que l'estimation statistique et la reconnaissance de formes, la prédiction d'événements économiques et l'estimation de la capacité des batteries lithium-ion, la mesure de distance , la catégorisation de texte et la classification multi-label [2,12].

Où la classification par les k plus proches voisins est illustrée en figure 1.9 :

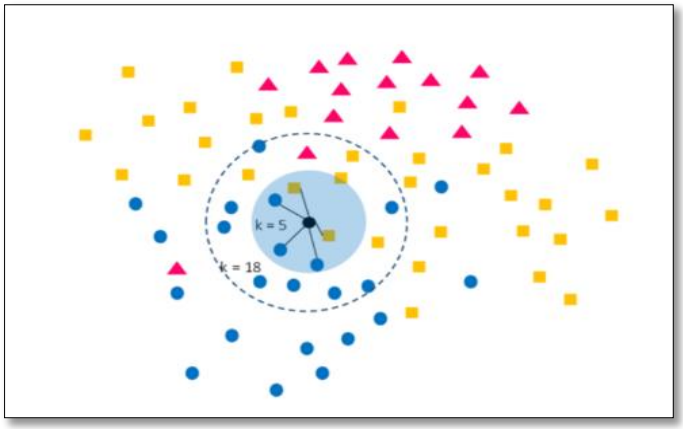

Figure1. 8 Classification avec l'algorithme des k plus proches voisins : le nouvel individu sera affecté à la classe "cercle bleu" dans les deux cas : k=5 ou k=18, car la majorité de ses k voisins appartiennent à cette classe [12].

1.5.3 Fenêtres de Parzen

La fenêtre de Parzen est une méthode d'apprentissage de voisinage (Chappelle, 2005), qui est proche de la méthode k du plus proche voisin. En considérant les instances où la proximité sera considérée comme « suffisante », de nouvelles instances peuvent être prédites. La différence entre la fenêtre de Parzen et la méthode du plus proche voisin k réside dans le concept de voisinage. Dans la méthode du voisin le plus proche k, il s'agit d'une constante (la valeur de k), tandis que dans le cas d'une fenêtre de Parzen, elle est définie par le noyau.

Notation : n représente le nombre d'instances disponibles dans la base de données (les exemples d'apprentissage), i représente l'index d'une des instances de la base de données, x représente une donnée pour laquelle on souhaite faire une prédiction. $K(x, x_i)$ représente le calcul de la fonction noyau (k) entre l'instance x et l'instance x_i. L'utilisation d'une fenêtre de Parzen en classification ou en régression est décrite dans le tableau 1.1 ci-dessous [13]:

Tableau 1. 1 . Utilisation de la fenêtre de Parzen[13].

Classification	Régression
$P(x\|y) = \frac{\sum_{i=1, y_i=y}^{n} K(x, x_i)}{\sum_{l=1}^{n} K(x, x_l)}$	$\hat{y} = \sum_{i=1}^{n} \left(\frac{K(x, x_i)}{\sum_{l=1}^{n} K(x, x_l)} \right) y_i$
La prédiction de la classe y pour l'instance x est la somme normalisée des fonctions noyau pour la classe considérée entre x et l'ensemble des instances de la base d'apprentissage	La prédiction de la valeur y de l'instance x est la somme normalisée des fonctions noyau entre cette instance x et l'ensemble des instances de la base d'apprentissage pondérée par leur valeur y

1.5.4 Mélanges de Gaussiennes

Les mélanges de Gaussiennes généralisent les fenêtres de Parzen (avec noyau Gaussien) pour l'estimation de densité, en écrivant la densité comme une somme pondérée de Gaussiennes [16] :

$$f(x) = \sum_{j=1}^{N} \alpha_j \mathcal{N}(x; \mu_j, \Sigma_j) \quad (22)$$

Où N est le nombre de composantes du mélange, et α_j le poids de la j^{ème} composante (les poids sont tels que $\alpha_j \geq 0$ et$\Sigma_j \alpha_j = 1$). L'interprétation dite "générative'' de cette équation est que le modèle suppose que chaque exemple observé a été généré de la façon suivante [9] :

- Une composante j est choisie aléatoirement, avec probabilité α_j.
- Un exemple est généré par une distribution Gaussienne centrée en μ_j avec covariance Σj.

Si N = *n*, $\alpha_j = n^{-1}$, $\mu_j = x_i$ et $\Sigma_j = \sigma^2\mathbf{I}$ on retrouve les fenêtres de Parzen non paramétriques vues précédemment. Mais on peut également apprendre un modèle paramétrique de mélange en fixant N et en optimisant les poids α_j, les centres μ_j et les covariances Σ_j de chaque Gaussienne. L'algorithme le plus populaire pour l'apprentissage d'un mélange de Gaussiennes est l'algorithme Espérance-Maximisation (EM) [17].

1.5.5 Méthodes à noyau

Plusieurs algorithmes mentionnés précédemment utilisent une fonction noyau $K(x_i, x_j)$ pour mesurer la similarité entre deux entrées x_i et x_j. Les méthodes dites "à noyau" incluent en particulier une catégorie d'algorithmes basés sur ce que l'on appelle "l'astuce du noyau", qui s'applique à tout algorithme qui peut s'exprimer uniquement à partir de produits scalaires de la forme $x_i^T x_j$. Cette astuce consiste à remplacer $x_i^T x_j$ dans l'algorithme d'origine par une fonction noyau $K(x_i, x_j)$, où K doit satisfaire certaines propriétés mathématiques (on dit que le noyau est défini positif – c'est le cas par exemple du noyau Gaussien que nous avons déjà utilisé). Cela revient à appliquer l'algorithme sur les données transformées implicitement et de manière non linéaire par une fonction ϕ telle que $\varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ (une telle fonction ϕ existe automatiquement si K est défini positif, et en pratique on n'a pas besoin de la calculer explicitement). L'intérêt du noyau est principalement d'effectuer une extraction de caractéristiques non linéaire à partir des entrées, ce qui peut améliorer les performances de l'algorithme [15,16].

La plus célèbre méthode à noyau exploitant cette astuce est l'algorithme de la machine à vecteurs de support (SVM, pour "Support Vector Machine" en anglais). Il s'agit d'un algorithme de classification basé sur l'idée de marge : pour obtenir une meilleure généralisation, il est préférable de laisser une marge entre les exemples et la surface de décision [15,16].

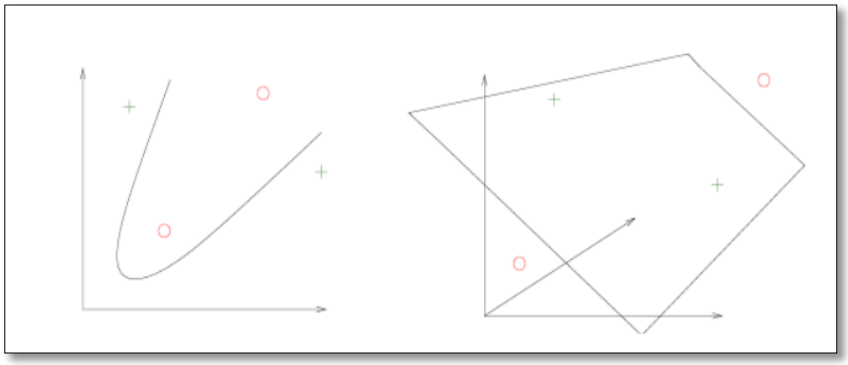

Figure1. 9 Exemple de méthode à noyaux [16].

1.5.6 Arbres de décision

L'arbre de décision est une technique d'approximation d'une fonction cible à valeur discrète qui représente la fonction apprise sous la forme d'un arbre de décision [2]. Un arbre de décision est un arbre où chaque nœud interne représente un test sur l'entrée x_i. L'exemple typique d'un arbre de décision est un arbre binaire où le test effectué à chaque nœud k est de la forme $x_{ij}<\theta_k$, c'est-à-dire. Qu'on compare la $j^{éme}$ coordonnée de x_i à un seuil θ_k : si elle est plus petite, on continue de parcourir l'arbre en suivant la première branche du nœud, sinon on suit la seconde branche. Lorsqu'on atteint finalement une feuille de l'arbre (un nœud sans enfants), on dit que l'exemple x_i appartient à cette feuille[2,12]. La figure 1.18 montre un tel arbre de décision.

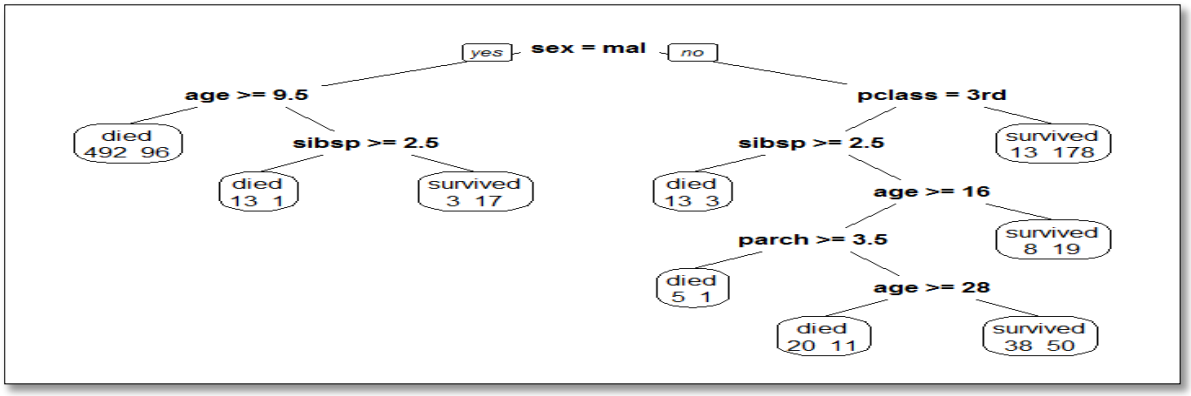

Figure1. 10 Arbre de décision

1.5.7 Méthodes Bayésiennes

Les méthodes Bayésiennes tirent leur nom de la célèbre règle de Bayes [2,9], alors elles sont basées sur La théorie des probabilités :

$$P(\mathrm{A}|B) = \frac{P(\mathrm{B}|A)P(\mathrm{A})}{P(B)} \quad (1)$$

Ou :

- A et B sont des événements
- $P(A)$ et $P(B)$ sont les probabilités d'observer A et B indépendamment l'une de l'autre.
- $P(A/B)$ est la probabilité conditionnelle, c'est-à-dire la probabilité d'observer A, étant donné que B est vrai.
- $P(B/A)$ est la probabilité d'observer B, étant donné que A est vrai.

1.5.8 Réseaux de neurones artificiels

L'unité de base du calcul dans un réseau de neurones artificiel est le neurone qui a été défini dans [12] en 1959. Un neurone artificiel reçoit des entrées de certains autres neurones ou d'une source externe ayant des valeurs $x1, x2 \ldots . . xn$ auxquels il est connecté par des synapses et calcule une sortie y. Chaque entréexia un poids associé Wi, qui est attribué en fonction de son importance relative par rapport aux autres entrées. La valeur d'entrée x du neurone correspond à la somme pondérée de ses entrées en ajoutant une autre entrée ayant un poids b appelé biais. Ensuite, le neurone applique une fonction f sur cette somme, comme illustré à la Figure 1.12.

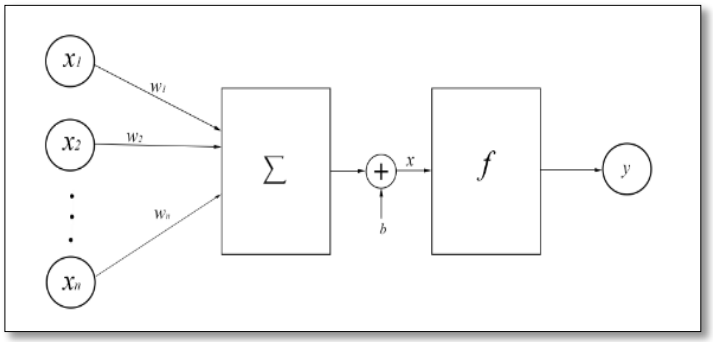

Figure1. 11 Modélisation d'un neurone artificiel.

Dans un réseau à propagation avant, les informations ne se déplacent que dans un seul sens, en partant des neurones d'entrée, passant par les neurones cachées (s'ils existent) et en

arrivant aux neurones de sortie. Un réseau de neurones peut être assimilé à un graphe orienté acyclique. Deux exemples de réseaux à propagation avant sont présentés dans la Figure 1.12 :

- Perceptron mono-couche - Il s'agit d'un réseau de neurones à propagation avant le plus simple qui contient une couche cachée (Figure 1.13.a).
- Perceptron multi-couches - Il comporte une ou plusieurs couches cachées. Dans le cadre de ces travaux, nous ne traiterons que le cas des perceptrons multi-couches, car ils sont plus utiles que les perceptrons mono-couches dans les applications actuelles (Figure 1.13.b).

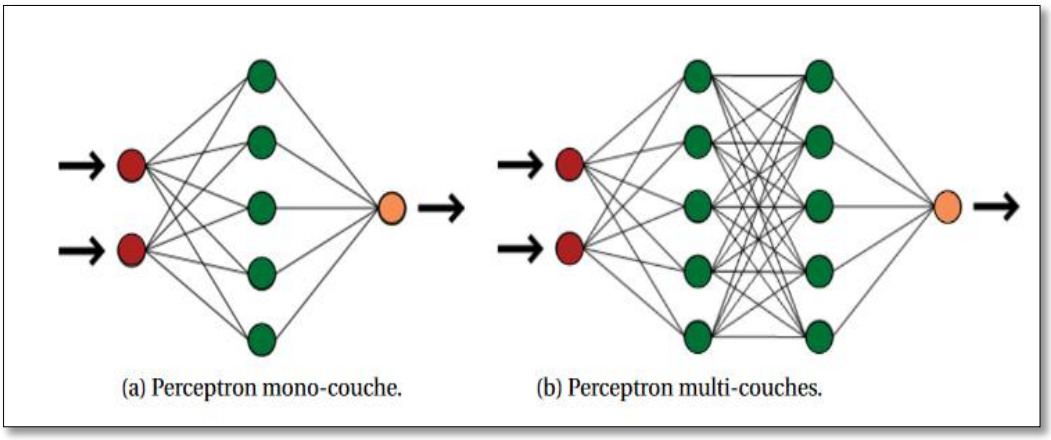

Figure1. 12 Deux types de perceptrons.

1.5.9 Deep Learning

L'apprentissage en profondeur est une classe d'algorithmes Machine learning qui utilise plusieurs couches pour extraire progressivement des caractéristiques de niveau supérieur de l'entrée brute. Par exemple, dans le traitement d'images, les couches inférieures peuvent identifier les contours, tandis que les couches supérieures peuvent identifier les concepts pertinents pour un humain, tels que les chiffres, les lettres ou les visages.

Le Deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à

d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ait jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant [14,18].

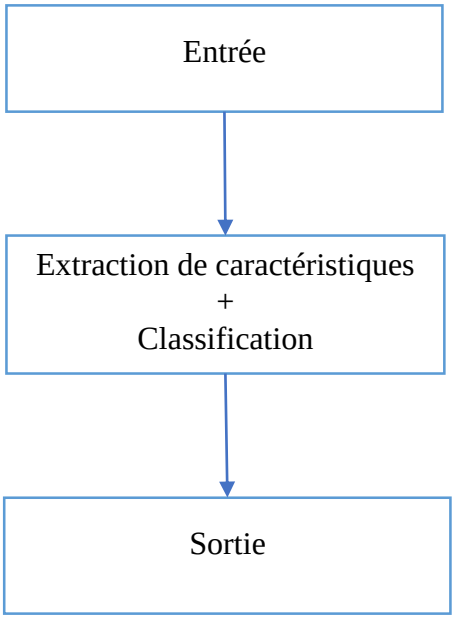

Figure1. 13 Les phases d'apprentissage profond

1.6 Domaine d'applications du Machine Learning

Le Machine learning est utilisée dans de nombreux domaines, on va définir quelques domaines et donne un exemple sur le domaine de détection d'objet qui est nous intéressé :

- Domaine médical : TRISS : Trauma & Injury Severity Score, qui est largement utilisé pour prédire la mortalité chez les patients blessés, a été développé à l'origine par Boyd et al. en utilisant la régression logistique. De nombreuses autres échelles médicales utilisées pour évaluer la gravité d'un patient ont été développées à l'aide de la régression logistique [2].
- Véhicules autonomes : Les modèles d'apprentissage automatique sont de nos jours appliqués pour conduire des véhicules autonomes comme les voitures, les drones, etc. Exemple : Google Driver Less Cars, Tesla Cars. Les techniques d'apprentissage automatique sont également très efficaces pour contrôler les applications basées sur des capteurs [2].

- Robotique et Intelligence Artificielle: L'apprentissage automatique est considéré comme une approche améliorée de la résolution de problèmes. En utilisant les connaissances de base et les données de formation avec des modèles d'apprentissage automatique, l'apprentissage peut être amélioré, ce qui portera la robotique et l'IA aux niveaux de la prochaine génération [2].
- Filtrage des e-mails (courriers indésirables) : L'apprentissage automatique peut être appliqué pour filtrer les e-mails de spam. Le modèle basé sur l'apprentissage automatique mémorisera simplement tous les e-mails classés comme spams par utilisateur. Lorsqu'un nouvel e-mail arrive dans la boîte de réception, le modèle basé sur l'apprentissage automatique recherche, compare et se base sur les e-mails de spam précédents. Si un nouvel e-mail correspond à l'un d'entre eux, il sera marqué comme spam ; sinon, il sera déplacé vers la boîte de réception de l'utilisateur [2].
- Détection d'objet : La détection d'objets est un domaine très actif de la recherche qui cherche à classer et localiser des régions/zones d'une image ou d'un flux vidéo. Ce domaine est à la croisée de deux autres : la classification d'image et la localisation d'objets. En effet, le principe de la détection d'objets est le suivant : pour une image donnée, on recherche les régions de celle-ci qui pourraient contenir un objet puis pour chacune de ces régions découvertes, on l'extrait et on la classe à l'aide d'un modèle de classification d'image. Les régions de l'image d'origine ayant de bons résultats de classification sont conservés et les autres jetés. Ainsi, pour avoir une bonne méthode de détection d'objets, il est nécessaire d'avoir un algorithme solide de détection de régions et un bon algorithme de classification.

Figure1. 14 Exemple de détection d'objet

1.7 Conclusion

Dans ce chapitre nous avons défini la notion de l'apprentissage artificiel et ses caractéristiques et nous avons introduit ses composants. Nous avons appris ses avantages. Ainsi, nous avons présenté quelques travaux antérieurs et récents ayant un lien avec notre travail de recherche.

Nous présenterons dans le chapitre suivant le domaine de vision par ordinateur et ses caractéristiques principales.

Chapitre 02 :

Vision par ordinateur

2.1 Introduction

Dans ce deuxième chapitre, nous définissons le domaine de vison par ordinateur et ses sous-concepts qui sont nous intéressé en notre travaille et on a expliqué la partie important dans chaque les sous-concepts pour la détection d'objets.

2.2 Définition

La vision par ordinateur est l'un des domaines de l'informatique qui vise à créer des applications intelligentes capables de comprendre le contenu des images tel que les gens les comprennent, il s'agit de la classification de l'image entière, comme dans un système de classification des photos téléchargées sur Internet (Facebook, Instagram),elle s'intéresse à la reconnaissance d'objets dans une image, comme la détection de visages ou de plaques d'immatriculation de voiture (Facebook, GoogleStreetView),il s'agit de la détection d'aspects d'une image, comme la détection du cancer dans les images biomédicales ou bien texte dans l'image de scéne [29-31].

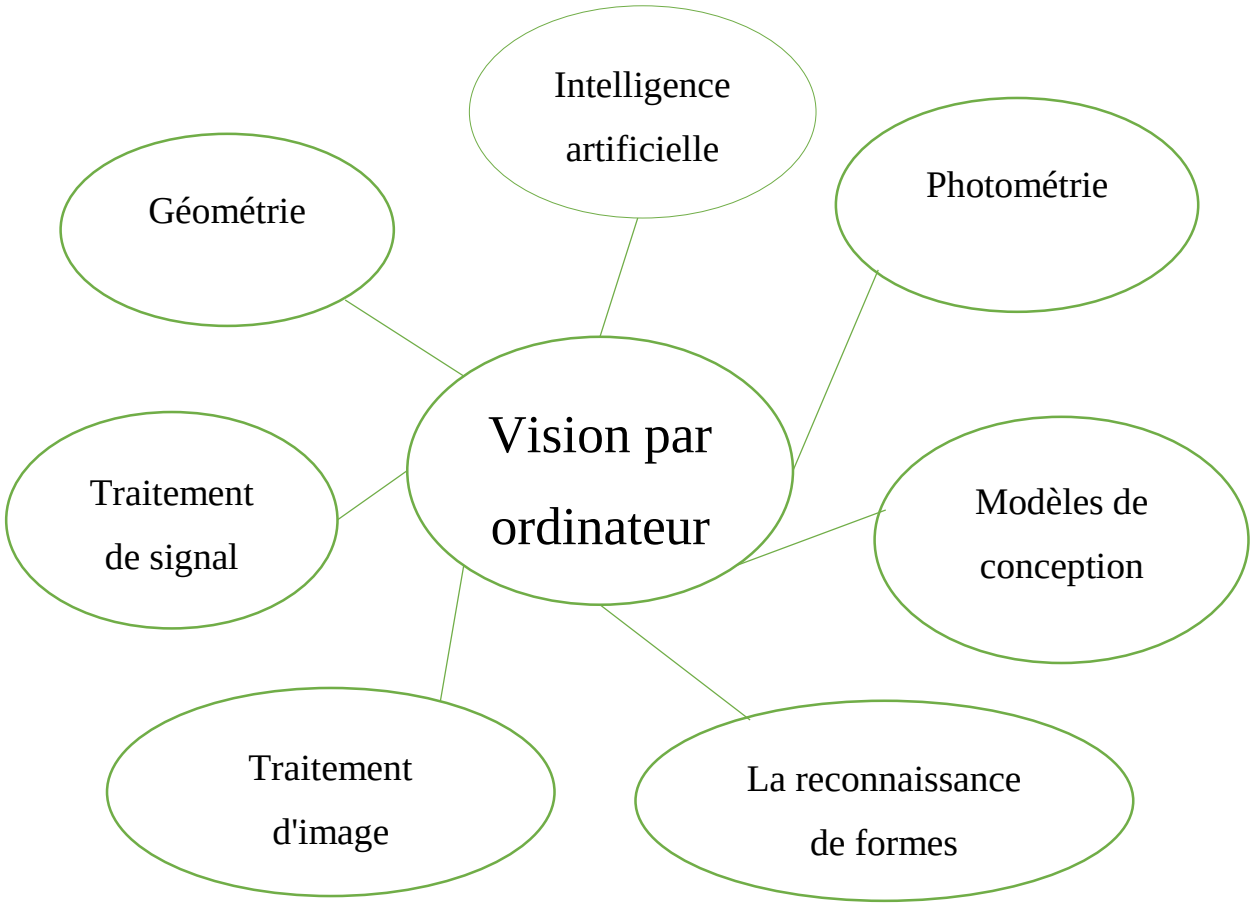

Figure 2. 1 Vision par ordinateur et ses sous-concepts

2.3 Historique

À la fin des années 1960, la vision par ordinateur a commencé dans les universités pionnières de l'intelligence artificielle. Il était censé imiter le système visuel humain, comme un tremplin pour doter les robots d'un comportement intelligent. En 1966, on croyait que cela pouvait être réalisé grâce à un projet d'été, en attachant une caméra à un ordinateur et en lui faisant « décrire ce qu'il a vu ».

Ce qui distinguait la vision par ordinateur du domaine prédominant du traitement d'images numériques à cette époque, c'était le désir d'extraire une structure tridimensionnelle des images dans le but d'obtenir une compréhension complète de la scène. Des études menées dans les années 1970 ont constitué les premières fondations de nombreux algorithmes de vision par ordinateur qui existent aujourd'hui, notamment l'extraction de bords d'images, l'étiquetage de lignes, la modélisation non polyédrique et polyédrique, la représentation d'objets en tant qu'interconnexions de structures plus petites, le flux optique et estimation de mouvement[29,30,31,32].

1974 a vu l'introduction de la technologie de reconnaissance optique de caractères (OCR), qui pouvait reconnaître le texte imprimé dans n'importe quelle police ou police de caractères. De même, la reconnaissance intelligente de caractères (ICR) pouvait déchiffrer le texte écrit à la main à l'aide de réseaux neuronaux. Depuis lors, OCR et ICR ont trouvé leur place dans le traitement des documents et des factures, la reconnaissance des plaques d'immatriculation, les paiements mobiles, la traduction automatique et d'autres applications courantes [29,30,31,32].

La décennie suivante a vu des études basées sur une analyse mathématique plus rigoureuse et des aspects quantitatifs de la vision par ordinateur. Ceux-ci incluent le concept d'échelle-espace, l'inférence de forme à partir de divers indices tels que l'ombrage, la texture et la mise au point, et les modèles de contour connus sous le nom de serpents. Les chercheurs ont également réalisé que bon nombre de ces concepts mathématiques pouvaient être traités dans le même cadre d'optimisation que la régularisation et les champs aléatoires de Markov. Dans les années 1990, certains des thèmes de recherche précédents sont devenus plus actifs que d'autres. Les recherches sur les reconstructions 3D projectives ont permis de mieux comprendre l'étalonnage des caméras. Avec l'avènement des méthodes d'optimisation pour l'étalonnage des caméras, on s'est rendu compte que de nombreuses idées étaient déjà explorées dans la théorie de l'ajustement des faisceaux du domaine de la photogrammétrie. Cela a conduit à des méthodes pour des reconstructions 3D clairsemées de scènes à partir de plusieurs images. Des progrès ont été réalisés sur le problème de la correspondance stéréo

dense et sur d'autres techniques stéréo multi-vues. Dans le même temps, des variations de coupe de graphe ont été utilisées pour résoudre la segmentation d'images. Cette décennie a également marqué la première fois que des techniques d'apprentissage statistique ont été utilisées dans la pratique pour reconnaître les visages dans les images .Vers la fin des années 1990, un changement important s'est produit avec l'interaction accrue entre les domaines de l'infographie et de la vision par ordinateur. Cela comprenait le rendu basé sur l'image, le morphing d'image, l'interpolation de vue, l'assemblage d'images panoramiques et le premier rendu de champ lumineux[29,30,31,32].

Des travaux récents ont vu la résurgence des méthodes basées sur les fonctionnalités, utilisées en conjonction avec des techniques d'apprentissage automatique et des cadres d'optimisation complexes. L'avancement des techniques d'apprentissage en profondeur a donné un nouveau souffle au domaine de la vision par ordinateur. La précision des algorithmes d'apprentissage en profondeur sur plusieurs ensembles de données de vision par ordinateur de référence pour des tâches allant de la classification, la segmentation et le flux optique a dépassé les méthodes antérieures[29,30,31].

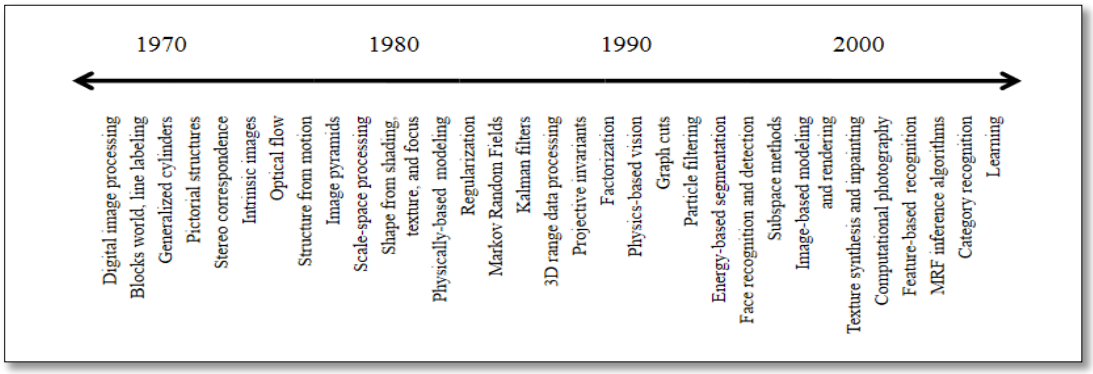

Figure 2. 2. Une chronologie approximative de certains des sujets de recherche les plus actifs en vision par ordinateur.

2.4 Reconnaissance des formes (RF)

2.4.1 Définition

La reconnaissance des formes ou bien reconnaissance de motifs est le processus de reconnaissance des formes à l'aide d'un algorithme d'apprentissage automatique. La reconnaissance de formes peut être définie comme la classification de données basée sur des connaissances déjà acquises ou sur des informations statistiques extraites de formes et/ou de leur représentation. L'un des aspects importants de la reconnaissance de formes est son potentiel d'application. Pour construire un bon système de vision par ordinateur, il faut une connaissance approfondie de la méthodologie de classification. Parfois, c'est même la partie la

plus importante du système de vision par ordinateur, comme dans le cas de la classification d'images, pour laquelle les réseaux neuronaux profonds ont produit la meilleure précision de classification jusqu'à présent[29,31,32].

2.4.2 Méthodes

Il y a plusieurs méthodes pour la reconnaissance de motifs, on a expliqué quelques dans chapitre de Machine learning. On mention quelques :

- Réseaux de neurones [12,14,18,33]
    a. Le Perceptron de ROSENBLATT [33]
    b. Adaline de Windrow–Hoff [33]
    c. Le modèle de KOHONEN [33]
    d. Le Perceptron multicouche [33]
    e. Le modèle de HOPFIELD [33]
- Méthodes à noyau [15,16,33]
- Les arbres de décision [2,12,33]
- K plus proches voisins [2,12,33]
- K plus proches voisins flous[33]
- Les modèles de Markov cachés[2,18,33]
- AdaBoost[33]

2.4.3 La reconnaissance de plusieurs objets dans une image

Une seule image peut être constituée d'un ou plusieurs objets [34]. Dans le cas où on désire détecter plusieurs objets dans une même image, on peut utiliser le procédé de reconnaissance multi objets représenté sur la figure 2.3 ci- dessous :

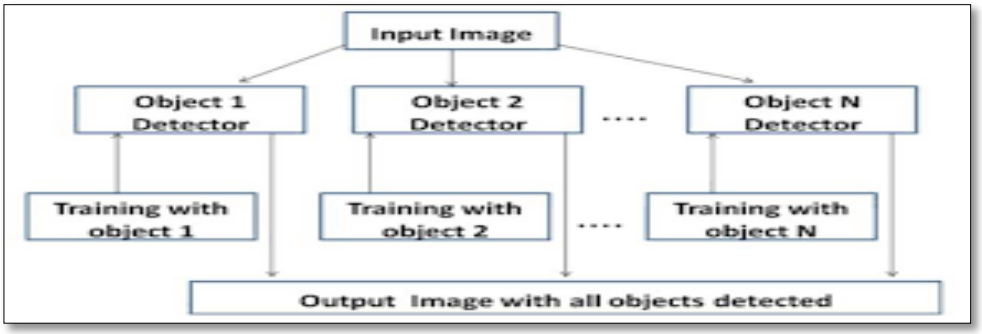

Figure 2. 3. La méthode de reconnaissance de multi objets dans une image[34].

2.4.4 Application Typique de la reconnaissance des formes

1. **Marketing**

La reconnaissance des formes est souvent utilisée pour classer les consommateurs selon les produits qu'ils sont susceptibles d'acheter. Elle est aussi utilisée par les sociétés de vente pour classer les clients selon qu'ils soient de bons ou mauvais payeurs, ou encore selon qu'ils vont oui ou non passer à la concurrence [34].

2. **Finances**

Les systèmes de reconnaissance des formes sont utilisés pour la détection de transactions bancaires frauduleuses ainsi que la prédiction des banqueroutes [34].

3. **Usinage**

La qualité des produits dépend souvent de paramétrisation correcte, et les relations exactes entre la qualité et les valeurs des paramètres n'est pas claire. Les systèmes de reconnaissance des formes sont utilisés pour classer les paramètres selon la qualité des produits qu'ils sont susceptibles de générer. Ils permettent ainsi de réduire le nombre d'essais ce qui fait gagner du temps et de l'argent [34].

4. **Energie**

Les systèmes de reconnaissance des formes sont utilisés pour prévoir la consommation électrique (réduite, normale, élevée), permettant ainsi aux clients de réduire si nécessaire leur consommation, et aux producteurs de mieux gérer leurs unités de production [34].

5. **Lecture automatisée**

Les systèmes de reconnaissance des formes permettent de numériser les anciens documents ainsi que les archives, non pas sous la forme d'images, mais plutôt sous une forme textuelle [34].

6. **Sécurité**

La reconnaissance vocale et rétinienne est un exemple d'applications typiques de la reconnaissance des formes pour l'authentification. La vérification des signatures est aussi très populaire [34].

2.4.5 Schéma général d'un système de Reconnaissance des Formes

La majorité des systèmes de RF ont le schéma de fonctionnement suivant :

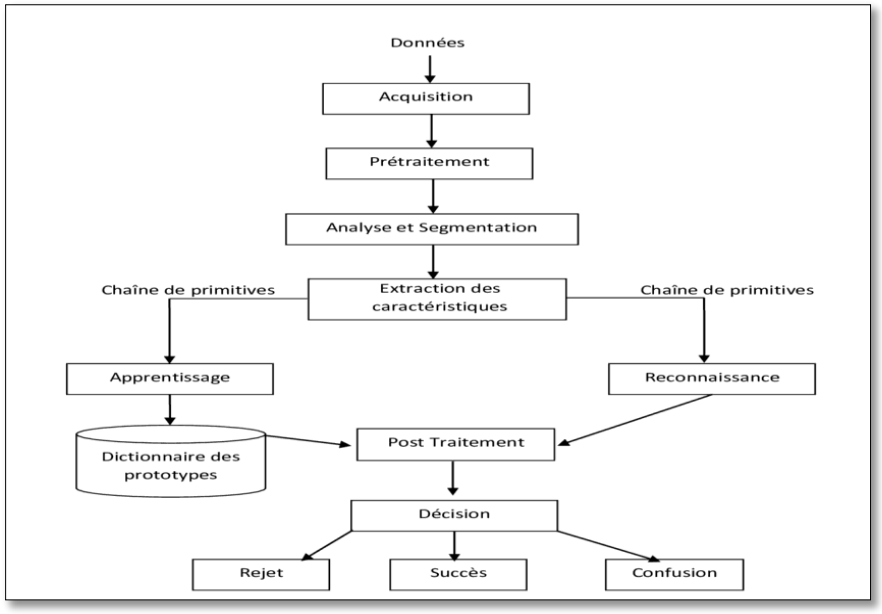

Figure 2. 4. Schéma général d'un système de reconnaissance des formes[33].

1. Prétraitement

Le prétraitement est une série d'opérations permettant de réduire la quantité de données à traiter et d'éliminer le bruit introduit par l'outil d'acquisition. Cette étape utilise un ensemble d'algorithmes et de filtres multiples, tels que la binarisation, la squelettisation, la détection des contours, le changement d'espace colorimétrique, la transformée de Fourier, la transformée en ondelettes, la transformée en cosinus discrète, le filtre autorégressif, la transformée géométrique et affine, le filtre médian, etc. Cette étape est très importante pour le succès du classificateur, et le retraitement utilisé varie d'une application à l'autre. [33].

2. Extraction des primitives (caractéristiques)

Les systèmes de reconnaissance utilisent des vecteurs de caractéristique pour pouvoir reconnaitre les formes ou chaque vecteur se comporte comme une signature d'une forme. La difficulté de cette étape est de déterminer quelles caractéristiques employées pour obtenir un bon résultat de classification.

L'objectif de l'extraction et de la sélection des caractéristiques est d'identifier les caractéristiques qui sont importantes dans la discrimination de classes de formes. Cela signifie de trouver, si c'est possible, les primitives les plus robustes, les plus pertinentes (afin de diminuer la taille du vecteur de primitives) et les moins coûteuses en calcul. Le résultat de cette phase d'extraction de primitives est une séquence de symboles qui représente la forme et qui traduit, soit la présence ou l'absence (cas booléen) ou la valeur associée (cas réel) a la caractéristique concernée. En fonction de l'objectif fixé et de la méthode d'extraction choisie, l'approche de l'extraction des primitives peut être systématique ou heuristique. La

modélisation et le codage conduisent à une approche systématique dans la mesure où l'objectif fixé est la détermination d'une représentation complète de la forme, même de façon approximative. Dans la modélisation, les primitives sont obtenues a posteriori, par le résultat de l'approximation, en ce qui concerne le codage, les catégories de primitives sont définies apriori. Un test, qui est par exemple réalisé à l'aide d'une sonde, permet de valider la présence de chacune des primitives sur l'ensemble de la forme. Le paramétrage conduit plutôt à une approche heuristique. Dans ce cas, on ne cherche pas nécessairement une représentation complète, mais seulement des indices significatifs. De la même façon que dans le cas du codage, ces indices sont des primitives définis a priori. L'une des raisons pour lesquelles cette étape pose un problème est que la plupart des techniques d'extraction s'accompagnent d'une perte d'information irrémédiable. De ce fait, il faut effectuer un compromis entre quantité et qualité de l'information, mais peu d'études théoriques sont faites sur ce sujet où l'intuition prédomine. Pour un problème de classification donné, la principale qualité recherchée pour un ensemble de caractéristiques est sa faculté de rassembler les objets appartenant à une même classe dans une même partition de l'espace de représentation, tout en éloignant autant que possible des autres. Cette qualité est communément appelée pouvoir discriminant de l'ensemble de caractéristiques[33].

3. Apprentissage et Classification

La classification est l'élaboration d'une règle de décision qui transforme les attributs caractérisant les formes en appartenance à une classe (passage de l'espace de codage, vers l'espace de décision). Avant qu'un modèle de décision ne soit intégré dans un système de reconnaissance des formes, il faut avoir procédé auparavant à deux étapes : l'étape d'apprentissage et l'étape de test. L'étape d'apprentissage consiste à caractériser les classes de formes de manière à bien distinguer les familles homogènes de formes. C'est une étape clé dans le système de reconnaissance. On distingue deux types d'apprentissages : apprentissage supervisé, apprentissage non supervisé et apprentissage semi-supervisé.

Dans le cas de l'apprentissage supervisé, un échantillon représentatif de l'ensemble des formes à reconnaitre est fourni au module d'apprentissage. Chaque forme est étiquetée par un opérateur appelé professeur, cette étiquette permet d'indiquer au module d'apprentissage la classe dans laquelle le professeur souhaite que la forme soit rangée. Cette phase d'apprentissage consiste à analyser les ressemblances entre les éléments d'une même classe et les dissemblances entre les éléments de classes différentes pour en déduire la meilleure partition de l'espace des représentations. Les paramètres décrivant cette partition sont stockés

dans une table d'apprentissage à laquelle le module de décision se référera ensuite pour classer les formes qui lui sont présentées. Dans le cas de l'apprentissage non supervisé, on fournit au système de reconnaissance un grand nombre de formes non étiquetées. L'étape de la classification va se charger d'identifier automatiquement les formes appartenant à une même classe. Le troisième type concerne les méthodes semi-supervisées qui utilisent les informations connues, c'est-à-dire les formes et classes connues, pour estimer les caractéristiques des classes et leurs fonctions d'appartenances tout en utilisant également l'apprentissage non supervisé pour détecter les nouvelles classes et apprendre leurs fonctions d'appartenance. L'étape de test permet d'évaluer la performance du classifieur pour un apprentissage donné. C'est une étape importante, car elle peut mettre en cause le choix des primitives ou le choix de la méthode d'apprentissage. En effet, il est difficile de trouver a priori les primitives pertinentes et la méthode d'apprentissage la plus adaptée au problème posé. D'où l'utilité de procéder par itérations successives. Ces itérations consistent à extraire des primitives jugées utiles au problème de reconnaissance à résoudre et à tester la performance du système avec cet ensemble de primitives. Au fur et à mesure que les performances du système souhaitées ne sont pas atteintes, alors il suffit de trouver à nouveau une nouvelle famille de primitives ou de combiner les primitives extraites avec de nouvelles primitives. Ces phases d'apprentissage et de test sont réalisées préalablement à l'intégration du module de décision dans le système de reconnaissance. Dans tous les cas, on peut permettre au système de reconnaissance d'itérer les phases d'apprentissage et de test tant qu'on n'a pas atteint les performances désirées. Le calcul de cette performance est le résultat du classifieur utilisé. Pour construire un classifieur, il existe trois approches :structurelle, statistique et hybride [33].

2.4.6 La vidéo

Le mot vidéo vient du latin vidéo qui signifie « je vois ». C'est l'apocope de vidéophonie ou vidéogramme. Les caméras vidéo créant des images destinées à la consommation humaine enregistrent des séquences d'images à une vitesse de 30 par seconde, permettant une représentation du mouvement de l'objet au fil du temps en plus des caractéristiques spatiales représentées dans des images individuelles d'images. Pour assurer une perception humaine fluide, 60 demi-images par seconde sont utilisées : ces demi-images sont toutes des rangées d'images impaires suivies de toutes les rangées d'images paires en alternance. Un signal audio est également codé. Les caméras vidéo créant des images pour la consommation des machines

peuvent enregistrer des images à n'importe quelle vitesse pratique et n'ont pas besoin d'utiliser la technique des demi-images [32].

- **Caractéristiques de la vidéo**

Les caractéristiques de la vidéo sont :

- Une image matricielle (de l'anglais bitmap : une image constituée d'une matrice de points colorés. C'est-à-dire, constituée d'un tableau, d'une grille, où chaque case possède une couleur qui lui est propre et est considérée comme un point [35].
- Balayage progressif: Le principe du balayage progressif est d'afficher la totalité de l'image en une seule fois, ce qui l'oppose au balayage entrelacé, dans lequel les lignes impaires de l'image sont affichées, suivies ensuite des lignes paires [35].
- Résolution : La définition au standard PAL peut atteindre 720 × 576 lignes (DVD). Il est différent de la norme de télédiffusion qui lui est associée (exemple: CCIR), laquelle définit la modulation des signaux [35].
- Fréquence réseau électrique / Image par seconde : au début du cinéma avec les films muets il n'y avait que 16i/s puis 18 i/s puis arrive le son. L'introduction du cinéma parlant, dans les années 1920, ne permettait plus de tolérer des variations images sons: nos oreilles sont sensibles aux modifications de la fréquence audio. Ainsila cadence de 24 images par seconde devint la norme, associée à la fameuse pellicule 35mm. Mais elle aussi est petit à petit en train de changer [35].
- Un codec (acronyme de codage-décodage) est un algorithme de compression / décompression d'un signal audiovisuel numérique [35].

2.5 Traitement d'image

Le traitement d'image est une méthode pour effectuer certaines opérations sur une image, afin d'obtenir une image améliorée ou d'en extraire des informations utiles. C'est un type de traitement du signal dans lequel l'entrée est une image et la sortie peut être une image ou des caractéristiques/caractéristiques associées à cette image. De nos jours, le traitement d'images fait partie des technologies en plein essor. Il constitue également un domaine de recherche central dans les disciplines de l'ingénierie et de l'informatique. Le traitement des images comprend essentiellement les trois étapes suivantes : Importation de l'image via des outils d'acquisition d'images, analyser et manipuler l'image, sortie dans laquelle le résultat peut être modifié, image ou rapport basé sur l'analyse d'image [32]. Les données quantitatives liées à la représentation des images sont représentées dans le tableau suivant (Tableau 2.1) :

Tableau 2. 1. Données quantitatives des images.

1 bit	2 couleurs (noir & blanc)
4 bits	16 couleurs
8 bits	256 couleurs ou niveaux de gris
16 bits	65536 couleurs
24 bits	16 777 216 couleurs (vraies couleurs)
32 bits	4 294 967 296 couleurs

2.5.1 Définition de l'image

- **Une image analogique** est une image 2D $F(x, y)$ qui a une précision infinie dans les paramètres spatiaux x et y et une précision infinie en intensité à chaque point spatial (x, y) [32].
- **Une image numérique** est une image 2D $I\ [r, c]$ représentée par un réseau 2D discret d'échantillons d'intensité, dont chacun est représenté avec une précision limitée [32].
- **Une image en niveaux de gris** est une image numérique monochrome $I[r, c]$ avec une valeur d'intensité par pixel [32].
- **Une image multi spectrale** est une image 2D $M[x, y]$ qui a un vecteur de valeurs à chaque point spatial ou pixel. Si l'image est en fait une image couleur , alors le vecteur a 3 éléments [32].
- **Une image binaire** est une image numérique avec toutes les valeurs de pixel 0 ou 1 [32].
- **Une image étiquetée** est une image numérique $L[r, c]$ dont les valeurs de pixels sont des symboles d'un alphabet fini. La valeur de symbole d'un pixel dénote le résultat d'une décision prise pour ce pixel. Les concepts associés sont l'image thématique et l'image pseudo-colorée [32].

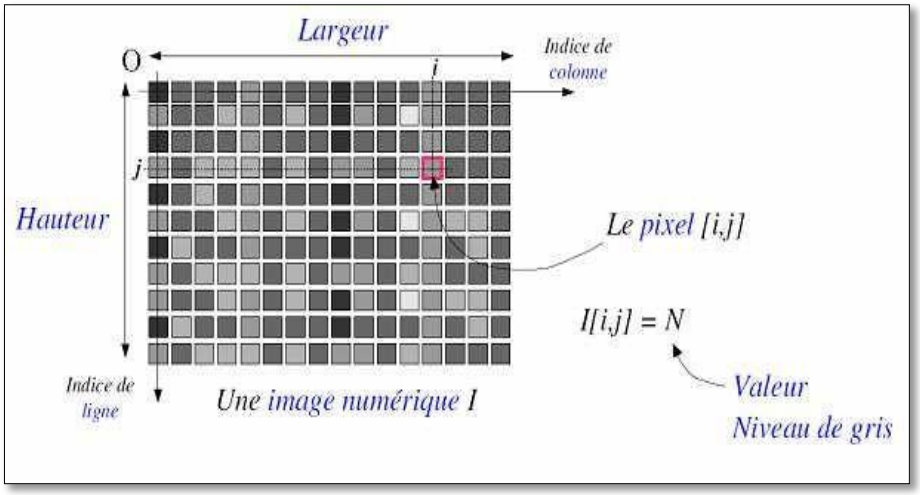

Figure 2. 5. Exemple de réseau de pixels [34].

Pixel : Le terme pixel est la contraction de l'anglais « Picture Element ». C'est le plus petit élément de l'image. Sortie de l'appareil photo une image est composée d'un nombre x de pixels. Un pixel a une couleur exprimée (codée) en langage binaire mathématique.

2.5.2 Acquisition d'une image

Il existe de nombreux appareils différents qui produisent des images numériques. Ils diffèrent par les phénomènes détectés ainsi que par leur conception électromécanique [32]. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales :

- Les caméras numériques,
- Et les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la caméra, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier [32].

2.5.3 Caractéristiques d'une image numérique

➢ **Dimension**

Une image informatique est une matrice (un tableau 2D) de pixels. La valeur de chaque pixel est proportionnelle à la luminosité du point correspondant de la scène ; sa valeur est,

bien entendu, généralement dérivée de la sortie d'un convertisseur A/N. La matrice de pixels, l'image, est généralement carrée et nous décrirons une image comme $N \times N$ pixels de m-bits où N est le nombre de points et m contrôle le nombre de valeurs de luminosité [31].

- **Résolution**

La transformée de Fourier d'une image tourne lorsque l'image source tourne. C'est normal puisque la décomposition en fréquence spatiale reflète l'orientation des caractéristiques au sein de l'image. En tant que telle, la dépendance à l'orientation est intégrée au processus de transformation de Fourier [31].

- **Bruit**

Parfois, les images sont « bruyantes ». Par exemple, la numérisation de photographies anciennes conduit souvent à des images dans lesquelles des pixels isolés se détachent de leur voisinage immédiat, un type de bruit également appelé bruit poivre et sel, car ces pixels ressorts apparaissent comme des grains de sel (lumineux) ou poivre (foncé) sur une image. Dans ce cas, nous pouvons prendre un filtre médian, un filtre qui remplace une valeur de pixel par la valeur médiane de son voisinage. Souvent, on choisit un voisinage 3x3, c'est-à-dire que la médiane est prise pour ces 9 valeurs seulement. De plus grands quartiers peuvent également être utilisés [29,30,31].

- **Histogramme**

L'histogramme d'intensité montre comment les niveaux de luminosité individuels sont occupés dans une image ; le contraste de l'image est mesuré par la plage de niveaux de luminosité. L'histogramme trace le nombre de pixels avec un niveau de luminosité particulier par rapport au niveau de luminosité. Pour les pixels 8 bits, la luminosité va de 0 (noir) à 255 (blanc). La figure 2.6 montre une image d'un œil et son histogramme. L'histogramme (Figure 2.6(b)) montre que tous les niveaux de gris ne sont pas utilisés et que les niveaux d'intensité les plus bas et les plus élevés sont rapprochés, reflétant un contraste modéré. L'histogramme a une région comprise entre 100 et 120 valeurs de luminosité, qui contient les parties sombres de l'image, telles que les cheveux (y compris les sourcils) et l'iris de l'œil. Les points les plus brillants concernent principalement la peau. Si l'image était globalement plus sombre, l'histogramme serait concentré vers le noir. Si l'image était plus lumineuse, mais avec un contraste plus faible, alors l'histogramme serait plus fin et concentré près des niveaux de luminosité les plus blancs [31].

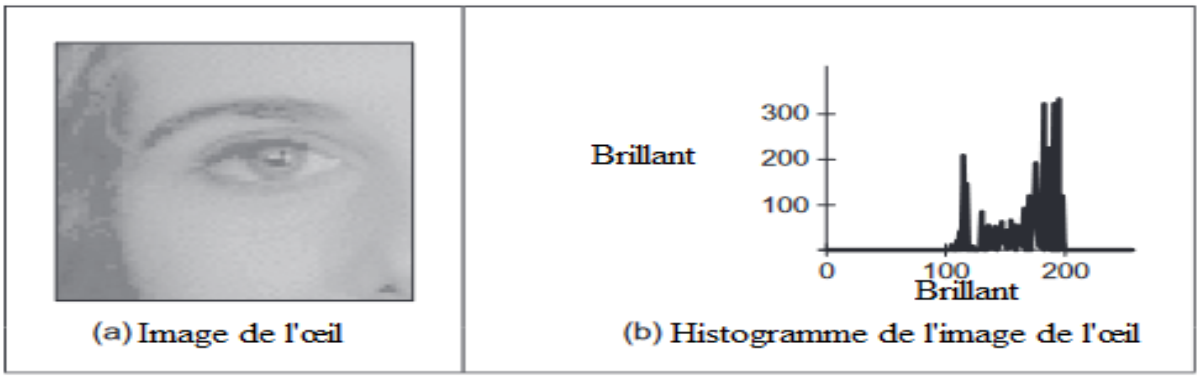

Figure 2. 6 Une image et son histogramme [31].

- **Luminance**

Les images ne peuvent exister sans lumière. Pour produire une image, la scène doit être éclairée par une ou plusieurs sources lumineuses. Les sources lumineuses peuvent généralement être divisées en sources lumineuses ponctuelles et surfaciques. Une source lumineuse ponctuelle provient d'un seul endroit dans l'espace (par exemple, une petite ampoule), potentiellement à l'infini (par exemple, le soleil). (Notez que pour certaines applications telles que la modélisation d'ombres douces (pénombres), le soleil peut devoir être traité comme une source lumineuse de zone.) En plus de son emplacement, une source lumineuse ponctuelle a une intensité et un spectre de couleurs, c'est-à-dire une distribution sur les longueurs d'onde $L(\lambda)$. L'intensité d'une source lumineuse diminue avec le carré de la distance entre la source et l'objet éclairé, car la même lumière est répartie sur une zone (sphérique) plus grande. Une source lumineuse peut également avoir une atténuation directionnelle (dépendance) [30].

- **Contraste**

Si une image présente un faible contraste, elle peut être manipulée pour afficher un contraste plus élevé en décalant ses valeurs de pixels selon une certaine distribution. Les manipulations sont généralement basées sur l'histogramme de l'image. Dans une image à faible contraste, l'histogramme montre un pic de chaque côté. Pendant le processus d'égalisation, ce pic d'histogramme est déplacé davantage vers le centre, sur la base d'une distribution d'histogramme d'entrée. Cette technique est parfois utilisée pour manipuler des images dans des collections afin d'afficher une distribution d'intensité à peu près égale, car cela permet d'améliorer les résultats d'apprentissage lors de la formation des systèmes de reconnaissance [29]. Le contraste C est défini par le rapport :

$$C = \frac{L_1 + L_2}{L_1 + L_2}$$

2.5.4 Système de traitement d'images

Un système de traitement numérique d'images est composé de:

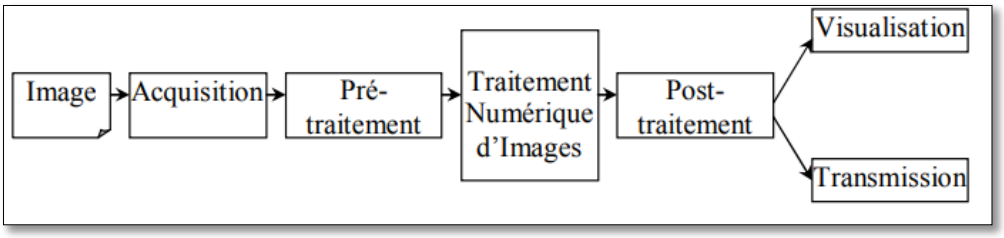

Figure 2. 7. Schéma d'un système de traitement d'images.

2.6 Segmentation des images

La segmentation est le partitionnement d'une scène en ses composants et elle la tâche de trouver des groupes de pixels qui « s'associent ». Par exemple, nous segmentons une scène de rue en ses objets et parties, tels que les voitures, les bâtiments, les piétons, les trottoirs, etc. Ou nous segmentons l'image de l'mélanome2 en deux régions, la peau saine et la lésion elle-même, la peau affectée. La segmentation sémantique peut être réalisée avec ou sans supervision :

- Supervisé : dans ce format, les images d'entraînement se présentent sous forme de paires d'images. Ce processus d'apprentissage est donc analogue à la tâche de classification, c'est pourquoi nous l'appelons supervisé, terme qui provient du domaine de l'apprentissage automatique. Dans un sens plus large, on peut aussi appeler cette segmentation top-down. car l'information est d'abord intégrée de manière hiérarchique, et dans un second temps dépliée à la taille de l'image, à savoir une carte de segmentation.
- Non supervisé : dans ce format, aucun ou presque aucun apprentissage n'a lieu - contrairement à la segmentation supervisée, c'est pourquoi on l'appelle aussi segmentation ascendante.

2.6.1 Types de Segmentation

a) Segmentation de texte en lignes

En général, le sous-titres ou le texte graphique ce compose de plusieurs lignes. En reconnaissance de texte, le texte se segmentant en lignes séparées. Il existe certaines méthodes utilisées à cet effet, telles que la projection horizontale.

b) Segmentation de lignes en caractères

Il s'agit ici de la segmentation de lignes en caractères individuels. Les points de segmentation sont identifiés à la fin d'un caractère et au début de la suivante.

2.6.2 Les principes de la segmentation

De nombreux travaux ont été réalisés sur ce sujet, dans des domaines aussi variés que le domaine médical, militaire, industriel, géophysique, … C'est toujours un sujet d'actualité et un problème qui reste ouvert où l'on retrouve de très nombreuses approches [34]:

- La segmentation par régions.
- La segmentation par seuillage.
- La segmentation par contours.
- La Transformée de Hough.
- La segmentation par étiquetage en composantes connexes.
- La segmentation par LPE.

Toutes ces approches visent à l'extraction des caractéristiques. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. Il n'existe pas d'algorithme universel de segmentation à chaque type d'images correspond une approche spécifique. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation [34].

2.7 Détection d'objet

2.7.1 Définition

La détection d'objets [29,30,31] est une technologie informatique liée à la vision par ordinateur et au traitement d'images qui traite de la détection d'instances d'objets sémantiques d'une certaine classe (comme les humains, les bâtiments ou les voitures) dans les images et vidéos numériques. Les domaines bien documentés de la détection d'objets comprennent la détection des visages et la détection des piétons. La détection d'objets a des applications dans de nombreux domaines de la vision par ordinateur, y compris la récupération d'images et la vidéosurveillance.

2.7.2 Méthodes

Les méthodes de détection d'objets relèvent généralement d'approches basées sur un réseau neuronal ou non neuronales. Pour les approches non neuronales, il devient nécessaire

de définir d'abord les caractéristiques en utilisant l'une des méthodes ci-dessous, puis en utilisant une technique telle que la machine à vecteurs de support (SVM) pour effectuer la classification. D'autre part, les techniques neuronales sont capables de détecter des objets de bout en bout sans définir spécifiquement de caractéristiques et sont généralement basées sur des réseaux de neurones convolutifs (CNN) [29,30,31], on mention quelques méthodes:

- Approches non neuronales:
  - Cadre de détection d'objets Viola-Jones basé sur les fonctionnalités de Haar
  - Scale-invariant feature transform (SIFT)
  - Histogramme des caractéristiques des gradients orientés (HOG)

- Approches des réseaux de neurones:
  - Propositions de région (R-CNN, Fast R-CNN, Faster R-CNN, cascade R-CNN)
  - Single Shot MultiBox Detector (SSD)
  - You Only Look Once (YOLO)
  - Single-Shot Refinement Neural Network for Object Detection (RefineDet)
  - Retina-Net
  - Réseaux convolutifs déformables

2.8 Travaux connexes

Cette partie est consacrée pour l'analyse des résultats de différents articles qui s'intéressant sur la détection de texte qui utilisent différentes méthodes de l'apprentissage automatique et l'apprentissage profonds.

a) Le travail de Wang et al

Dans la recherche de Wang et al [36] le système réalisé propose une détection de texte en utilisant le modèle PSENet pour la classification. Pour l'apprentissage du modèle, 3 base de données (ICDAR17, ICDAR15, ICDAR13) ont été utilisées ce qui détecte des mots. Le modèle est basé sur architecture FPN et colonne vertébrale ResNet.

b) Le travail de Liu et al

Dans la recherche de Liu et al [36] le système réalisé propose une détection de texte en utilisant le modèle PMTD pour la classification. Pour l'apprentissage du modèle, 3 base de

données (ICDAR17, ICDAR15, ICDAR13) ont été utilisées ce qui détecte des mots. Le modèle est basé sur architecture Mask-RCNN et colonne vertébrale ResNet-50.

c) Le travail de Zhou et al

Dans la travaille de Zhou et al [36] le système réalisé propose une détection de texte en utilisant le modèle EAST pour la classification. Pour l'apprentissage du modèle, 2 base de données (ICDAR15, COCO ou M500) ont été utilisées ce qui détecte des mots ou bien ligne de texte. Le modèle est basé sur architecture FCN et colonne vertébrale VGG-16.

1.5.10. Comparaison entres les travaux réalisés

Les travaux cités dans la partie précédents concernant le domaine de la détection de texte utilisent des différentes bases de données, la comparaison entre eux est faite en fonction des résultats obtenus et le nombre d'échantillons utilisés. Le but est de nous aider à proposer une approche en appliquant des nouvelles techniques sur une base de données plus riches avec des nombres, des lettres (majuscules et minuscules) et les marques de ponctuation. Le tableau suivant donne un résumé des résultats des recherches mentionnés Où P est pourcentage de précision, R est pourcentage de rappel et F est pourcentage de mesure-F :

Tableau 2. 2.Les résultats des travaux connexes.

Méthode	**ICDAR 2013**			**ICDAR 2015**			**COCO-Text**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
PSENet	81.04%	62.46%	70.55%	84.69%	77.51%	80.94%	60.58%	49.39%	54.42%
EAST	84.86%	74.24%	79.20%	84.64%	77.22%	80.76%	55.48%	32.89%	41.30%
PMTD	92.49%	83.29%	87.65%	92.37%	84.59%	88.31%	61.37%	59.46%	60.40%

1.6. Conclusion

La vision par ordinateur met en œuvre plusieurs étapes : segmentation des objets (analyse d'images), extraction de caractéristiques (géométrie, invariants, …), classification (supervisée ou non, méthodes probabilistes, statistiques, …). Les applications sont très variées et nécessitent des connaissances expertes du domaine d'application. Les méthodes sont nombreuses mais les principes de base sont assez stables.

Dans le prochain chapitre, Nous expliquant la conception générale de notre système d'extraction de texte, ainsi les approches utilisées et on montre les résultats obtenus.

Chapitre03 : Conception et implémentation.

3.1 Introduction

Ce dernier chapitre est consacré à la conception et la mise en place de notre projet qui permettra d'identifier des extractions texte dans des images en utilisant modèle de détection d'objets qui basé sur méthode prédire de la boite englobant.Dans ce chapitre nous allons décrire également les différentes parties de notre système, les détails relatifs à chaque phase ainsi que leurs interactions sont présentés dans les sous-sections suivantes.

3.2 Conception générale de notre système

Le processus d'extraction de texte dans des images se compose deux phases qui sont la phase d'apprentissage et la phase de test comme le montre la **figure 3.1**, on va expliquer les deux phases :

1. **La phase d'apprentissage :**

En propagation vers l'avant après réception de l'image à traiter en entrée, Notre système divise l'image en cellules pour calculer la congruence du prédire de les boites englobant avec le vérité terrain (les coordonnée de les objets), ceci est calculé par la fonction de perte, si le résultat est inférieur au seuil de l'arrêt de l'entraînement(Vrai), notre système passe à la phase de test sinon(faux) il se propage en arrière puis pente à propagation vers l'avant et répéter les étapes .

2. **La phase de test :**

A cette phase, notre système enregistre le modèle, il y a plusieurs boite où chaque boite englobant a six paramètres (x, y, h, w, C, P) ;

- x, y : Les coordonnées du centre du boite englobant
- h, w: Hauteur et largeur de la boite englobant
- C : Classification de l'objet détecté, on a une seule classe dans notre modèle
- P : La possibilité de l'existence d'un objet ou non

Generalized Intersection over Union « GIoU » maximise la zone de chevauchement de la vérité terrain et de la boîte englobant prédite, NMS réduit les mauvaises boites puis il nous donne le résultat de détection de texte.

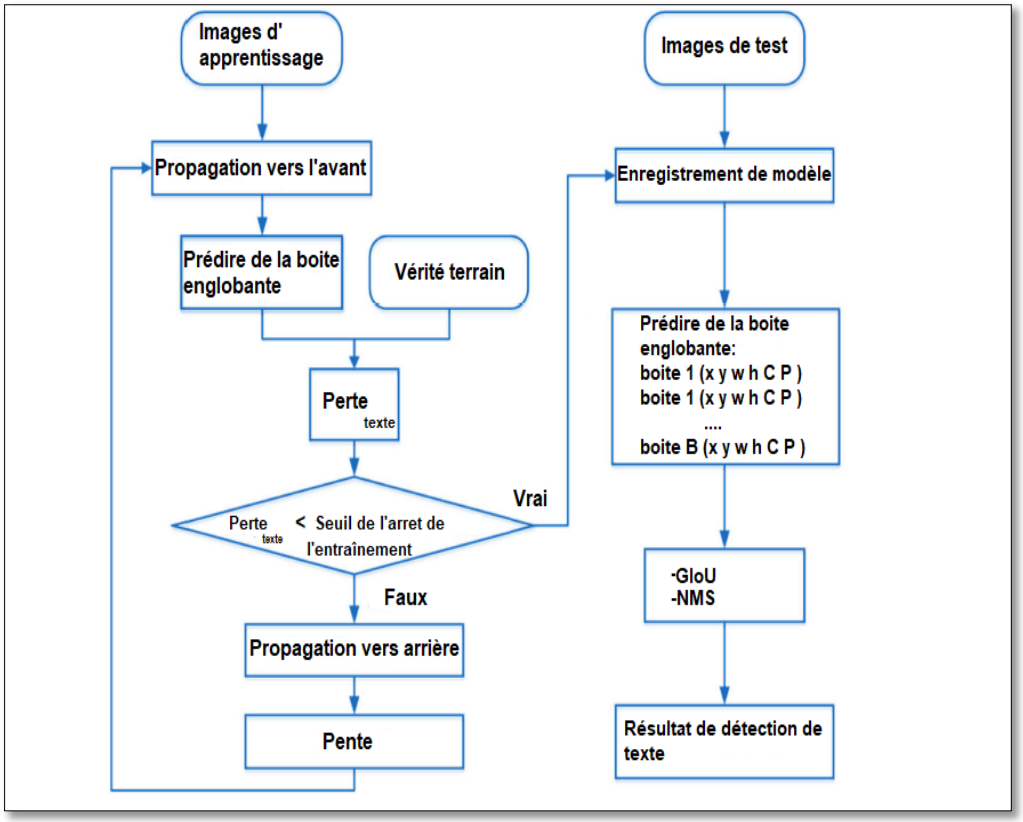

Figure 3. 1 Notre système d'extraction de texte.

3.3 Conception détaillé

Yolo est un modèle de détection d'objets open source rapide et compact basé sur CNN où la dernière version de Yolo est Yolov5, alors premièrement on va expliquer CNN et d'autre quelques notions qui ont une relation avec l'architecture de yolov5. De nombreux algorithmes de détection de cibles différents ont également obtenu des résultats remarquables. Depuis lors, formant deux concepts de détection d'objets architecturaux: One-stage detector and Two-stage detector test comme le montre la figure 3.2, le concept de Yolov5 est One-stage detector.

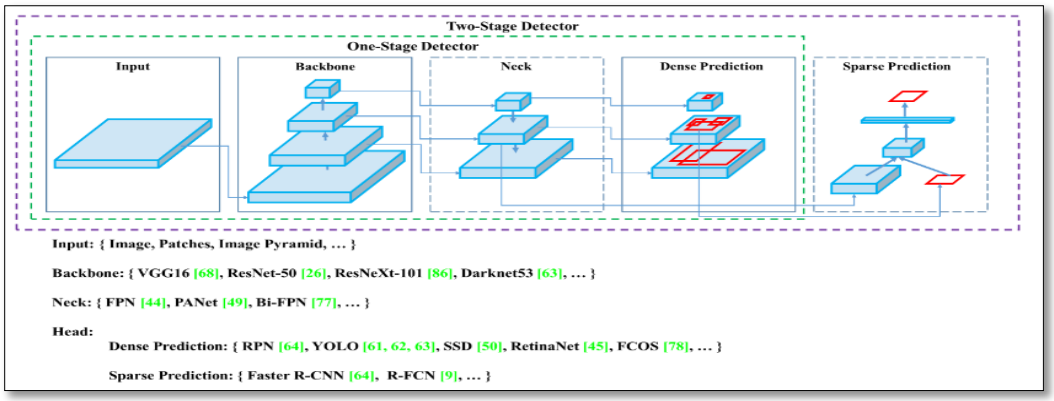

Figure 3. 2 Détecteur d'objets[23].

3.3.1 Réseaux de neurones convolutifs(CNN)

a) Couche de convolution

Un réseau de neurones convolutif [12,14,18] est un réseau de neurones qui utilise une opération mathématique qui s'appelle convolution ou produit de convolution. Il s'agit d'une opération linéaire. Chaque réseau de neurones convolutif contient au moins une couche de convolution. Une couche de convolution est caractérisée par :

- Les dimensions des noyaux de convolution, généralement une convolution à une dimension égale à 2 avec des noyaux carrés.
- Le nombre des filtres de convolution C, c'est le nombre de cartes d'activations, ou cartes de caractéristiques, en sortie de la couche. Ces cartes sont représentées sous la forme de tenseurs de dimension 3 (H, W, C) avec H la hauteur des cartes, W la largeur et C le nombre de canaux.
- Le pas de convolution –ou stride-s. C'est le pas de décalage du noyau de convolution à chaque calcul.
- Le padding p. C'est le paramètre permettant de dépasser la taille de l'image pour appliquer la convolution en ajoutant des pixels autour de l'image.

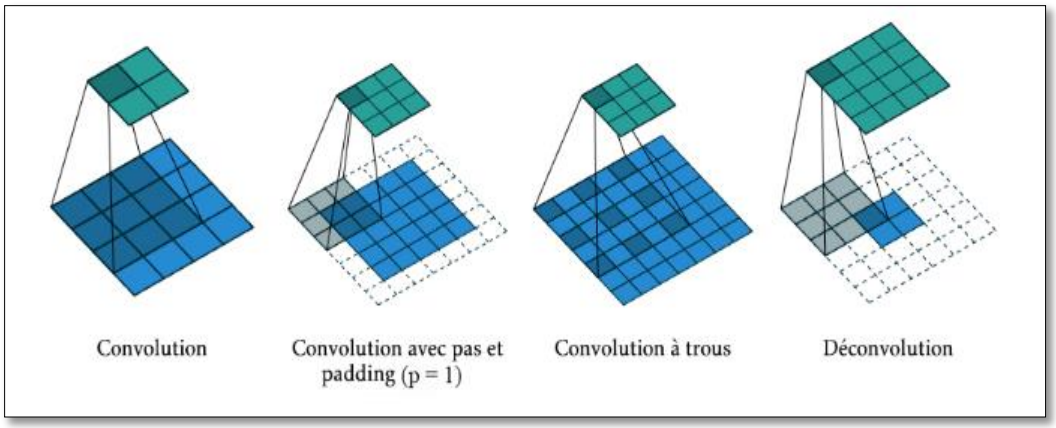

Figure 3. 3 Différents types de convolution [12].

b) Couche d'échantillonnage (Pooling)

Semblable à la couche de convolution, la couche d'échantillonnage est chargée de réduire la taille spatiale des cartes de caractéristiques, mais elle conserve les informations les plus importantes. Il existe différents types d'échantillonnage dont l'échantillonnage maximum-ou Max Pooling-, l'échantillonnage moyen ou Average Pooling[12,14].

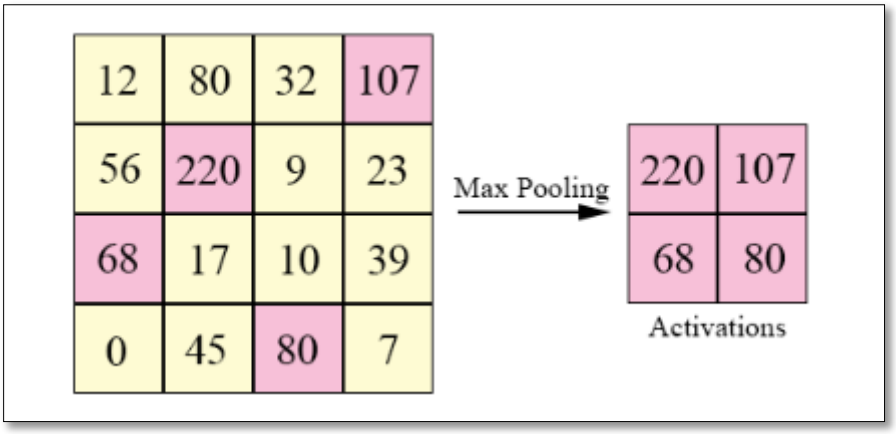

Figure 3. 4 Exemple d'une opération de pooling de taille 2×2[12].

c) Couche complètement connectée

La couche complètement connectée est un Perceptron multi-couches traditionnel qui utilise une fonction d'activation (par exemple softmax) sur le vecteur de sortie afin d'ajouter la non-linéarité. Le terme « complètement connecté » implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante[12,14]. Leurs activations peuvent donc être calculées avec une multiplication matricielle suivie d'un offset de biais. L'équation de softmax écrit sous la forme :

$$softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

3.3.2 Architecture de Yolov5

Généralement architecture de Yolo se compose de trois pièces principales :

- **Colonne vertébrale (Backbone)** : Un réseau de neurones convolutifs qui agrège et forme des caractéristiques d'image à différentes granularités.
- **Cou (Neck)** : Une série de couches pour mélanger et combiner les caractéristiques de l'image pour les transmettre à la prédiction.
- **Tête (Head)** : Consomme les caractéristiques du cou et effectue des étapes de prédiction de boîte et de classe.

L'architecture de modèle YOLOv5 se composé comme le montre la **figure 3.5**:

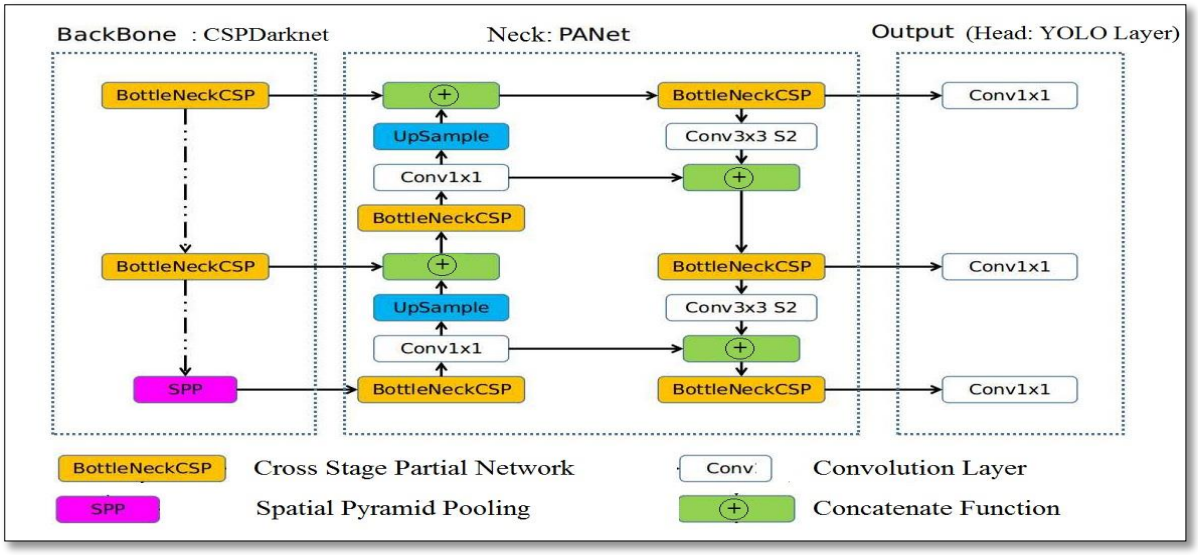

Figure 3. 5 L'architecture de modèle YOLOv5.

3.3.3Colonne vertébrale (Backbone)

a) Cross Stage Partial Network

Les modèles CSP (Cross Stage Partial Darkent) est dérivé de l'architecture DenseNet qui utilise l'entrée précédente et la concatène avec l'entrée actuelle avant de passer à la couche dense [24].DenseNet a été conçu pour connecter des couches dans un réseau de neurones très profond dans le but d'atténuer les problèmes de gradient de disparition [25].

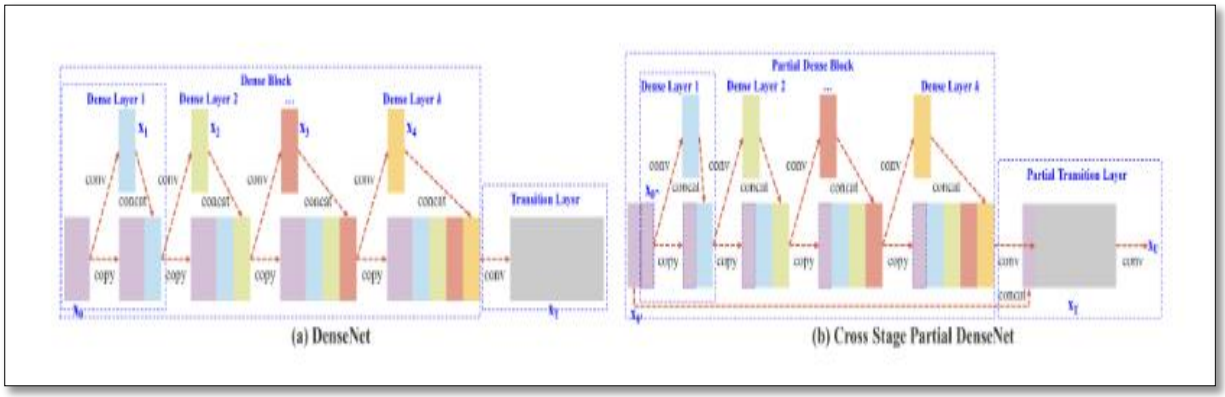

Figure 3. 6 DenseNet : Illustrations de (a) DenseNet et (b) de Cross Stage Partial DenseNet (CSPDenseNet). CSPNet sépare la carte des caractéristiques de la couche de base en deux parties, une partie passera par un bloc dense et une couche de transition ; l'autre partie est ensuite combinée avec la carte des caractéristiques transmise à l'étape suivante[25].

DenseNet : Chaque étape d'un DenseNet contient un bloc dense et une couche de transition, et chaque bloc dense est composé de k couches denses. La sortie de la i^{th} couche dense sera concaténée avec l'entrée de la couche dense i^{th}, et le résultat concaténé deviendra l'entrée de la $(i+1)^{th}$couche dense. Les équations montrant le mécanisme mentionné ci-dessus peuvent être exprimées comme suit [25]:

$$x_1 = w_1 * \quad x_0$$

$$x_2 = w_2 * [\quad x_0, x1\,] \qquad (3.1)$$

$$x_k = w_k * [\quad x_0\,, x1, \ldots\ldots,\ x_{k-1}]$$

Où $*$ représente l'opérateur de convolution, et $[\quad x_0\,, x1, \ldots\ldots]$ signifie concaténer $x_0\,, x1, \ldots$.et w_i et x_i sont respectivement les poids et la sortie de la i^{th} couche dense. Si l'on utilise un algorithme de propagation vers l'avant pour mettre à jour les poids, les équations de mise à jour des poids peuvent s'écrire comme [25]:

$$w_1 = f(\,w_1,\quad g_0)$$

$$w_2 = f(\,w_2,\quad g_0,\quad g_1) \qquad (3.2)$$

$$w_k = f(\,w_k,\quad g_0,\quad g_1 \ldots\ldots,\quad g_{k-1})$$

Où f est la fonction de mise à jour du poids, et g_i représente le gradient propagé à la i^{th} couche dense. Nous pouvons constater qu'une grande quantité d'informations de gradient est réutilisée pour mettre à jour les poids de différentes couches denses. Il en résultera que les différentes couches denses apprendront à plusieurs reprises les informations de gradient copiées.

Cross Stage Partial Dense Net : L'architecture à une étape du CSPDenseNet proposé est illustrée à la figure 3.6 (b). Une étape de CSPDenseNet est composée d'un bloc dense partiel et d'une couche de transition partielle. Dans un bloc dense partiel, les cartes de caractéristiques de la couche de base dans une étape sont divisées en deux parties via le canal $x_0 = [\quad x_0{}',\quad x_0{}'']$. Entre $x_0'\ et\quad x_0''$, le premier est directement lié à la fin de l'étage, et le second va traverser un bloc dense. Toutes les étapes impliquées dans une couche de transition partielle sont les suivantes : Premièrement, la sortie des couches denses, $[\,x_0''\,,\, x_1 \ldots\ldots\, x_k]$subira une couche de transition. Deuxièmement, la sortie de cette couche de transition, $x_T$subira, sera concaténée avec x_0'' subira et subira une autre couche de transition, puis générera la sortie x_U subira. Les équations de la réussite et de la mise à jour du poids de CSPDenseNet sont présentées dans les équations 3.3 et 3.4, respectivement [25].

$$x_K = w_K * [\ x_{0'},\ x_1, \ldots\ldots,\ x_{k-1}]$$

$$x_T = w_T * [\ x_0,\ x_1, \ldots\ldots,\ x_k\ ] \qquad (3.3)$$

$$x_U = w_U * [\ x_{0'},\ x_T]$$

$$w_k = f(w_k,\ g_0,\ g_1 \ldots\ldots,\ g_{k-1})$$

$$w_T = f(w_T,\ g_0,\ g_1 \ldots\ldots,\ g_k) \qquad (3.4)$$

$$w_U = f(w_U,\ g_0,\ g_T)$$

Le CSP maintient les caractéristiques par propagation, encourage le réseau à réutiliser les caractéristiques et réduit le nombre de paramètres réseau, aide à préserver les caractéristiques à grain fin pour les transmettre plus efficacement aux couches plus profondes. Considérant qu'une augmentation excessive des couches convolutives densément connectées peut conduire à une diminution de la vitesse de détection, seul le dernier bloc convolutif qui peut extraire les caractéristiques sémantiques les plus riches du réseau dorsal Darknet-53 est amélioré pour devenir un bloc dense [25].

b) CSPDarknet

Il s'agit d'un réseau de neurones convolutifs et d'une dorsale pour la détection d'objets qui utilise DarkNet-53. Il utilise une stratégie CSPNet pour partitionner la carte des caractéristiques de la couche de base en deux parties, puis les fusionne via une hiérarchie à plusieurs niveaux. L'utilisation d'une stratégie de division et de fusion permet un plus grand flux de gradient à travers le réseau. Les modèles utilisent les connexions CSP avec le Darknet-53 (Figure 3.7) ci-dessous comme épine dorsale dans l'extraction de fonctionnalités.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3. 7 Darknet-53

c) **Spatial Pyramid Pooling**

Un bloc de mise en commun de pyramide spatiale « **SPP »** avec trois couches de mise en commun maximum illustré sur la figure 3.8 est introduit entre le bloc DC et la couche de détection d'objet dans le réseau. La convolution 1×1 est utilisée pour réduire le nombre de cartes de caractéristiques en entrée de 1024 à 512. Après cela, les cartes de caractéristiques sont regroupées à différentes échelles ; $size_{pool} \times size_{pool}$ représente la taille des fenêtres coulissantes, $size_{fmap} \times size_{fmap}$ représente la taille des cartes de caractéristiques, puis $size_{pool} = [\, size_{fmap} / n_i \,]$ [21].

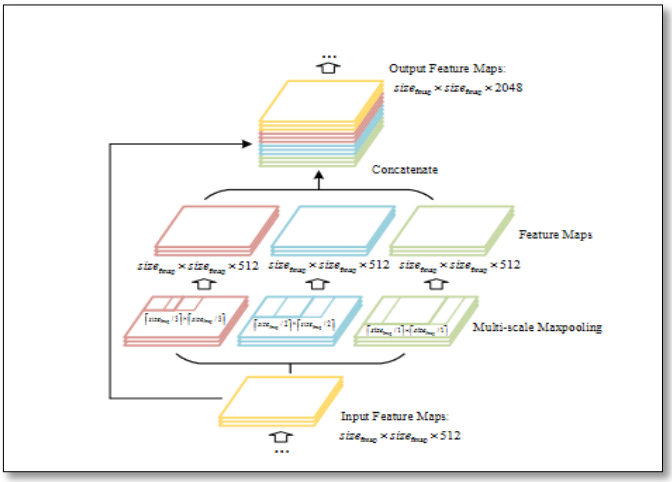

Figure 3. 8 Améliorée SPP [21].

3.3.4 Cou (Neck)

Yolov5 utilise PANet comme **Cou(Neck)** pour agréger les fonctionnalités et il est basé sur frameworks FPN, tout en améliorant la diffusion de l'information.

L'architecture FPN a mis en œuvre un chemin descendant pour transférer les caractéristiques sémantiques (de la couche de haut niveau), puis les concaténer en caractéristiques à grain fin (de la couche de bas niveau dans la dorsale) pour prédire les petits objets dans le détecteur à grande échelle figure 3.9.

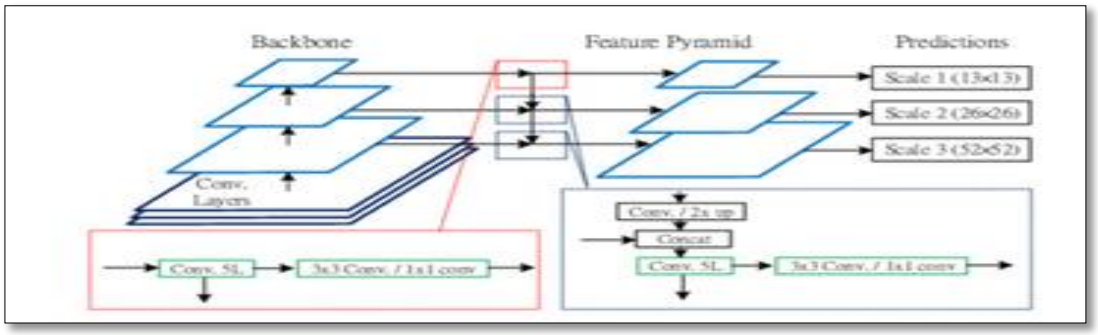

Figure 3. 9 Architecture de FPN de Yolov3 [26]

Le réseau d'agrégation de chemins « PAN » est une version avancée de FPN et il fonctionne de la même manière que les FPN, mais ils ont ajouté un chemin d'augmentation ascendant comme le montre la figure 3.10(b) (ci-dessous) afin que les réponses de texture fortes à partir de bas niveaux puissent être directement fusionnées avec réponses sémantiquement riches présentes dans N5 en utilisant un chemin de raccourci[27].

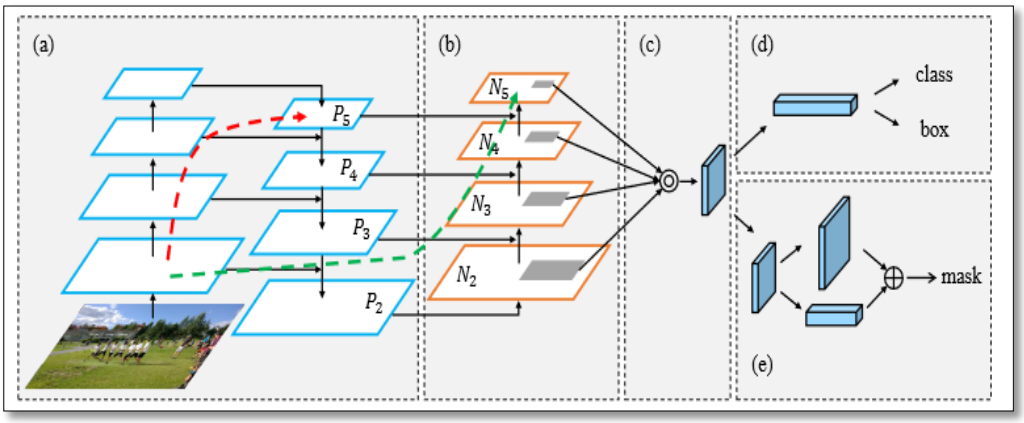

Figure 3. 10 PAN : a) Colonne vertébrale FPN. (b) Augmentation de la trajectoire ascendante. (c) Mise en commun des fonctionnalités adaptatives. (d) Branche-boîte. (e) Fusion entièrement connectée [27].

3.3.5 Tête (Head)

La tête est principalement utilisée dans la partie de détection finale. Il applique une boîte d'ancrage sur la carte des caractéristiques et génère le vecteur de sortie final avec la probabilité de classe, le score de l'objet et la boîte englobant.

3.3.6 La boîte d'ancrage (Anchor box)

La boîte d'ancrage est une liste de boîtes prédéfinies qui correspondent le mieux aux objets souhaités. Les boîtes englobantes n'ont pas seulement été prédites en fonction des boîtes de vérité terrain, mais également des boîtes d'ancrage prédéfinies, en Yolov5 la boîte d'ancrage est automatiquement apprise en fonction des données d'entraînement.

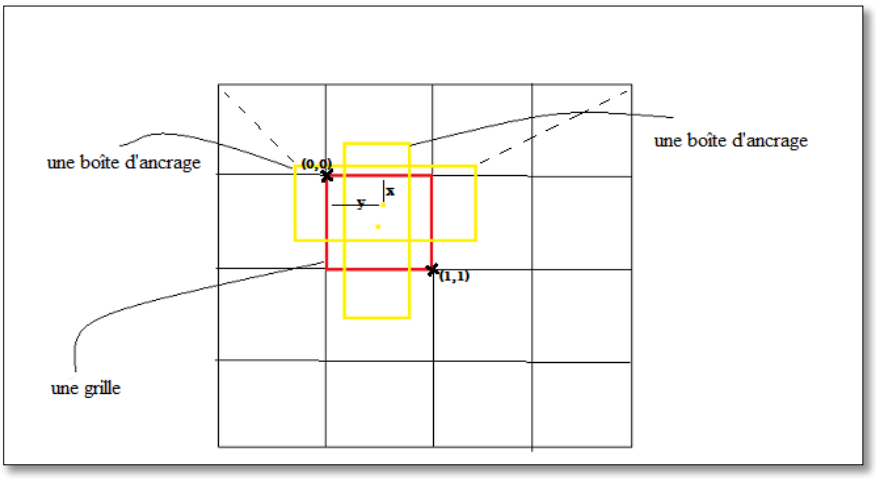

Figure 3. 11 Les boîtes d'ancrages

L'idée de la boîte d'ancrage est de combiner les dimensions de sortie de plusieurs objets dans une seule cellule de grille.

3.3.7 Generalized Intersection over Union

L'idée de Generalized Intersection over Union « GioU » est de rapprocher la boîte prédite de la vérité terrain malgré l'absence de chevauchement qui est fonction de perte où se formule comme suit [28]:

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \setminus (A \cup B)|}{|C|} = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \qquad (3.5)$$

Où :

- $A\ et\ B$ sont les boîtes englobantes de prédiction et de vérité terrain.
- C est la plus petite enveloppe convexe qui englobe à la fois $A\ et\ B$.

3.3.8 Lissage des étiquettes de classe (Class label smoothing)

C'est une méthode de régularisation. Si le réseau de neurones est trop adapté ou trop confiant, nous pouvons tous essayer de lisser les étiquettes. C'est-à-dire qu'il peut y avoir des erreurs dans les étiquettes pendant l'apprentissage, et nous pouvons « trop » croire aux étiquettes des échantillons d'apprentissage et, dans une certaine mesure, ne pas examiner la complexité d'autres prédictions. Par conséquent, afin d'éviter l'excès de croyance, il est plus raisonnable de coder la représentation de l'étiquette de classe pour évaluer l'incertitude dans une certaine mesure.

3.4 Les bases de données

Nous avons sélectionné trois ensembles de données contenant du texte directionnel : ICDAR2015 [31], MSRA-TD500 [38] et ICDAR2017-MLT [39] pour des expériences visant à évaluer les performances sur divers textes directionnels. Pour démontrer davantage la polyvalence de Yolov5, nous avons également mené des expériences sur un ensemble de données de texte horizontal populaire, ICDAR2013 [30]. Une brève description de tous les ensembles de données pertinents est donnée ci-dessous :

- ICDAR2013 [20,36] : L'ensemble de données ICDAR2013 contient 233 images de test et 229 images d'entraînement, ce qui constitue le texte de scène clé du concours de lecture robuste ICDAR. Le texte de la scène est horizontal et étiqueté avec une boîte rectangulaire horizontale, comprenant le sommet supérieur gauche et le sommet inférieur droit du rectangle.
    - ICDAR2015 [20,36] : L'ensemble de données de texte de scène ICDAR2015 provient du défi 4 du concours de lecture robuste ICDAR2015. L'ensemble de données comprend 1 000 images d'entraînement et 500 images de test, qui ont été capturées à l'aide de lunettes Google avec des résolutions relativement faibles. Les annotations d'instance de texte ont quatre sommets, qui forment une boîte englobante quadrilatérale irrégulière avec des informations d'orientation.
    - ICDAR2017-MLT [20,36] : L'ICDAR2017-MLT est un ensemble de données textuelles multilingues à grande échelle, qui contient 7 200 images pour la formation, 1 800 images pour la validation et 9 000 images pour les tests. L'ensemble de données se compose d'images de scènes naturelles contenant des textes en neuf langues avec des orientations multiples. Certaines langues sont étiquetées au niveau de la ligne comme le chinois, le coréen et le japonais, tandis que d'autres sont étiquetées au

niveau du mot comme l'anglais, le français, l'arabe et le bengali. Les différentes distributions de longueur de texte dans différentes langues rendent la tâche de détection beaucoup plus difficile.

- MSRA-TD500 [20 ,36] : L'ensemble de données MSRA-TD500 contient 200 images de test et 300 images d'entraînement, qui contiennent du texte arbitrairement orienté en chinois et en anglais. Les textes sont étiquetés avec des cases inclinées constituées par le coin supérieur gauche du rectangle, la largeur et la hauteur, et l'angle de rotation au niveau de la phrase. De longues lignes de texte droites apparaissent dans le jeu de données.

Nous avons redimensionné les images de tous base de donnés au 416 × 416 et ses annotations à l'annotation de yolov5 (x ,y , largeur , hauteur) où x et y sont les coordonnées de texte dans une image.

3.5 Langage, Logiciels et librairies utilisés dans l'implémentation

3.5.1 Python

Python [39] est un langage de programmation de haut niveau Crée par Guido van Rossum et sorti en1991, on l'a choisi parce que il a obtenu un grand succès dans le domaine de Machine learning grâce aux développeurs qui ont développé de nombreuses bibliothèques, ce qui rend l'utilisation de ce langage facile.

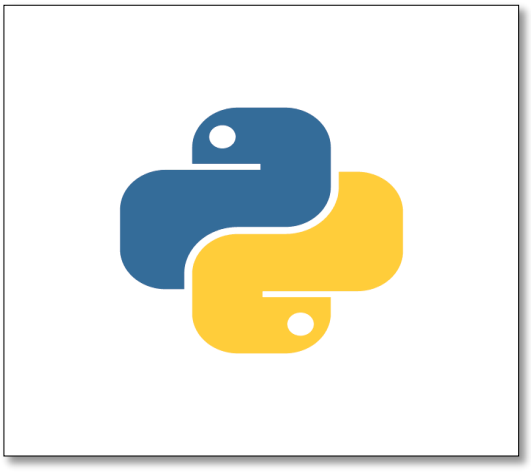

Figure 3. 12 Logo de Python

3.5.2 PyTorch

PyTorch [37] est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch (en) développée par Facebook. PyTorch permet d'effectuer les calculs tensoriels nécessaires notamment pour l'apprentissage profond « Deep learning ». Ces calculs

sont optimisés et effectués soit par le processeur « CPU » soit, lorsque c'est possible, par un processeur graphique « GPU » supportant CUDA. Il est issu des équipes de recherche de Facebook, et avant cela de Ronan Collobert dans l'équipe de Samy Bengio à l'IDIAP.

Figure 3. 13 Logo de Pytorch

3.5.3 PIL

Pillow est une bibliothèque d'imagerie Python « PIL », qui prend en charge l'ouverture, la manipulation et l'enregistrement d'images. La version actuelle identifie et lit un grand nombre de formats. La prise en charge de l'écriture est volontairement limitée aux formats d'échange et de présentation les plus couramment utilisés [41].

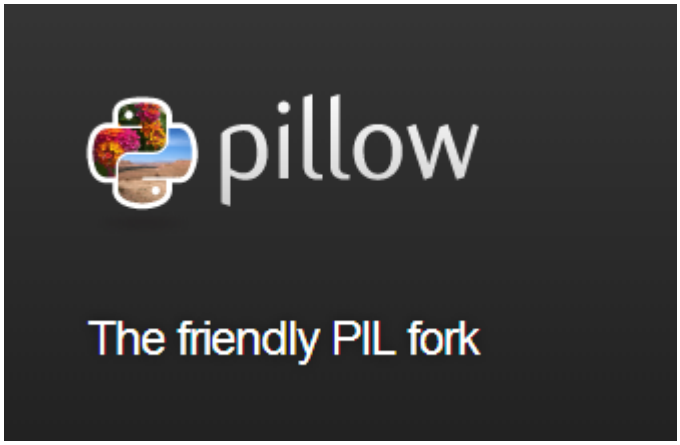

Figure 3. 14. Logo de pillow

3.5.4 OpenCV

OpenCV [40] est une bibliothèque de fonctions de programmation principalement destinées à la vision par ordinateur en temps réel. Lancé officiellement en 1999, le projet OpenCV était initialement une initiative d'Intel Research pour faire progresser les applications gourmandes en CPU, faisant partie d'une série de liaisons en Python, Java et MATLAB/OCTAVE. Dans d'autres projets.

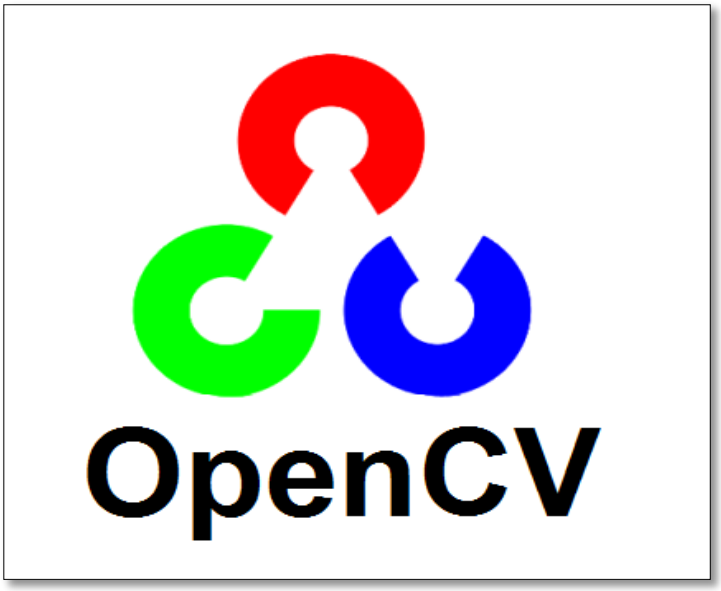

Figure 3. 15. Logo de OpenCV

3.5.5 Google Colab

Cloab ou bien Google Colaboratory [38] permet d'entraîner des modèles de Machine Learning directement dans le Cloud , on l' a choisi parce que la rapidité en l'exécution puisque il utilise CPU et GPU où l'accès gratuits aux GPU et facile pour partager notre travaille.

Figure 3. 16. Logo de Colab

3.5.6 Configuration utilisée dans l'implémentation

Nous ne pouvons pas utiliser des ordinateurs normaux car les algorithmes de Deep learning nécessitent que le système ait une vitesse et une puissance de traitement élevé (généralement basées sur le GPU), c'est pourquoi nous avons choisi Colab qui fournit :

- GPU (Tesla V100) et TPU (TPUv2) sur le Cloud
- 25 Go de RAM

- 150 Go de disque principal

3.6 Résultat

Les protocoles d'évaluation classiques pour la détection de texte, le repérage de mots et la reconnaissance de bout en bout reposent tous sur la précision « P », le rappel « R » et la mesure F « F ». La précision représente le rapport entre le nombre de régions de texte correctement détectées et le nombre total de régions de texte détectées. Le rappel représente le rapport entre le nombre de régions de texte correctement détectées et le nombre total de régions de texte dans l'ensemble de données. La F-mesure est une mesure unique de qualité créée en combinant rappel et précision. Ces protocoles d'évaluation s'expriment ainsi :

$$P = \frac{TP}{TP+FP} \qquad (3.6)$$

$$R = \frac{TP}{TP+FN} \qquad (3.7)$$

$$F = 2 \times \frac{P \times R}{P+R} \qquad (3.8)$$

Où :

- TP (True-Positive) est le nombre de vraies prédictions, c'est-à-dire les emplacements de texte corrects.
- TN (True-Negative) est le nombre de fausses prédictions, c'est-à-dire les emplacements de texte faux.
- FN (False-Negative) n'est pas les emplacements de texte.

La précision moyenne « AP » est la zone sous la courbe précision-rappel. AP combine à la fois précision et rappel. Il prend une valeur comprise entre 0 et 1 (le plus élevé est le meilleur).

Pour obtenir $AP = 1$, nous avons besoin que la précision et le rappel soient égaux à 1. Mean Average Precision « MAP » est la moyenne de l' AP calculé pour toutes les classes.

3.6.1 Les étapes pour faire l'apprentissage de modèle

Étant donné que Colab est un service Google, il permet de se lier à un compte Google Drive personnel pour obtenir des données de Drive (figure 3.17) à utiliser dans l'entraînement du modèle ainsi que pour enregistrer les résultats ultérieurement. Après cela, nous copierons la base de données de Google Drive vers Colab, puis l'extrairons (figure 3.18), le résultat sera 3 dossiers (train, test et valid).

```
[ ]  from google.colab import drive
     drive.mount('/content/gdrive')
```

Figure 3. 17. Montage sur Google Drive.

```
[ ]  %cd /content
     !cp /content/gdrive/MyDrive/dataset-icdar2013/c13-train.zip /content/
     !cp /content/gdrive/MyDrive/dataset-icdar2013/v2/c13-test.zip /content/
     !unzip /content/c13-test.zip; rm c13-test.zip
     !unzip /content/c13-train.zip; rm c13-train.zip
```

Figure 3. 18. Copier la base de données ICDAR2013 de Google Drive vers Colab puis l'extraire.

Après avoir installé les laboratoires et fait la configuration du modèle, nous l'entraînons (figure 3.19).

```
[ ]
   %%time
   %cd /content/yolov5/
   !python train.py --img 416 --batch 16 --epochs 200 --data '../data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results  --cache
```

Figure 3. 19. Mettre en œuvre le processus de formation.

Où :

- **img :** pour définir la taille de l'image d'entrée.
- **batch :** pour déterminer la taille du lot.
- **epochs :** pour définir le nombre d'époques d'entraînement.
- **data :** le chemin d'accès au fichier data.yaml contenant le résumé de l'ensemble de données
- **cfg :** pour spécifier notre chemin de configuration de modèle.
- **wheights :** pour spécifier un chemin vers les poids.
- **name :** nom du dossier de résultats.
- **cache:**images de cache pour une formation plus rapide.

3.6.2 L'étape pour faire l'extraction de texte

Nous pouvons choisir les meilleurs poids ou les derniers poids où les poids entraînés peuvent être utilisés pour extraire du texte à partir d'une image, d'une vidéo ou d'une caméra.

```
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source image.jpg
```

Figure 3. 20 Insterction pour faire le test.

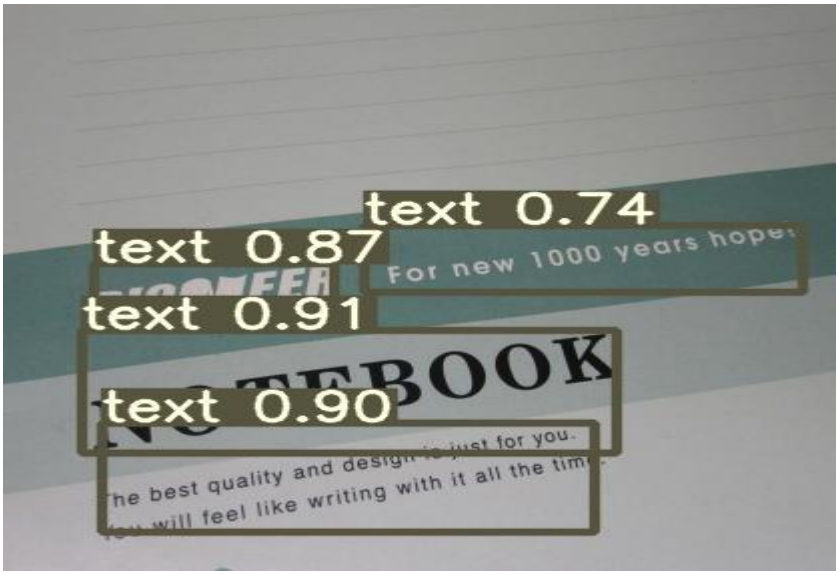

Figure 3. 21 Exemple sur l'extraction de texte d'une image de notre modèle

Figure 3. 22 Exemple sur l'extraction de texte de la vidéo de notre modèle

3.6.3 Évaluation sur long texte

Nous évaluons notre modèle sur l'ensemble de données MSRA-TD500. Le modèle est affiné pour 500 époques sur l'ensemble de données d'entraînement de MSRA-TD500. Pendant la phase de réglage, le taux d'apprentissage commence à partir de $1{,}0 \times 10^{-3}$ et est multiplié par 2 après 500 époques.

Les résultats quantitatifs sont répertoriés dans les courbes au-dessous où P est 86.13 %, R est 53.74 % et MAP est 58.28%. Notre modèle atteint une mesure F de 66.20% et une vitesse de calcul de 130 fps.

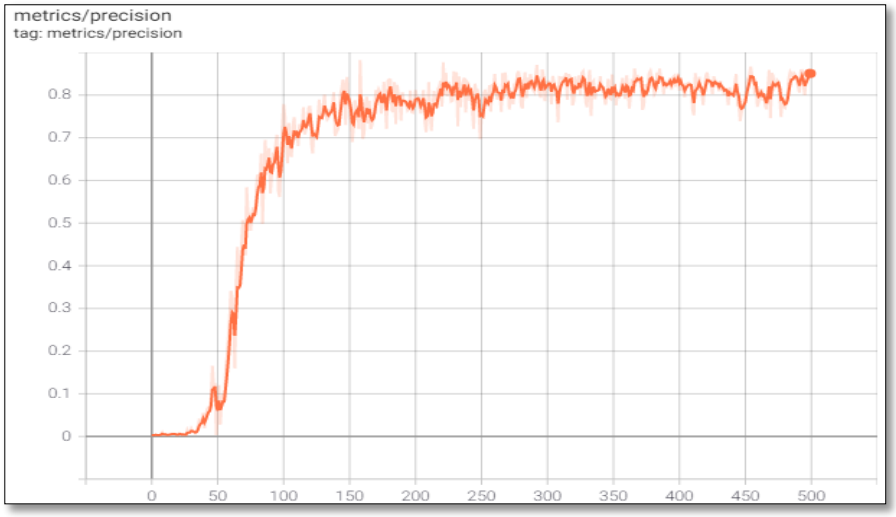

Figure 3. 23 La courbe de la précision de MSRA-TD500

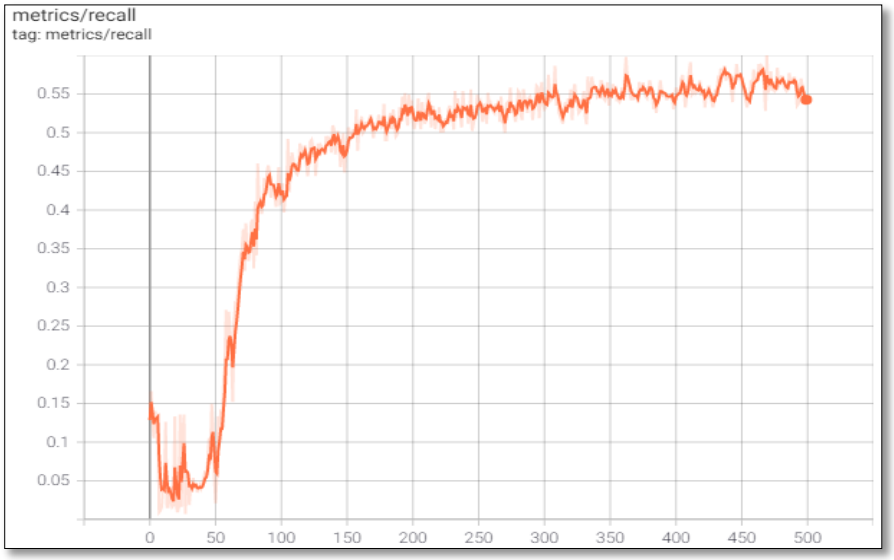

Figure 3. 24 La courbe du rappel de MSRA-TD500

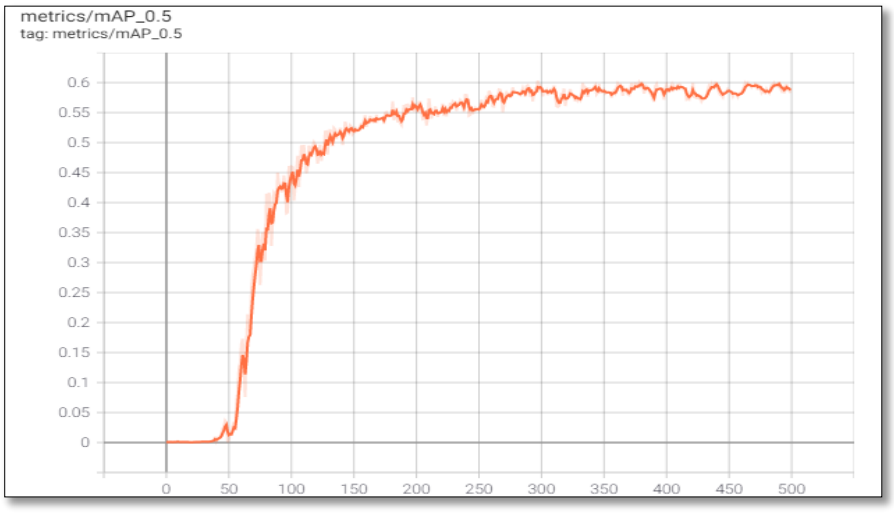

Figure 3. 25 La courbe de la Map de MSRA-TD500

3.6.4 Évaluation sur horizontal texte

Nous avons également mené des expériences sur ICDAR2013 pour tester la capacité d'adaptation générale de notre modèle. Cette base de données s'est concentrée sur les images de texte de scène où le texte dans les images est horizontal. Pendant la phase de réglage, le taux d'apprentissage commence à partir de $1{,}0 \times 10^{-3}$ et est multiplié par 2 après 500 époques.

Les résultats quantitatifs sont répertoriés dans les courbes au-dessous où P est 82.85 %, R est 69.71 % et MAP est 72.40 %. Notre méthode atteint une mesure F de 75.71 % et la vitesse ne change pas.

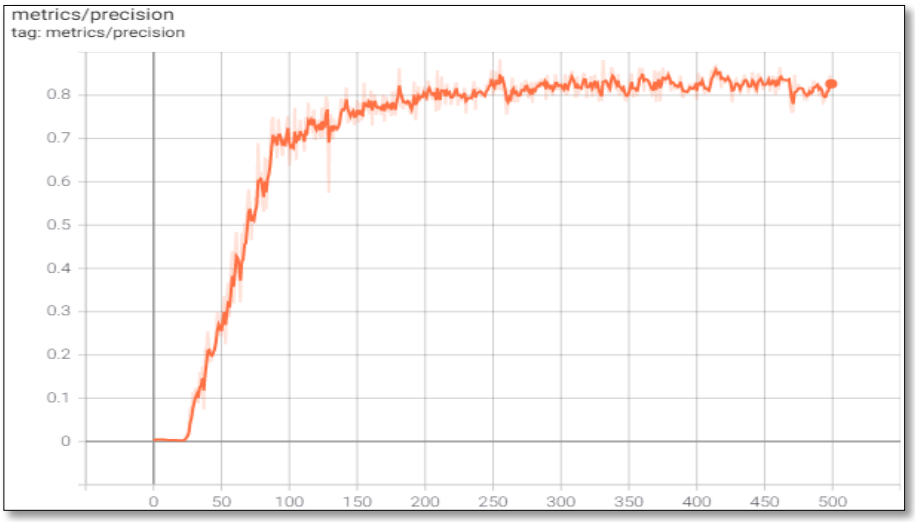

Figure 3. 26 La courbe de la précision de ICDAR2013

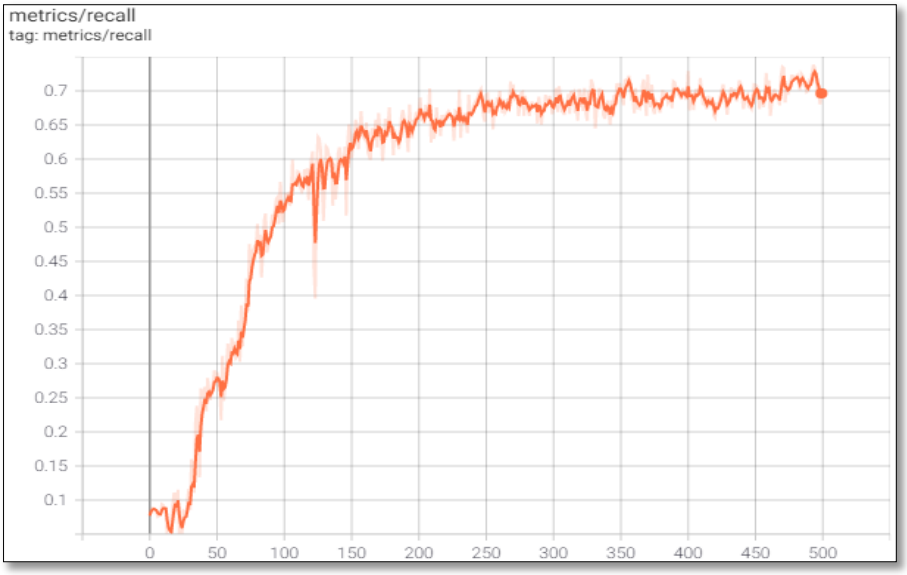

Figure 3. 27 La courbe du rappel de ICDAR2013

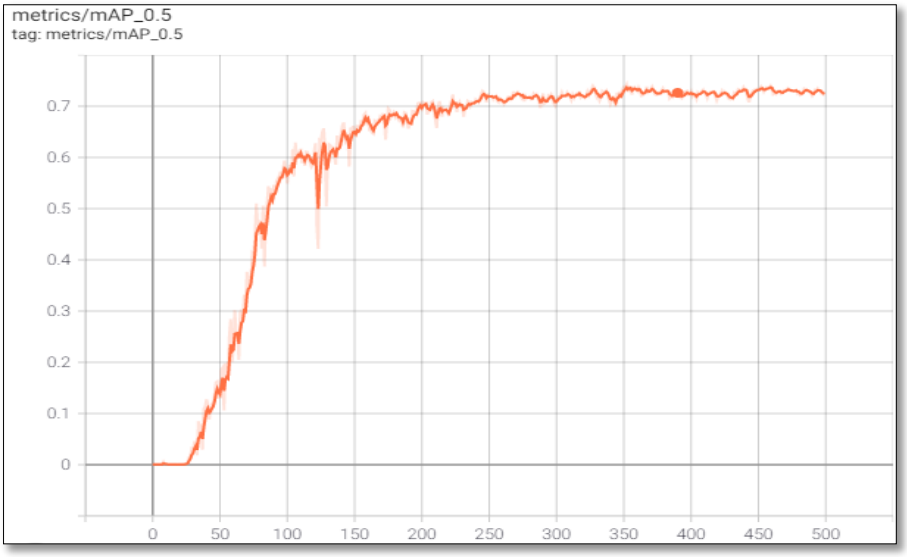

Figure 3. 28 La courbe de la Map de ICDAR2013

3.6.5 Évaluation sur orienté texte

Nous évaluons notre modèle sur l'ensemble de données ICDAR2015 aussi. Le modèle est affiné pour 500 époques sur l'ensemble de données d'entraînement de ICDAR2015. Pendant la phase de réglage, le taux d'apprentissage commence à partir de $1{,}0 \times 10^{-3}$ et est multiplié par 2 après 500 époques.

Les résultats quantitatifs de la méthode proposée sont répertoriés dans les courbes au-dessous où P est 65.46%, R est 42.27 % et MAP est 44.91 %. Notre méthode atteint une mesure F de 51.37 % et la vitesse ne change pas.

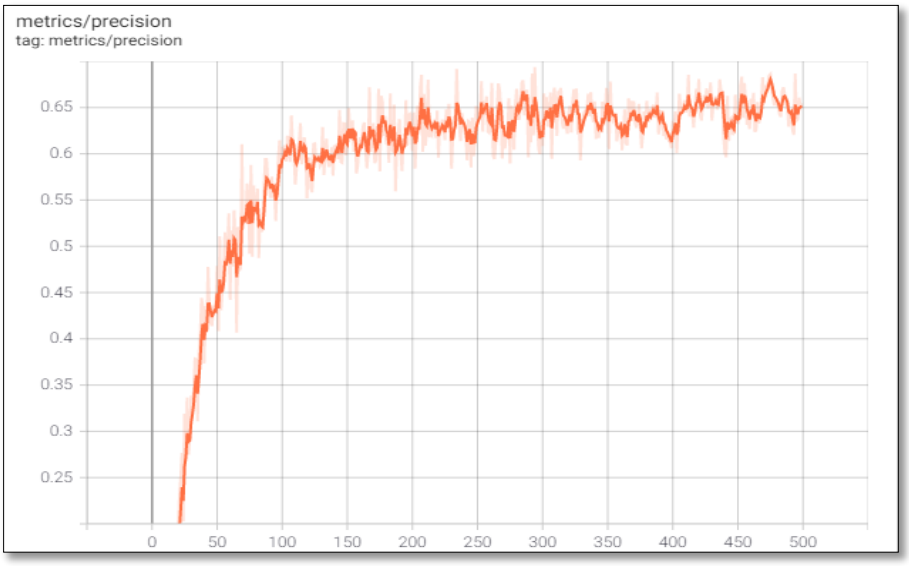

Figure 3. 29 La courbe de la précision de ICDAR2015

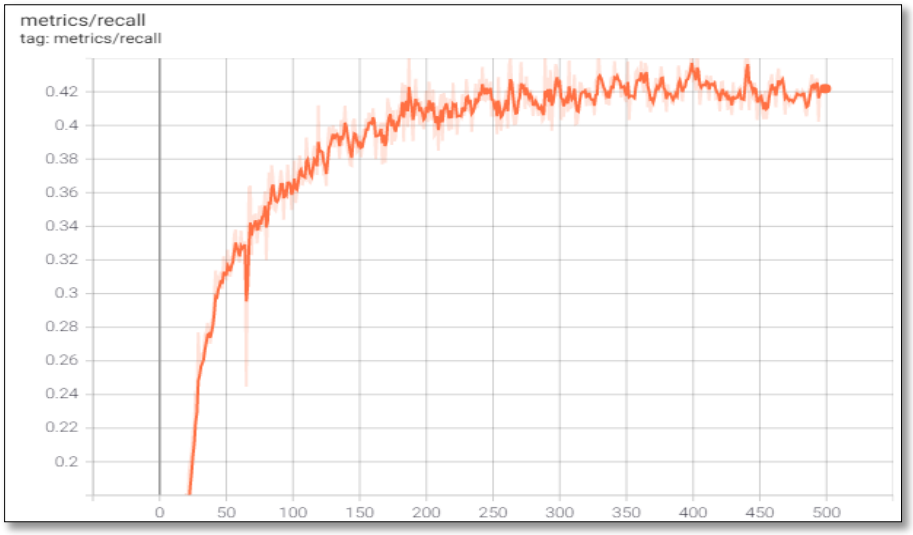

Figure 3. 30 La courbe du rappel de ICDAR2015

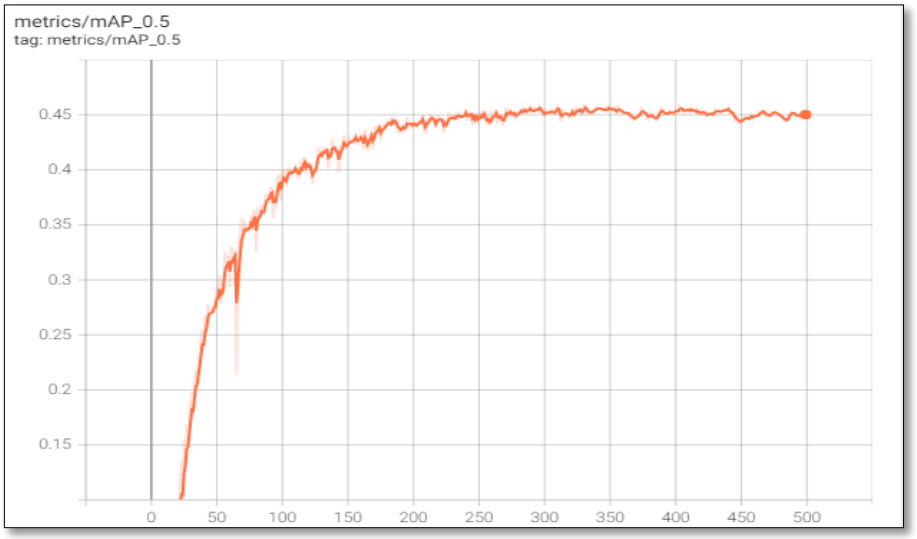

Figure 3. 31 La courbe de la Map de ICDAR2015

3.6.6 Évaluation sur multilingue texte

Nous évaluons notre modèle sur l'ensemble de données ICDAR2017. Le modèle est affiné pour 100 époques sur l'ensemble de données d'entraînement de ICDAR2017 parce que ça nous donne un bon résultat. Pendant la phase de réglage, le taux d'apprentissage commence à partir de $1{,}0 \times 10^{-3}$ et est multiplié par 2 après 100 époques.

Les résultats quantitatifs sont répertoriés dans les courbes au-dessous où P est 80.79%, R est 60.08 % et MAP est 65.62 %. Notre méthode atteint une mesure F de 68.91 %.

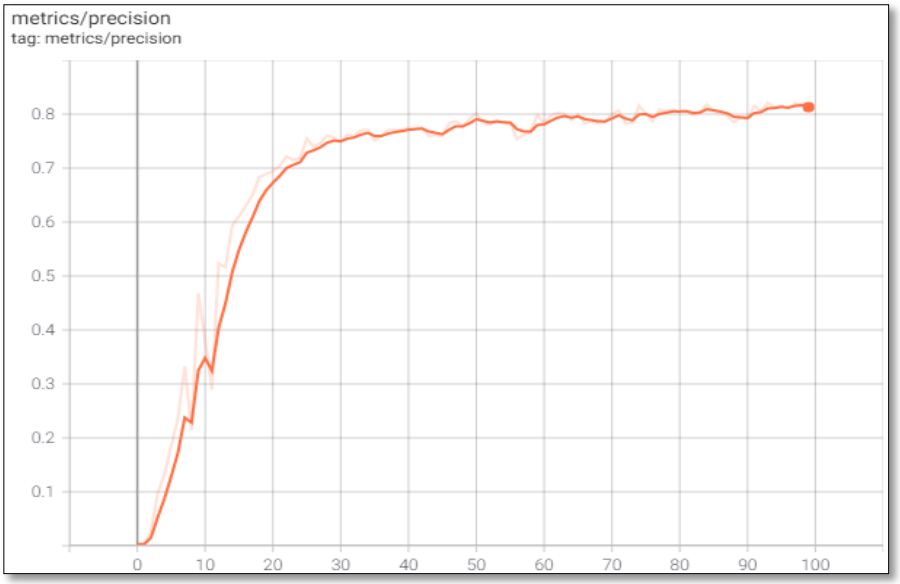

Figure 3. 32 La courbe de la précision de ICDAR2017

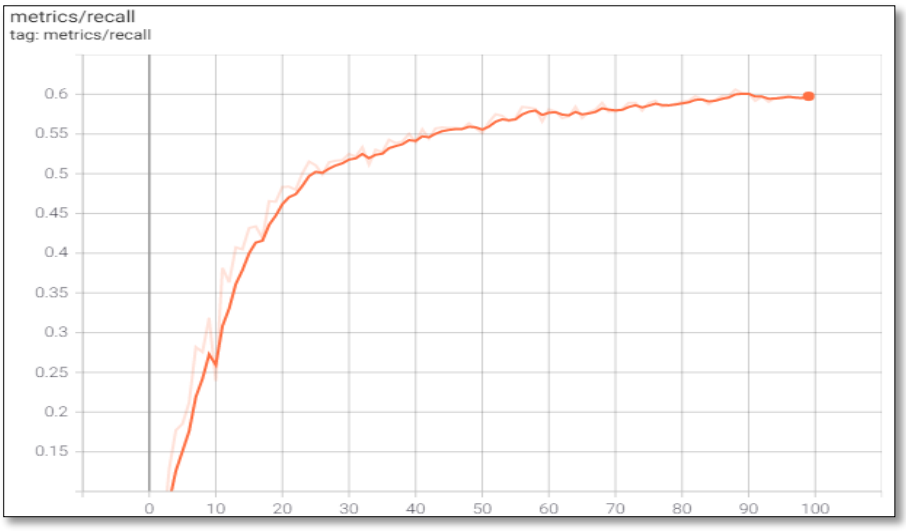

Figure 3. 33 La courbe du rappel de ICDAR2017

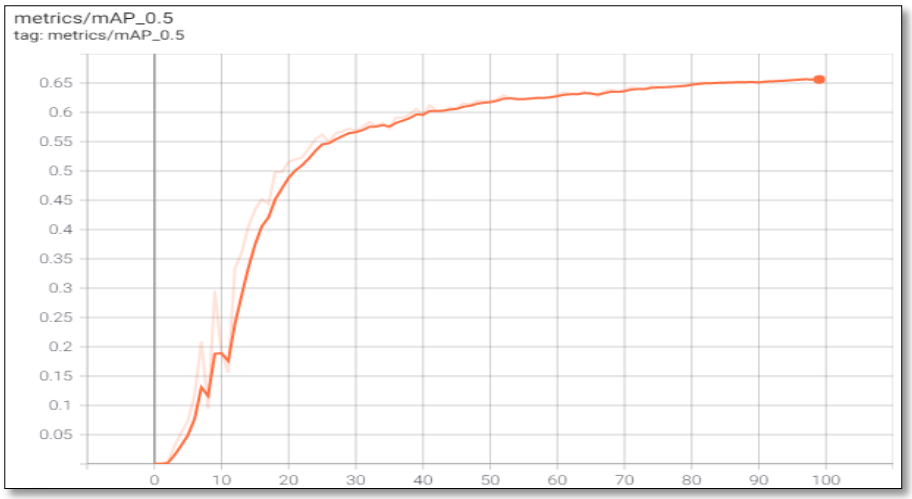

Figure 3. 34 La courbe de la Map de ICDAR2017

3.7 .Conclusion

Nous avons présenté dans ce chapitre la conception et l'implémentation de notre modèle proposé, on a vu aussi les détails de deux phase de notre modèle et s'architecture qui est basé sur CNN.

Ensuite , on a présenté les outils et les jeux de données utilise, ainsi que analysé les résultats de l'extraction de texte de chaque base de données après leur avoir appliqué notre modèle où les époques ont un rôle important pour obtenir des meilleurs résultats.

Conclusion Générale

Notre travail a pour but de concevoir et de réaliser une modèle pour l'extraction des textes dans les images en utilisant les méthodes de deep learning où on a fait modèle basé sur Réseaux de neurones convolutifs pour traiter la problème de extraction ou bien détection de texte non structuré ,qui donnait une excellent résultat du côté de vitesse 130 fps en temps réel dans des environnements complexes, en ce qui concerne le résultat de la précision, elle est plus 80% pour les différents textes comme long, horizontal et multilingue textes.

Notre thèse est organisée en 3 chapitres avec introduction générale et conclusion générale où on a défini les notions de ML et ses caractéristiques dans premier chapitre, nous avons parlé sur la vision par ordinateur et ses sous-concepts dans la deuxième chapitre et le troisième chapitre est notre conception et implémentation de notre modèle.

Pour les perspectives et les travaux de futur qui peuvent améliorer notre modèle de l'extraction des textes dans les images, nous avons des idées telles que :

- Utilisation de la boîte englobant inclinée pour améliorer la précision de orienté texte.
- Ajouter plus des fonctionnalités (mobile et site web) parce que le volume des poids de notre modèle est 14 MB.
- Appliquer notre modèle sur autres bases de données contient autre espèce de texte.
- Ajouter des méthodes de reconnaissance de texte.

Bibliographie

[1].Judith Hurwitz , D. K. (2018). "Machine Learning For Dummies.".

[2].Alzubi, J. (2018). "Machine Learning from Theory to Algorithms.".

[3].Madan Somvanshi, S. T., S.V. Shinde, Pranjali Chavan "A Review of Machine Learning Techniquesusing Decision Tree and Support VectorMachine ".

[4].Delalleau, O. (2008). "Extraction hiérarchique de caractéristiques pour l'apprentissagea partir de données complexes en haute dimension.".

[5].Roelofs, R., et al. (2019). "A meta-analysis of overfitting in machine learning." 32: 9179-9189.

[6].Park, H., et al. (2014). "Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over KOSPI 200 Index options." 41(11): 5227-5237.

[7].Charpentier, A., et al. (2017). "Econom\'etrie et Machine Learning." .

[8].Farhadi, F. (2017). Learning activation functions in deep neural networks. , École Polytechnique de Montréal.

[9].Delalleau, O. (2012). "Apprentissage machine efficace: théorie et pratique.".

[10].Juhel, J. (2013). "La recherche d'invariants différentiels dans les variations développementales: de la population à l'individu et réciproquement." : 13-41.

[11].Paul Smolensky, D. E. R., James L. McLelland (1986). "Explorations in parallel distributed processing: computational models of cognition and perception.".

[12].Dahmane, K. (2020). Analyse d'images par méthode de Deep Learning appliquée au contexte routier en conditions météorologiques dégradées. Université Clermont Auvergne.

[13].Lemaire, V., et al. (2008). "Réglage de la largeur d'une fenêtre de parzen dans le cadre d'un apprentissage actif: une évaluation.".

[14].Deng, L. and D. Yu (2014). "Deep learning: methods and applications." 7(3–4): 197-387.

[15].RASMA, C. (2008). "Méthodes à noyaux en apprentissage statistique ".

[16].Hasan, M. and F. Boris (2006). "Svm: Machines à vecteurs de support ou séparateurs à vastes marges." 64.

[17].Nacereddine, N., et al. (2009). L'algorithme EM et le Modèle de Mélanges de Gaussiennes Généralisées pour la Segmentation d'images. Application au contrôle des joints

soudés par radiographie.Traitement et Analyse de l'Information: Méthodes et Applications-TAIMA 2009.
[18].Ahmad, J., et al. (2019). Deep learning methods and applications. Deep Learning: Convergence to Big Data Analytics, Springer: 31-42.
[19].Yu, J. and W. Zhang (2021). "Face mask wearing detection algorithm based on improved YOLO-v4." Sensors 21(9): 3263.
[20].Wang, X., et al. (2021). "R-YOLO: A Real-Time Text Detector for Natural Scenes with Arbitrary Rotation." Sensors 21(3): 888.
[21].Huang, Z., et al. (2020). "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection." Information Sciences 522: 241-258.
[22].Raisi, Z., et al. (2020). "Text Detection and Recognition in the Wild: A Review." arXiv preprint arXiv:2006.04305.
[23].Bochkovskiy, A., et al. (2020). "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934.
[24].Huang, G., et al. (2017). Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition.
[25].Wang, C.-Y., et al. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops.
[26].Chen, P.-Y., et al. (2019). Smaller object detection for real-time embedded traffic flow estimation using fish-eye cameras. 2019 IEEE International Conference on Image Processing (ICIP), IEEE.
[27].Liu, S., et al. (2018). Path aggregation network for instance segmentation.Proceedings of the IEEE conference on computer vision and pattern recognition.
[28]. [En ligne]. Available:https://giou.stanford.edu/
[29].Rasche, C. (2019). "Computer Vision.".
[30].Szeliski, R. (2010). Computer vision: algorithms and applications, Springer Science & Business Media.
[31] .Nixon, M. S. and A. S. Aguado (2012). Feature extraction & image processing for computer vision, Academic Press.
[32].SHAPIRO, L. (2000). Computer Vision.
[33].FAROU, M. B. (1945). Multimédia mining Reconnaissance des formes dans une vidéo. Université Badji Mokhtar-Annaba.

[34].DJABEUR DJEZZAR Mohammed Rafik, B. F.-a. (2017). "Mise au Point d'une Application de Reconnaissance de Formes."Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique.
[35].Paour, A. (2016). "Les caractéristiques de la vidéo numérique.".
[36].Raisi, Z., et al. (2020). "Text Detection and Recognition in the Wild: A Review."arXiv preprint arXiv:2006.04305.
[37] [En ligne]. Available:https://pytorch.org/
[38] [En ligne]. Available: https://colab.research.google.com/
[39] [En ligne]. Available: https://www.python.org/
[40] [En ligne]. Available: https://opencv.org/
[41] [En ligne]. Available: https://pillow.readthedocs.io/en/stable/