



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: SIOD07/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Système d'Information Optimisation et Décision (SIOD)

**Contrôle d'accès des véhicules par la reconnaissance automatique de leur
plaque d'immatriculation**

Par :

MABROUK WASSIM

Soutenu le 26/06/021 devant le jury composé de :

Nom Prénom	Grade	Président
BENAMEUR Sabrina	MCB	Rapporteur
Nom Prénom	Grade	Examineur

Année universitaire 2020-2021

DEDICASE

Je dédie ce modeste travail

A mes très chers parents CHAHRAZED et MAHMOUD pour leur soutien et encouragement durant toutes mes années d'études et sans lesquels je n'aurais jamais réussi, A mon frère AMINE et mes sœurs hala et shirine et Lylia. A mon encadreur SABRINA BENAMEUR qui me donne tout pour réussir, A tous mes amis Oussama , Zizou , ikbal , Abedlhak , Ali, Guerri et à tous mes enseignants durant mes années d'études avec lesquels j'ai beaucoup appris.



Remerciement

Avant tout, ma profonde et sincère louange à Allah le Tout-Puissant pour m'avoir fourni la force et la patience pour accomplir ce travail.

Mes remerciements vont à mon superviseur Benameur Sabrina pour ses précieux conseils. Des remerciements particuliers seront adressés aux membres du jury, à savoir, pour les efforts qu'ils ont déployés pour évaluer mon travail de fin d'études.

J'exprime également ma gratitude à mes camarades de classe Aymen, Oussama, Habib pour leurs conseils qui m'ont permis d'améliorer ce travail.



Résumé

La reconnaissance automatique des plaques d'immatriculation (ALPR) est devenue très importante dans notre vie quotidienne en raison de la croissance illimitée des voitures et des systèmes de transport, ce qui rend impossible la gestion et la surveillance de nombreux domaines par les humains, tels que la sécurité routière, la gestion des parkings, la poursuite des criminels. Dans ce mémoire, nous présenterons un système ALPR.

Comme tout système ALPR, nous avons passé par un ensemble d'étapes, nous commençons d'abord par la capture de l'image d'un véhicule, puis nous exécutons une méthode de détection des plaques d'immatriculation construit par l'apprentissage profond utilisant les réseaux neuronaux(CNN). Par la suite on coupe cette plaque et on la convertit en une image en niveaux de gris, Les numéros de plaque d'immatriculation sont ensuite extraits en utilisant une fonction de la bibliothèque OCR tesseract, finalement on compare le contenu de notre plaque par les plaques qui sont déjà enregistrés dans la base de donnée du système et tout ça pour avoir le contrôle d'accès des véhicules dans un parking moderne.

Mots clés: Reconnaissance automatique de plaque d'immatriculation (ALPR), Réseaux neuronaux Convolutionnel (CNN), Reconnaissance optique des caractères Tesseract (OCR), Apprentissage Profond, Control d'accès.

Abstract

The recognition of license plates (ALPR) has become very important in our daily life because of the unlimited growth of cars and transport systems, making it impossible for humans to manage and monitor many areas, such as: road safety, parking management, and criminal prosecution. In this memory, we will present an ALPR system.

As any ALPR system, we will go through a set of steps. First, capturing the vehicle's image, then; running a method of detecting license plates built by deep learning using neural networks (CNN), then we cut this plate and convert it into a grayscale image. The license plate numbers are then extracted using a function of the tesseract OCR library, finally; we compare the contents of our plate by the plates that are already registered in the database of the system, and all of this is to have access control of vehicles in a modern parking.

Key-words: Automatique License Plate Recognition (ALPR), Convolutional Neural Network (CNN), Optical Character Recognition Tesseract (OCR), deep learning, Access control.

Table des matières

Table des matières	I
Liste des figures.....	V
Introduction générale	1
Chapitre I :.....	3
La reconnaissance de plaque d'immatriculation: Etat de l'art	3
1 Introduction	3
2 Etat de l'art.....	3
2.1 Un aperçu sur les plaque d'immatriculations	3
2.1.1 Définition.....	3
2.1.2 Historique.....	3
2.1.3 La délivrance	3
2.1.4 Identification	4
2.1.5 Attribution des numéros.....	4
2.1.6 Type des plaques.....	5
2.2 Un aperçu sur les parkings	6
3 La reconnaissance des formes	7
3.1 Définition.....	7
3.2 Système de reconnaissance des formes	7
3.2.1 Acquisition.....	8
3.2.2 Prétraitement.....	8
3.2.3 Segmentation	8
3.2.4 Extractions des primitives	9
3.2.5 Classification	9
3.2.5.1 L'apprentissage	10
3.3 Méthodes de reconnaissance des formes	11
3.3.1 Méthode de correspondance de modèle	11
3.3.1.1 Corrélation	12
3.3.1.2 Convolution	13
3.3.2 Méthode statistique.....	14

Table des matières

3.3.3	Approche syntaxique et structurale	14
3.3.4	Approche de réseaux de neurones.....	15
4	L'importance du système de reconnaissance des plaques d'immatriculation dans l'entrée et la sortie du stationnement.....	16
5	Les dernières technologies utilisées pour détecter les matricules	17
5.1	LAPI (Lecture Automatique de Plaques d'Immatriculation).....	17
5.1.1	Caractéristiques.....	17
5.1.2	LAPI-ACCES	17
5.1.3	LAPI-ENGINE	18
5.1.4	LAPI-SMARTPHONE	19
5.2	Système VIRONIC.....	19
A.	Fonctionnement	19
B.	Avantages	19
5.3	Système Auto-Vu	20
A.	Fonctionnalités	20
5.4	OpenALPR.....	21
6	Autres applications utilisant le numéro d'immatriculation	22
7	Conclusion	22
Chapitre 2 :		23
L'apprentissage profond.....		23
1	Introduction.....	23
2	Réseaux neuronaux	23
2.1	Définition	23
2.2	Architectures de réseaux de neurone	23
2.2.1	Réseau d'anticipation à une seule couche	23
2.2.2	Réseaux d'alimentation multicouches	24
2.2.3	Réseaux récurrents.....	25
2.3	Types d'apprentissage automatique	26
2.3.1	Apprentissage supervisé.....	26
2.3.2	Apprentissage non supervisé	27
2.3.3	Apprentissage du renforcement	27
2.3.4	Apprentissage semi-supervisé.....	28
3	L'apprentissage profond.....	28
3.1	Définition	28

Table des matières

3.2	Architectures d'apprentissage profond	29
3.2.1	Réseaux préétablis non supervisés	29
3.2.2	Réseaux neuronaux récurrents(RNN)	31
3.2.3	Réseaux neuronaux récurrents (RNRs)	32
3.2.4	Réseaux neuronaux convolutifs (RNC).....	33
➤	Hyper-paramètres de couche convolutionnelle :	37
➤	Couches groupées :	37
4	Domaine d'application de l'apprentissage profond.....	40
4.1	Bio-informatique	40
4.2	Santé	40
4.3	Robotique.....	41
4.4	Visual and vocal recognition	41
4.5	Reconnaissance visuelle et vocale	41
5	Les défis de l'apprentissage profond	42
5.1	La quantité de données utilisées	42
5.2	Le temps d'apprentissage trop élevé.....	42
5.3	Optimisation de l'hyper-paramètre	42
5.4	Difficulté à comprendre comment l'information arrive	42
6	L'apprentissage profond est sensible :	43
7	Conclusion.....	43
Chapitre 3 :		44
Conception et implémentation.....		44
1	Introduction	44
2	Conception générale de notre système	44
3	Conception détaillé	45
3.1	L'apprentissage du modèle	46
3.1.1	La collecte et la préparation des ensembles de données (Description des données « dataset »)	46
3.1.2	Prétraitement des ensembles de données	47
3.1.2.1	Importer des bibliothèques.....	48
3.1.2.2	Importer les ensembles de données.....	48
3.1.2.3	Prendre soin des données manquantes dans l'ensemble de données.....	48
3.1.2.4	Encodage des données catégoriques.....	49
3.1.2.5	Diviser l'ensemble de données en ensemble de formation et ensemble de tests.....	49

Table des matières

3.1.2.6	La mise à l'échelle des données	49
3.1.3	Entraînement du model (Training)	50
3.1.3.1	Conception de l'architecture du modèle.....	51
3.1.3.2	L'entraînement du model.....	51
3.1.4	Utilisation du model obtenu.....	51
3.2	La reconnaissance de plaque.....	52
3.2.1	La segmentation des chiffres.....	52
3.2.2	La reconnaissance des chiffres	52
3.3	Le contrôle d'accès	52
4	Implémentation.....	53
4.1	Environnements et Outils de développement	53
4.2	Chargement de l'ensemble de données (transfert les données vers le colab).....	59
4.3	Création du modèle.....	62
4.4	Algorithmes	63
4.4.1	Algorithme principal	63
4.4.2	Algorithme d'apprentissage	63
4.4.3	Algorithme de reconnaissance de plaque	64
4.4.4	Algorithme de Contrôle d'accès	64
5	Résultats	64
5.1	Architecture du modèle.....	65
5.2	Présentation de l'application	66
5.3	Discussion des résultats.....	67
6	Conclusion	69
	Conclusion générale	70
	Bibliographie.....	72

Liste des figures

Figure I.1 Plaque d'immatriculation algérienne	5
Figure I.2 Plaque d'immatriculation canadienne	6
Figure I.3 Système de reconnaissance des formes.....	7
Figure I.4 Les étapes de classification.....	9
Figure I.5 Etape d'apprentissage de la méthode de correspondance de modèle et de corrélation	11
Figure I.6 Phase de la classification de la méthode de correspondance de modèle et de corrélation	11
Figure I.7 Le processus de numérisation des images effectué à l'aide de la convolution	13
Figure I. 8 Numérisation de l'image à l'aide de convolution	13
Figure I.9 Séparation des bonnes propriétés "a" et de mauvaises propriétés "b"	14
Figure I.10 Composants de réseau de neurone artificiel.....	15
Figure I.11 Une photo du parking moderne.	16
Figure I.12 Exemple d'architecture matérielle de système LAPI	18
Figure I.13 Exemple d'architecture matérielle di système LAPI-NETWORK	18
Figure I.14 Utilisation du système VIRONIC pour la surveillance routière	20
Figure II. 1 Réseau d'alimentation avec une seule couche de neurones (Subarna, 2019)	24
Figure II. 2 Réseau d'alimentation entièrement connecté avec une couche cachée et une couche de sortie (Subarna, 2019)	24
Figure II. 3 Réseau récurrent sans boucles d'auto rétroaction et sans neurones cachés (Haykin, 2008)	25
Figure II. 4 Réseau récurrent avec neurones cachés (Haykin, 2008)	26
Figure II. 5 Apprentissage supervise.....	27
Figure II. 6 Apprentissage non-supervise	27
Figure II. 7 Apprentissage du renforcement (Reinforcement learning, 2019).	28
Figure II. 8 Structures d'auto-encodeurs (Baldi P. , jan 2012).	29
Figure II. 9 Architecture RBD (Patterson, 2017).	30
Figure II. 10 Idée de base des PAC (Gersey, 2018).	31
Figure II. 11 Exemple de réseau neuronal récurrent entièrement connecté (R. Medsker, jan 1999).....	31
Figure II. 12 Exemple de réseau récurrent simple (R. Medsker, jan 1999)	32
Figure II. 13 Une architecture de réseau neuronal récursif simple (Recursive neural network, n.d.)	32
Figure II. 14 RNC et la vision par ordinateur (Patterson, 2017).	33
Figure II. 15 Architecture générale de haut niveau de RNC (Patterson, 2017).	34
Figure II. 16 Volume 3D de la couche d'entrée (Patterson, 2017)	35
Figure II. 17 Couche de convolution avec volumes d'entrée et de sortie (Patterson, 2017)	36
Figure II. 18 Convolutions et cartes d'activation (Patterson, 2017).	36
Figure II. 19 Mise en commun maximale (Prabhu, 2021).....	38
Figure II. 20 La couche de mise en commun (Prabhu, 2021)	38
Figure II. 21 Architecture complète de RNC (Prabhu, 2021).....	39

Liste des figures

Figure III. 1 Schéma de conception global	45
Figure III. 2 Schéma de conception détaillée.....	46
Figure III. 3 Échantillon de données utilisées	47
Figure III. 4 Les étapes de Prétraitement des ensembles de données	48
Figure III. 5 Cycle de la vie d'entraînement	51
Figure III. 6 Exemple de teste du model.....	52
Figure III. 7 Le logo du python.....	53
Figure III. 8 Le logo du Qt Creator	54
Figure III. 9 Le logo du TensorFlow	54
Figure III. 10 Le logo du Keras	55
Figure III. 11 Le logo de l'Anaconda	55
Figure III. 12 Le logo du NumPy	56
Figure III. 13 Le logo du pandas.	56
Figure III. 14 Le logo du matplotlib	57
Figure III. 15Le logo du Google Colab.....	57
Figure III. 16 Le logo de l'OpenCV	57
Figure III. 17 Le logo du Tesseract OCR	58
Figure III. 18 Le logo du PyCharm.....	58
Figure III. 19 Le logo du CSS	59
Figure III. 20 Le logo du PyCharm.....	59
Figure III. 21 La premiere partie de donnée (images)	59
Figure III. 22 Code image DataGenerator	60
Figure III. 23 Code du flow-from-directory.....	61
Figure III. 24 Code de compilation.	62
Figure III. 25 Code de déclanchement l'entraînement.	62
Figure III. 26 Architecture de notre modèle.	65
Figure III. 27 Présentation Graphique de la précision	65
Figure III. 28 Présentation Numérique (Accuracy & Loss).....	66
Figure III. 29 Première page de l'interface.	66
Figure III. 30 Un résultat d'un test d'accès valide.....	66
Figure III. 31 Un résultat d'un test d'accès non valide.....	67
Figure III. 32 Le bouton d'ajout à la BD.	67
Figure III. 33 Exemple d'une mauvaise détection.....	68
Figure III. 34 Exemple d'une mauvaise détection.....	68
Figure III. 35 Exemple d'une bonne détection.....	68

Introduction générale

À la fin du XXe siècle, le monde a connu une croissance démographique accompagnée d'un niveau économique élevé et d'un pouvoir d'achat accru des particuliers, ce qui a conduit à une augmentation du nombre de voitures achetées par des personnes en plus des véhicules de transport en commun, ce qui a entraîné l'apparition de nombreuses difficultés avec ce grand nombre de voitures, ce qui nécessite une base de données spéciale pour faciliter le processus d'accès aux informations permettant de connaître les voitures et leurs propriétaires, avec précision et rapidité. Ces facteurs et d'autres ont conduit à un besoin urgent d'un système automatisé capable de détecter les plaques d'immatriculation des voitures. Ceci est dû à la difficulté de gérer manuellement le grand nombre de voitures, car cela demande beaucoup de temps et d'efforts, un travail ardu ainsi qu'une faible efficacité. En plus de l'inexactitude des informations obtenues (Griffon, (2006)).

Ce système de détection aide les personnes à détecter automatiquement la plaque d'immatriculation sans intervention ni supervision humaine directe. Ainsi que pour surveiller le trafic et le suivi des voitures volées, et est utilisée dans les portails électroniques pour s'ouvrir automatiquement aux membres autorisés aux zones aménagées et au contrôle des frontières ou bien dans les contrôles d'accès aux parkings, Par conséquent (Moati, 2001), il existe une grande tendance mondiale au cours des deux dernières décennies à proposer et développer des techniques automatisées dans le but de détecter et d'identifier les plaques d'immatriculation de voiture utilisant des systèmes inspirés du l'apprentissage profond (deep Learning).

Nous nous sommes intéressés à la détection automatique des plaques d'immatriculation. Ce travail a donc pour but de construire et d'implémenter un modèle de détection et de reconnaissance des plaques d'immatriculation afin de développer une application de contrôle d'accès dans un parking moderne en utilisant un apprentissage approfondi.

En plus de l'introduction générale, notre mémoire est organisée en 3 chapitres,

Introduction générale

Dans le premier chapitre nous donnons un aperçu sur les plaques d'immatriculation, nous détaillons ensuite la reconnaissance des formes enfin et nous citons les derniers techniques utilisées pour la détection des plaque d'immatriculations.

Dans le deuxième chapitre nous définissons des notions liées à l'apprentissage automatique et ses caractéristiques tout en détaillons les réseaux neuronaux et ses architectures.

Le troisième chapitre est réservé à la conception et l'implémentation de notre système. Dont nous commençons par la présentation de la conception générale et détaillée de notre application ensuite nous détaillons les étapes d'implémentation. Les résultats obtenus ainsi qu'une évaluation sont illustrés à la fin du chapitre.

Une conclusion ainsi que des perspectives à ce travail sont présentées à la fin de ce mémoire.

Chapitre I :

La reconnaissance de plaque d'immatriculation: Etat de l'art

1 Introduction

En raison du grand nombre de véhicules, les villes modernes nécessitent de mettre en place des systèmes automatiques efficaces pour la gestion de la circulation et l'ordonnement. L'un des systèmes les plus utiles est le système de détection des plaques d'immatriculation des véhicules (VLP).

Dans ce chapitre nous commençons par un aperçu sur les plaques d'immatriculation, nous détaillons ensuite la reconnaissance des formes, à la fin du chapitre nous présentons les derniers techniques pour la détection des plaques d'immatriculations.

2 Etat de l'art

2.1 Un aperçu sur les plaque d'immatriculations

2.1.1 Définition

Une plaque d'immatriculation est une plaque portant une combinaison unique de chiffres ou de lettres (pour une zone géographique donnée), destinée à identifier facilement un véhicule terrestre (automobile, motocyclette, camion, véhicule agricole, etc ...) (Ait benamar & Ouheb , 2019).

2.1.2 Historique

En 1783, un arrêt du Conseil du roi de France fait obligation aux rouliers et cochers de fixer sur les attelages une plaque de métal portant nom et adresse de leurs propriétaires. C'est en 1891, à Lyon, dans le parc de la Tête d'Or, que le premier système d'immatriculation du monde est mis en place) (Ait benamar & Ouheb , 2019).

2.1.3 La délivrance

Les numéros des plaques sont délivrés par une administration. Dans certains pays, comme la France, le propriétaire du véhicule est généralement responsable de l'achat des plaques, qu'il fait réaliser par un prestataire de services (garage, centre auto...). Dans d'autres, comme en Suisse, le jeu de plaques est délivré par l'administration qui reste propriétaire de celles-ci, ou comme en

Belgique où la plaque arrière est délivrée par l'administration qui en est propriétaire tandis que la plaque avant doit être achetée dans le commerce.

En Amérique du Nord, chaque État, province ou territoire émet sa propre plaque d'immatriculation (Lebib & Ait Menguellet , 2016).

2.1.4 Identification

Il existe deux concepts d'utilisation de plaques : certains pays, comme la France, adoptent le principe d'utiliser les plaques pour identifier un véhicule. Quel que soit le propriétaire, cette plaque accompagne le véhicule tout le long de sa vie ; le cas particulier des déménagements, dans les pays où l'indication géographique est également indiquée, est réglé par des règles spécifiques (Lebib & Ait Menguellet , 2016).

Dans d'autres pays, comme la Belgique ou la Suisse, la plaque est attribuée au propriétaire, qui la conserve lorsqu'il change de véhicule. Identifiant alors le propriétaire, la plaque fait également office d'attestation d'assurance, le propriétaire devant rendre les plaques à l'administration lorsqu'il n'est plus assuré. La notion de plaques interchangeable peut alors y exister, le même jeu de plaques est utilisé pour les différents véhicules d'un même propriétaire, celui-ci ne pouvant utiliser, ou autoriser l'utilisation à des tiers, qu'un seul de ses véhicules à la fois (Lebib & Ait Menguellet , 2016).

Pour la Belgique, la plaque est attribuée à une personne physique ou une personne morale. Quand le véhicule est revendu, le propriétaire de la plaque peut la conserver et la transférer sur son nouveau véhicule ou la remettre à l'Administration et demander une nouvelle plaque (GADOUI, KEBIR, & MAMOUNI, 2020).

2.1.5 Attribution des numéros

La plupart du temps, pour une administration donnée (une par État dans l'Union européenne ou aux États-Unis, parfois à l'échelle d'une sous division administrative de l'État dans certains pays), il ne peut y avoir deux véhicules différents portant le même numéro.

Selon les systèmes, un numéro est attribué par véhicule indépendamment de son propriétaire (comme au Royaume-Uni, en France ou en Italie), ou un propriétaire se voit attribuer un numéro par véhicule (comme en Belgique, en Suisse ou aux États-Unis).

Dans le premier cas, lorsqu'un véhicule change de propriétaire, il ne change généralement pas de numéro. Lorsque le véhicule est détruit ou change de numéro, le numéro qu'il portait est perdu et ne peut plus être réutilisé.

Dans le second cas, lorsqu'un véhicule change de propriétaire ou est détruit, son numéro reste attribué à son ancien propriétaire. Le nouveau propriétaire peut alors soit réutiliser un

numéro disponible qui lui aurait précédemment été attribué, soit acquérir un nouveau numéro (GADOUI, KEBIR, & MAMOUNI, 2020).

2.1.6 Type des plaques

Selon les systèmes, la plaque peut comporter des chiffres et des lettres dans un ordre spécifique ou dans un ordre quelconque. Pour anecdote, une plaque composée de trois chiffres et de trois lettres (i et o exclus) peut avoir 13 824 000 combinaisons différentes ($24 \times 24 \times 24 \times 10 \times 10 \times 10$) mais certaines combinaisons de lettres sont exclues, si celles-ci peuvent porter préjudice à son détenteur. Ainsi, en Belgique par exemple, des combinaisons de trois lettres telles que AAP (singe, en néerlandais), ANE, GAG, CON, CUL, DOM (crétin, en néerlandais), SEX, etc. ne sont pas délivrées par l'administration (sauf dans les cas de plaques personnalisées). Les combinaisons qui forment les initiales de partis politiques comme CDH (Centre démocrate humaniste) ou VLD (Libéraux et démocrates flamands) sont strictement interdites (Ait benamar & Ouheb, 2019).

➤ **En Algérie :**

Un à cinq chiffres, trois chiffres, deux chiffres. La première série de chiffres est le numéro de série d'enregistrement, la seconde est un nombre composé du type de véhicule (1 : véhicule de tourisme, 2 : camion, 3 : véhicule utilitaire, 4 : transport en commun, 5 : tracteur de semi-remorque, 6 : tracteur agricole 7 : véhicules spéciaux, 8 : remorque, 9 : motos) suivi des deux derniers chiffres de l'année de mise en circulation, et la troisième correspond au code du département (wilaya) d'immatriculation.



Figure I.1 Plaque d'immatriculation algérienne

➤ **En Canada :**

Diffère selon la province. Par exemple :

Au Québec, depuis 1983, les plaques d'immatriculation des véhicules de promenade (les plus répandues) comportent 6 caractères bleu foncé sur fond gris : trois lettres, une espace et trois chiffres. Quand la séquence a été épuisée, la Société de l'assurance automobile du Québec (SAAQ) a émis l'inverse, soit trois chiffres, une espace suivie de trois lettres. Depuis l'automne 2009, la limite de cette séquence a aussi été atteinte, alors la nouvelle séquence disponible est composée d'une lettre et de deux chiffres, puis d'une espace suivie de trois lettres. Il existe encore

quelques exemplaires du format datant de 1979, soit 7 caractères bleus sur fond blanc : 3 chiffres, une lettre et 3 chiffres.

En Ontario, les plaques pour véhicules de particuliers ("véhicules de tourisme") comportent trois ou quatre lettres suivies de trois chiffres. Les lettres et chiffres sont séparés par une couronne. Les plaques pour voitures et camions en l'Ontario existent en quatre principaux motifs, généralement semblables en apparence mais de couleurs différentes. Les plaques blanches, avec une combinaison de lettres et de chiffres bleus, sont pour les voitures de tourisme et les remorques. Les plaques des camions et véhicules commerciaux et agricoles sont blancs avec le texte en noir. Les véhicules diplomatiques ont des plaques rouges au texte blanc. Les plaques amovibles pour vendeurs de voitures et garages sont jaunes avec texte noir. Le Ministère des transports de l'Ontario offre aux particuliers la possibilité d'acheter des plaques personnalisées moyennant un supplément. Ces plaques peuvent soit afficher de deux à huit lettres ou chiffres seuls, ou de deux à six chiffres en plus d'une image puisée à même un catalogue d'environ 60 choix (2015). Il est à noter la légende sur les plaques ontariennes est en anglais («Yours to discover»), sauf sur la plaque personnalisée arborant le drapeau franco-ontarien qui elle a une légende française («Tant à découvrir»). Le ministère se réserve le droit de refuser les choix pour une variété de motifs, dont l'incitation à la haine, les propos politiques, sexuels ou prônant la consommation de drogues et d'alcool. Des plaques personnalisées sont aussi disponibles pour les motos¹³, Les gouvernements des Territoires du Nord-Ouest et du Nunavut émettent une plaque en forme d'ours polaire (GADOUI, KEBIR, & MAMOUNI, 2020).



Figure I.2 Plaque d'immatriculation canadienne

2.2 Un aperçu sur les parkings

Le parking d'automobiles est un terrain (une surface clôturée) de stationnement d'un ensemble des voitures immatriculées, pendant un temps déterminé. Il peut se trouver dans un immeuble bâti en superstructure (partie en élévation à l'air libre) ou en infrastructure (partie enterrée ou en dessous du sol artificiel), Sur une aire aménagée ou non pour le stationnement, sur une terrasse d'un immeuble, sous un immeuble bâti (sur pilotis ou en encorbellement) (Amnache & Ouadi , 2011).

3 La reconnaissance des formes

3.1 Définition

La reconnaissance de forme est l'une des branches de l'apprentissage automatique qui met l'accent sur la reconnaissance des motifs et des régularités dans les données. Il est également considéré comme le processus de classification des données d'entrée dans certains modèles en fonction de caractéristiques de base. Dans certains cas, la reconnaissance de forme est synonyme d'apprentissage automatique. Les techniques de reconnaissance des formes formalisent, expliquent et visualisent généralement les formes; alors que l'apprentissage automatique se concentre sur la maximisation des taux de reconnaissance. Les algorithmes de reconnaissance de formes fournissent normalement de meilleurs résultats pour les données fournies en entrée. Ils effectuent une correspondance des entrées en tenant compte de la variation statistique; ils ont donc pu obtenir de meilleurs résultats (Theodoridis & K. Koutroumbas, 2003) (Dutt, Vinita, Vikas , & Khan., 2012).

3.2 Système de reconnaissance des formes

La figure I.3 présente le système de reconnaissance des formes.

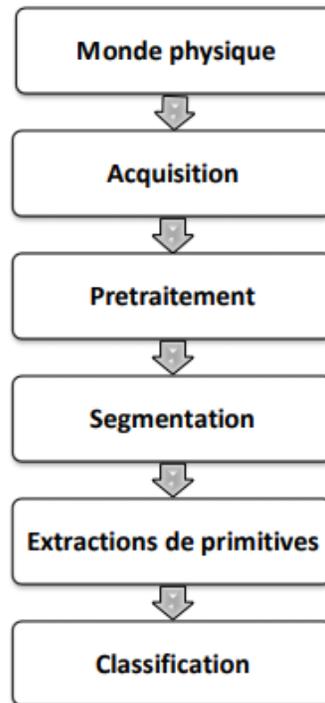


Figure I.3 Système de reconnaissance des formes

Rappelons, en premier lieu, le rôle de chaque étape dans l'élaboration du processus complet de reconnaissance des formes :

3.2.1 Acquisition

C'est la première étape du processus de reconnaissance du modèle, elle permet de transformer l'objet à reconnaître d'une forme réelle ou dite physique en une forme numérique compris par les ordinateurs. Cela peut être réalisé par un appareil photo ou un scanner placé dans un endroit ou un microphone pour la parole ou le son. Les données doivent être collectées et obtenues avec le plus grand soin et tout bruit ou biais dans les mesures effectuées à ce niveau est préjudiciable au système final. Il est également important de contrôler le processus d'acquisition et sa fiabilité pour que toutes les données soient collectées de la même manière et sans écart (Dutt, Vinita, Vikas , & Khan., 2012).

L'hypothèse importante est que les systèmes RF sont développés et distribués indépendamment. Les données doivent être indépendantes et produites par le même processus de base. (Dutt, Vinita, Vikas , & Khan., 2012)

3.2.2 Prétraitement

Les données d'entrée du monde réel contiennent toujours une certaine quantité de bruit et un prétraitement est nécessaire pour réduire son effet. Le terme "bruit" doit être compris au sens large: tout ce qui empêche un système de reconnaissance de formes de s'acquitter de ses tâches peut être considéré comme du bruit, quelle que soit la nature intrinsèque de ce "bruit" dans la nature des données. Certaines propriétés souhaitables des données peuvent également être améliorées avec un prétraitement avant que les données ne soient entrées dans le système de reconnaissance. (Dutt, Vinita, Vikas , & Khan., 2012)

Le prétraitement est normalement effectué par une méthode de filtrage simple sur les données. En reconnaissance d'image, l'image peut être filtrée à l'aide d'un filtre médian pour éliminer le bruit de point parasite qui pourrait gêner le processus de segmentation. Cette étape de prétraitement avantageuse pour les images couleur est la décorrélation des composants couleur. Un tel processus transfère une image initialement dans les coordonnées (Rouge, Verte, Bleu) de manière linéaire au YIQ (luminosity- inphase-quadrature). (Dutt, Vinita, Vikas , & Khan., 2012).

3.2.3 Segmentation

Les données d'entrée enregistrées et prétraitées doivent être divisées en sous-parties, ce qui donne des entités significatives pour la classification. Cette étape du traitement s'appelle la segmentation. Il peut s'agir d'un processus clairement séparé ou étroitement imbriqué avec les processus précédents ou suivants. Dans les deux cas, une fois que le système de reconnaissance de formes a terminé le traitement d'une totalité de données, la segmentation résultante des données en ses sous-parties peut être révélée. Selon la manière dont l'application a été réalisée, le bloc de

segmentation peut soit ajouter les informations concernant les limites des segments au flux de données, soit alternativement copier tous les segments dans des mémoires tampons séparés et les transférer un par un à l'étape suivante. (Cornelius & T. Leondes, 2012)

3.2.4 Extractions des primitives

L'extraction de primitives, également appelées caractéristiques, est une étape clé du processus de reconnaissance des formes. Il consiste à représenter les données d'entrées (mots, caractères, phonème, graphème, etc.) sous une forme de vecteur de primitives de dimension fixe, pour pouvoir reconnaître les formes où chaque vecteur se comporte comme une signature d'une forme. C'est une étape cruciale dans tout système de reconnaissance des formes. Le but de cette étape est la sélection des informations pertinentes, discriminantes et moins volumineuses afin de limiter les ressources de la machine nécessaires à la classification, tout en évitant de perdre des informations importantes et significatives. La plupart des techniques d'extraction de primitives dans le domaine de la reconnaissance de formes sont généralement accompagnées d'une perte irréversible d'informations. Il est donc nécessaire de faire un compromis entre quantité et qualité d'informations. (Rachid, 2018).

3.2.5 Classification

L'étape de classification joue un rôle primordial dans le processus complet d'un processus de reconnaissance de formes, en désignant l'appartenance d'une forme à une classe. En effet, la classification consiste à élaborer une règle de décision qui va transformer les attributs caractérisant les formes en fonction d'appartenance, c'est à dire un 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 de formes, il faut avoir procédé auparavant à deux étapes (Figure I. 1):

- L'étape d'apprentissage.
- L'étape de test.

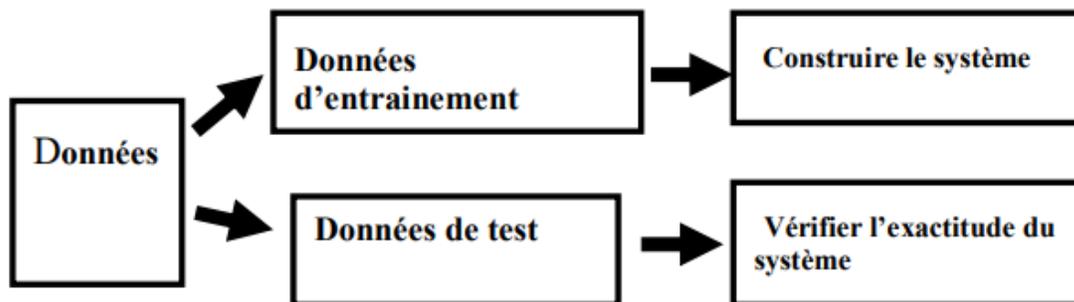


Figure I.4 Les étapes de classification

3.2.5.1 L'apprentissage

Est une étape clé dans le système de reconnaissance. Il consiste à fournir au système un ensemble de formes connues a priori (on connaît la classe de chacune d'elles). C'est cet ensemble d'apprentissage qui va permettre de régler le système de reconnaissance de façon à ce qu'il soit capable de reconnaître ultérieurement des formes de classes inconnues. On distingue généralement deux types d'apprentissage: apprentissage supervisé et apprentissage non supervisé (Rachid, 2018).

3.2.5.1.1 L'apprentissage supervisé

Consiste à fournir au module d'apprentissage un échantillon représentatif de l'ensemble des formes à reconnaître. Où l'opérateur de supervision ou professeur, indique l'étiquette correcte de chaque exemple, qui sera utilisée par le module d'apprentissage pour identifier la classe dans laquelle l'opérateur de supervision, souhaite que l'exemple soit arrangé. Donc la phase d'apprentissage, a pour objectif d'analyser les ressemblances entre les formes d'une même famille et les dissemblances entre les formes de Familles différentes pour en déduire les meilleures séparations de l'espace de représentations. Alors l'objectif général des méthodes d'apprentissage supervisé, est de construire ou approximer à partir de la base d'apprentissage, une règle ou une fonction de classification qui permet à partir de la description d'une forme, d'affecter la bonne étiquette ou classe à cette forme inconnue par le module d'apprentissage.

3.2.5.1.2 L'apprentissage non supervisé

On ne dispose d'aucune information de supervision, donc, on fournit au système de reconnaissance un grand nombre de formes non étiquetées. Et, c'est l'étape de la classification qui va se charger de regrouper automatiquement les formes jugées similaires (homogènes) dans la même partition de l'espace de représentation et placer les formes jugées dissimilaires (hétérogènes) dans des partitions distinctes. Ce type d'apprentissage, évite l'assistance d'un opérateur de supervision, mais il n'assure pas toujours, une classification correcte. L'étape de test permet l'évaluation de la performance du classifieur pour un apprentissage donné. C'est une étape très importante, car elle peut conditionner le choix des méthodes d'extraction de primitives ou le choix de la procédure d'apprentissage. En effet, la détermination a priori de l'ensemble des caractéristiques les plus discriminatives et la méthode d'apprentissage la plus adaptée pour un problème de classification donné, est une tâche très ardue, d'où l'utilité d'une procédure de test ou de validation qui procède par itérations successives. Ces itérations consistent à extraire des caractéristiques les plus discriminatives au problème de reconnaissance à résoudre et à tester la fiabilité du système avec cet ensemble de caractéristiques. Ces tests sont itérés tant qu'on n'a pas atteint les performances désirées. Ces itérations ont pour objectif, de régler quelques paramètres ou de tester une nouvelle famille de primitives, ou de combiner les primitives extraites avec d'autres types de primitives afin de perfectionner le module de classification. Comme nous avons

déjà mentionné ci-dessus, ces deux modules d'apprentissage et de test sont réalisés avant l'intégration du module de décision dans le système de reconnaissance de formes (Ahmed, 2016).

3.3 Méthodes de reconnaissance des formes

3.3.1 Méthode de correspondance de modèle

L'un des plus simples et les premières approches de la reconnaissance des formes est basé sur la correspondance de modèle. La correspondance est une opération générique dans la reconnaissance de modèle qui est utilisée pour déterminer la similarité entre deux entités (points, courbes ou formes) du même type. L'étape d'apprentissage de cette méthode consiste à stocker dans l'ordinateur un ensemble de modèles ou de prototypes de chaque classe, comme indiqué dans La Figure I.5 (Rachid, 2018).

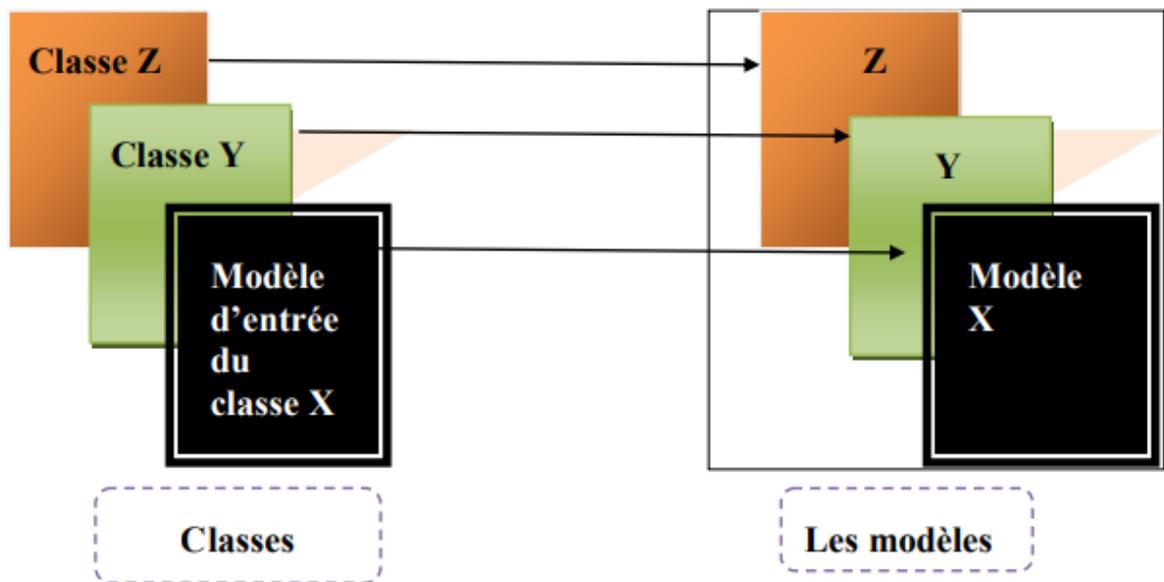


Figure I.5 Etape d'apprentissage de la méthode de correspondance de modèle et de corrélation

Lors de la phase de classification, une comparaison de l'image avec les modèles est effectuée: si la comparaison avec la catégorie X est supérieure au résultat de la comparaison avec la catégorie Y, elle est classée dans la catégorie X, etc. Comme le montre dans (la figure I.6).

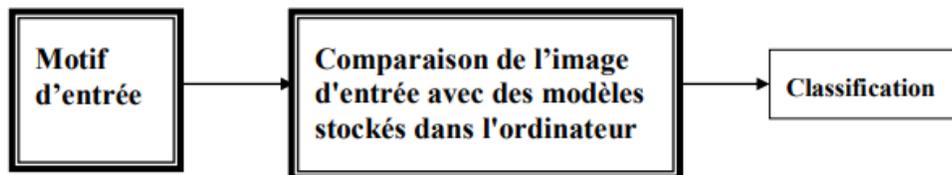


Figure I.6 Phase de la classification de la méthode de correspondance de modèle et de corrélation

La comparaison est effectuée après le stockage de l'image sous forme de matrice, elle est donc comparée aux modèles pixel par pixel et donne une valeur de comparaison. La seule difficulté de cette méthode réside dans le bon choix des modèles de chaque classe ainsi que dans la détermination des critères de comparaison, en particulier si l'image impliquée porte des distorsions. Par exemple, si nous utilisons cette méthode pour identifier les criminels, nous devons prendre plusieurs photos pour chaque criminel à stocker sur l'ordinateur, deux photos des deux côtés, une pour chaque côté, une de prise de vue avant et deux photos à un angle de 45° par rapport à la camera. Vous pouvez imaginer l'espace de stockage nécessaire pour tous ces modèles (Luo & Daisheng, 1998).

3.3.1.1 Corrélation

Pour savoir ou déterminer si l'image contient une zone similaire à une région, la corrélation est l'une des méthodes utilisées. Par exemple, nous avons une image numérique (x, y) avec une échelle de $M * N$ et nous voulons déterminer si cette image contient une zone similaire à une région de (x, y) avec une échelle de $J * K$ ou $J < M$ et $K < N$. Nous allons utiliser la corrélation entre ces deux régions comme dans l'équation suivante.

$$R(m, n) = \sum \sum f(x, y) \cdot (x - m, y - n) \dots (I.1)$$

Où : $n = 0, 1, 2, \dots, N - 1, m = 0, 1, 2, \dots, M - 1$

Pour toute valeur de (m, n) à l'intérieur de $F(x, y)$ Nous appliquons l'équation (I.1) pour obtenir une valeur pour R, et avec le changement de n,m, w(x,y) se déplace autour de la zone d'image en utilisant de la convolution, On obtient alors la fonction R(m,n), La valeur maximale de R(m,n) indique alors la position où les deux images w(x,y) et f(x,y) ont été appariées, Le processus de numérisation est identique à celui des deux figures (I.6.I.7) (qui montrant la numérisation d'image à l'aide de la convolution), Bien que la fonction de corrélation dans l'équation (I.1) soit simple par nature, il existe une fonction (I.2) qui est plus efficace en utilisation elle est comme suit.

$$R(m, n) = \frac{\sum_x \sum_y [f(x,y) - \bar{f}(x,y)] \cdot [w(x-m, y-n) - \bar{w}]}{\sum_x \sum_y [f(x,y) - \bar{f}(x,y)]^2 \cdot \sum_x \sum_y [w(x-m, y-n) - \bar{w}]^2}$$

Où : $n = 0, 1, 2, \dots, N-1, m = 0, 1, 2, \dots, M-1$

w est la valeur moyenne du modèle (cette valeur est calculée une fois) f(x,y) est la valeur moyenne de f(x,y) Dans la zone correspondante avec w(x,y) la valeur de R(x,y) est comprise dans la plage (-1, 1) où les valeurs positives élevées indiquent que la corrélation est positive et élevée entre les deux images) et les valeurs négatives élevées indiquent que la corrélation est négative et élevée entre les deux images, Tandis que les valeurs proches de zéro indiquent un manque de

corrélation entre les deux images. (Fernández Sánchez & z, Leticia, 2018)

3.3.1.2 Convolution

Où un cadre est défini pour l'image à partir de la première et de la dernière ligne, ainsi que la première et la dernière colonne et que le modèle est placé sur l'image à partir du premier élément comme indiqué dans les deux figures suivantes (I.7, I.8) c'est-à-dire que c'est une méthode de correspondance binaire entre les grands éléments d'image et les petits éléments d'image. Ensuite, nous déplaçons le modèle vers la droite d'un seul élément et renvoyons le processus de comparaison des éléments du modèle avec les éléments de la grande image jusqu'à la fin de la ligne, Passez ensuite à la ligne suivante pour répéter le processus, et ainsi de suite jusqu'à la fin de la grande image (Fernández Sánchez & z, Leticia, 2018).

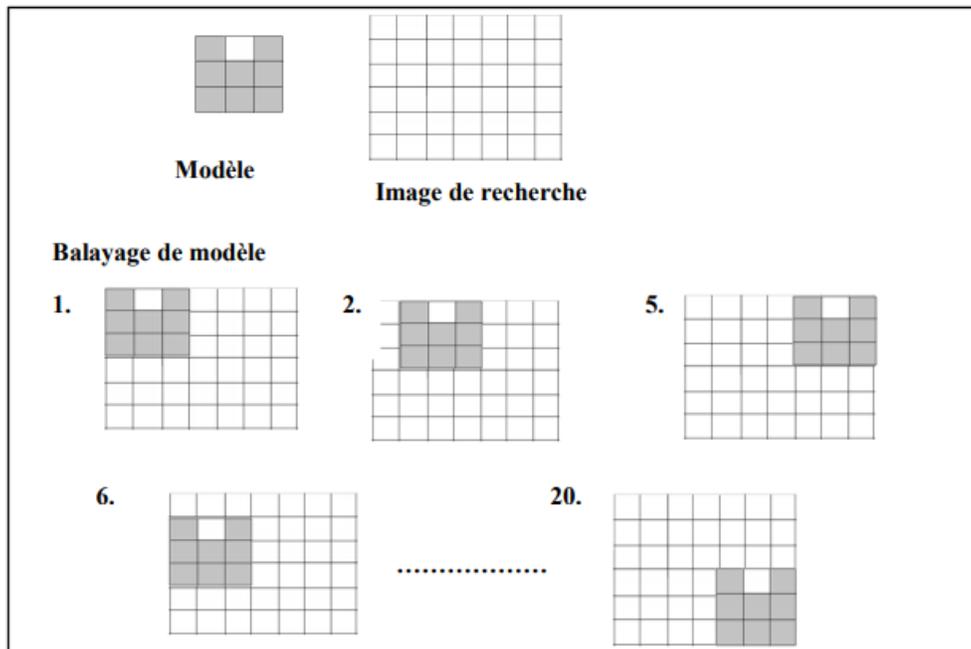


Figure I.7 Le processus de numérisation des images effectué à l'aide de la convolution

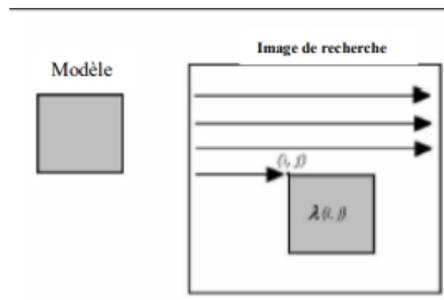


Figure I. 8 Numérisation de l'image à l'aide de convolution

3.3.2 Méthode statistique

Cette méthode crée un modèle statistique de l'objet à distinguer au lieu de créer un modèle en collectant un ensemble de motifs afin de former un ensemble de données statistiques, qui permet de créer un mécanisme de prise de décision. De cette manière, Le motif est représenté par un faisceau de propriétés extraites de l'élément à distinguer, de sorte qu'il apparaisse comme un point dans un espace multidimensionnel et que le nombre de dimensions correspond au nombre de propriétés du motif (Muallim & Mohammad Tarek, 2017).

Un ensemble de modèles forment un seul élément pour donner des informations sur cet élément , il est important de sélectionner ou de créer des propriétés permettant aux motifs d'appartenir au différents groupes et d'occuper des régions comprimées et non imbriquées dans l'espace. La figure (I.9) montre un exemple de séparation des bonnes propriétés et de mauvaises propriétés. Après le processus de filtrage, les limites de la résolution sont définies dans l'espace des propriétés, de sorte que les motifs appartenant à différentes classes soient séparés. (Muallim & Mohammad Tarek, 2017).

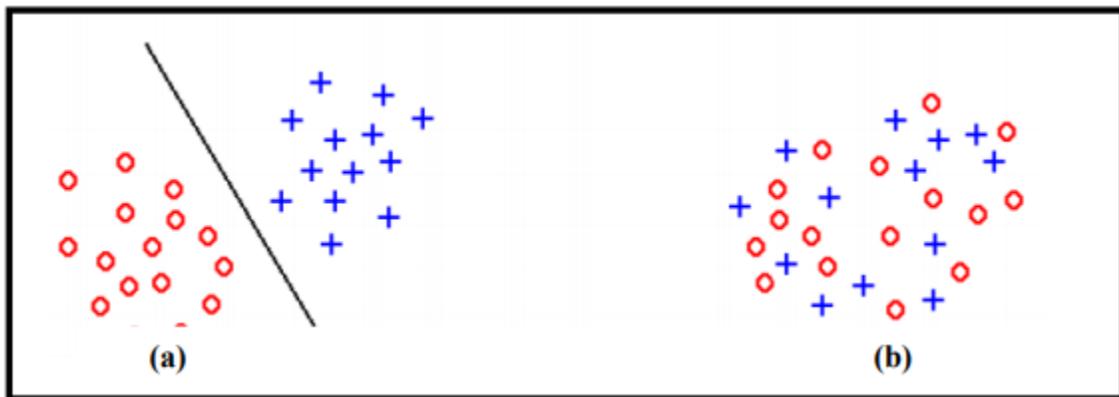


Figure I.9 Séparation des bonnes propriétés "a" et de mauvaises propriétés "b"

Dans la méthode de décision statistique, les limites de la décision sont déterminées par la distribution de probabilité des modèles appartenant à la même classe. Toutefois, si la distribution de probabilité des éléments est inconnue, les limites de décision sont construites à l'aide d'algorithmes d'apprentissage, qui sont divisés en l'apprentissage supervisé et l'apprentissage non supervisé.

3.3.3 Approche syntaxique et structurelle

Dans cette méthode, nous ne limitons pas seulement les valeurs numériques de chaque classe, mais ajoutons les relations entre les propriétés de chaque catégorie, qui nous fournissent les informations structurelles nécessaires pour identifier des modèles.

Dans la phase d'apprentissage de cette méthode, le modèle est généralement représenté sous la forme d'un arbre, d'un graphe ou une chaîne littérale des éléments primaires et des relations.

Le processus de classification dans cette méthode se fait à travers l'analyse des structures, l'analyse syntaxique ou, en d'autres termes, le programme d'analyse d'arabisation. Le rapport de comparaison le plus élevé de chaque image d'entrée avec chaque graphique ou une chaîne littérale ou un arbre spécifiant la catégorie à laquelle l'image d'entrée appartient. Prenons l'exemple d'une image d'entrée avec deux cercles (propriétés) si la distance les séparant est de 1 à 2 cm. il est possible de classer l'image comme une image des lunettes prenant en compte les autres caractéristiques et de leurs relations. Si la distance entre eux est d'environ un mètre il peut être classé comme feux de la voiture en tenant compte des autres caractéristiques.

Cette méthode est utilisée pour identifier des cibles, ou missiles, ainsi que pour la reconnaissance des caractères (Dutt, Vinita, Vikas , & Khan., 2012).

3.3.4 Approche de réseaux de neurones

Les réseaux de neurones sont un type d'algorithme assez complexe utilisé pour résoudre des problèmes non soumis à des lois fixes. Ils simulent la façon dont le cerveau humain reconnaît les sons, la parole et les images via un processus énorme et distribué, constitué de simples unités de traitement appelées neurones. Qui a une caractéristique neurologique car il stocke les connaissances scientifiques et les informations expérimentales et les met à la disposition de l'utilisateur en ajustant les poids.

Le réseau de neurones crée des neurones interconnectés avec des poids différents et chaque neurone est responsable d'une entrée. Lors du processus de formation, le réseau ajuste en permanence les poids associés à chaque neurone pour que la sortie soit aussi proche que possible.

Le processus de formation continue sur toutes les entrées disponibles et les poids associés à chaque neurone sont ajustés, à condition que la valeur résultante soit proche de la valeur de sortie réelle. la figure I.10 montre les composants de réseau de neurone artificiel. (yadav, 2019).

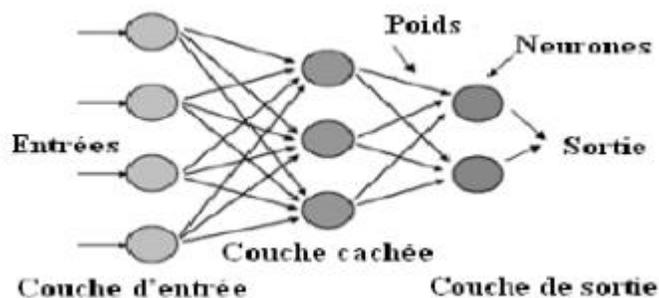


Figure I.10 Composants de réseau de neurone artificiel

4 L'importance du système de reconnaissance des plaques d'immatriculation dans l'entrée et la sortie du stationnement

L'utilisation d'un système intelligent avancé de reconnaissance des plaques d'immatriculation dans les stationnements a apporté beaucoup de commodité pour les gens à se garer. Cependant, la demande actuelle de stationnement est en constante évolution avec l'augmentation des véhicules chaque jour. En tant que seul système de recharge de stationnement dans le stationnement, il doit non seulement adopter un modèle de gestion avancé, mais aussi d'assurer les avantages du parking, L'ensemble du processus de chargement doit être très rigoureux. Le système moderne de recharge de stationnement implique tous les aspects de la gestion, dont la gestion des véhicules est un aspect important, en particulier dans certaines occasions spéciales, il nécessite une gestion stricte de divers véhicules en temps réel, et un contrôle strict de leur temps d'accès surveiller et enregistrer et gérer divers véhicules. Cependant, pour certains grands stationnements, il y a beaucoup de véhicules qui entrent et sortent. Juger et traiter par l'homme est non seulement long, mais aussi pas propice à la gestion et à l'enquête, ce qui rend le travail inefficace



Figure I.11 Une photo du parking moderne.

La gestion traditionnelle du parking adopte principalement le type de carte/billet. Le propriétaire doit s'arrêter et récupérer la carte lors de glisser dans et hors de passer. Cependant, en cas de pluie et de neige, la carte est facile à mouiller lorsque la carte est glissée et. Pour le stationnement en montée et en descente avec carte de balayage, il est plus susceptible de causer des accidents tels que le roulement et la collision. Aujourd'hui, la transformation intelligente des parkings est dans une période dorée, et comme la technologie de base des parkings intelligents, la valeur des systèmes automatiques de reconnaissance des plaques d'immatriculation est devenue de plus en plus élevée. Intégrant l'intelligence et la haute définition, la technologie de reconnaissance des plaques d'immatriculation nous a aidés à résoudre de nombreux problèmes dans la vie. (Goy, 2019)

5 Les dernières technologies utilisées pour détecter les matricules

La recherche scientifique et le besoin urgent dans ce domaine ont contribué au développement du système DRPI (détection et reconnaissance plaque d'immatriculation) en aboutissant à des solutions exigeant des plateformes et des logiciels tels que :

5.1 LAPI (Lecture Automatique de Plaques d'Immatriculation)

La reconnaissance de plaques minéralogiques ou LAPI repose sur le procédé informatique de Reconnaissance Optique de Caractères (ROC). Ce procédé permet d'isoler et de capturer l'image d'une plaque minéralogique, d'en extraire les caractères alphanumériques et de les convertir en données numériques afin de les analyser et de les comparer avec une base de données. Les algorithmes des logiciels LAPI sont en mesure de gérer la position du véhicule (orientation, distance, vitesse...), les problèmes d'éclairage et de contraste ainsi que les éventuelles barrières physiques (immatriculation couverte de poussière, police d'écriture fantaisie...). Leur vitesse d'exécution permet d'analyser un nombre élevé d'images prises en temps réel sur des axes de circulation à grande vitesse. (Ahmed, 2016)

5.1.1 Caractéristiques

Le système LAPI est en mesure de posséder les caractéristiques suivantes [3] :

- Lecture et reconnaissance de plaques d'immatriculation des pays de l'Union Européenne et d'Afrique.
- Système automatique 24h/24 h et indépendant du matériel .La solution fonctionne avec tout type de caméras IP répondant à nos préconisations techniques.
- Interface simple à configurer disponible sur PC Windows et PC linux en 2 langues (Français/Anglais)
- Gestion des autorisations d'entrées et sorties
- Possibilité de chiffrer les données (numéros de plaques et photos)
- Les produits qui appliquent système LAPI :

5.1.2 LAPI-ACCES

C'est une solution logicielle de contrôle d'accès par lecture automatique de plaques d'immatriculation s'adressant au marché de la sécurité. Le logiciel intègre de nombreuses innovations algorithmiques permettant la reconnaissance en temps réel, de jour comme de nuit, des plaques d'immatriculation de tous types de véhicules (autos, motos, poids lourds) passant dans les champs de vision des caméras. LAPI ACCES est un produit autonome et indépendant du matériel disposant d'un taux de reconnaissance optimal. (Ćelić, 2012)

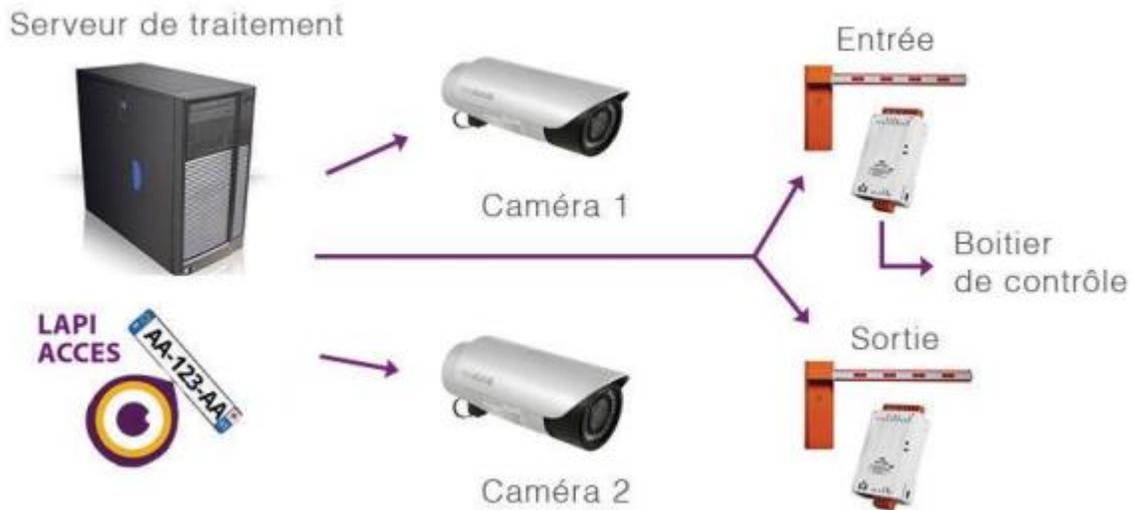


Figure I.12 Exemple d'architecture matérielle de système LAPI

5.1.3 LAPI-ENGINE

Le produit LAPI ENGINE représente le cœur technologique permettant la Lecture Automatique de Plaques d'Immatriculation (LAPI - ANPR) qui a été intégré dans l'ensemble de solutions. Principalement dédiée à l'identification et la traçabilité de véhicules, LAPI ENGINE est un produit autonome pouvant s'adapter à un large éventail d'applications. Il s'intègre facilement dans des solutions "métiers" comme des logiciels de contrôle d'accès, de pesée industrielle ou de vidéosurveillance (Ćelić, 2012)

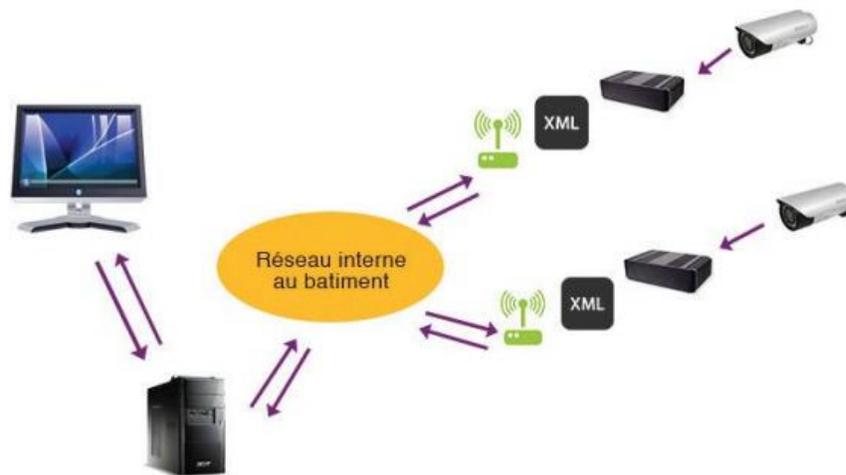


Figure I.13 Exemple d'architecture matérielle di système LAPI-NETWORK

5.1.4 LAPI-SMARTPHONE

La solution LAPI SMARTPHONE permet la détection et la reconnaissance automatique de plaques d'immatriculation à partir de l'analyse d'un flux vidéo ou d'une photo, en temps réel. Dédiée principalement à la gestion et au contrôle de stationnement, cette application permet le contrôle d'une plaque minéralogique par rapport à une base de données locale ou distante et s'intègre facilement à d'autres systèmes de gestion (Ćelić, 2012).

5.2 Système VIRONIC

Les systèmes de reconnaissance des plaques d'immatriculation POLISCAN SURVEILLANCE de VITRONIC trouvent leur place partout où l'accès est réservé à un type ou à une catégorie de véhicules. (Mirwanti, 2020)

A. Fonctionnement

Le système POLISCAN SURVEILLANCE combine la technologie de reconnaissance performante. VITRONIC et un logiciel back office qui effectue des comparaisons entre les plaques d'immatriculation relevées et celles d'une base de données. Les plaques d'immatriculation qui ne correspondent pas à la base de données sont immédiatement supprimées. Les applications de type liste noire du système POLISCAN SURVEILLANCE incluent l'aide à la recherche, les contrôles aux frontières et la localisation de véhicules volés ou avec défaut d'assurance (Mirwanti, 2020).

Dans le cas d'une procédure dite « de liste blanche », la base de données contient des immatriculations autorisées. Avec le contrôle d'accès par liste blanche, le système signale tout véhicule qui n'est pas enregistré. Le système POLISCAN SURVEILLANCE ne se limite pas à la reconnaissance des plaques d'immatriculation. Il peut aussi identifier d'autres caractéristiques graphiques, telles que des vignettes ou des macarons, et surveiller la circulation en continu. Il permet donc d'identifier, par exemple, tout véhicule sans pastille verte dans la file de circulation. Il peut aussi surveiller les véhicules qui empruntent les couloirs réservés aux bus. (Mirwanti, 2020)

B. Avantages

- Requête automatique dans des bases de données d'immatriculation.
- Prise en charge des procédures de type liste noire/liste blanche.
- Taux de lecture extrêmes, y compris à vitesse élevée.
- Cryptage et transfert sélectif pour la protection et la sécurité des données.



Figure I.14 Utilisation du système VIRONIC pour la surveillance routière

5.3 Système Auto-Vu

C'est le système de reconnaissance de plaques d'immatriculation (RAPI) sur IP de Security Center, la plate-forme de sécurité unifiée de Genetec. Que ce soit dans votre véhicule ou au bureau, Auto-Vu vous aide à automatiser l'identification des plaques d'immatriculation des véhicules. Auto-Vu est une solution éprouvée qui permet aux organisations d'améliorer diverses applications, telles que la surveillance urbaine, la sécurité, le contrôle d'accès des véhicules, le maintien de l'ordre et le stationnement. (Servais, 2019)

A. Fonctionnalités

Système intégré de vidéosurveillance : Fusionnez le système de reconnaissance automatique de plaques d'immatriculation avec le contrôle d'accès et la vidéosurveillance dans Security Center. Ainsi effectuer une surveillance vidéo en direct à l'aide des lectures de plaques d'immatriculation et recevoir des alertes lorsque le système Auto-Vu repère un véhicule recherché. Suivi du comportement et des caractéristiques des véhicules : Auto-Vu analyse les caractéristiques et les comportements des véhicules scannés, vous permettant de suivre les tendances et d'identifier les véhicules suspects. Outre la capture précise des numéros des plaques d'immatriculation, Auto-Vu peut identifier la marque et le sens de déplacement des véhicules, estimer leur vitesse et détecter l'État ou le pays d'origine de leur plaque. Event-to-Action : Utilisez les correspondances d'événements et lectures RAPI pour déclencher des actions avancées, telles que l'enregistrement vidéo ou l'activation de barrières via le puissant module Event-to-Action de Security Center. (Servais, 2019)

B. Avantages

- Obtenez des lectures de plaques d'immatriculation précises
- Assurez des performances ininterrompues
- Fournisseur de solutions de RAPI complètes
- Unification avec la vidéosurveillance
- Combinez la RAPI fixe et mobile

5.4 OpenALPR

C'est une bibliothèque open source de reconnaissance automatique de plaques d'immatriculation écrite en C++ avec des liaisons en C#, Java et Python. Ce logiciel est distribué à la fois dans une version commerciale basée sur le cloud et dans une version open source. OpenALPR utilise les bibliothèques OCR OpenCV et Tesseract. Il peut être exécuté comme un utilitaire de ligne de commande, une bibliothèque autonome ou un processus d'arrière-plan. Le logiciel s'intègre également aux systèmes de gestion vidéo (VMS) tels que Milestone XProtect. La bibliothèque analyse les images et les flux vidéo pour identifier les plaques d'immatriculation. La sortie est la représentation textuelle de tout caractère de plaque d'immatriculation. OpenALPR comprend un utilitaire de ligne de commande. Il suffit de taper "alpr [chemin du fichier image]" pour commencer à reconnaître les images de la plaque d'immatriculation. (Singh, 2019)

A. Avantages

Le logiciel peut être utilisé de différentes manières. Par exemple, avec OpenALPR, vous pouvez:

- Reconnaître les plaques d'immatriculation des flux de caméras. Les résultats sont consultables et peuvent déclencher des alertes. Le référentiel de données peut être dans le cloud ou être entièrement stocké dans votre réseau sur site.
- Reconnaître les plaques d'immatriculation des flux de caméras et envoyer les résultats à votre propre application.
- Traiter un fichier vidéo et stocker les résultats de la plaque d'immatriculation dans une base de données CSV et SQLite.
- Analyser des images fixes à partir de la ligne de commande
- Intégrer la reconnaissance de plaque d'immatriculation dans votre application directement dans le code (C / C++, C#, VB.NET, Java, Python, Node.js).

6 Autres applications utilisant le numéro d'immatriculation

Il existe beaucoup d'applications qui bénéficient de la reconnaissance des plaques d'immatriculation, nous citons quelques-unes d'entre elles :

a) Contrôles de paiement

Pour calculer les frais de voyage on utilise le matricule et c'est le cas des passages de péage autoroutiers

b) Contrôle des frontières

Le numéro d'immatriculation est enregistré dans une base de données avec autres informations comme le passeport pour contrôler tous les passagers frontaliers et surveiller son déplacement.

c) Protection de voitures

Le vol est l'une des choses les plus importantes que les propriétaires craignent, donc la reconnaissance d'immatriculation peut être utilisée pour l'empêcher, ce système réalise une comparaison entre les voitures qui passent et la liste des voitures volées puis une alerte est déclenchée pour informer l'agent de police de la voiture détectée.

d) Les systèmes de gestion de la circulation

Qui calculent la vitesse de circulation en mesurant le temps entre les passages devant deux points de lecture.

e) Système d'information Schengen

Détection de matricule pour le stocker dans une base de données des VISA à des raisons de sécurité.

7 Conclusion

Dans ce chapitre nous avons défini les notions de bases liées à la détection et la reconnaissance de plaque d'immatriculation, dont nous avons réservé une partie à la reconnaissance des formes et aux techniques existant pour la détection automatique des plaques d'immatriculation.

Dans le prochain chapitre nous allons présenter en détail les réseaux neuronaux ainsi que les caractéristiques et les architectures de l'apprentissage profond.

Chapitre 2 :

L'apprentissage profond

1 Introduction

Ces dernières années, une grande partie de l'innovation moderne en matière de la technologie d'apprentissage. Un type avancé d'apprentissage automatique est la merveille moderne de l'intelligence artificielle. L'approche d'apprentissage profond de la reconnaissance d'image implique l'utilisation de réseaux neuronaux convolutionnels.

Dans ce chapitre nous allons définir les réseaux neuronaux et ses architectures ensuite nous allons détailler les notions de l'apprentissage profond.

2 Réseaux neuronaux

2.1 Définition

Les réseaux neuronaux sont un modèle informatique qui partage certaines propriétés avec l'être humain (le cerveau) dans lequel de nombreuses unités fonctionnent en parallèle, sans unité de contrôle centralisée. Les pondérations entre les unités sont les principaux moyens de calcul à long terme. Pour stocker des informations dans les réseaux neuronaux. La mise à jour des poids est le principal moyen d'apprentissage de nouvelles informations par le réseau neuronal (Patterson, 2017).

2.2 Architectures de réseaux de neurone

La façon dont les neurones d'un réseau neuronal sont structurés est intimement liée à l'algorithme d'apprentissage utilisé sur le réseau, nous concentrons l'attention sur les architectures de réseau (structures). En général, nous pouvons identifier trois classes fondamentalement différentes d'architectures de réseaux :

- Réseaux d'anticipation à couche unique.
- Réseaux d'anticipation multicouches.
- Réseaux récurrents.

2.2.1 Réseau d'anticipation à une seule couche

Les neurones sont organisés en couches. Dans le plus simple en tant que réseau superposé, nous avons une couche d'entrée de nœud source qui projette directement vers une couche de neurones de sortie (nœuds de calcul). Elle est illustrée dans la figure II.1.

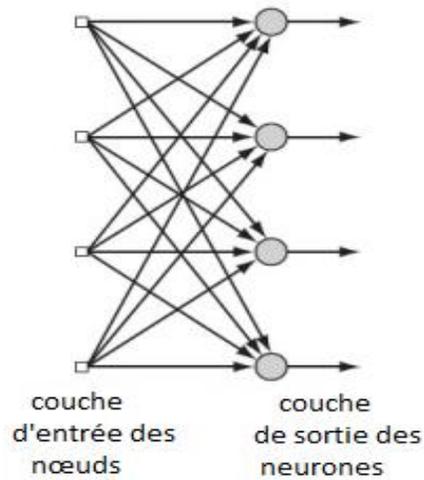


Figure II. 1 Réseau d'alimentation avec une seule couche de neurones (Subarna, 2019)

2.2.2 Réseaux d'alimentation multicouches

Les réseaux d'alimentation multicouches se distinguent par la présence d'une ou de plusieurs couches cachées. La carte architecturale de la figure II.2. Illustre la structure d'un réseau neuronal d'anticipation multicouche. Pour le cas d'une seule couche cachée.

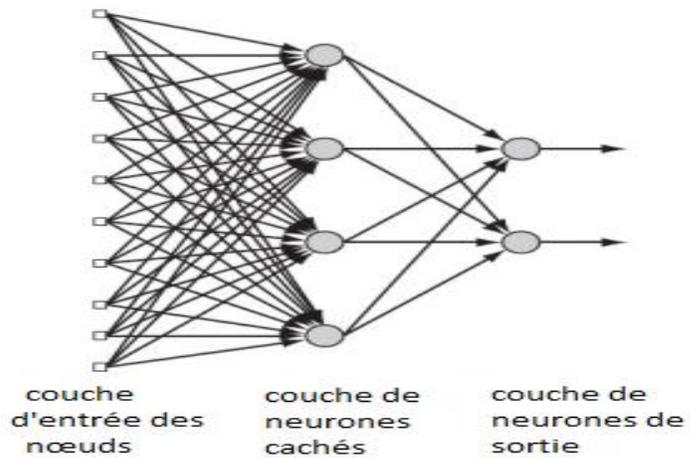


Figure II. 2 Réseau d'alimentation entièrement connecté avec une couche cachée et une couche de sortie (Subarna, 2019)

2.2.3 Réseaux récurrents

Un réseau neuronal actuel se distingue d'un réseau neuronal anticipatif en ce qu'il possède au moins une boucle de rétroaction (Haykin, 2008). Par exemple, un réseau récurrent peut consister en une seule couche de neurones, chaque neurone fournissant son signal de sortie aux entrées de tous les autres neurones, comme le montre le graphique architectural de la figure II.3.

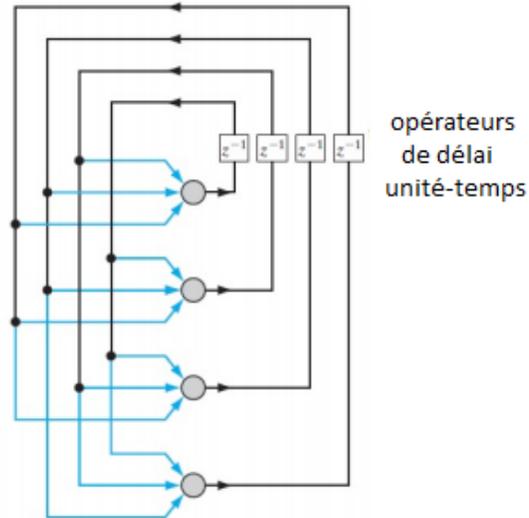


Figure II. 3 Réseau récurrent sans boucles d'auto rétroaction et sans neurones cachés (Haykin, 2008)

Dans la figure II.4., nous illustrons une autre classe de réseaux récurrents avec des neurones cachés. Les connexions de rétroaction illustrées proviennent également de neurones cachés des neurones de sortie.

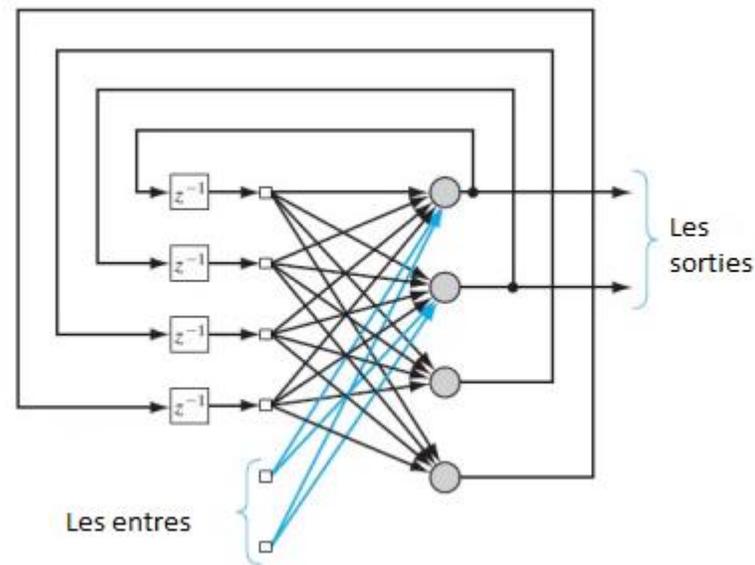


Figure II. 4 Réseau récurrent avec neurones cachés (Haykin, 2008)

2.3 Types d'apprentissage automatique

Les types les plus importants et essentiel pour l'apprentissage sont :

- Apprentissage supervisé.
- Apprentissage non supervisé.
- Renforcement de l'apprentissage.
- Apprentissage semi-supervisé.

2.3.1 Apprentissage supervisé

L'apprentissage supervisé est une fonction de prévision composée d'exemples, qui constituent une base d'apprentissage. Le but de l'apprentissage supervisé est alors de construire la base d'apprentissage, des classificateurs, ou des fonctions de classement. Une telle fonction nous a permis de reconnaître un objet que nous avons sa description, Figure II.5.

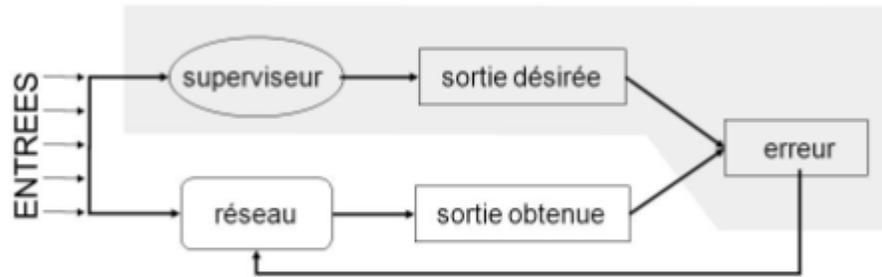


Figure II. 5 Apprentissage supervise

2.3.2 Apprentissage non supervisé

L'apprentissage non supervisé est l'inverse de l'apprentissage supervisé. il s'agit de trouver une structure qui correspond au problème à prévoir mais à partir des données non annotées, il n'est pas possible de faire de l'algorithme un résultat souhaité, l'absence de données annotées ou de base d'apprentissage, ce qui signifie un apprentissage non supervisé, Figure II.6.

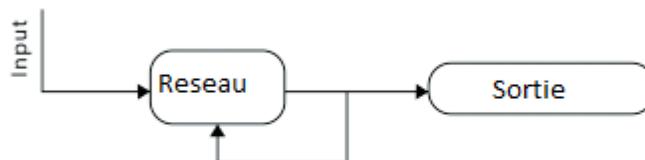


Figure II. 6 Apprentissage non-supervise

2.3.3 Apprentissage du renforcement

Pour l'apprentissage du renforcement, pour un agent autonome (robot, etc.), apprendre les actions à prendre, à partir d'expériences, afin d'optimiser une récompense quantitative au fil du temps. L'agent est immergé dans un environnement et prend ses décisions en fonction de son état actuel. En retour, l'environnement offre à l'agent une récompense, qui peut être positive ou

négative. L'agent recherche, à travers des expériences itératives, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associant à l'état actuel l'action à effectuer) optimal, en ce qu'il maximise la somme des récompenses au fil du temps (Mitchell, 1997). Figure II.7.

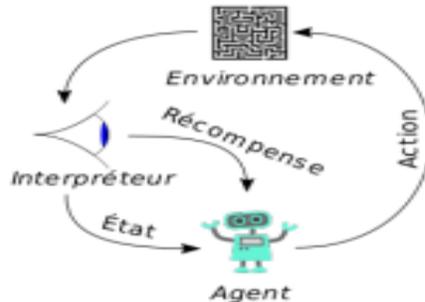


Figure II. 7 Apprentissage du renforcement (Reinforcement learning, 2019).

2.3.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé utilise un ensemble de données étiquetées et non étiquetées; c'est donc entre l'apprentissage supervisé qui utilise uniquement des données étiquetées et l'apprentissage non supervisé qui n'utilise que des données non étiquetées (Blum, 2000). L'apprentissage semi-supervisé vise à résoudre les problèmes de reconnaissance des formes avec une petite quantité de données étiquetées pour un grand nombre de points sans étiquettes. Ces derniers points utilisés pour démêler la structure sous-jacente des données, il est pertinent d'utiliser des graphiques de similitude pour représenter cette structure (Belkin, 2006). Il a été démontré que l'utilisation de données non étiquetées en combinaison avec des données étiquetées améliore considérablement la qualité de l'apprentissage. Un autre avantage est que l'étiquetage des données nécessite un utilisateur humain. Lorsque les ensembles de données deviennent très grands, cette opération peut être fastidieuse.

3 L'apprentissage profond

3.1 Définition

L'apprentissage profond est un type spécial d'apprentissage automatique qui atteint une grande puissance et flexibilité dans l'apprentissage pour représenter le monde comme une hiérarchie imbriquée de concepts, chaque concept étant défini en fonction de concepts plus simples et de représentations plus abstraites calculées en termes moins abstraits (Goodfellow, 2021).

3.2 Architectures d'apprentissage profond

L'architecture majeure de l'apprentissage profond est comme suite :

- Réseaux préétablis non supervisés.
- Réseaux neuronaux récurrents.
- Réseaux de neurones récursifs.
- Réseaux neuronaux convolutifs (RNC).

3.2.1 Réseaux préétablis non supervisés

Dans ce type nous allons traiter 3 architectures spécifiques :

- Auto-encodeurs.
- Réseaux de croyances profondes.
- Les réseaux génératifs accusatoires.

A. Auto encodeurs

Les auto-encodeurs sont des circuits d'apprentissage simples qui visent à transformer les entrées en sorties avec le moins de distorsion possible. Bien que conceptuellement simple, ils jouent un rôle important dans l'apprentissage automatique. Les encodeurs automatiques ont été introduits dans les années 1980 par Hinton et le PDP pour s'attaquer au problème de la rétro-propagation sans enseignant, en utilisant les données d'entrée en tant qu'enseignant (Baldi P. , jan 2012). Ceci est résumé dans la figure II.8.

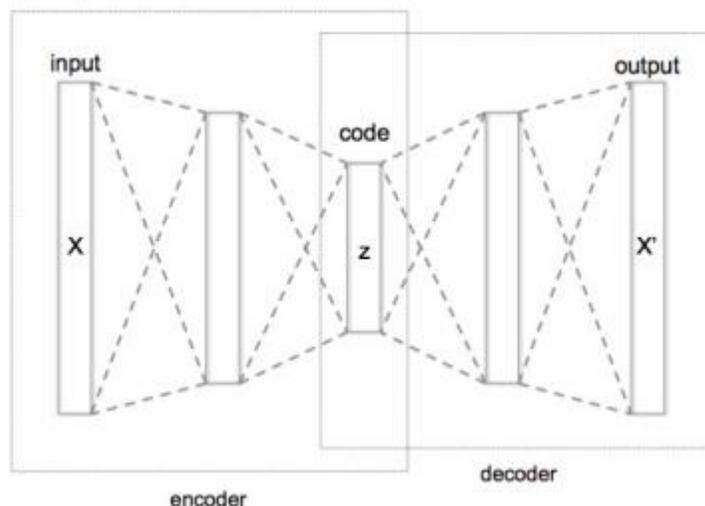


Figure II. 8 Structures d'auto-encodeurs (Baldi P. , jan 2012).

Réseaux de croyances profondes

Réseaux de croyances profondes (RCP) est une pile de nombreuses machines Boltzmann restreintes (MBRs) telles que la couche cachée de chaque MBR agit comme une couche visible pour le prochain MBR. La couche visible du premier MBR est également une couche visible de RCP et toutes les autres couches sont des couches masquées. RCP est formé en formant un MBR à la fois. Une fois que le premier MBR a été formé, les échantillons sont simplement acheminés vers lui et la sortie produite à la couche cachée de celui-ci est servi comme entrée à la couche visible du prochain MBR et ainsi de suite. C'est ce qu'on appelle une couche de préformation de RCP (Khan, 2018). La figure II.9. montre l'architecture de base du RCP.

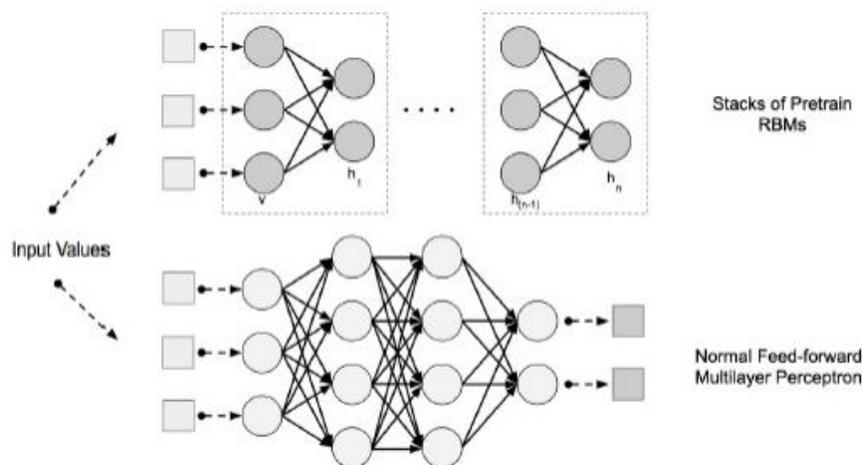


Figure II. 9 Architecture RBD (Patterson, 2017).

B. Réseaux génératifs accusatoires

Les réseaux de confrontation génériques (RGA) sont une technique émergente d'apprentissage semi-supervisé et non supervisé. Ils réalisent cette modélisation implicite des distributions de grandes données. Offres en 2014 (Creswell, 2017). La figure II.10. pourrait nous donner une idée générale du fonctionnement des (RGA).

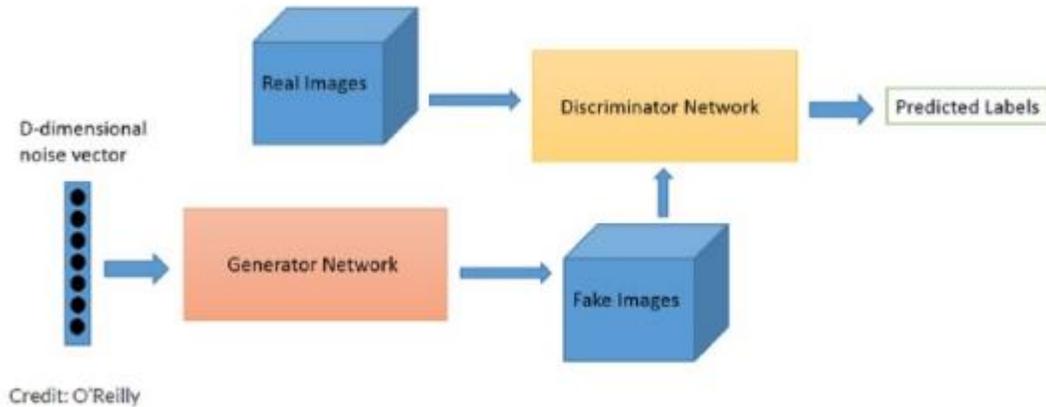


Figure II. 10 Idée de base des PAC (Gersey, 2018).

3.2.2 Réseaux neuronaux récurrents(RNN)

Les réseaux neuronaux récurrents appartiennent à la famille des réseaux neuronaux d'avance. Ils se distinguent des autres réseaux de prédictif par leur capacité à envoyer de l'information au fil du temps (Patterson, 2017). Les ARN techniques ont été appliqués à une grande variété de problèmes (R. Medsker, jan 1999). Les réseaux récurrents peuvent généralement être classés en réseaux récurrents globaux, dans lesquels les connexions de rétroaction entre chaque neurone sont autorisées, et les réseaux récurrents localement, globalement et prospectivement avec la dynamique réalisée au sein des modèles neuronaux (Du, dec 2012). Les RNN ont des architectures varier la figure II.11. Montre un exemple du RNN complètement connecté les uns aux autres.

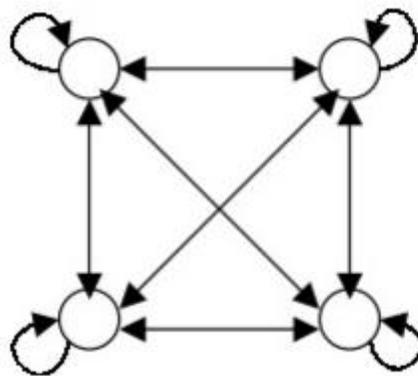


Figure II. 11 Exemple de réseau neuronal récurrent entièrement connecté (R. Medsker, jan 1999)

La figure II.12. Montre un exemple de RNN ayant un lien simple et partiel, bien que certains nœuds fassent partie d'une structure de prédictif.

Les poids sont représentés par (C1 et C2).

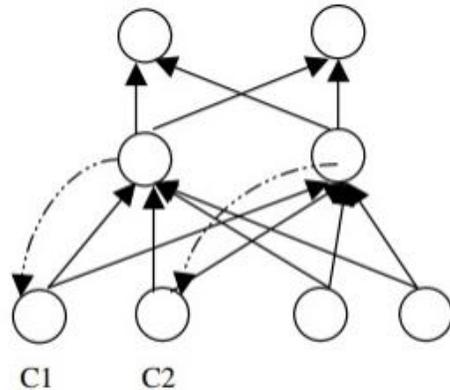


Figure II. 12 Exemple de réseau récurrent simple (R. Medsker, jan 1999)

3.2.3 Réseaux neuronaux récurrents (RNRs)

Les réseaux neuronaux récurrents sont des modèles adaptatifs non linéaires qui sont : capables d'apprendre des informations structurées en profondeur (China Manrique de Lara, Nov 2009). Semble très approprié pour la classification. Les réseaux de neurones récurrents ont été utilisés dans certaines applications, notamment l'analyse des sentiments, la description des images et la détection de la paraphrase (R. Bowman, June 2014). La figure II.13.montre l'architecture la plus simple pour les RNRs.

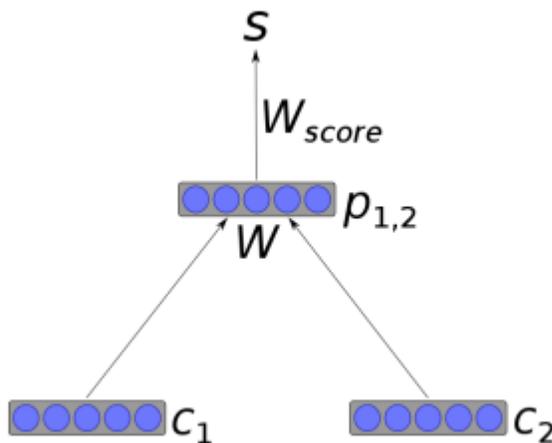


Figure II. 13 Une architecture de réseau neuronal récurrent simple (Recursive neural network, n.d.)

3.2.4 Réseaux neuronaux convolutifs (RNC)

Les RNCs (CNNs) sont un type de réseau neuronal artificiel acyclique (Prédicatifs), dans lequel le schéma de connexion entre les neurones est inspiré par le cortex visuel de l'animal. Les neurones de cette région du cerveau sont disposés de manière à correspondre aux régions qui se chevauchent lors du pavage du champ visuel (Réseau neuronal convolutif, Visited 01-03-2021.).

Le but d'un CNN est d'apprendre des caractéristiques d'ordre supérieur dans les données via des convolutions. Ils sont bien adaptés à la reconnaissance des objets avec des images et à des concours de classement des images toujours supérieurs. Ils peuvent identifier les visages, les individus, les panneaux de rue, les ornithorynques et de nombreux autres aspects des données visuelles. Les CNN se chevauchent avec l'analyse de texte par reconnaissance optique des caractères, mais ils sont également utiles pour analyser les mots en tant qu'unités textuelles discrètes. Ils sont aussi bons pour analyser le son. La Figure II.14 illustre que les CNN sont bons pour établir la position et (quelque peu) les caractéristiques invariantes de rotation à partir des données d'image brutes.

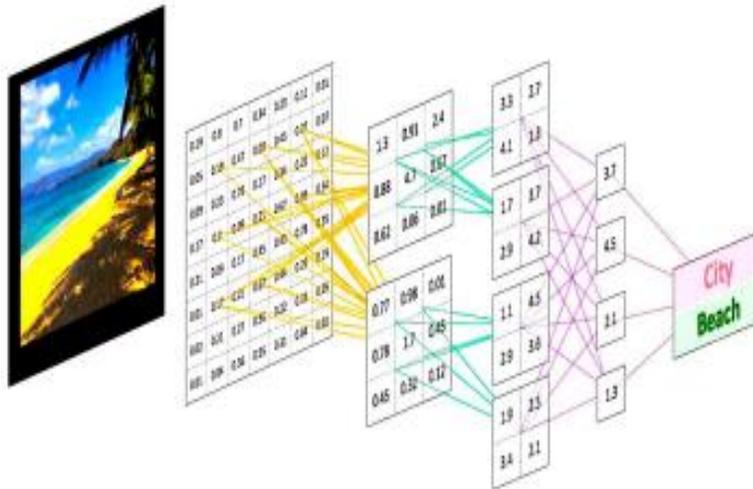


Figure II. 14 RNC et la vision par ordinateur (Patterson, 2017).

A. Inspiration biologique

L'inspiration biologique des RNC est le cortex visuel chez les animaux. Les cellules du cortex visuel sont sensibles aux petites sous-régions de l'entrée. Nous appelons cela le champ

visuel (ou champ réceptif). Ces sous-régions plus petites sont carrelées ensemble pour couvrir l'ensemble du champ visuel. Les cellules sont bien adaptées pour exploiter la forte corrélation spatiale locale trouvée dans les types d'images que nos cerveaux traitent, et agissent comme des filtres locaux sur l'espace d'entrée. Il y a deux classes de cellules dans cette région du cerveau. Les cellules simples s'activent lorsqu'elles détectent des motifs en forme de bord, et les cellules plus complexes qui s'activent lorsqu'elles ont un champ réceptif plus grand et sont invariantes à la position du motif (Patterson, 2017).

B. Présentation de l'architecture de RNC(CNN)

Les RNCs transforment les données d'entrée de la couche d'entrée à travers toutes les couches connectées en un ensemble de scores de classe donnés par la couche de sortie. Il existe de nombreuses variantes de l'architecture de RNC, mais elles sont basées sur le modèle des couches, comme le montre la figure II.15.

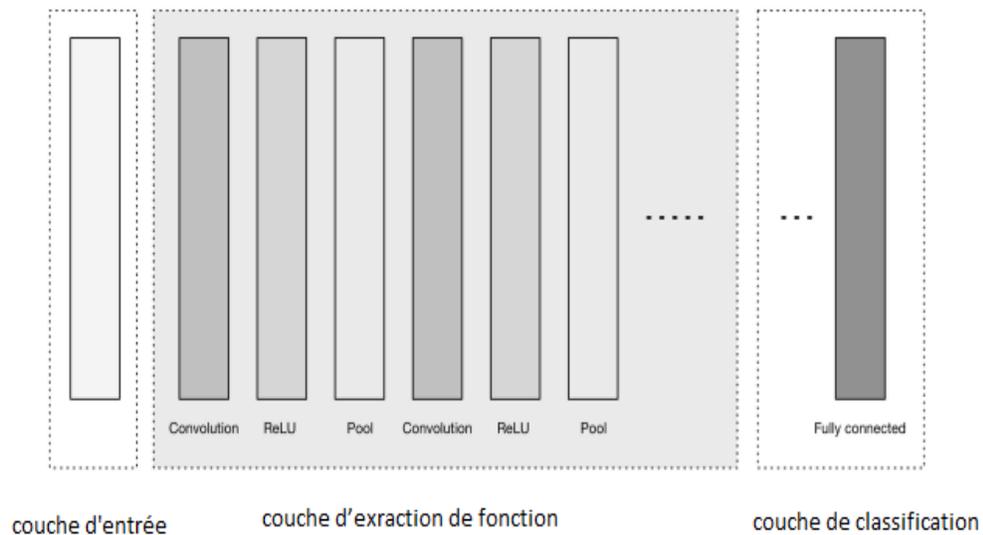


Figure II. 15 Architecture générale de haut niveau de RNC (Patterson, 2017).

➤ Couches d'entrée

Les couches d'entrée sont là où nous chargeons et stockons les données d'entrée brutes de l'image pour le traitement dans le réseau. Ces données d'entrée spécifient la largeur, la hauteur et le nombre de canaux. Typiquement, le nombre de canaux est de trois, pour les valeurs RVB pour chaque pixel.

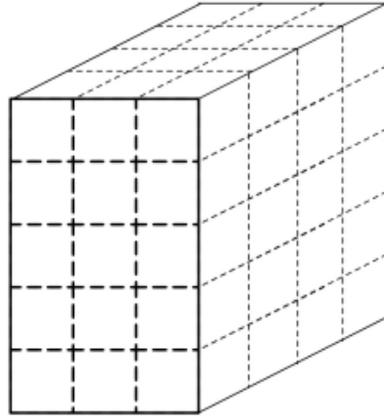


Figure II. 16 Volume 3D de la couche d'entrée (Patterson, 2017)

➤ Couches convolutionnelles

Les couches convolutionnelles sont considérées comme les éléments de base des architectures RNC. Comme la montre la figure II.17. Les couches convolutionnelles transforment les données d'entrée en utilisant un patch de neurones connectés localement de la couche précédente. La couche calculera un produit de point entre la région des neurones dans la couche d'entrée et les poids auxquels ils sont connectés localement dans la couche de sortie.

➤ Convolution :

Opération mathématique décrivant une règle de fusion de deux ensembles d'information. La figure II.18 est connue comme le détecteur de fonction d'un CNN. L'entrée à une convolution peut être des données brutes ou une sortie de carte de fonction d'une autre convolution. Il est souvent interprété comme un filtre dans lequel le noyau filtre les données d'entrée pour certains types d'informations, par exemple, un noyau de bord laisse passer seulement l'information du bord d'une image (Patterson, 2017).

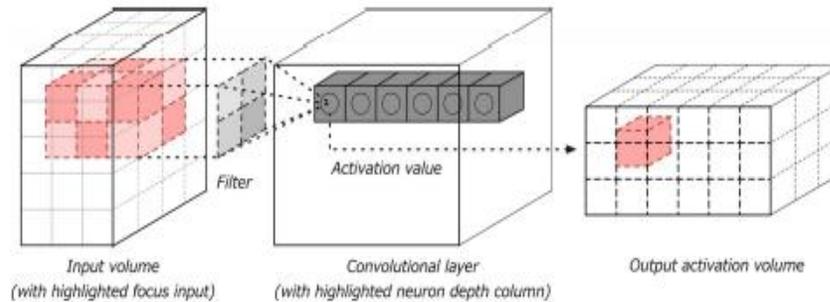


Figure II. 17 Couche de convolution avec volumes d'entrée et de sortie (Patterson, 2017)

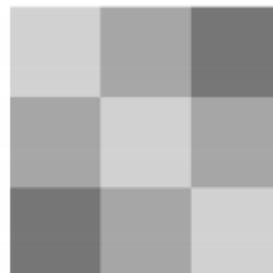
➤ Filtre :

Les paramètres d'une couche convolutionnelle configurent l'ensemble de couches de filtres. Les filtres sont une fonction dont la largeur et la hauteur sont inférieures à la largeur et à la hauteur du volume d'entrée. Les filtres (p. ex., convolutions) sont appliqués sur la largeur et la hauteur du volume d'entrée de façon coulissante, comme le montre la figure II.18. Des filtres sont également appliqués pour chaque profondeur du volume d'entrée. Nous calculons la sortie du filtre en produisant le produit point du filtre et la région d'entrée (Patterson, 2017).

Cartes d'activation : Nous glissons chaque filtre à travers les dimensions spatiales (largeur, hauteur) du volume d'entrée pendant le passage de l'information dans le CNN. Cela produit une sortie bidimensionnelle appelée carte d'activation pour ce filtre spécifique. La figure II.18 illustre la relation entre cette carte d'activation et notre concept de fonction complexe précédemment introduit. La carte d'activation à droite de la figure II.18 est rendue différemment pour illustrer comment les cartes d'activation convolutionnelles sont couramment rendues dans la littérature.

0.5	0.2	0.1
0.2	0.5	0.2
0.1	0.2	0.5

caractéristiques
convolutionnelle



Cartes d'activation

Figure II. 18 Convolutions et cartes d'activation (Patterson, 2017).

➤ **Partage de paramètres :**

Les RNC utilisent un schéma de partage de paramètres pour contrôler le nombre total de paramètres. Cela permet de gagner du temps de formation, car nous utiliserons moins de ressources pour apprendre l'ensemble de données de formation. Pour implémenter le partage de paramètres dans les RNC, nous désignons d'abord une seule coupe bidimensionnelle de profondeur comme une coupe de profondeur. Nous contraignons ensuite les neurones de chaque coupe de profondeur à utiliser les mêmes poids et biais. Cela nous donne beaucoup moins de paramètres (ou de poids) pour une couche convolutionnelle donnée (Patterson, 2017).

➤ **Hyper-paramètres de couche convolutionnelle :**

Voici les hyper-paramètres qui déterminent la disposition spatiale et la taille du volume de sortie d'une couche convolutionnelle :

- Taille du filtre (ou du noyau) (taille du champ).
- Profondeur de sortie.
- Foulée.
- Rembourrage zéro (Machine Learning & Deep Learning Fundamentals, n.d.).

➤ **Couches groupées :**

Les couches groupées sont généralement insérées entre des couches convolutionnelles successives. Nous voulons suivre des couches convolutionnelles avec des couches groupées pour réduire progressivement la taille spatiale (largeur et hauteur) de la représentation de données. Les couches de mise en commun réduisent progressivement la représentation des données sur le réseau et aident à contrôler l'ajustement. La couche de mise en commun fonctionne indépendamment sur chaque tranche de profondeur de l'entrée (Patterson, 2017). La mise en commun des couches réduirait le nombre de paramètres lorsque les images sont trop grandes. La mise en commun spatiale, aussi appelée sous-échantillonnage ou contre-échantillonnage, réduit la dimensionnalité de chaque carte tout en conservant les informations importantes. La mise en commun spatiale peut être de différents types (Prabhu, 2021):

Mise en commun maximale

Mise en commun moyenne

Quelque Pooling.

La mise en commun maximale prend le plus grand élément de la carte des caractéristiques rectifiées. La prise du plus grand élément pourrait également prendre la mise en commun moyenne, la somme de tous les éléments de la carte des caractéristiques.

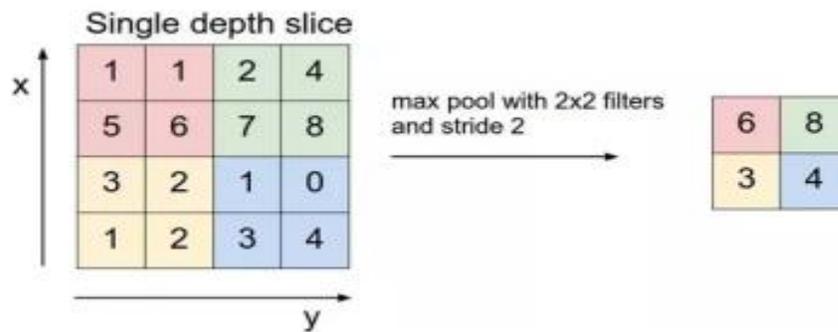


Figure II. 19 Mise en commun maximale (Prabhu, 2021)

➤ Couche entièrement connectée

Nous utilisons cette couche pour calculer les scores de classe que nous utiliserons comme sortie du réseau (par exemple, la couche de sortie à la fin du réseau). Les dimensions du volume de sortie sont $[1 * 1 * N]$, où N est le nombre de classes de sortie que nous évaluons. Cette couche a une connexion entre tous ses neurones et chaque neurone de la couche précédente.

Les couches entièrement connectées ont les paramètres normaux pour la couche et les hyper paramètres. Les couches entièrement connectées effectuent des transformations sur le volume de données d'entrée qui sont une fonction de l'activation dans le volume d'entrée et les paramètres (poids et biais des neurones).

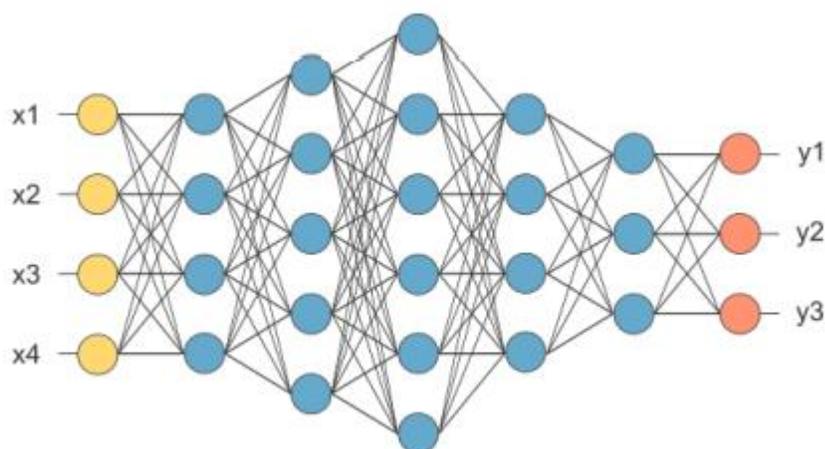


Figure II. 20 La couche de mise en commun (Prabhu, 2021)

Dans le diagramme ci-dessus, la matrice de la carte des caractéristiques sera convertie en vecteur (x_1, x_2, x_3, \dots). Avec les couches entièrement connectées, nous avons combiné ces caractéristiques pour créer un modèle. Enfin, nous avons une fonction d'activation telle que soft max ou sigmoïde pour classer les sorties comme chat, chien, voiture, camion etc...

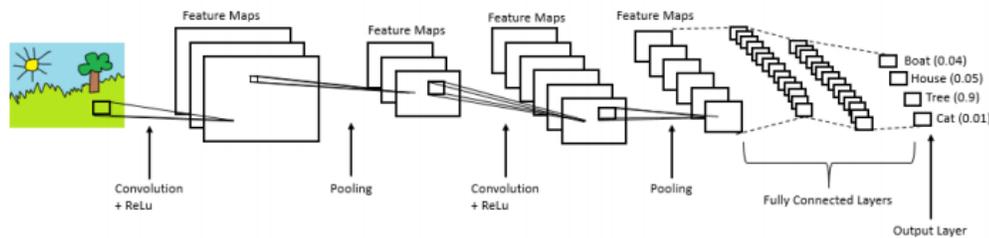


Figure II. 21 Architecture complète de RNC (Prabhu, 2021).

➤ Architectures populaires des RNC (CNN)

Voici une liste de certaines des architectures les plus populaires des CNN.

- **LeNet** (Lecun, Y., Bottou, L., Bengio, Y., & Haffner, Dec 1998).

- Une des premières architectures réussies des RNCs.
- Développé par Yann Lecun
- Utilisé à l'origine pour lire les chiffres dans les images.

- **AlexNet** (Krizhevsky, A., & Sutskever, I. & E. Hinton, Jan 2012).

- Aide à populariser les CNN en vision par ordinateur
- Développé par Alex Krizhevsky, Il y a Sutskever et Geoff Hinton
- A remporté le ILSVRC 2012

- **ZF Net** (Zeiler, Jan 2013).

- A remporté le ILSVRC 2013
- Conçu par Matthew Zeiler ET Rob Fergus
- Présentation du concept de visualisation du réseau de déconvolution

- **GoogLeNet** (Szegedy, June 2015).

- A remporté le ILSVRC 2014
- Développé par Christian Szegedy et son équipe chez Google
- Nom de code « Inception », une variation comporte 22 couches
- VGGNet (Simonyan, Sep 2014).
- Finaliste de l'ILSVRC 2014.
- Élaboré par Karen Simonyan et Andrew Zisserman.
- A montré que la profondeur du réseau était un facteur critique pour une bonne performance

- ResNet (He, June 2016).
- Formation sur des réseaux très profonds (jusqu'à 1200 couches)
- Première victoire dans la tâche de classification ILSVRC 2015.

4 Domaine d'application de l'apprentissage profond

4.1 Bio-informatique

La bio-informatique est un domaine multidisciplinaire de la recherche en biotechnologie. Plus généralement, la bio-informatique est l'application de la statistique et de l'informatique aux sciences biologiques. Les biologistes, médecins, informaticiens, mathématiciens, physiciens et bio-informaticiens travaillent ensemble pour résoudre un problème scientifique posé par la biologie.

Le terme bio-informatique peut également décrire, par abus de langage, toutes les applications informatiques résultant de cette recherche (Hogeweg, Mar 2011). Ce domaine va de l'analyse du génome à la modélisation de l'évolution d'une population animale dans un environnement donné, y compris la modélisation moléculaire, l'analyse d'images, l'assemblage et la reconstruction du génome (Baldi P. &., Jan 2001).

4.2 Santé

L'apprentissage profond a été particulièrement efficace en imagerie médicale, grâce à la disponibilité d'images de haute qualité et à la capacité des réseaux neuronaux convolutionnels à

classer les images. Par exemple, l'apprentissage profond peut être aussi efficace (ou même plus efficace) qu'un dermatologue dans la classification des cancers de la peau. Plusieurs fournisseurs ont déjà reçu l'approbation de la FDA pour l'utilisation d'algorithmes d'apprentissage profond pour effectuer des diagnostics, y compris l'analyse d'images pour l'oncologie et les maladies rétinienne. L'apprentissage profond fait également de grands progrès dans l'amélioration de la qualité des services de santé en anticipant les événements médicaux au moyen de dossiers médicaux électroniques ("Deep learning").

4.3 Robotique

Un grand nombre de développements récents dans le domaine de la robotique a été réalisé grâce à l'intelligence artificielle et à l'apprentissage profond. Par exemple, l'IA a permis aux robots de détecter leur environnement et d'y réagir. Cette capacité étend la gamme de leurs fonctions, des maisons mobiles sur les étages au tri et à la manipulation des objets irréguliers, fragiles ou emmêlés. La cueillette d'une fraise est une tâche facile pour les humains, mais beaucoup plus compliquée pour les robots. Les progrès de l'IA feront évoluer les capacités des robots. L'évolution de l'IA permettra aux robots du futur de devenir de meilleurs assistants pour l'homme. Ils ne serviront pas seulement à comprendre et à répondre aux questions, comme c'est le cas pour certains robots. Ils seront également en mesure de répondre aux commandes vocales et aux gestes, et même d'anticiper le prochain mouvement d'une personne avec qui ils travaillent. Aujourd'hui, les robots collaboratifs travaillent déjà aux côtés des humains pour accomplir des tâches distinctes adaptées à leurs capacités (Qu'est-ce que l'apprentissage profond?).

4.4 Visual and vocal recognition

En général, une méthode qui contient une méthode différente et une utilisation différente, par exemple, la reconnaissance visuelle d'un panneau routier qui est des éléments de panneaux routiers placés sur le côté des routes. Ils désignent à la fois le dispositif sur lequel est implanté un signal routier et le signal lui-même, par un robot ou une voiture autonome (Cireşan, Feb 2012).

La reconnaissance automatique de la parole (souvent appelée incorrectement reconnaissance de la parole) est une technique informatique qui permet d'analyser la voix humaine captée par un microphone en vue d'une transcription sous forme de texte lisible par machine (Cai, Dec 2013).

4.5 Reconnaissance visuelle et vocale

En général, une méthode qui contient différentes méthodes et différentes utilisations par exemple, la reconnaissance visuelle d'un panneau routier qui est un des panneaux routiers placés sur les côtés des routes. Ils désignent à la fois, le dispositif sur lequel est implanté un signal routier

et le signal lui-même, par un robot ou une voiture autonome (Cireşan, Feb 2012). La reconnaissance automatique de la parole (souvent appelée incorrectement reconnaissance de la parole) est une technique informatique qui permet d'analyser la voix humaine captée par un microphone en vue d'une transcription sous forme de texte lisible par machine (Cai, Dec 2013).

5 Les défis de l'apprentissage profond

5.1 La quantité de données utilisées

Les algorithmes d'apprentissage profond sont formés pour apprendre progressivement en utilisant des données. De grands ensembles de données sont nécessaires pour la machine pour fournir les résultats souhaités. Puisque le cerveau humain a besoin de nombreuses expériences pour apprendre et dériver l'information, le réseau neuronal artificiel nécessite une grande quantité de données. Plus l'abstraction que vous voulez est puissante, plus vous devez ajuster les paramètres et plus les paramètres nécessitent plus de données.

5.2 Le temps d'apprentissage trop élevé

Une fois les ensembles de données en main, leur utilisation pour former des réseaux d'apprentissage profond peut nécessiter plusieurs jours sur de grands groupes de processeurs et de GPU, des techniques émergentes comme l'apprentissage par transfert et les réseaux génératifs accusatoires promettent de surmonter ce défi.

5.3 Optimisation de l'hyper-paramètre

L'une des raisons pour lesquelles l'apprentissage profond fonctionne si bien est le grand nombre de neurones interconnectés, ou paramètres libres, qui capturent des nuances subtiles et des variations dans les données. Cependant, cela signifie aussi qu'il est plus difficile d'identifier les hyper-paramètres, paramètres dont les valeurs doivent être définies avant l'entraînement. Le processus est plus un art que la science. Il existe également un risque de sur-adaptation des données, surtout lorsque le nombre de paramètres dépasse largement le nombre d'observations indépendantes.

5.4 Difficulté à comprendre comment l'information arrive

En raison du grand nombre de couches, de nœuds et de connexions, il est difficile de comprendre comment les réseaux d'apprentissage profond accèdent à l'information. Bien que cela ne soit peut-être pas si important dans des applications comme l'étiquetage des photos sur les médias sociaux, la compréhension du processus décisionnel devient très importante dans des applications critiques comme la maintenance prédictive ou la prise de décisions.

6 L'apprentissage profond est sensible :

Les réseaux d'apprentissage profond sont très sensibles à l'effet papillon de petites variations dans les données d'entrée peuvent conduire à des résultats radicalement différents, les rendant instables en conséquence. Cette instabilité ouvre également de nouvelles surfaces d'attaque pour les pirates. Dans les attaques dites conflictuelles, les chercheurs ont montré qu'en ajoutant une quantité imperceptible de bruit, il est possible d'induire en erreur les réseaux d'apprentissage profond pour aboutir à des informations complètement incorrectes. Tout sans même accéder au système.

7 Conclusion

Dans ce chapitre nous avons fait une étude théorique sur les réseaux neuronaux, ainsi que sur l'apprentissage profond et ses architectures, à la fin du chapitre nous présentons les domaines d'application et les défis de l'apprentissage profond.

Dans le prochain chapitre nous allons présenter la méthodologie suivie tout au long du développement de ce projet, commençons par la conception et passons par l'implémentation et finissons par les résultats.

Chapitre 3 :

Conception et implémentation

1 Introduction

L'un des principaux objectifs de la détection et de la reconnaissance automatiques des plaques d'immatriculation de voitures est de réduire la main d'œuvre, le coût, le temps et de rendre le processus rapide et hautement disponible. Il est nécessaire d'identifier et de lire le numéro de la plaque d'immatriculation du véhicule à partir de photo prise par des dispositifs de surveillance. Il faut donc convertir l'image en format texte avant de pouvoir l'utiliser par d'autres parties du système, sans intervention humaine.

Dans ce chapitre nous allons présenter la conception et l'implémentation de notre système qui concerne la détection et la reconnaissance de plaques d'immatriculation ainsi que le control d'accès d'un véhicule à un parking, par la suite nous allons illustrer les résultats obtenus.

2 Conception générale de notre système

L'architecture globale de notre système de détection et de reconnaissance des plaques d'immatriculation à partir d'images capturé est illustrée dans la figure III.1.

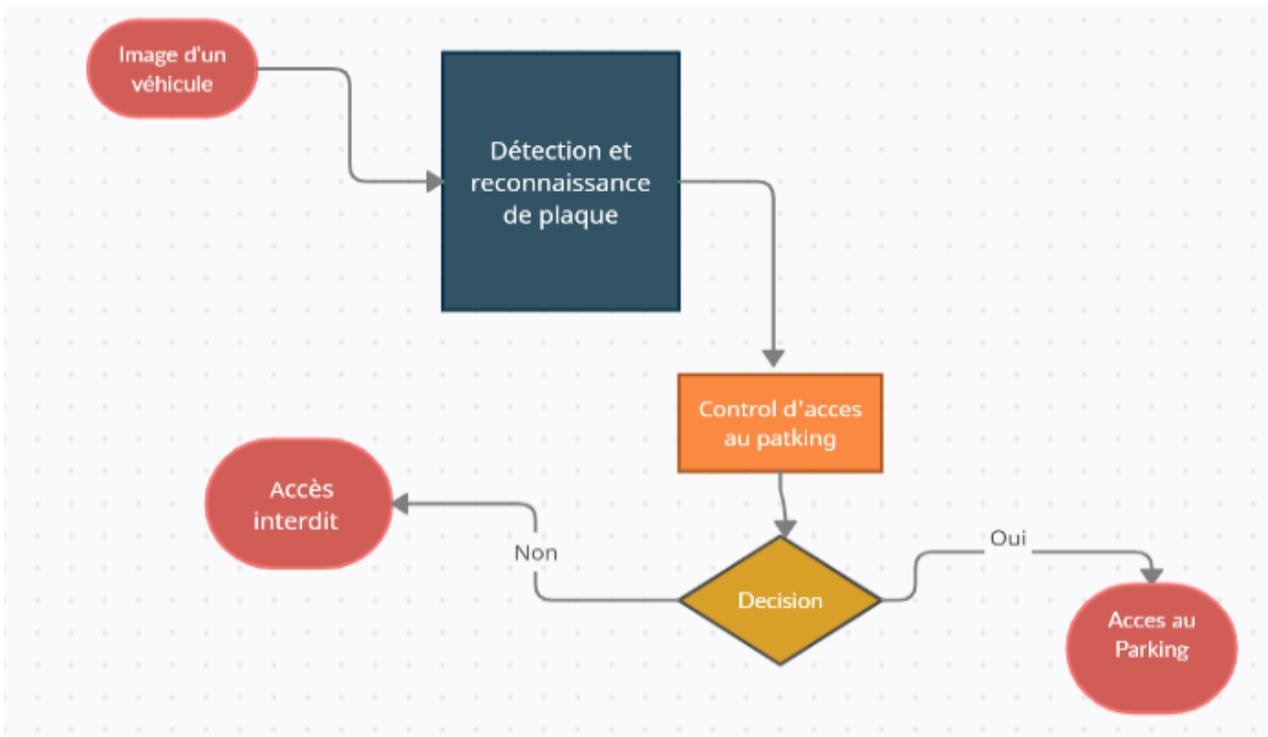


Figure III. 1 Schéma de conception global

3 Conception détaillé

Dans ce paragraphe nous allons présenter la conception de notre système, dont nous allons expliquer les composants principaux du système à savoir la détection et la reconnaissance de plaque ainsi que le contrôle d'accès au parking (la vérification de l'existence du matricule dans la base de données). Et puisque notre système nécessite une étape d'apprentissage au début, nous allons bien détailler cette importante étape. La Figure III.2 ci-dessous illustre notre conception détaillée.

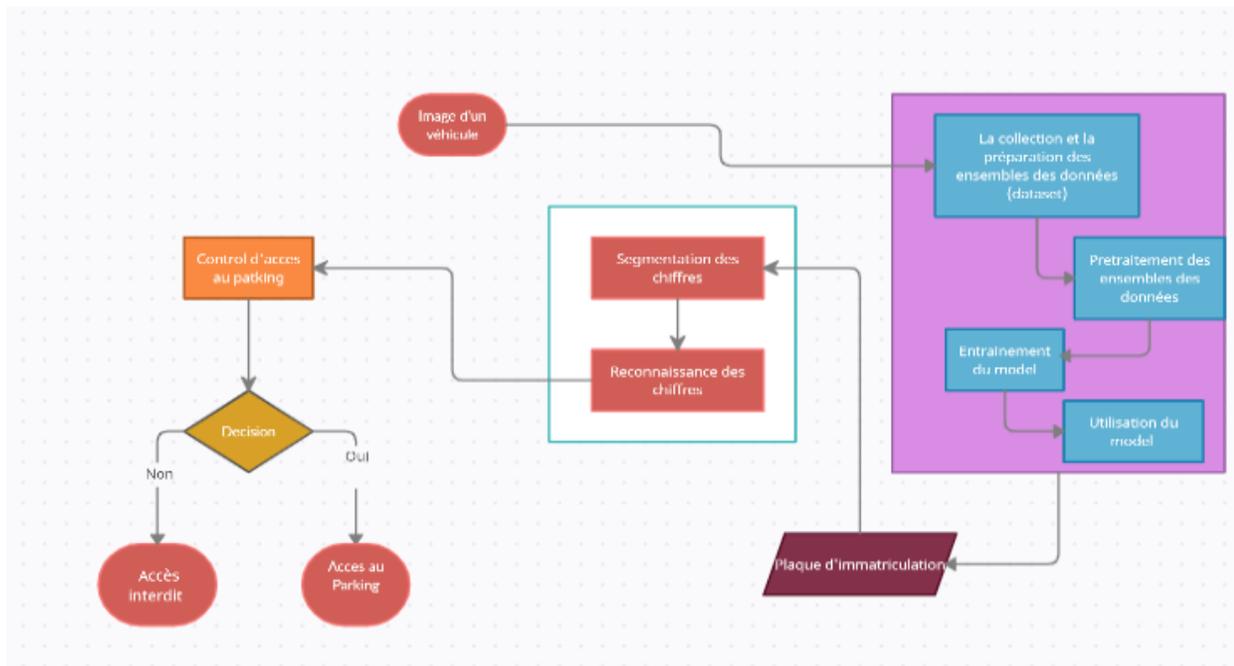


Figure III. 2 Schéma de conception détaillée

3.1 L'apprentissage du modèle

Dans ce paragraphe nous allons détailler les différentes étapes d'apprentissage, dans lesquelles le système va faire un entraînement pour qu'il apprenne à détecter des plaques d'immatriculation à partir d'une image d'un véhicule donné.

3.1.1 La collecte et la préparation des ensembles de données (Description des données « dataset »)

Dans notre système, nous allons travailler sur trois ensembles de données (dataset) différents provenant de la même source, la plupart des méthodes actuelles de détection et de reconnaissance des plaques d'immatriculation sont évaluées à l'aide d'un petit ensemble de données habituellement non représentatif, puisqu'il n'y a pas de grands ensembles de données diversifiés accessibles au public

Dans ce travail, nous avons utilisé le Chinese City Parking Dataset (CCPD) qui est un ensemble de données pour la détection et la reconnaissance des plaques d'immatriculation. Ces données sont vaste et complet et bien structuré (toutes les images ont les mêmes structure), toutes les images sont prises manuellement par les travailleurs d'une société de gestion de stationnement au bord de la route et sont annotées. À notre connaissance, CCPD est le plus grand ensemble de données LP accessible au public à ce jour avec plus de 250k images uniques de voiture, l'emplacement de 2 points de rectangle englobant la plaque est sous forme XML, il est le seul qui fournit des annotations de localisation de sommets.

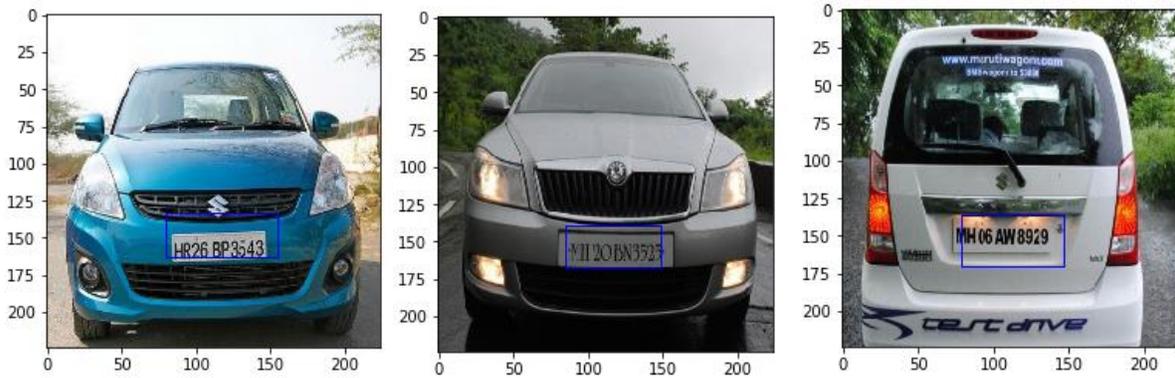


Figure III. 3 Échantillon de données utilisées

3.1.2 Prétraitement des ensembles de données

Le prétraitement des données est la première étape (et sans doute la plus importante) vers l'élaboration d'un modèle d'apprentissage profond fonctionnel (Bonner, 2021), c'est essentiel. Le prétraitement des données est une technique d'extraction de données qui consiste à transformer les données brutes en un format compréhensible. Les données réelles sont souvent incomplètes, incohérentes et/ou manquantes dans certains comportements ou tendances, et sont susceptibles de contenir de nombreuses erreurs. Le prétraitement des données est une méthode éprouvée pour résoudre de tels problèmes (Sharma, 2021).

Dans le monde réel, les données sont généralement incomplètes : absence de valeurs d'attributs, absence de certains attributs d'intérêt, ou ne contenant que des données agrégées. Bruyant : contient des erreurs ou des valeurs aberrantes. Incohérent : contient des incohérences dans les codes ou les noms.

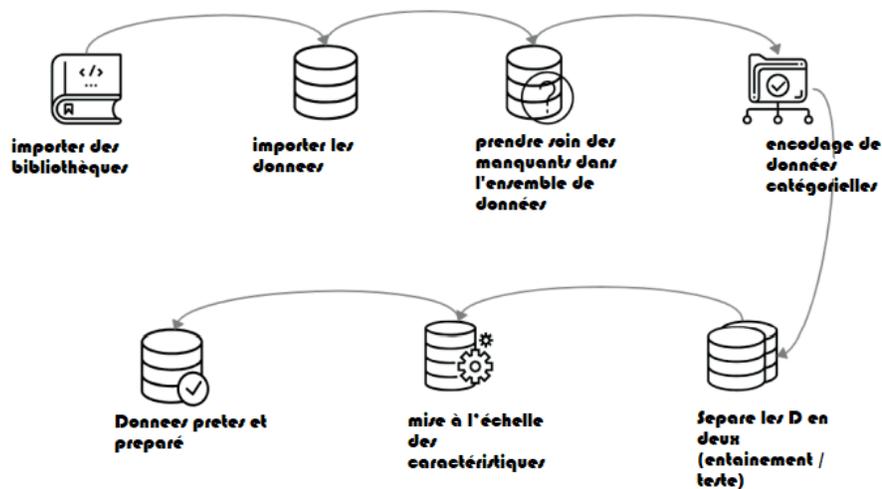


Figure III. 4 Les étapes de Prétraitement des ensembles de données

3.1.2.1 Importer des bibliothèques

La première étape consiste habituellement à importer les bibliothèques qui seront nécessaires dans le programme. Une bibliothèque est essentiellement un ensemble de modules qui peuvent être appelés et utilisés.

Beaucoup de choses dans le monde de la programmation n'ont pas besoin d'être écrites explicitement jamais le temps qu'elles sont nécessaires. Il y a des fonctions pour eux, qui peuvent simplement être invoquées, aussi les bibliothèques dont nous avons besoin pour charger des ensembles de données et travailler sur eux.

3.1.2.2 Importer les ensembles de données

Beaucoup d'ensembles de données sont offerts dans différents formats. Nous devons d'abord localiser le répertoire des fichiers des jeux de données (il est plus efficace de conserver l'ensemble de données dans le même répertoire que votre système) et les lire à l'aide d'une méthode qui peut être trouvée dans les bibliothèques dans les variables en mémoire, en outre utilisé dans les pré-étapes de traitement.

3.1.2.3 Prendre soin des données manquantes dans l'ensemble de données

Il arrive parfois qu'il manque des données dans l'ensemble de données. Nous devons être équipés pour régler le problème lorsque nous les rencontrons. L'une des idées les plus courantes pour traiter le problème est de prendre une moyenne de toutes les valeurs de la même colonne et de l'avoir pour remplacer les données manquantes.

3.1.2.4 Encodage des données catégoriques

Parfois, nos données sont qualitatives, c'est-à-dire que nous avons des images comme données. Nous pouvons trouver des catégories sous forme d'image. Maintenant, il devient plus facile pour les machines de traiter l'image, plutôt que des nombres, puisque les modèles sont basés sur des équations mathématiques et des calculs. Par conséquent, nous devons encoder les données catégoriques.

3.1.2.5 Diviser l'ensemble de données en ensemble de formation et ensemble de tests

Nous devons diviser notre ensemble de données en deux ensembles : un ensemble de formation et un ensemble de tests. Nous allons former nos modèles d'apprentissage automatique sur notre ensemble de formation, nos modèles d'apprentissage profond vont essayer de comprendre les corrélations dans notre ensemble de formation et ensuite nous allons tester les modèles sur notre ensemble de test pour vérifier la précision qu'il peut prévoir. Une règle générale consiste à allouer 80 % de l'ensemble de données à l'ensemble de formation et les 20 % restants à l'ensemble de tests.

3.1.2.6 La mise à l'échelle des données

La mise à l'échelle des données marque la fin du prétraitement des données dans L'apprentissage automatique. C'est une méthode pour standardiser les variables indépendantes d'un ensemble de données dans une plage spécifique. En d'autres termes, la mise à l'échelle des entités limite la plage de variables afin que vous puissiez les comparer sur des bases communes. Il existe des méthodes pour effectuer l'étape de mise à l'échelle des entités en fonction des données telles que la normalisation, et la normalisation moyenne. Nous travaillons avec des images donc nous avons utilisé la méthode de normalisation qui est également appelé Redimensionnement (normalisation min-max) qui s'incarne dans la formule suivante :

$$d(A, B) = p (x_1 - x_2)^2 + (y_1 - y_2)^2 \quad (3.1)$$

Redimensionnement (normalization min-max)

Aussi connu sous le nom de mise à l'échelle min-max ou normalisation min-max, est la méthode la plus simple et se compose en redimensionnant la plage de fonctionnalités pour la mettre à l'échelle dans [0, 1] ou [-1, 1]. La sélection de la plage cible dépend de la nature des données. La formule générale est donnée comme suit :

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.2)$$

Où x est une valeur originale, x' est la valeur normalisée. Par exemple, supposons que nous ayons les données sur le poids des élèves et que les poids des élèves [160 livres, 200 livres]. Pour

redimensionner ces données, nous soustrayons d'abord 160 du poids de chaque élève et divisons le résultat par 40 (la différence entre les poids maximum et minimum).

- **Mean normalization**

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)} \quad (3.3)$$

Où x est une valeur originale, x' est la valeur normalisée.

- **Normalisation**

La méthode générale de calcul consiste à déterminer la moyenne de distribution et l'écart-type pour chaque caractéristique. Ensuite, nous soustrayons la moyenne de chaque caractéristique. Ensuite, nous divisons les valeurs (la moyenne est déjà soustraite) de chaque caractéristique par son écart-type.

$$x' = \frac{x - \bar{x}}{\sigma} \quad (3.4)$$

Où x est le vecteur caractéristique d'origine, $\bar{x} = \text{moyenne}(x)$ est la moyenne de ce vecteur caractéristique, et σ est son écart-type.

3.1.3 Entraînement du model (Training)

C'est la phase importante de notre système, dont nous avons formé un modèle de réseaux neuronaux convolution avec une architecture spécifique pour atteindre le meilleur résultat de précision et moins de perte de données, qui peut être utilisé dans la reconnaissance des plaques d'immatriculation d'image, la phase d'entraînement pourrait être divisée en deux étapes illustrées à la figure III.5.

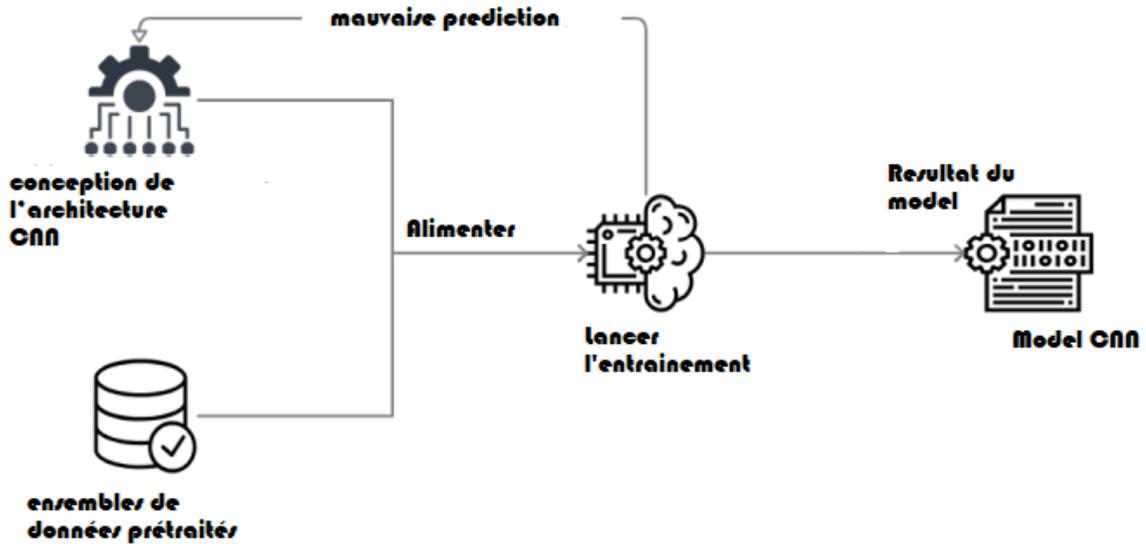


Figure III. 5 Cycle de la vie d'entraînement

3.1.3.1 Conception de l'architecture du modèle

Comme nous connaissons déjà la méthode d'apprentissage en profondeur, nous allons travailler avec les "Réseaux neuronaux de convolution", Nous concevons les paramètres du modèle (nombre de couches, taille de couche, type de chaque couche et la fonction de définition de couche).

3.1.3.2 L'entraînement du model

Après avoir préparé et conçu l'architecture du modèle, Et les ensembles de données prétraités, Maintenant, nous configurons les paramètres de la phase de formation tels que : Taille des lots, Nombre d'époques, définissant également les données à utiliser, après avoir traité la formation en attente de bons résultats, dans un cas de mauvais résultats, nous revenons à l'étape 1 : Modéliser la conception de l'architecture, et nous nous entraînons à nouveau jusqu'à ce que de bons résultats soient obtenus.

3.1.4 Utilisation du model obtenu

Une fois la phase de formation terminée, bien sûr, peu importe les meilleurs résultats de modèle ont été atteints, nous sauvegardons notre modèle pour la reconnaissance de caractère d'image, la [figure III.6](#) montre un exemple d'étapes tout en testant notre modèle de sortie de résultats avec une image d'entrée, après cela, nous obtenons les résultats de la reconnaissance.

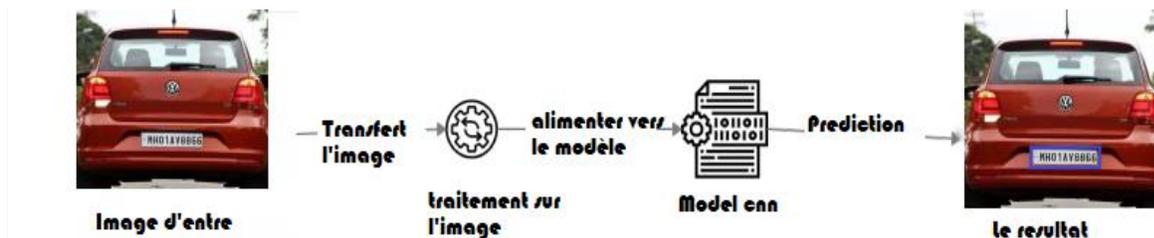


Figure III. 6 Exemple de teste du model

3.2 La reconnaissance de plaque

L'étape d'apprentissage nous a permis d'obtenir un modèle capable de détecter une plaque d'immatriculation à partir d'une image. Une fois la plaque est sélectionné il faut passer à l'étape de reconnaissance, ceci est réalisé en deux phases :

- La segmentation des chiffres
- La reconnaissance des chiffres

3.2.1 La segmentation des chiffres

Elle consiste à faire une segmentation de l'image du matricule sélectionné pour extraire les chiffres et éventuellement les lettres qui la composent.

3.2.2 La reconnaissance des chiffres

La reconnaissance consiste à interpréter les chiffres et les lettres pour qu'on puisse par la suite les traiter autant que chiffres/lettres et non pas comme image. Ceci va nous permettre de comparer la plaque d'immatriculation détecté avec celles de notre base de données.

3.3 Le contrôle d'accès

C'est la dernière étape dans notre système, dont nous allons faire une vérification de l'existence, de la plaque d'immatriculation détecté, dans la base de données contenant la liste des numéros d'immatriculation ayant une autorisation d'accès au parking. Le résultat sera donc soit le

numéro de la plaque existe (ce véhicule peut accéder au parking), ou il n'existe pas dans ce cas le système interdit son accès au parking mais il lui offre la possibilité d'ajouter son numéro à notre base.

4 Implémentation

Nous allons décrire la mise en œuvre des différentes étapes de notre système conçu dans le paragraphe précédent. Nous commençons par la justification de l'environnement et les outils de développement utilisés, ensuite nous détaillons le processus de chargement de données pour la création de notre modèle. Et enfin nous présentons l'algorithme principal nécessaire à l'implémentation de notre système.

4.1 Environnements et Outils de développement

Pour développer notre système, nous allons utiliser différents environnements et outils pour le langage de programmation y compris des API, bibliothèques, langage de balisage aussi la feuille de style avec différentes bibliothèques et cadres, travaillé sur de nombreux éditeurs, IDE etc... .

- **Python**



Figure III. 7 Le logo du python

Python est un langage de programmation interprété, de haut niveau et généralisé. Créé par Guido van Rossum et publié pour la première fois en 1991, la philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'espaces blancs significatifs. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire du code clair et logique pour des projets de petite et de grande envergure.

- **Qt Creator**

Qt Creator est un environnement de développement intégré multiplateforme (IDE) conçu pour une expérience de développement maximale. Qt Creator fonctionne sur les systèmes d'exploitation de bureau Windows, Linux et macOS, et permet aux développeurs de créer des

applications sur des plateformes de bureau, mobiles et embarquées, Développé par deux ingénieurs logiciels norvégiens Haavard Nord et Eirik Chambe-Eng, le cadre Qt a été rendu public en mai 1995. La première version publique de Qt a été publiée par la société appelée Trolltech.



Figure III. 8 Le logo du Qt Creator

- **Tensorflow**

TensorFlow est une bibliothèque de logiciels libre et open-source pour le flux de données et la programmation différentiable sur une gamme de tâches. Il s'agit d'une bibliothèque mathématique symbolique, qui est également utilisée pour des applications d'apprentissage automatique comme les réseaux neuronaux. Il est utilisé à la fois pour la recherche et la production chez Google.

TensorFlow a été développé par l'équipe de Google Brain pour une utilisation interne de Google. Il a été publié sous la licence Apache 2.0 le 9 novembre 2015.



Figure III. 9 Le logo du TensorFlow

- **Keras**

Chapitre 3 : Conception et implémentation

Keras est une bibliothèque de réseau neuronal open-source écrite en Python. Il est capable de fonctionner sur TensorFlow, Microsoft Cognitive Toolkit, Theano, ou PlaidML. Conçu pour permettre une expérimentation rapide des réseaux neuronaux profonds, il se concentre sur une utilisation conviviale, modulaire et extensible. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son auteur principal et responsable est François Chollet, un ingénieur de Google. Chollet est également l'auteur du modèle de réseau neural profond Xception.

En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow.

Chollet a expliqué que Keras a été conçu pour être une interface plutôt qu'un cadre d'apprentissage automatique autonome. Il offre un ensemble d'abstractions plus intuitives et de plus haut niveau qui facilitent le développement de modèles d'apprentissage profond, quel que soit le moteur de calcul utilisé. Microsoft a ajouté un backend CNTK à Keras ainsi, disponible à partir de CNTK v2.0.



Figure III. 10 Le logo du Keras

- **Anaconda**

Anaconda est une distribution libre et open-source des langages de programmation Python et R pour l'informatique scientifique, qui vise à simplifier la gestion et le déploiement des paquets. Les versions des paquets sont gérées par le système de gestion des paquets conda.



Figure III. 11 Le logo de l'Anaconda

- **Numpy**

NumPy est une bibliothèque pour le langage de programmation Python, ajoutant le support pour les grands tableaux et matrices multidimensionnels, avec une grande collection de fonctions mathématiques de haut niveau pour fonctionner sur ces tableaux.



Figure III. 12 Le logo du NumPy

- **Pandas**

Dans la programmation informatique, pandas est une bibliothèque logicielle écrite pour le langage de programmation Python pour la manipulation et l'analyse de données. Il offre notamment des structures de données et des opérations de manipulation de tables numériques et de séries chronologiques. C'est un logiciel libre publié sous la licence BSD à trois clauses.



Figure III. 13 Le logo du pandas.

- **Matplotlib**

Matplotlib est une bibliothèque de tracés pour le langage de programmation Python et son numérique mathématique extension NumPy. Il fournit une API orientée objet pour intégrer des tracés dans des applications utilisant des outils GUI généraux comme Tkinter, wxPython, Qt ou GTK+.



Figure III. 14 Le logo du matplotlib

- **Colab**

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté à la machine learning, à l'analyse de données et à l'éducation.



Figure III. 15 Le logo du Google Colab

- **OpenCV**

OpenCV est une bibliothèque de fonctions de programmation principalement destinées à la vision informatique en temps réel. Initialement développé par Intel, il a été plus tard pris en charge par Willow Garage puis Itseez. La bibliothèque est multiplateforme et gratuite pour une utilisation sous licence open-source BSD.



Figure III. 16 Le logo de l'OpenCV

- **Tesseract**

Chapitre 3 : Conception et implémentation

Tesseract est un logiciel de reconnaissance optique de caractères sous licence Apache. Conçu par les ingénieurs de Hewlett Packard de 1985 à 1995, Son développement est abandonné pendant les dix années suivantes, En 2005, les sources du logiciel sont publiées sous licence Apache et Google poursuit son développement.

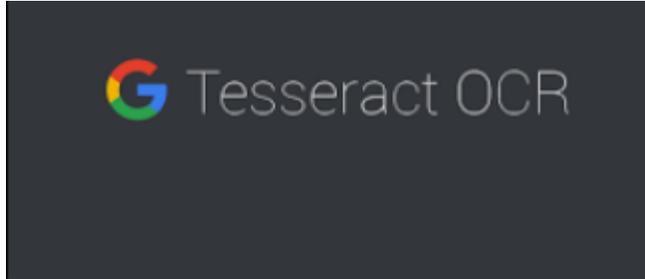


Figure III 1. Le logo du Tesseract OCR

Figure III. 17 Le logo du Tesseract OCR

- **PyCharm**

PyCharm est un environnement de développement intégré utilisé dans la programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains.



Figure III. 18 Le logo du PyCharm

- **CSS**

CSS (Cascading Style Sheets) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage comme HTML. CSS est une technologie de base du World Wide Web, aux côtés de HTML et JavaScript.



Figure III. 19 Le logo du CSS

- **PySide2**

PySide est une liaison Python de la boîte à outils multiplateforme GUI Qt développée par The Qt Company, dans le cadre du projet Qt pour Python. C'est l'une des alternatives au paquetage standard Tkinter. Comme Qt, PySide est un logiciel libre. PySide prend en charge Linux/X11, macOS et Microsoft Windows.



Figure III. 20 Le logo du PyCharm

4.2 Chargement de l'ensemble de données (transfert les données vers le colab)

Dans cette étape nous avons fait un chargement des données c'est-à-dire nous avons les transféré vers le colab. Voici le code permettant de réaliser cette opération.

```
] img_dir = "/content/drive/MyDrive/dataset/archive/images"
data_path = os.path.join(img_dir, '*g')
files = glob.glob(data_path)
files.sort()
X=[]
for f1 in files:
    img = cv2.imread(f1)
    img = cv2.resize(img, (IMAGE_SIZE,IMAGE_SIZE))
    X.append(np.array(img))
```

Figure III. 21 La premiere partie de donnée (images)

Nous avons besoin des coordonnées (Min et Max) des sommets du rectangle de la plaque détecté, Pour ce faire nous avons implémenté le code suivant :

```
[ ] from lxml import etree
def resizeannotation(f):
    tree = etree.parse(f)
    for dim in tree.xpath("size"):
        width = int(dim.xpath("width")[0].text)
        height = int(dim.xpath("height")[0].text)
    for dim in tree.xpath("object/bndbox"):
        xmin = int(dim.xpath("xmin")[0].text)/(width/IMAGE_SIZE)
        ymin = int(dim.xpath("ymin")[0].text)/(height/IMAGE_SIZE)
        xmax = int(dim.xpath("xmax")[0].text)/(width/IMAGE_SIZE)
        ymax = int(dim.xpath("ymax")[0].text)/(height/IMAGE_SIZE)
    return [int(xmax), int(ymax), int(xmin), int(ymin)]
```

Figure III 2.L'emplacement des deux points pour le rectangle

```
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.1)
```

Figure III. 22 Code image DataGenerator

L'augmentation des données d'image crée des versions transformées d'images dans l'ensemble de données qui appartiennent à la même classe que l'image originale

- **Arguments :**

- **Rescale**

Facteur d'annulation. Valeur par défaut None. Si None ou 0, aucun rescaling n'est appliqué, sinon nous multiplions les données par la valeur fournie (après avoir appliqué toutes les autres transformations).

- **Validation split**

Float. Fraction des images réservées à la validation (strictement entre 0 et 1) et nous réserverons pour validation 20 % de l'ensemble de données.

```
train_generator = datagen.flow_from_dataframe(
    reduced_df,
    directory="data/",
    x_col="image_name",
    y_col=["top_x", "top_y", "bottom_x", "bottom_y"],
    target_size=(WIDTH, HEIGHT),
    batch_size=32,
    class_mode="other",
    subset="training")

validation_generator = datagen.flow_from_dataframe(
    reduced_df,
    directory="data/",
    x_col="image_name",
    y_col=["top_x", "top_y", "bottom_x", "bottom_y"],
    target_size=(WIDTH, HEIGHT),
    batch_size=32,
    class_mode="other",
    subset="validation")
```

Figure III. 23 Code du flow-from-directory

Prend le chemin d'accès à un répertoire génère des lots de données augmentées.

➤ **Directory**

String, chemin vers le répertoire cible. Il doit contenir un sous-répertoire par classe. Et les images à l'intérieur de chaque arborescence de sous-répertoires seront incluses dans le générateur.

➤ **Target size:**

Tuple d'entiers (hauteur, largeur), par défaut à (256, 256). Les dimensions auxquelles toutes les images trouvées seront redimensionnées.

➤ **Batch size**

Taille des lots de données (par défaut : 32).

➤ **Subset**

Sous-ensemble de données (« formation » ou « validation ») utilisé uniquement si le split de validation est défini dans ImageDataGenerator.

4.3 Création du modèle

Le modèle est compilé avec la fonction de perte d'entropie croisée catégorique et l'adam l'optimiseur est utilisé. La mesure de la « précision » est utilisée pour évaluer le modèle.

Formation du modèle à l'aide d'un GPU de Google colab accélère le processus de formation, le nombre d'époques à 50 et utiliser une méthode de régularisation pour réduire le taux d'apprentissage.

```
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
```

Figure III. 24 Code de compilation.

```
[ ] train = model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=50, shuffle=true, batch_size=32, verbose=1 )
```

Figure III. 25 Code de déclenchement l'entraînement.

Nous utilisons la fonction fit pour former le modèle

- **Arguments de fit :**
 - **Train data**

Liste des données de formation.

- **Validation data**

Liste des échantillons utilisés pour évaluer la perte et les mesures du modèle après toute époque ont pris fin.

- **Epochs**

Un nombre entier est un nombre d'époques pour lesquels nous voulons former notre modèle.

- **Verbose**

Specifies verbosity mode, 1 to show progress bar and details par epoch.

- **Shuffle**

Utilisez les données au hasard.

4.4 Algorithmes

Dans cette section nous allons présenter le principe des algorithmes de notre application.

4.4.1 Algorithme principal

Nous allons essayer à travers cet algorithme d'expliquer le principe de notre système

Début

Lecture d'une image () ; // l'entrée est une image d'une voiture

Détection de la plaque d'immatriculation () ; // appel d'une fonction de notre modèle

// permettant de détecter la plaque d'immatriculation, le résultat est une image

// contenant un rectangle englobant la plaque détecté ()

Reconnaissance de la plaque d'immatriculation () ; // appel de la fonction de

//reconnaissance de la plaque

Contrôle d'accès () ; //appel de la fonction de contrôle d'accès du véhicule au parking

Fin

4.4.2 Algorithme d'apprentissage

C'est l'algorithme de base de notre système, nous avons au début implémenté un modèle d'apprentissage profond basé sur les réseaux neuronaux convolutionnel. Ce modèle va nous permettre de détecter une plaque d'immatriculation à partir d'une image fourni en entrée

Début

// l'entrée est un ensemble de donnée contenant des images de véhicule

Collection et préparation de données

Prétraitement des ensembles de données // extraction de données c-à-d transformer les

//données brutes en un format compréhensible

Entraînement du modèle // Faire un entraînement du modèle pour qu'il soit capable de

// détecter les plaques d'immatriculation

Utilisation du modèle // tester le modèle pour voir sa capacité de détection

// La sortie est un modèle de détection de plaque d'immatriculation

Fin

4.4.3 Algorithme de reconnaissance de plaque

Début

```
// L'entrée et image de la plaque détecté  
Segmenter la plaque détectée ()  
Lire le contenu de la plaque avec OCR tesseract () // appel d'une fonction de  
//reconnaissance de caractère le résultat est l'ensemble des lettres et des chiffres  
// composant la plaque
```

Fin

4.4.4 Algorithme de Contrôle d'accès

Début

```
// L'entrée est une matricule en lettres et en chiffres  
Effectuer une recherche dans la base de données // afin de vérifier l'existence de la  
// plaque dans notre base de données ou non  
Si (la plaque existe)  
    Accès accepté  
Sinon  
    Accès refusé // accès interdit au véhicule mais on lui donne la possibilité de  
    //s'ajouter dans notre base
```

Finsi

Fin

5 Résultats

Travaillant sur une combinaison matérielle et logicielle spécifique, pour notre configuration i5 6K CPU et 8 Go de mémoire RAM, windows 10 et Nvidia 1080 GTX utilisés pour la phase d'entraînement, en prenant un avantage de parallélisme pour l'entraînement rapide (gagner du temps et des ressources), nous obtenons de très bons résultats sur les différents modèles que nous avons formés.

5.1 Architecture du modèle

La figure III.26 est la visualisation détaillée de notre modèle avec tous les paramètres utilisés.

```
58892288/58889256 [=====] - 1s 0us/step
58900480/58889256 [=====] - 1s 0us/step
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
vgg16 (Functional)          (None, 7, 7, 512)          14714688
-----
flatten (Flatten)           (None, 25088)               0
-----
dense (Dense)                (None, 128)                 3211392
-----
dense_1 (Dense)              (None, 128)                 16512
-----
dense_2 (Dense)              (None, 64)                  8256
-----
dense_3 (Dense)              (None, 4)                   260
-----
Total params: 17,951,108
Trainable params: 3,236,420
```

Figure III. 26 Architecture de notre modèle.



Figure III. 27 Présentation Graphique de la précision

```
87/100 [=====>....] - ETA: 6s - loss: 0.0063 - accuracy: 0.8391WARNING
100/100 [=====] - 42s 412ms/step - loss: 0.0063 - accuracy: 0.8391
Test results
Loss: 0.006349066272377968
Accuracy 0.8390804529190063
```

Figure III. 28 Présentation Numérique (Accuracy & Loss)

5.2 Présentation de l'application



Figure III. 29 Première page de l'interface.

On ajoute une photo pour tester le programme.

Test 1 : le premier exemple :



Figure III. 30 Un résultat d'un test d'accès valide.

En chargeant l'image le système fait la détection et la reconnaissance du matricule, ensuite il fait la vérification dans la base de données qui a une forme CSV, pour tester son existence ou non dans la base, dans cet exemple il a confirmé son existence et il lui donne le permis d'accès.

Test 2 : Deuxième exemple



Figure III. 31 Un résultat d'un test d'accès non valide.

Le contenu de cette plaque d'immatriculation n'est pas enregistré dans la base de données qui a une forme CSV, l'utilisateur n'aura pas le permis d'accès. Dans ce cas l'utilisateur doit faire une demande d'enregistrement par un clique sur le bouton qui est en gris pour avoir le permis d'accès. Figure III.32.

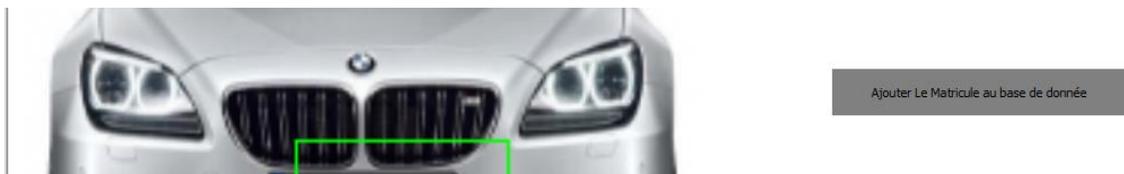


Figure III. 32 Le bouton d'ajout à la BD.

5.3 Discussion des résultats

Nous avons été en mesure de construire un système de détection et reconnaissance des plaques d'immatriculation en utilisant la technique d'apprentissage profond avec les réseaux neuronaux convolutionnel, après plusieurs tests, nous avons pu obtenir des résultats très satisfaisants. Notre système détecte environ 83% des plaques, puisque il y a des cas où le système n'arrive pas à détecter correctement la plaque voilà des exemples.

Test 3 : exemple de fausse détection



Figure III. 33 Exemple d'une mauvaise détection.

Dans cet exemple, On a obtenu une mauvaise détection car l'image est capturée du côté droit par conséquent la plaque d'immatriculation est partiellement visible.

Test 4 : Exemple de fausse détection et sa correction



Figure III. 34 Exemple d'une mauvaise détection Figure III. 35 Exemple d'une bonne détection.

On a seulement changé l'angle de la capture pour trouver une solution.

6 Conclusion

Ce chapitre présente la méthodologie qui a été suivie tout au long du développement de ce projet, il explique en détail comment le système de reconnaissance automatique des plaques d'immatriculation a été développé, tout en décrivant en détail la mise en œuvre de chacune des différentes étapes du système commençons par l'entrée (image) jusqu'à le résultat du model et finalement la vérification.

Conclusion générale

Les développements technologiques de ces vingt dernières années ont permis aux systèmes numériques d'envahir notre vie quotidienne. Il y a plus à démontrer que le numérique occupe aujourd'hui une place de choix dans le monde. Parmi les composantes majeures des systèmes numériques, une grande importance est accordée à l'image. La représentation et le traitement des images numériques font l'objet de recherches très actives à l'heure actuelle. Le traitement des images est un très vaste domaine qui a connu, et connaît encore, une évolution importante depuis quelques décennies.

Dans ce projet de master, nous avons concentré sur la détection des plaques d'immatriculation pour développer une application de vérification dans un parking moderne en utilisant un apprentissage approfondi, Pour cela nous avons proposé une nouvelle architecture dans la méthode CNN et un ensemble des données (dataset) trouver à partir du CCPD.

Durant ce travail nous avons essayé d'approfondir nos connaissances dans ce domaine, pour cela nous avons étudié les plaques d'immatriculation sur lesquelles nous avons appliqué notre système de détection. Et nous avons fait aussi une étude théorique sur les systèmes de reconnaissance de forme et d'apprentissage.

Ce travail avait pour but de concevoir et implémenter un modèle de détection des plaques d'immatriculation, Le procédé commence par le prétraitement des images ensuite la détection de l'emplacement de plaque d'immatriculation puis on la coupe pour obtenir une image contenant seulement la plaque. Par la suite on fait la lecture du contenu de la plaque qui est détecté avec une fonction de Tesseract OCR. Après la détection et l'extraction du contenu de la plaque on finit notre travail par la vérification de son existence dans la base de données qui a un type CSV.

Nos résultats sont très encourageants par rapport à d'autres travaux dans le même thème (On a une précision de 83%). Nous avons créé un modèle basé sur une approche de détection très puissante (CNN) et l'appliqué sur la base de donnée, on a pu accomplir notre objectif avec un minimum d'erreur et un taux de précision très élevé.

Conclusion générale

En ce qui concerne nos perspectives pour les travaux futurs, notre idée et notre objectif seront d'améliorer l'ensemble du système de détection en ajoutant quelque composants afin de:

- Rendre notre système capable de détecter des plaques dans des cas plus difficile.
- Améliorer notre base de données avec d'autres plaques de différente région.
- Ajouter plusieurs modes d'entrée (Vidéo par exemple).

Bibliographie

- Theodoridis, S., & K. Koutroumbas. (2003). *Pattern Recognition*. Elsevier: Academic Press.
- "Deep learning". (n.d.). In *les dessous d'une technologie de rupture*. Futurible.
- Ahmed, A. &. (2016). *Réalisation d'un système de lecture des plaques d'immatriculation Algérienne*.
Doctoral dissertation, Université Mouloud Mammeri.
- Ait benamar , D. K., & Ouheb , M. (2019). *Reconnaissance automatique de plaque d'immatriculation véhicules pour smart parkings*. Adresse Universite Mouloud MAMMERI Tizi-Ouzou.
- Amnache , A., & Ouadi , A. (2011). *Gestion d'un ensemble de parkings automobiles*. Récupéré sur dl.ummt0: <https://dl.ummt0.dz/handle/ummt0/12930>
- Baldi, P. &. (Jan 2001). *Bioinformatics: The Machine Learning Approach*. MIT Press.
- Baldi, P. (jan 2012). Autoencoders, unsupervised learning, and deep architectures. *Workshop and Conference Proceedings 27* (pp. 37–50). *Journal of Machine Learning Research*,.
- Belkin, M. N. (2006). A Geometric Frame-work for Learning from Labeled and Unlabeled Examples. . *Journal of Machine Learning*, 2399–2434.
- Blum, A. &. (2000). Combining Labeled and Unlabeled Data with Co-Training. *doi:10.1145/279943.279962*
- Bonner, A. (2021, 05 02). *The complete beginner's guide to data cleaning and preprocessing*. Retrieved from <https://towardsdatascience.com/the-complete-beginners-guide-to-data-cleaning-and-preprocessing-2070b7d4c6d>.
- Cai, M. S. (Dec 2013). Deep maxout neural networks for speech recognition. *ASRU 2013 -Proceedings* (pp. 291–296). *IEEE Workshop on Automatic Speech Recognition and Understanding*,.
- Ćelić, J. &. (2012). *SYSTEM FOR AUTOMATIC ROAD TRAFFIC MONITORING*. *Scientific Journal of Maritime Research*,.
- Chinea Manrique de Lara, A. (Nov 2009). Understanding the Principles of Recursive Neural Networks. *doi:10.1007/978-3-642-04274-*. A Generative Approach to Tackle Model Complexity.
- Cireşan, D. M. (Feb 2012). Multi-Column Deep Neural Network for Traffic Sign Classification. (pp. 333–8). *Neural networks : the official journal of the International Neural Network Society* 32.
- Cornelius, & T. Leondes. (2012). *Image Processing and Pattern Recognition*.
- Creswell, A. e. (2017). *Generative Adversarial Networks: An Overview*. ArXiv e-prints .

Bibliographie

- Du, K.-L. &. (dec 2012). . isbn: 978-1-4471-5570-6. doi:10.1007/, (pp. 337–353).
- Dutt, Vinita, Vikas , C., & Khan., I. (2012). "Pattern recognition: an overview. American Journal of Intelligent Systems.
- Fernández Sánchez, & z, Leticia. (2018). *Automatic Number Plate Recognition (ANPR) System*.
- GADOUI, S., KEBIR, F., & MAMOUNI, E. (2020). *Lecture Automatique de Plaques d'Immatriculation par les Machines à Vecteurs de Support*.
- Gersey, B. (2018). *Generative Adversarial Networks*. degree of Master of Advanced Study in Mathematics.
- Goodfellow, I. B. (2021, 14 04). *Deep Learning*. Retrieved from <http://www.deeplearningbook.com>
- Goy, C. (2019). *Montbéliard–Parking de Velotte*. Lyon: Archéologie de la France-Informations. une revue Gallia.
- Griffon, M. O. ((2006).). *Nourrir la planète*.
- Haykin, S. (2008, 01). *Neural Networks and Learning Machine*.
- He, K. Z. (June 2016). Deep Residual Learning for Image Recognition,, (pp. 770–778).
- Hogeweg, P. (Mar 2011). The Roots of Bioinformatics in Theoretical Biology. e1002021. PLoS computational biology 7.
- Khan, A. Z. (2018). Deep Belief Networks Based Feature Generation and Regression for Predicting Wind Power Architectures. *ArXiv e-prints*.
- Krizhevsky, A., , & Sutskever, I. & E. Hinton. (Jan 2012). G. ImageNet Classification with Deep Convolutional Neural Networks. doi:10.1145/3065386. Neural Information Processing Systems 25.
- Lebib , S., & Ait Menguellet , F. (2016). *Détection de plaque d'immatriculation de véhicule basée sur MapReduce dans une séquence Vidéo*. Tizi-Ouzou 15000 Algérie: Adresse Université Mouloud MAMMERI Tizi-Ouzou .
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner. (Dec 1998). P. Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE 86*, (pp. 2278–2324).
- Luo, , & Daisheng. (1998). *Pattern recognition and image processing*.
- Machine Learning & Deep Learning Fundamentals*. (n.d.). Retrieved from Deeplizard.: [http : / / deeplizard.com/learn/video/qSTv_m-KFk0](http://deeplizard.com/learn/video/qSTv_m-KFk0)
- Mirwanti, R. S. (2020). *The Impact of CPR Position (Kneeling, Footstool and Standing Beside Bed) on Cardiopulmonary Resuscitation Quality: A Literature Review*.
- Mitchell, T. M. (1997). *Machine Learning*.
- Moati, P. O. (2001). *L'avenir de la grande distribution*.
- Muallim, & Mohammad Tarek. (2017). *AI. "Pattern Recognition using Artificial Immune"*.

Bibliographie

- Patterson, J. &. (2017). *Deep Learning: A Practitioner's Approach* (" O'Reilly Media).
- Prabhu. (2021, 04 22). Retrieved from Understanding of Convolutional Neural Network (CNN) Deep Learning: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>.
- Qu'est-ce que l'apprentissage profond?* (n.d.).
- R. Bowman, S. P. (June 2014). . Recursive Neural Networks Can Learn Logical. *doi:10.18653/v1/W15-4002*.
- R. Medsker, L. &. (jan 1999). *Recurrent Neural Networks: Design and Applications*.
- Rachid, Z. (2018). *les techniques de reconnaissance de forme, application : a la. université* 08 mai 1945.
- Recursive neural network*. (n.d.). Retrieved from Visited 01-03-2019.: https://en.wikipedia.org/wiki/Recursive_neural
- Reinforcement learning*. (2019, 02 25). Retrieved from https://en.wikipedia.org/wiki/Reinforcement_learning.
- Réseau neuronal convolutif*. (Visited 01-03-2021.). Retrieved from https://fr.wikipedia.org/wiki/Réseau_neuronal_convolutif
- Servais, S. &. (2019). *Quelle est l'influence des caméras ANPR sur l'organisation du travail policier? Surveiller plus ou surveiller mieux?*.
- Sharma, M. (2021, 05 02). *What Steps should one take while doing Data Preprocessing?* Retrieved from <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa>.
- Simonyan, K. &. (Sep 2014). Very Deep Convolutional Networks for Large-Scale Image recognition ., (p. 1409.1556).
- Singh, J. &. (2019). Real Time Indian License Plate Detection using Deep Neural Networks and Optical Character Recognition using LSTM Tesseract. (pp. 347-352). *IEEE*.
- Subarna, D. (2019, 02 25). *artificial- intelligence- 2*. Retrieved from <http://www.engineeringenotes.com>
- Szegedy, C. e. (June 2015). Going deeper with convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1–9).
- yadav, A. (2019). « *Digital image processing* ». new delhi boston usa.
- Zeiler, M. &. (Jan 2013). Visualizing and understanding convolutional networks. *European Conference on Computer Vision(ECCV) 8689*, (pp. 818–833).