



DEPARTEMENT GENIE ELECTRONIQUE

RAPPORT DE STAGE DE FIN D'ETUDES

En vue de l'obtention du diplôme de Licence en :
ELECTRONIQUE, ELECTROTECHNIQUE ET AUTOMATIQUE
Systèmes embarqués

**Réalisation d'un Robot de Reconnaissance des
Panneaux routiers basé sur l'Intelligence
Artificielle**

Elaboré par :

**GABBOUJ Safa
HAJRI Mohamed Mortadha**

Encadré par :

HAJJAJI Mohammed Ali

**Institut Supérieur des Sciences Appliquées et de
Technologie de Sousse**

BOUAAFIA Soulef

**Laboratoire d'Electronique et
Microélectronique**

Année Universitaire : 2021 / 2022

Code Sujet: [L-SE22-032]

Dédicace

Qu'il me soit permis d'exprimer ma plus profonde gratitude envers ceux qui m'ont encouragé et qui ont toujours été là pour moi je dédie ce travail.

À mon cher père Gabbouj Ahmed, qui décède trop tôt. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'un fils qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout-puissant, l'avoir en sa sainte miséricorde.

À ma chère mère Zardoub Najet que nulle dédicace ne puisse exprimer ce que je lui dois, pour sa bienveillance depuis ma plus jeune enfance, son soutien et son affection. En témoignage de mon profond amour et ma gratitude pour les sacrifices qu'elle avait consenti que n'a cessé de m'assister, me soutenir et m'encourager. À celui qui a sacrifié ses plus belles années pour embellir les miennes, je lui dois ma réussite, aucun mot ne serait assez pour témoigner l'étendue des sentiments que j'éprouve à son égard. J'aime infiniment. Que Dieu vous préserve, vous accorde santé, bonheur et vous protège de tout mal.

À mes chers frères Wided et Hamdi, je vous remercie pour vos encouragements et vos soutiens, tout au long de mon cursus universitaire, je vous dédie ce travail en souhaitant un Radieux, plein de bonheur et de succès. Vous comptez énormément pour moi, je t'estime beaucoup et je vous aime beaucoup. Je vous souhaite beaucoup de succès, de prospérité et une vie pleine de joie et de bonheur.

GABBOUJ Safa

Dédicace

Je dédie ce modeste travail :

L'être le plus cher au monde en témoignage de mon respect, à mon amour et

mon plus grand rattachement, à l'âme de mon très cher père RIDHA.

La femme la plus affectueuse et la plus douce au monde, l'ange le plus tendre

Qui a été toujours pour moi une source d'amour, de pitié et d'espoir, ma très

Chère mère Noura.

À ma sœur Sayda, qui a été toujours à mes côtés.

A mes très chers amis qui m'ont aidé et encouragé tout le long de ce chemin et

Qui sont la source de ma confiance.

HAJRI Mohamed Mortadha

Remerciement

Je remercie Dieu le plus puissant de m'avoir donné la santé et la volonte d'entamer et de terminer ce projet de fin d'études.

Mes sincères remerciements s'adressent d'abord aux membres respectés de jury qui me font le grand honneur d'évaluer ce modeste travail. Je remercie infiniment **Mr HAJJAJI Mohamed Ali** mon encadrant académique à l'ISSAT qui m'a l'honneur et le plaisir de m'encadrer, pour précieux conseils et ses efforts durant la réalisation de ce travail et de diriger avec amabilité et patience.

Au terme de ce travail, je tiens aussi à exprimer tous mes remerciements à **Mme BOUAAFIA Soulef** mon encadrant professionnel pour son accueil chaleureux, pour ses encouragements et pour les conseils qu'il m'a apportés lors de ses suivis.

Je remercie aussi Mme /Mr pour l'honneur qu'il m'a fait en acceptant de présider le jury de soutenance.

Je remercie Mr /Mme d'avoir accepté de participer à ce jury.

Enfin je n'oublie pas de remercier tous les enseignants qui ont participé à ma formation de licence durant toutes mes années d'études à l'ISSAT.

GABBOUJ Safa et HAJRI Mohamed Mortadha

Sommaire

Introduction Générale	1
Chapitre 1 : Etat de l'Art sur la Reconnaissance Intelligente des Panneaux Routiers	2
1. Introduction.....	2
2. Robot intelligent.....	2
3. Intelligence Artificielle.....	3
4. Apprentissage automatique.....	3
4.1. Apprentissage supervisé.....	4
4.2. Apprentissage non supervisé	5
4.3. Apprentissage semi supervisé	7
4.4. Apprentissage par renforcement	7
5. Apprentissage profond	8
5.1. Réseau de neurones	8
5.2. Perceptron	9
5.3. Perceptron multicouche	9
5.4. Réseau de neurones profond (DNN)	10
6. Réseau de neurones convolutifs pour la reconnaissance des panneaux routiers	10
6.1. Couche de convolution	11
6.2. Couche de mise en commun: pooling.....	11
6.3. Couche Flatten.....	12
6.4. Couche Entièrement connecté (fully-connected)	12
7. Fonctions d'activation	13
7.1. Relu.....	13
7.2. Softmax	14
8. Phase d'apprentissage : Algorithme de rétro propagation.....	14
9. Problématique	16
10. Conclusion	17
Chapitre 2 : Conception du système intelligent pour la reconnaissance des panneaux routiers.....	18
1. Introduction.....	18
2. Modèle proposé pour la reconnaissance des panneaux routiers	18
2.1. Phase d'entraînement.....	19
2.2. Phase de test	19

3.	Architecture du modèle CNN proposé	19
4.	Base de données « German_Traffic_signs »	20
5.	Phase d'apprentissage	22
5.1.	Environnement utilisé	22
5.2.	Paramètre d'apprentissage	22
6.	Metrics dévaluation	23
6.1.	Confusion Matrix.....	23
6.2.	Accuracy score	23
6.3.	Précision.....	24
6.4.	Recall score	24
6.5.	F1-Score.....	24
7.	Résultats obtenus et discussion.....	24
7.1.	Accuracy et Loss pour Training et validation	25
7.2.	Confusion Matrix pour le modèle CNN proposé	26
7.3.	Précision.....	26
7.4.	Recall	27
7.5.	F1-Score.....	28
8.	Phase de Test	28
9.	Conclusion	29
Chapitre3 : Réalisation d'un Robot Intelligent		30
1.	Introduction.....	30
2.	Choix du matériel.....	30
2.1.	Raspberry Pi 3 (Model B)	30
2.2.	Arduino Uno.....	31
2.3.	Raspberry Pi Camera V1	31
2.4.	Module L298	31
2.5.	Module suiveur de ligne.....	32
3.	Premier démarrage de la carte Raspberry	32
3.1.	Création carte SD	32
3.2.	Configuration et démarrage de la carte.....	33
4.	Relation entre Raspberry et Arduino.....	34
5.	Implémentation du modèle au carte et simulation réel du système	34
5.1.	Conception du système	34
5.2.	Implémentation du modèle sur la carte.....	35

6.	Assemblage du Robot.....	36
6.1.	Réalisation en ISIS	36
6.2.	Réalisation réelle	37
7.	Conclusion	38
	Conclusion Générale	39

Liste des figures

Figure 1: Domaine d'application de l'intelligence artificielle	3
Figure 2: Exemple d'apprentissage supervisé	4
Figure 3: Exemple de classification.....	5
Figure 4: Exemple de la régression linéaire	5
Figure 5: Exemple d'algorithme de regroupement	6
Figure 6: Exemple de la réduction en dimension.	6
Figure 7 : Apprentissage semi-supervisé	7
Figure 8: Exemple de l'apprentissage par renforcement	8
Figure 9 : Architecture de perceptron	9
Figure 10 : Architecture de perceptron multicouche	10
Figure 11: Architecture de réseau de neurone profond (DNN)	10
Figure 12 : Architecture Réseau de neurones convolutifs (CNN).....	11
Figure 13: Exemple de produit de convolution	11
Figure 14 : Exemple de Couche de Max pooling et Average pooling.....	12
Figure 15 : Architecture de la couche entièrement connecté	13
Figure 16: Fonction ReLU	13
Figure 17: Fonction Softmax.....	14
Figure 18 : Exemple de la rétropropagation.....	14
Figure 19 : Cercle statistique pour Les raisons de conception de ce robot.....	16
Figure 20 : Architecture globale d'un modèle CNN.....	18
Figure 21 : Architecture de modèle CNN proposé	20
Figure 22 : Les nombres d'images dans chaque classe.....	20
Figure 23 : Test.csv.....	21
Figure 24 : Base de données de German Traffic Signs.....	21
Figure 25 : Problème de classification binaire (matrice 2x2)	23
Figure 26 : Accuracy et taux d'erreur (Loss) pour le Modèle	25
Figure 27 : Matrice de confusion de modèle CNN	26
Figure 28 : Histogramme de précision pour 43 classes	27
Figure 29 : Histogramme de Recall score pour 43 classes.....	27
Figure 30 : Histogramme de F1-score pour 43 classes	28

Figure 31 : Interface graphique pour la classification de notre model	29
Figure 32 : Tester en temps réel avec camera webcam	29
Figure 33 : Composition générale d'une Raspberry Pi 3	30
Figure 34 : Chemin d'un carte Arduino Uno et ses composants	31
Figure 35 : Raspberry Pi Camera.....	31
Figure 36 : Module L298	32
Figure 37 : Module suiveur de ligne	32
Figure 38 : Raspberry pi imager	33
Figure 39 : Affichage du Raspberry pi au VNC VIEWER	34
Figure 40 : Relation entre Raspberry Pi3 et Arduino Uno	34
Figure 41 : Schéma de système	35
Figure 42 : Simulation de système sur Raspberry pi 3	36
Figure 43 : Réalisation en ISIS.....	37
Figure 44 : Robot Intelligent	37

Liste des tableaux

Tableau 1 : Environnement utilisé pour la phase d'apprentissage	22
Tableau 2: paramètre nécessaire pour la phase d'apprentissage du modèle CNN proposé.....	22
Tableau 3: Résultat d'apprentissage du notre modèle	24
Tableau 4 : Résultat de Metrics de précision de quelques classes du modèle CNN proposé	26
Tableau 5: Résultat de Metrics de Recall de quelques classes du modèle CNN proposé	27
Tableau 6 : Résultat de Metrics de F1-score de quelques classes du modèle CNN proposé	28

Introduction Générale

L'intelligence artificielle est un domaine qui ne cesse de progresser, elle consiste à mettre en œuvre un certain nombre de techniques et de développer des programmes informatiques pour traiter des problèmes difficiles résolus par l'homme et pour permettre aux machines de réaliser des fonctions conformes aux formes d'intelligence réelles, dont les ordinateurs ont été capables de faire des tâches et des calculs complexes d'où la solution nécessite des connaissances, de la perception, du raisonnement et de l'apprentissage.

On peut exploiter cette technologie dans des domaines différents tels que la robotique, les voitures autonomes, la santé etc. Dans nos jours le nombre des accidents a été augmenté d'une façon prodigieuse, c'est pourquoi les chercheurs ont inventé la technologie de l'autonomie dans les voitures ou d'une autre façon la voiture autonome. Les systèmes autonomes, ont un rôle important dans la réduction de taux d'accidents dans le cours, la régulation routière, la sécurité économique ainsi qu'ils facilitent la mobilité (en particulier pour les personnes handicapées), etc.... C'est dans ce contexte on a créé un robot autonome pour la reconnaissance des panneaux routiers, et on lui a appris à reconnaître et à détecter les panneaux de signalisation (Stop, tournez à droite et à gauche, etc.).

Pour réaliser ce projet de fin d'études nous avons divisé notre travail en trois chapitres. Tout d'abord on commence par le premier chapitre dont nous avons donné une présentation générale de l'intelligence artificielle et ses différents algorithmes d'apprentissage utilisés jusqu'à arriver aux algorithmes de l'apprentissage profond et plus précisément au réseau des neurones convolutif (CNN) qui est classé l'une des méthodes la plus efficace et la plus utilisée dans cet apprentissage pour faire la reconnaissance des différents objets. Pour cela nous lui avons adapté cette méthode comme une solution adaptable pour notre projet grâce à sa simplicité et son efficacité. Ensuite, dans le deuxième chapitre on a présenté la conception détaillée de notre modèle CNN, au sein de ce chapitre nous avons étudié l'architecture générale de notre modèle, en présentant les différents logiciels et bibliothèques utilisées pour réussir notre conception.

Et finalement pour le dernier chapitre nous avons présenté la conception détaillée de robot on définit les différents composants matériels et les configurations logicielles faites pour réaliser le robot à travers des captures d'écran. Et nous achevons notre travail par une conclusion générale.

Chapitre 1 : Etat de l'Art sur la Reconnaissance Intelligente des Panneaux Routiers

1. Introduction

Depuis l'avènement de la robotique, les chercheurs en informatique tentent d'injecter le concept d'intelligence ou d'intelligence humaine dans les machines, et cela doit être fait par l'homme, cette technologie s'appelle l'intelligence artificielle, ce qui signifie que cette technologie est le résultat de l'homme ayant besoin d'effectuer des calculs complexes, en effet, cette technologie est devenue un sujet avec de nombreux problèmes et fait partie de notre vie quotidienne, cette technologie se développe chaque jour, elle comprend l'apprentissage automatique et l'apprentissage profond.

2. Robot intelligent

Le cinéma et les films d'animations depuis les années 1980 ne sont pas en reste. L'on pense bien évidemment au Terminator, à The Matrix, ou à Goldorak et à tout l'univers Manga qui en a découlé, mais l'on peut lister également dans un contexte plus léger des œuvres filmographiques qui traitent du côté sentimental de la relation homme-intelligence artificielle (par exemple, La Belle et l'Ordinateur ou dans sa version originale Electric dreams). A cette époque, est apparue KITT dans la série Knight Rider (K2000), voiture équipée d'une intelligence artificielle la rendant autonome et capable de prendre des décisions. Notre étude porte sur les robots intelligents, mais il en existe une très large variété.

Un robot est un système appliquant des programmes informatiques capable de capter, stocker, traiter et communiquer des informations, de décider et d'agir afin de remplir une mission en interaction avec l'humain. L'apprentissage automatique, qu'il soit supervisé ou non, permet de mettre en œuvre des mécanismes d'adaptation à l'évolution d'un environnement qui peuvent être intégrés à ces robots pour les rendre intelligents. Parmi les outils techniques et scientifiques permettant à un robot d'apprendre, on trouve tous les outils relevant de l'intelligence artificielle (apprentissage par renforcement, techniques de classification, réseaux de neurones, logique floue, systèmes multi-agents, etc.) [1].

L'apprentissage permet à un système d'améliorer ses performances au fil du temps avec l'expérience en appliquant des décisions qui peuvent évoluer à plus ou moins long terme, même si le contexte décisionnel reste le même.

De notre point de vue, un robot devient intelligent dès lors qu'il met en œuvre des techniques d'intelligence artificielle pour apprendre à améliorer ses décisions au fil du temps. Un tel robot

intelligent doit présenter un degré d'autonomie dans ses décisions suffisant pour lui permettre premièrement de décider par lui-même sans qu'il soit possible systématiquement, d'un œil externe en mettre comme exemple le robot autonome pour la détection des panneaux routiers.

3. Intelligence Artificielle

Le concept de l'intelligence artificielle consiste à faire penser les machines comme les humains, c'est-à-dire à effectuer des tâches telles que le raisonnement, la planification, l'apprentissage et la compréhension de notre langage. En d'autres termes, cette technologie implique la mise en œuvre de nombreuses techniques permettant aux machines d'imiter une certaine forme de Langue. Véritable intelligence, en fait, cette technologie a d'abord été inventée par le mathématicien "Alan Turing" en 1950, puis par le scientifique américain "Marvin Lee Minsky" en 1956 [2].

Il est clair que l'intelligence artificielle est un domaine qui implique et réunit l'informatique, les mathématiques, les neurosciences et la philosophie, et cette technologie devient de plus en plus utile à notre époque, entrant dans nos vies par le biais d'assistants personnels, d'apprentissage en profondeur et de réseaux de neurones. On peut exploiter l'IA dans différents domaines, comme l'indique la figure 1 suivante.

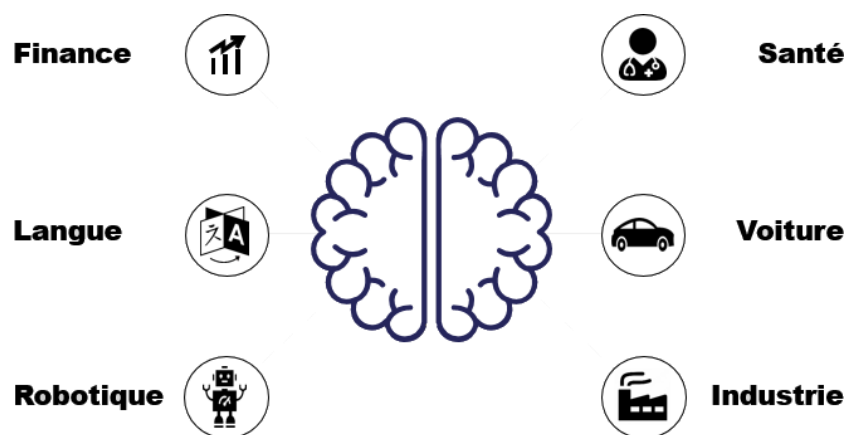


Figure 1: Domaine d'application de l'intelligence artificielle

4. Apprentissage automatique

L'apprentissage automatique est une branche de l'intelligence artificielle qui vise à donner aux ordinateurs la capacité d'apprendre. Un ordinateur n'est pas intelligent, il exécute simplement des tâches. Le programme lui a appris quoi faire et comment le faire. C'est ce qu'on appelle la programmation. La machine Learning aborde des sujets complexes pour lesquels la programmation traditionnelle trouve ses limites. Par exemple, construire un programme qui pilote une voiture est très difficile, voire impossible. En raison du grand nombre d'instructions, avec l'aide de ML, nous pouvons aborder ce problème de différentes manières. Le programme ne décrit pas ce qu'il faut faire, mais

s'apprend plutôt à conduire à travers des expériences "d'observation". Cela vous permet d'évaluer l'optimalité de certaines méthodes par rapport à d'autres. Il existe 4 catégories de l'apprentissage automatique :

- ✓ Apprentissage supervisé
- ✓ Apprentissage non supervisé
- ✓ Apprentissage semi-supervisé
- ✓ Apprentissage par renforcement

4.1. Apprentissage supervisé

L'apprentissage supervisé est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples étiquetée, au contraire de l'apprentissage non supervisé. Les algorithmes d'apprentissage supervisé prennent un ensemble de données et utilisent ses fonctionnalités pour apprendre une relation avec un ensemble d'étiquettes correspondant. Ce processus est connu sous le nom de formation et une fois terminé, nous espérons que notre algorithme ferait un bon travail de prédiction des étiquettes de toutes nouvelles données dans lesquelles l'algorithme n'a aucune connaissance explicite de la véritable étiquette [3].

L'algorithme tente alors de trouver la règle permettant de relier l'entrée à la sortie, comme il est indiqué dans la figure 2.

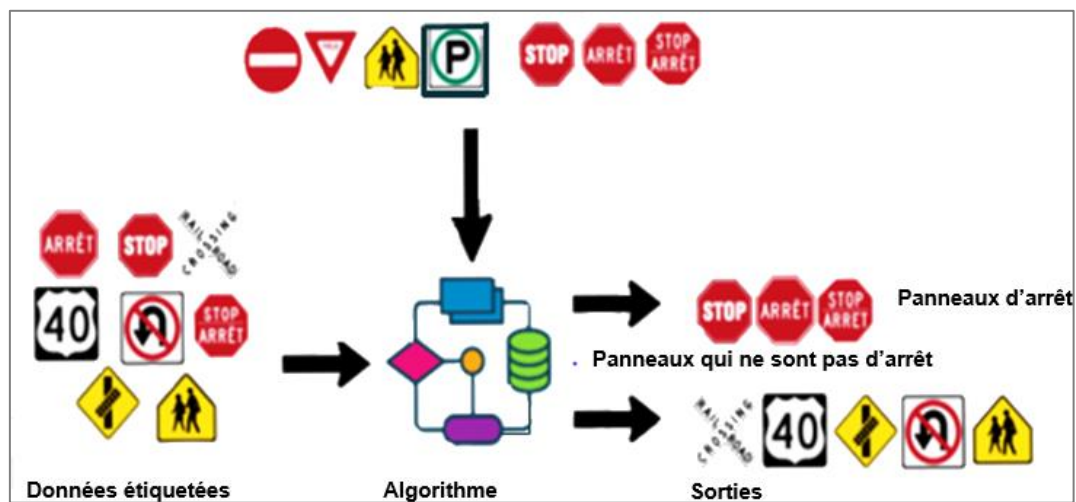


Figure 2: Exemple d'apprentissage supervisé

➔ L'apprentissage supervisé comprend deux catégories d'algorithmes :

✓ Classification

Nous utilisons cette méthode lorsque nous avons plusieurs entrées divisées en deux classe ou plus, et que ces données doivent être disponibles, nous classons donc les données (texte, images, vidéos) dans des dossiers. Par exemple, nous nous concentrerons sur la classification des images, cela inclut les éléments de marquage comme l'indique la figure 3.

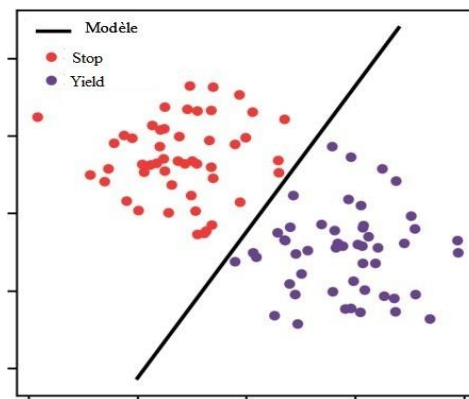


Figure 3: Exemple de classification

✓ Régression

Dans le cas où les étiquettes sont à valeurs réelles, on parle de régression. L'idée de la régression linéaire est simplement de trouver une ligne qui s'adapte (ou correspond) le mieux aux données, comme il est indiqué dans la figure 4.

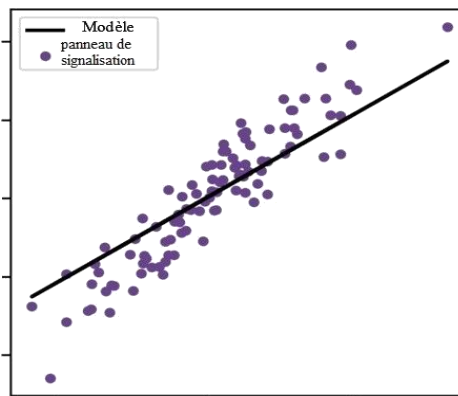


Figure 4: Exemple de la régression linéaire

4.2. Apprentissage non supervisé

L'apprentissage non supervisé permet aux algorithmes de faire ces prédictions sur les données elles-mêmes. Ils sont très utiles pour fournir aux humains des informations sur la signification des données [4]. Exemple d'une voiture autonome. Dans l'apprentissage automatique non supervisé, le système ne reçoit aucune image d'entraînement et les résultats attendus sont non étiquetés ou inconnus. Ou en

d'autres termes, la sortie est indéfinie. Ainsi, l'algorithme prend les données et recherche ensuite des modèles par lui-même. L'apprentissage non supervisé comprend 2 types (clustering et réduction de dimension).

✓ Clustering

Tout d'abord, le clustering consiste à identifier des groupes dans les données. Cela permet de comprendre leurs caractéristiques générales, et éventuellement d'inférer les propriétés d'une observation en fonction du groupe auquel elle appartient, voir l'exemple dans la figure 5.

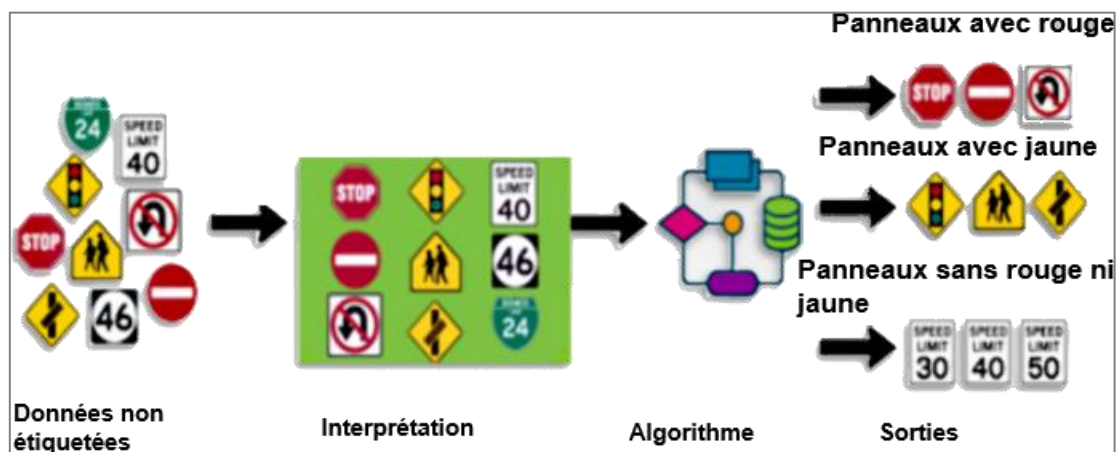


Figure 5: Exemple d'algorithme de regroupement

✓ Réduction de dimension

La réduction de dimension est une autre famille importante de problèmes d'apprentissage non supervisé. Il s'agit de trouver une représentation des données dans un espace de dimension plus faible que celle de l'espace dans lequel elles sont représentées à l'origine. Cela permet de réduire les temps de calcul et l'espace mémoire nécessaire au stockage des données, mais aussi souvent d'améliorer les performances d'un algorithme d'apprentissage supervisé entraîné par la suite sur ces données, comme indique à la figure 6.

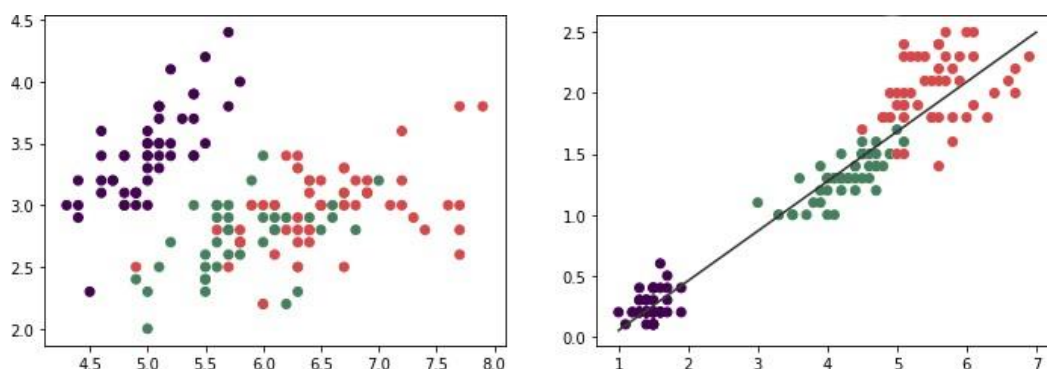


Figure 6: Exemple de la réduction en dimension.

➔ A la différence de l'apprentissage supervisé qui fait appel à des données étiquetées ou annotées pour réaliser des prédictions, l'apprentissage non supervisé n'a pas besoin d'étiquette. Puisque les données ne sont pas étiquetées, il n'est pas possible pour le modèle de calculer des scores de réussite. En conséquence, alors que les systèmes supervisés se concentrent sur les tâches de régression et classification, la technique non supervisée est utilisée pour effectuer des regroupements ou des mises en grappe de données en fonction de leurs ressemblances ou différences.

4.3. Apprentissage semi supervisé

L'apprentissage semi-supervisé est une approche de l'apprentissage automatique qui combine une petite quantité de données étiquetées avec une grande quantité de données non étiquetées pendant la formation. Comme le montre la figure 7. L'apprentissage semi-supervisé se situe entre l'apprentissage non supervisé (sans données de formation étiquetées) et l'apprentissage supervisé (avec uniquement des données de formation étiquetées). C'est un cas particulier de faible supervision [5].

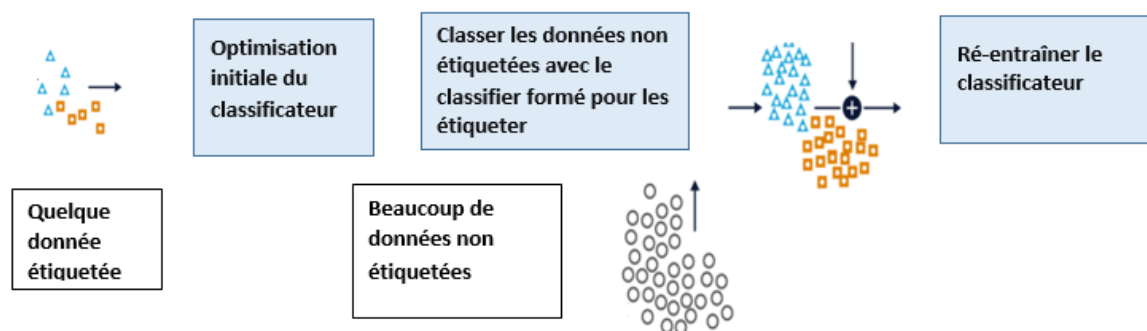


Figure 7 : Apprentissage semi-supervisé

4.4. Apprentissage par renforcement

L'apprentissage par renforcement est une méthode de Machine Learning. Elle consiste à entraîner des modèles d'intelligence artificielle d'une manière bien spécifique. L'agent IA doit apprendre à atteindre un objectif au sein d'un environnement incertain et potentiellement complexe. Pour y parvenir, l'ordinateur essaye toutes les façons possibles et apprend de ses erreurs. À chaque tentative, l'IA reçoit une récompense ou une punition en fonction des actions effectuées. Elle est programmée pour maximiser sa récompense, et tentera donc de trouver la méthode le lui permettant [3]. Comme indique La figure 8.



Figure 8: Exemple de l'apprentissage par renforcement

5. Apprentissage profond

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique, qui est essentiellement un réseau de neurones composé de trois couches ou plus. Ces réseaux de neurones parviennent à imiter le comportement du cerveau humain et lui permettent d'apprendre à partir de grandes quantités de très grandes données [6] [7]. Alors qu'un réseau de neurones avec une seule couche peut encore faire des sacrifices ou des prédictions approximatives, l'utilisation de couches cachées supplémentaires peut aider à améliorer et à affiner pour une plus grande précision. L'apprentissage en profondeur est à la base de nombreuses applications et services d'intelligence artificielle (IA) qui existent dans tous nos services quotidiens (tels que les assistants numériques, les objets télécommandés pour le contrôle vocal et la détection de mouvement) et les technologies émergentes (telles que la conduite autonome).

5.1. Réseau de neurones

Un réseau de neurones est un modèle informatique similaire au réseau des neurones du cerveau au niveau de structure en couches, il est constitué par des couches connectées. Un réseau neurones peut apprendre à partir de données ; il peut ainsi être entraîné à reconnaître des objets, classer des données et prévoir des résultats et des événements à venir. Un réseau de neurones décompose vos données en couches d'abstraction. Il peut être entraîné sur de nombreux exemples en vue de reconnaître des modèles au niveau de la parole ou des images, par exemple, le cerveau humain. Son comportement est défini par la façon dont ses éléments individuels sont reliés et par la solidité (ou poids) de ces liaisons. Ces poids sont automatiquement ajustés au cours de l'entraînement selon une règle d'apprentissage spécifiée jusqu'à ce que le réseau de neurones exécute correctement la tâche souhaitée [8].

5.2. Perceptron

Un perceptron est un algorithme d'apprentissage supervisé dont le rôle est de séparer linéairement deux classes (classificateurs binaires), et il a n entrées (X_1 à X_n), qui sont typiquement des entiers ou des nombres réels alimentés au perceptron. Chaque entrée X est associée à un poids W qui sera utilisé pour calculer la sortie Y (comme il est indiqué dans la figure 9). Enfin, une connexion de polarisation (b) est ajoutée car elle est nécessaire au bon fonctionnement du perceptron [9]. Le calcul effectué pour obtenir la sortie Y est le suivant :

$$F(x) = \sum_{k=0}^n (x^k * wk) + \text{bias}$$

$$F(x) = 1 \text{ si } \sum_{k=0}^n (x^k * wk) \geq 0$$

$$F(x) = 0 \text{ si } \sum_{k=0}^n (x^k * wk) \leq 0$$

La figure 13 explique l'architecture de perceptron et comment ça passe.

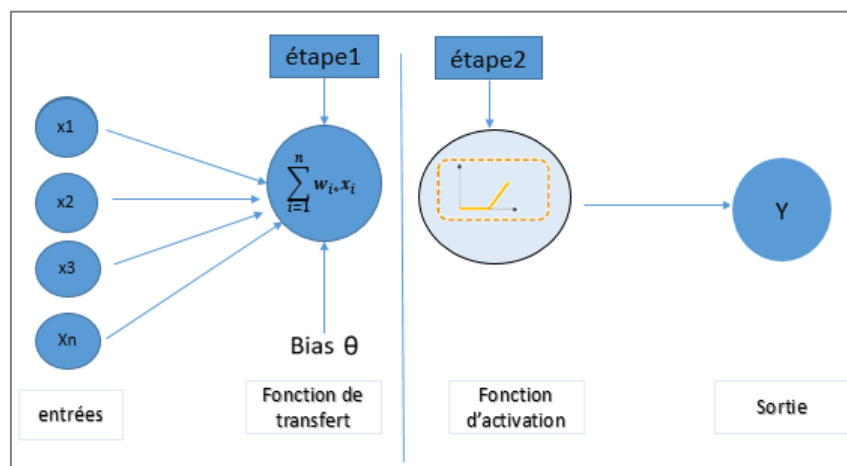


Figure 9 : Architecture de perceptron

5.3. Perceptron multicouche

Les perceptrons multicouches ont été les premiers réseaux de neurones à découvrir de nombreuses applications pratiques, telles que la reconnaissance des fleurs, la détection des fraudes, etc. Il peut être utilisé pour toutes les tâches de classification supervisées. Actuellement, c'est l'un des modèles les plus populaires et il est implémenté par de nombreuses bibliothèques telles que TensorFlow, Weka, Scikit-Learn, etc. La particularité topologique de ce réseau est que tous les neurones d'une couche sont connectés à tous les neurones de la couche suivante, ce qui donne un phénomène appelé propagation directe. Ainsi, chaque neurone a n entrées, où n est le nombre de neurones présents dans la couche précédente, et une sortie qui est envoyée à tous les neurones de la couche suivante, comme le montre la figure 10 [9].

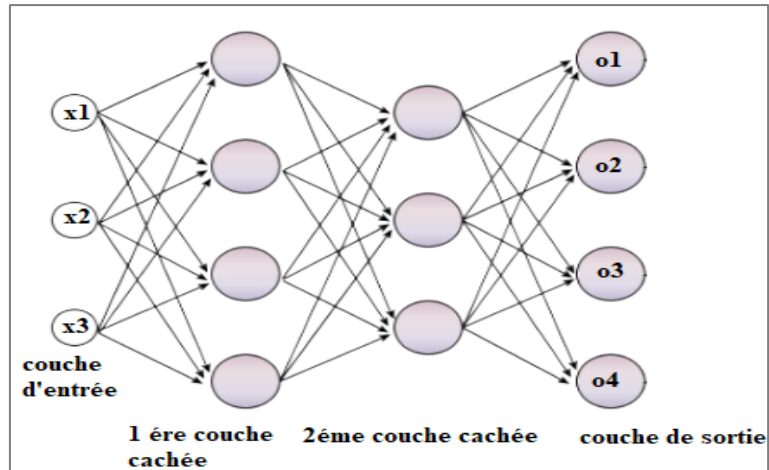


Figure 10 : Architecture de perceptron multicouche

5.4. Réseau de neurones profond (DNN)

Ces réseaux de neurones multicouches peuvent contenir des millions de neurones répartis sur des dizaines de couches. Les réseaux de neurones profonds utilisent une modélisation mathématique sophistiquée pour traiter les données de manière complexe, comme le montre la figure 11.

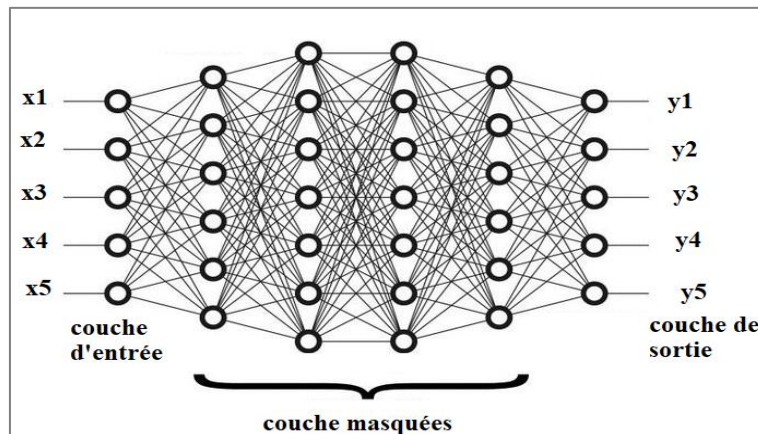


Figure 11: Architecture de réseau de neurone profond (DNN)

6. Réseau de neurones convolutifs pour la reconnaissance des panneaux routiers

Le CNN (Convolutional Neural Networks) est utilisé pour remplacer les réseaux de neurones traditionnels car les réseaux de neurones traditionnels ne sont pas idéaux pour le traitement d'images et doivent fournir des images à faible résolution. Le CNN (Figure 12) améliore l'intelligence de la voiture, en collectant plus d'informations, en les filtrant et en les classant. À l'aide de CNN nous cherchons à créer des modèles bien conçus pour entraîner les programmes des voitures [10].

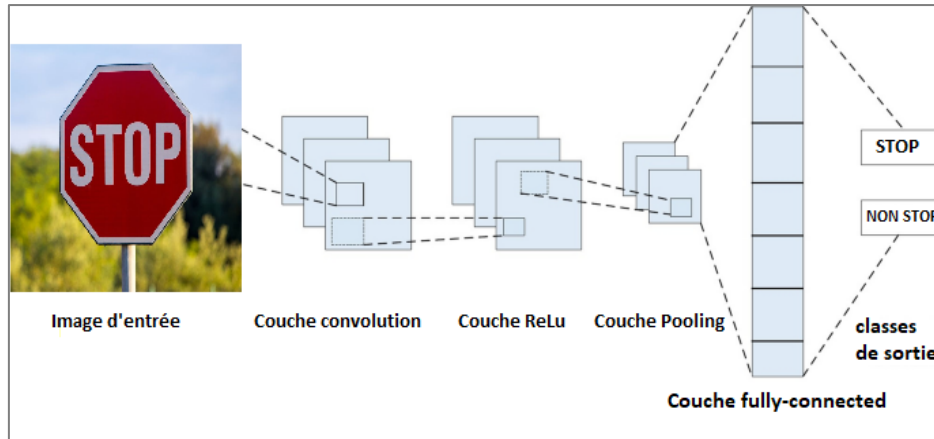


Figure 12 : Architecture Réseau de neurones convolutifs (CNN)

6.1. Couche de convolution

La couche convolutive est la composante la plus importante de toute architecture CNN. Elle contient un ensemble de noyaux convolutifs (également appelés filtres), qui sont convolués avec l'image d'entrée (métriques à N dimensions) pour générer une carte de caractéristiques [10].

$$O = \frac{\omega + 2P - f}{S}$$

La taille du filtre, padding, stride sont respectivement désignés par : f , P , S . Les nombres canaux doivent être les mêmes pour l'image d'entrée et le filtre. Voir l'exemple de la figure ci-dessous pour une image avec 3 canaux (R, G, B) de dimensions $6 \times 6 \times 3$. Le filtre à des dimensions $3 \times 3 \times 3$, où les 3 derniers sont le nombre de canaux. L'image résultante à la dimension 4×4 , en supposant $s=1$ et $p=0,0$, comme le montre la figure 13.

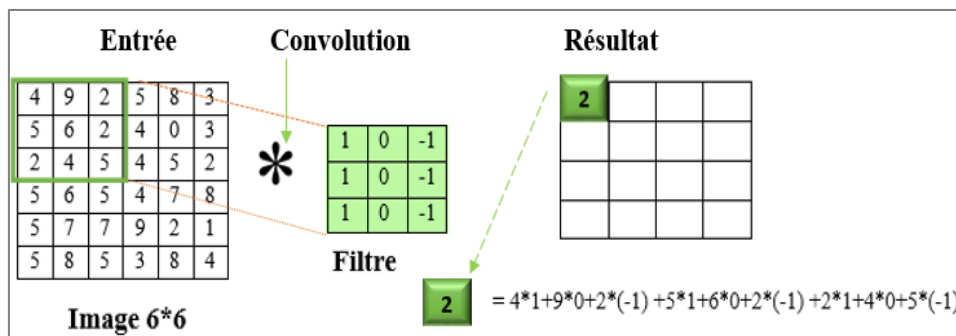


Figure 13: Exemple de produit de convolution

6.2. Couche de mise en commun: pooling

Cette étape permet de réduire la taille des images d'entrées, de réduire la charge de travail en diminuant les paramètres du réseau et donc la charge de calcul, tout en gardant les principales caractéristiques de l'image. Cette étape est composée de deux parties :

✓ Mise en commun moyenne

La mise en commun moyenne fonctionne en calculant la moyenne des valeurs de pixel dans le champ réceptif. Pour 4 pixels avec les valeurs 3, 9, 0 et 6, la couche de regroupement moyenne donnera une sortie de 4,5. Arrondir à un nombre entier nous donne 5.

✓ Mise en commun maximale

Dans le pool maximal, le filtre sélectionne simplement la valeur de pixel maximale dans le champ réceptif. Par exemple, si vous avez 4 pixels dans le champ avec les valeurs 3, 9, 0 et 6, vous sélectionnez 9, comme l'indique la figure 14.

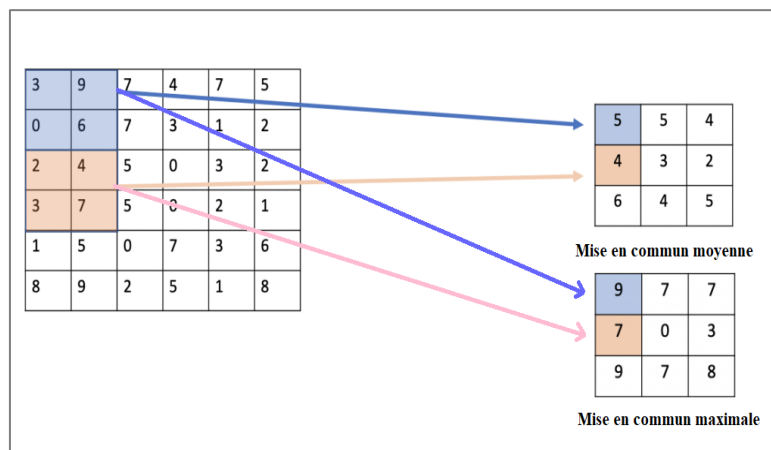


Figure 14 : Exemple de Couche de Max pooling et Average pooling

Bien que beaucoup d'informations soient perdues dans la couche de mise en commun, elles présentent également un certain nombre d'avantages pour CNN. Ils aident à réduire la complexité, à améliorer l'efficacité et à limiter le risque de sur-ajustement.

6.3. Couche Flatten

Une couche Flatten a pour rôle de transformer les données de deux dimensions en un vecteur d'une seule dimension de caractéristiques issu des convolutions à une dimension.

6.4. Couche Entièrement connecté (fully-connected)

Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme Fully Connected). Après avoir reçu un vecteur en entrée, la couche FC applique successivement une combinaison linéaire puis une fonction d'activation dans le but final de classifier l'input image. Elle renvoie enfin en sortie un vecteur de taille d correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'input image d'appartenir à une classe, voir la figure 15.

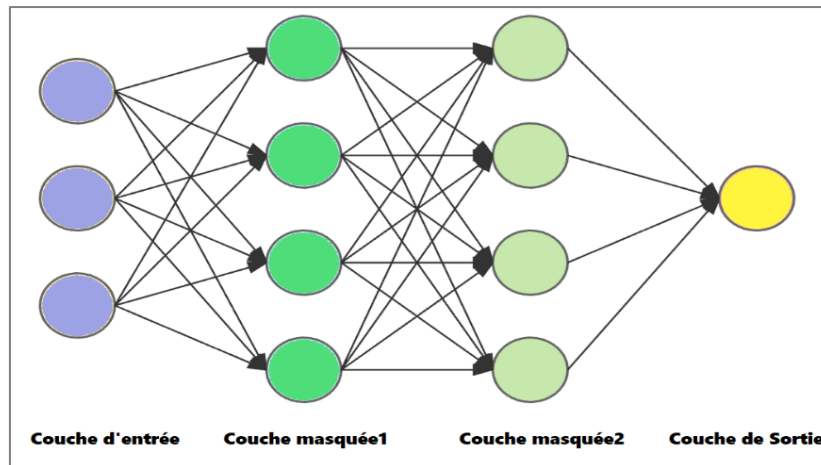


Figure 15 : Architecture de la couche entièrement connecté

7. Fonctions d'activation

Enfin, l'un des paramètres les plus importants d'un modèle CNN est la fonction d'activation. Ils sont utilisés pour apprendre et approximer tout type de relation continue et complexe entre les variables du réseau. En bref, il décide quelles informations du modèle doivent se déclencher vers l'avant et lesquelles ne doivent pas se déclencher à la fin du réseau. Cela ajoute de la non-linéarité au réseau. Les fonctions d'activation couramment utilisées sont les fonctions ReLU, Softmax et Sigmoid. Chacune de ces fonctions a un but précis.

7.1. Relu

Il est possible d'améliorer l'efficacité du traitement en intercalant une couche entre les couches de traitement qui effectuera une fonction mathématique (fonction d'activation) décrite dans la Figure 16 sur les signaux de sortie. La fonction ReLU [10].

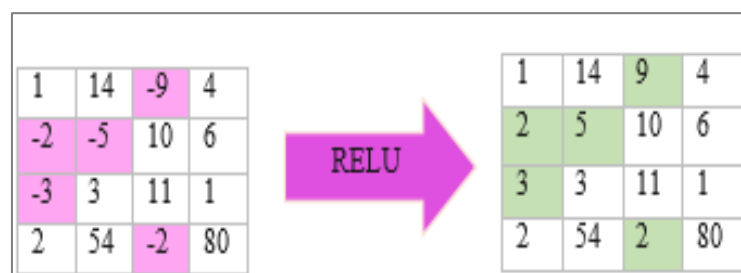


Figure 16: Fonction ReLU

7.2. Softmax

On peut utiliser La fonction mathématique Softmax dans l'apprentissage automatique pour convertir les scores en probabilités pour la classification multi classe en se basant sur l'équation $\sigma(\mathbf{z})_j =$

$\frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$, Pour tout $j \in \{1, \dots, k\}$, Comme l'indique la figure 17.

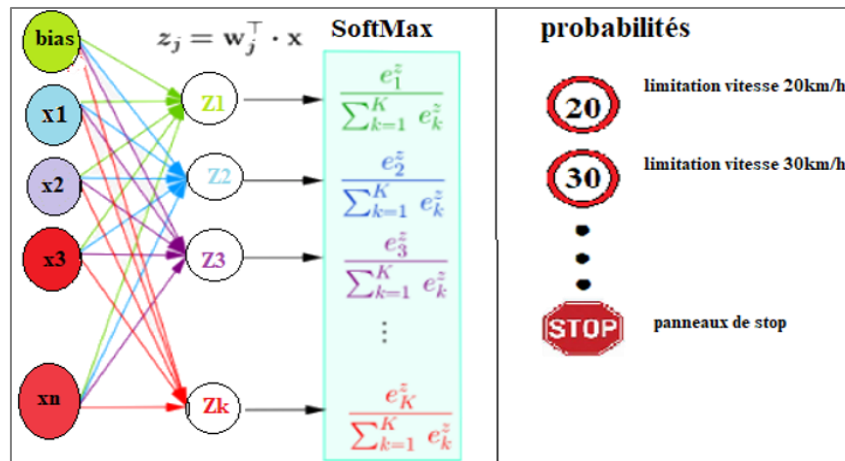


Figure 17: Fonction Softmax

8. Phase d'apprentissage : Algorithme de rétro propagation

La rétropropagation de gradient est une méthode d'entraînement d'un réseau de neurones qui consiste à mettre à jour les poids de chaque neurone de la dernière couche à la première comme l'indique la figure 18. Il est conçu pour corriger les erreurs en fonction de l'importance de chacun de ces éléments intermédiaires. Dans le cas des réseaux de neurones, les poids qui contribuent davantage aux erreurs varieront plus que les poids qui causent de petites erreurs [11].

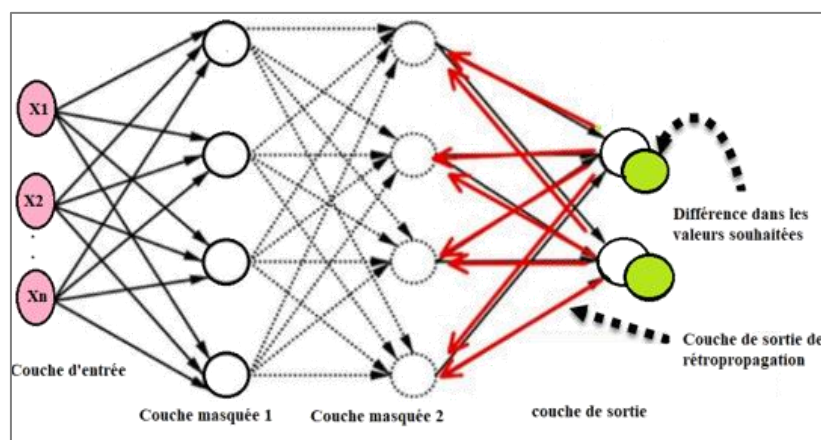


Figure 18 : Exemple de la rétropropagation

Cet algorithme fonctionne de manière suivante :

- Au début on va initialiser les poids du MLP (multi Layer perceptron) [12].

- Propagation en avant : pour chaque vecteur de donnée :

- Affecter $X_k = (n = 1, \dots, N)$; envoyer les signaux aux neurones cachés.

- chaque neurone caché calcule son entrée

$$Z_{in_j} = v_{j0} + \sum_{n=1}^N X_n V_{jn}$$

- Appliquer sa fonction d'activation pour calculer sa sortie

$$Z_j = f(Z_{in_j})$$

- chaque neurone de sortie calcule son entrée

$$Y_{inM} = w_{m0} + \sum_{j=1}^J Z_j W_{mj}$$

- Appliquer sa fonction d'activation pour calculer $y_k = f(Y_{inM})$

- **Rétropropagation** : chaque neurone de sortie reçoit son étiquette

- calculer le gradient $\delta_m = (t_m - y_m) f'(Y_{inM})$.

- calculer les incréments $\Delta w_{mj} = \eta Z_j \delta_m$.

- retropropager les gradients δ_m vers la couche cachée qui précède

$$\delta_m = \frac{1}{M} \sum_{i=1}^M (a_{m1} X_i + b - y_i)$$

- chaque neurone caché calcule son δ_{in_j} correspond selon

$$\delta_{in_j} = \sum_{m=1}^M \delta_m W_{mj}$$

- chaque neurone caché calcule son gradient $\delta_j = \delta_{in_j} f'(Z_{in_j})$.

- calcule les incréments $\Delta v_{jn} = \eta X_n \delta_j$.

- mise à jour des poids et biais :

$$V_{jn} (new) = \Delta V_{jn} + V_{jn} (old)$$

$$W_{mj} (new) = \Delta W_{mj} + W_{mj} (old)$$

- Algorithme se répète a mise à jour les poids jusqu'à arriver au critère d'arrêt satisfait.

9. Problématique

Dans nos jour le nombre des accidents a été augmenté d'une façon prodigieuse, c'est pourquoi les chercheurs ont inventé la technologie de l'autonomie dans les voiture ou d'une autre façon la voiture autonome, en fait si tous les voitures seront autonome le nombre d'accident va chuterait car 90%, de ce phénomène est liée à l'être humain.

Grâce à la technologie de l'intelligence artificielle (et plus précisément l'apprentissage automatique et l'apprentissage profond) qui est embarqué dans la voiture, cette technologie permettre aux voitures de réagiraient plus rapidement et surtout dans le cas où il y'a un danger par exemple il n'est pas toujours facile de distinguer les panneaux de signalisation, l'obstacle etc. En plus si tout le parc automobile était remplacé par des voitures autonomes, le trafic dans les villes serait bien plus fluide. Il n'y aurait presque plus d'embouteillages plus de voitures garées n'importe où, celles-ci étant capables d'aller se garer elles-mêmes dans le parking le plus proche après vous avoir déposé.

En outre les systèmes de vision sont de plus en plus utilisés dans le contexte routier. Il existe des capteurs météorologiques pour faire la mesure physique, mais ils sont très coûteux. Et heureusement il existe des solutions pour être sûre de respecter le code de la route (détecter les obstacles, le danger, détection de piétons, mesure des écarts de trajectoire, gestion des feux de route) grâce aux caméras qui sont installé aux voitures, ou sur les téléphones. Ce système de vision est utilisé également pour scanner les panneaux de signalisations la fonction est nommée TSR Le capteur vidéo détecte les limitations de vitesse et les affiche sur le tableau de bord.

Il s'agit d'une aide utile pour les conducteurs qui ne sont pas nécessairement en mesure de voir le panneau dans la circulation environnante. Selon le système, la voiture peut même comparer sa vitesse à la limite effective et alerter le conducteur en cas d'excès de vites. Voir figure 19 ci-dessous pour connaitre l'importance d'utilisation de technologie de l'intelligence artificielle

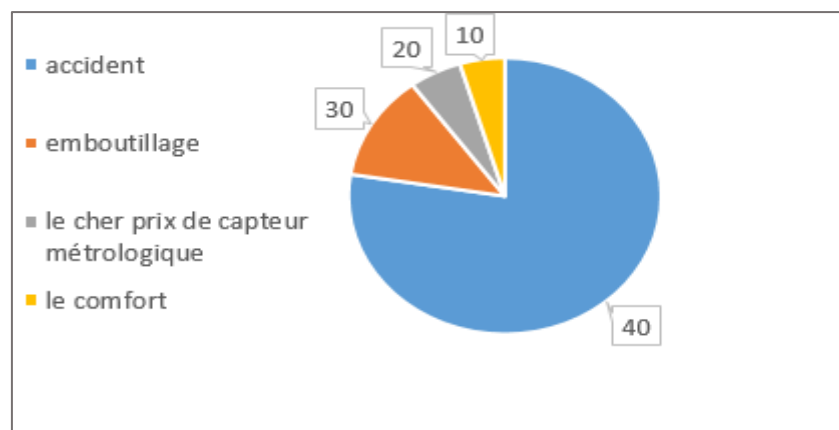


Figure 19 : Cercle statistique pour Les raisons de conception de ce robot

10. Conclusion

Dans ce chapitre en été consacré à la description de notre conception et la mise en œuvre de notre système de reconnaissance des panneaux routiers en utilisant l'apprentissage profond et spécifiquement les réseaux neurones convolutifs et pour cela en définit Quesque l'intelligence artificielle et ces domaines d'application après ça en définit l'apprentissage automatique et ces types comment l'apprentissage supervisé qui nous allons utiliser ultérieurement dans notre projet et l'apprentissage non et Semi supervise et l'apprentissage par renforcement, puis en discute sur l'apprentissage profond et ces domaines d'application, après en définissent ce quoi les réseaux neurones et ces types en concentraient sur le réseau de neurones convolutifs et son architecture et en finissent ce chapitre avec une phase d'apprentissage et une problématique.

Chapitre 2 : Conception du système intelligent pour la reconnaissance des panneaux routiers

1. Introduction

Dans ce chapitre, nous nous concentrerons sur notre conception et montrerons comment créer un classifieur et créer un modèle. Ensuite, nous évaluerons et créerons un modèle en utilisant une architecture CNN, que nous appliquerons ensuite sur nos données (dataset) « German_Traffic_signs ». Pour cela, nous utiliserons le langage de programmation python et des bibliothèques comme Tensorflow et Keras, et pour apprendre et classer et améliorer les performances du modèle, nous utiliserons quelques techniques simples et efficaces comme Dropout.

2. Modèle proposé pour la reconnaissance des panneaux routiers

Au cours de nos expérimentations, nous avons créé un modèle CNN où on a appliqué la base d'images « German Traffic Signs », dans ce qui suit on présentera l'architecture de modèle et le résultat obtenu. Le modèle est composé de quatre couches de convolution et deux couches de maxpooling et deux couches de fully connected. Avant d'entrer dans l'implémentation, nous allons vous expliquer comment concevoir notre programme d'après la figure 20.

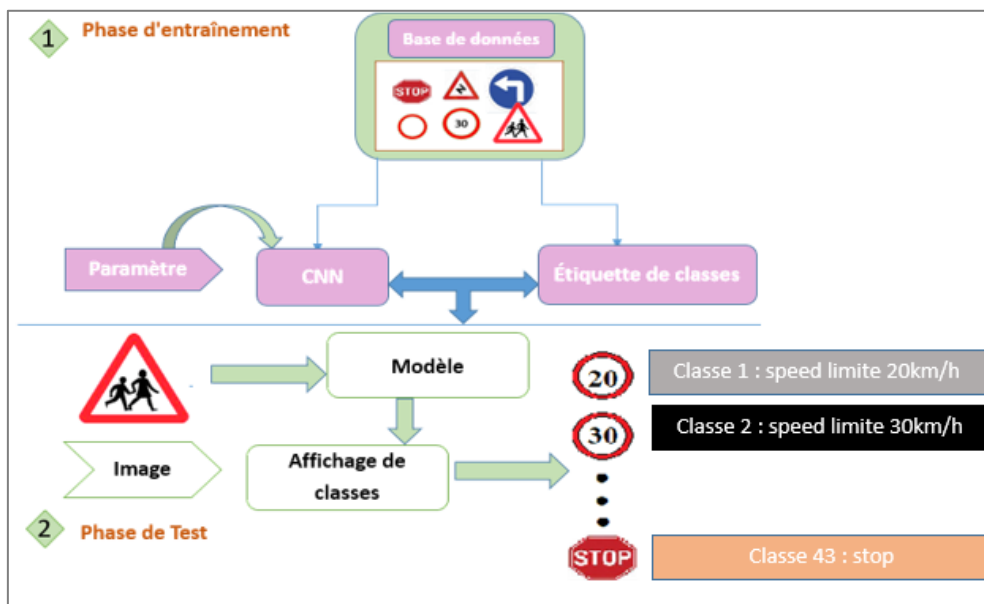


Figure 20 : Architecture globale d'un modèle CNN

On peut remarquer qu'on a 2 parties ou processus qui dans notre architecture :

2.1. Phase d'entraînement

C'est le processus le plus important car nous allons créer le modèle avec une configuration précise.

- ✓ La Dataset : c'est une base de données d'images divisé en classes. Par exemple si on prend notre dataset On remarque qu'il y'a 43 classes différents
- ✓ Labels classes : c'est un fichier.csv texte qui portera les noms des classes de notre dataset.
- ✓ CNN et paramètres : c'est notre algorithme de création d'un réseau neurones convolutifs qui sera configuré avec des paramètres, par exemple : nombre d'époques, nombre de filtre, nombre de couche ...etc.

Nous exécuterons l'ensemble de données sur notre algorithme CNN, qui sera configuré pour générer un modèle, que nous utiliserons ensuite pour les tests.

2.2. Phase de test

Dans le processus Test on retrouve :

- ✓ L'image : Il s'agit de l'entrée pour le test. Ce sont plusieurs images ou une seule.
- ✓ Le modèle : c'est le fichier généré après le training.
- ✓ L'affichage de la classification : son nom résume son travail, dans cette étape on va montrer et afficher la sortie du modèle qui est le nom d'une classe.

3. Architecture du modèle CNN proposé

Dans nos expériences, nous avons créé un modèle CNN dans lequel nous avons appliqué la base de données d'images "German_Traffic_signs". Nous présentons ci-dessous l'architecture du modèle et les résultats obtenus.

Le modèle se compose de quatre couches convolutifs, de deux couches de regroupement maximum (Max Pooling) et de deux couches entièrement connectées (Fully connected).

La taille de l'image d'entrée est de 30*30 et l'image passe d'abord dans la première couche convolutifs. Cette couche est constituée de 32 filtres de taille 5*5, chaque couche est suivie d'une fonction d'activation ReLU qui force le neurone à retourner une valeur positive, après cette convolution, 32 filtres de taille 26 seront créés.

Prenez les 26 cartes de caractéristiques précédemment obtenues comme entrée de la deuxième couche convolutif, qui se compose également de 32 filtres, appliquez la fonction d'activation RELU à la couche convolutif, Ensuite, appliquez Max Pooling pour réduire les paramètres de taille d'image et le calcul. En sortie de cette couche, nous aurons 32 feature maps de taille 11*11, pour éviter le sur

ajustement, nous utilisons la méthode Dropout. Nous répétons la même chose avec les couches convolutif trois et quatre, ces couches sont constituées de 64 filtres et la fonction d'activation ReLU est toujours appliquée à chaque convolution.

Une couche Maxpooling est appliquée après la quatrième couche convolutif. couches de convolution, nous utilisons un réseau de neurones composé de deux couches fully connected. La première couche comporte 256 neurones, la fonction d'activation utilisée est ReLU, et la deuxième couche est Softmax, ce qui permet de calculer la distribution de probabilité (nombre de classe dans la base d'image GTS), comme l'indique la Figure 21.

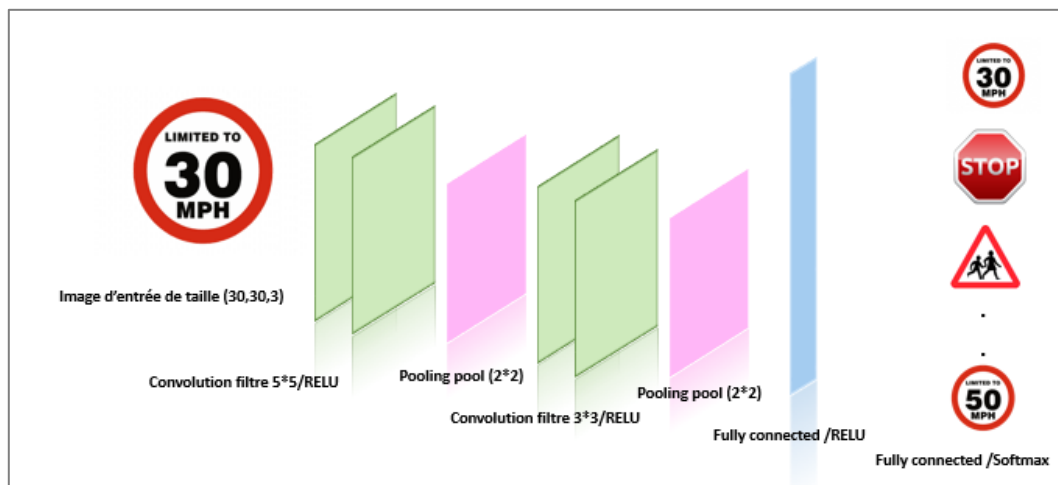


Figure 21 : Architecture de modèle CNN proposé

4. Base de données « German_Traffic_signs »

La base de données d'images est composée de plus de 50 000 images de divers panneaux de signalisation (limitation de vitesse, passage à niveau, feux de circulation, etc.). Environ 43 classes différentes sont présentes dans la base de données qu'ils varient en taille, car certaines classes ont peu d'images tandis que d'autres ont un grand nombre d'images, comme l'indique la figure 22.

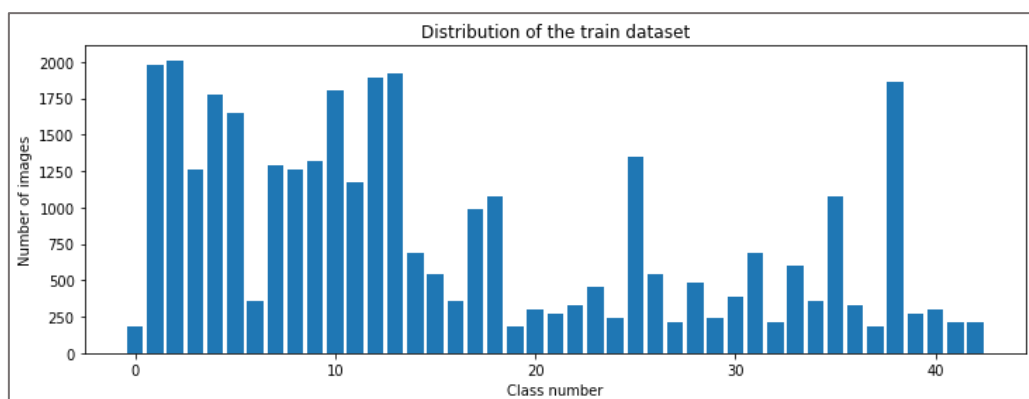


Figure 22 : Les nombres d'images dans chaque classe

Le Base de données ne prend pas beaucoup de temps et d'espace à télécharger car la taille du fichier est d'environ 314,36 Mo. Il contient trois dossiers distincts, Train, Meta, Test où le dossier train est composé de classes, et chaque catégorie contient diverses images. D'où 39209 images de taille 30x30x3 sont spécifiée pour le dossier de train et validation avec 27446 image d'entraînement ,11763 images de validation et le 30% restante pour le test, cette division est effectué par une méthode nommée split data.

Cette base de données est composé un fichier csv nommé Test.csv qui contient le chemin d'accès de chaque image et leurs classes (de 0 à 42), comme l'indique la Figure 23.

Test.csv

	A	B	C	D	E	F	G	H	I
1	Width	Height	Roi.X1	Roi.Y1	Roi.X2	Roi.Y2	ClassId	Path	
2	53	54	6	5	48	49	16	Test/00000.png	
3	42	45	5	5	36	40	1	Test/00001.png	
4	48	52	6	6	43	47	38	Test/00002.png	
5	27	29	5	5	22	24	33	Test/00003.png	
6	60	57	5	5	55	52	11	Test/00004.png	
7	52	56	5	5	47	51	38	Test/00005.png	
8	147	130	12	12	135	119	18	Test/00006.png	
9	32	33	5	5	26	28	12	Test/00007.png	

Figure 23 : Test.csv

On peut télécharger cette base de donn   d'apr  s le site kaggle sur le lien «[GTSRB - German Traffic Sign Recognition Benchmark | Kaggle](#)» [13]. Comme l'indique la Figure 24.

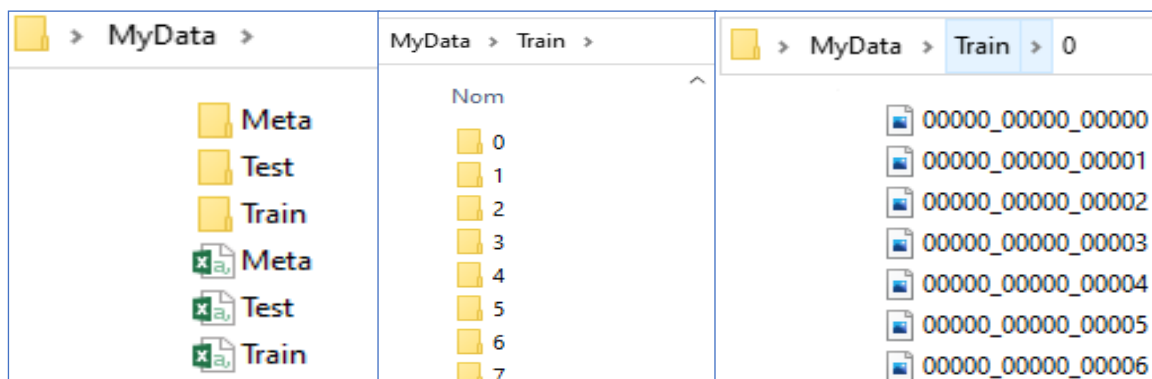


Figure 24 : Base de donn  es de German Traffic Signs

5. Phase d'apprentissage

5.1. Environnement utilisé

Tout d'abord pour implémenter dans la phase d'apprentissage nous utilisons quelques outils nécessaires (CPU, GPU, Libraires, etc...), le tableau 1 explique clairement l'environnement utilisé pour faire cette phase d'apprentissage.

Tableau 1 : Environnement utilisé pour la phase d'apprentissage

Configuration Utilisé dans l'implémentation	Logiciel	Libraires
Un PC portable ASUS AMD Ryzen 5 CPU 2.10 GHZ Carte graphique Nvidia GeForce Mx230	Python 3.9 Pycharm Google Collab (GPU)	Tensorflow Keras Scikit-Learn NumPy Matplotlib

5.2. Paramètre d'apprentissage

Pour faire l'apprentissage il est nécessaire d'utiliser quelques paramètres comme le montre le tableau 2 pour compiler et entraîner le modèle avant d'observer l'évaluation du modèle.

Tableau 2: paramètre nécessaire pour la phase d'apprentissage du modèle CNN proposé

Epoch	Algorithme d'optimisation	Batch_size	Loss	Metrics
15	Adam	32	Categorical_crossentropy	Accuracy

Explication de ces paramètres :

Epoch : Nombre d'époque égal au nombre de fois que l'algorithme voit l'ensemble de données complet. Ainsi, chaque fois que l'algorithme a vu tous les échantillons de l'ensemble de données, une époque s'est terminée.

Algorithme d'optimisation « Adam » : Adam est un processus important qui optimise les poids d'entrée en comparant la prédiction et la fonction de perte, l'optimisation d'Adam est une méthode de descente de gradient stochastique basée sur l'estimation adaptative des moments de premier et de second ordre.

Batch_size : La taille du lot définit le nombre d'échantillons qui seront propagés sur le réseau.

Loss : Loss est une fonction utilisée pour trouver une erreur ou une déviation dans le processus d'apprentissage, on utilise `Catégorical_crossentropy` car on a une classification multi classe.

Metrics : Les Metrics sont utilisées pour évaluer les performances de votre modèle, Il est similaire à la fonction de perte.

6. Metrics dévaluation

6.1. Confusion Matrix

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle comme l'indique la figure 25.

		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negative (TN)	False Positive (FP) Type I Error
	Positive +	False Negative (FN) Type II Error	True Positive (TP)

Figure 25 : Problème de classification binaire (matrice 2x2)

Les vrais positifs ou TP (True Positive) : indiquent les cas où les prédictions et les valeurs réelles sont effectivement positives.

Les vrais négatifs ou TN (True Négative) : indiquent par contre les cas où les prédictions et les valeurs réelles sont toutes les deux négatives.

Les faux positifs ou FP (False Positive) : indiquent quant à eux une prédiction positive contraire à la valeur réelle qui est négative.

Les faux négatifs ou FN (False Négative) : font référence aux cas où les prédictions sont négatives alors que les valeurs réelles sont positives.

6.2. Accuracy score

L'indicateur le plus simple est l'Accuracy : il indique le pourcentage de bonnes prédictions. C'est un très bon indicateur parce qu'il est très simple à comprendre

$$\text{Accuracy score} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

6.3. Précision

La précision est une bonne mesure pour déterminer quand les coûts des faux positifs sont élevés. Dans notre exemple un faux positif signifie qu'un étudiant appartient aux bases de données. Mais le modèle à le classer comme n'appartient pas aux bases de données, le système est précis si la précision n'est pas élevée.

$$\text{Précision} = \frac{TP}{TP + FP}$$

6.4. Recall score

Le rappel (ou Recall en anglais) quantifie le nombre de prédictions de classe positives faites à partir de tous les exemples positifs de la base de données. Cela nous aide à mesurer combien d'échantillons positifs ont été correctement classés par le modèle d'apprentissage automatique.

$$\text{Recall} = \frac{TP}{\text{Total actual positives}}$$

6.5. F1-Score

Le F1-score est une moyenne harmonique de la précision et du rappel. Il équivaut au double du produit de ces deux paramètres sur leur somme. Sa valeur est maximale lorsque le rappel et la précision sont équivalents. Dans certains modèles, on cherche à connaître la distribution des faux positifs et des faux négatifs. La métrique score F1 est alors utilisée pour évaluer la performance de l'algorithme. De même, il est particulièrement difficile de comparer deux modèles avec une faible précision et un rappel élevé. Le contraire est également vérifié. Dans ces conditions, le score F1 permet de mesurer ces deux paramètres simultanément.

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

7. Résultats obtenus et discussion

Après la configuration de notre modèle, nous intéressons aux résultats d'exécution de l'approche développée dans le tableau 3.

Tableau 3: Résultat d'apprentissage du notre modèle

Layer (type)	Output Shape	Paramètres
conv2d (Conv2D)	(None, 26, 26, 32)	2432
conv2d_1 (Conv2D)	(None, 22, 22, 32)	25632
max_pooling2d (MaxPooling2D)	(None, 11, 11, 32)	0
dropout (Dropout)	(None, 11, 11, 32)	0

conv2d_2 (Conv2D)	(None, 9, 9, 64)	18496
conv2d_3 (Conv2D)	(None, 7, 7, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 64)	0
dropout_1 (Dropout)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
Dense (Dense)	(None, 256)	147712
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 43)	11051

7.1. Accuracy et Loss pour Training et validation

Afin de montrer les résultats obtenus par le modèle, on illustre dans ce qui suit les résultats en termes de précision, perte, score de précision, score de rappel, f1 score et ainsi que les matrices de confusion par rapport au nombre d'époques (Une époque décrit le nombre de fois que l'algorithme passe sur l'ensemble de données).

Après l'analyse des résultats obtenus, On constate les remarques suivantes : La précision de l'apprentissage et de test augmente avec le nombre d'époque jusqu'à atteindre presque la valeur ou (100%) dans notre modèle on obtient une 'Accuracy=0.9945' , ceci reflète qu'à chaque époque le modèle apprend plus d'informations.

Si la précision est diminuée alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque et vice versa de même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époque jusqu'à atteindre presque la valeur ou (0%), pour notre modèle on obtient une valeur de 'Loss=0.0225', comme l'indique la figure 26.

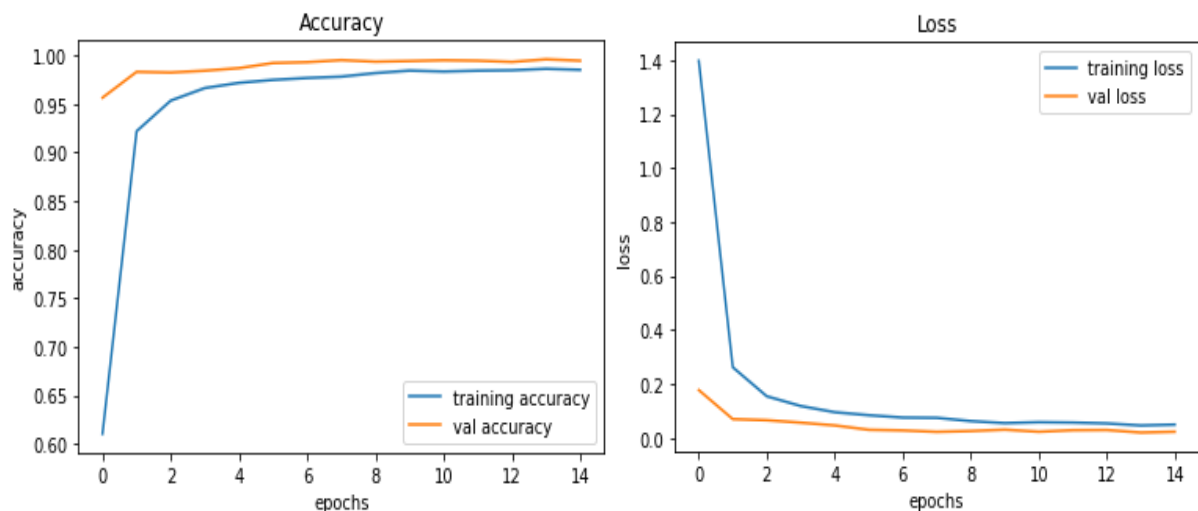


Figure 26 : Accuracy et taux d'erreur (Loss) pour le Modèle

7.2. Confusion Matrix pour le modèle CNN proposé

Une matrice de confusion est un résumé tabulaire du nombre de prédictions correctes et incorrectes faites par un classifieur. Il est utilisé pour mesurer les performances d'un modèle de classification, pour notre modèle nous obtenons les résultats dans la figure 27 ci-dessous.

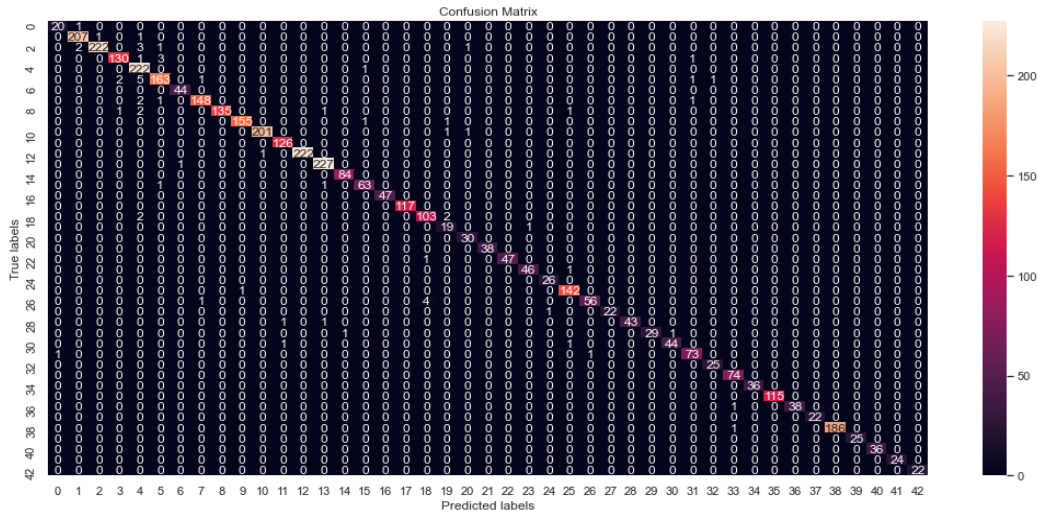


Figure 27 : Matrice de confusion de modèle CNN

7.3. Précision

C'est une mesure de l'exactitude qui est atteinte dans la vraie prédiction. En termes simples, il nous indique combien de prédictions sont réellement positives sur l'ensemble des prédictions positives totales, la Figure 28 exprime l'évaluation de la précision de 43 classes de notre modèle CNN proposé.

Tableau 4 : Résultat de Metrics de précision de quelques classes du modèle CNN proposé

Classe	Précision	Classe	Précision	Classe	Précision
1	0.38	4	0.86	7	0.98
2	0.82	5	0.99	8	0.98
3	0.97	6	0.68	9	0.96
Précision moyenne des classes (43)			0.67		

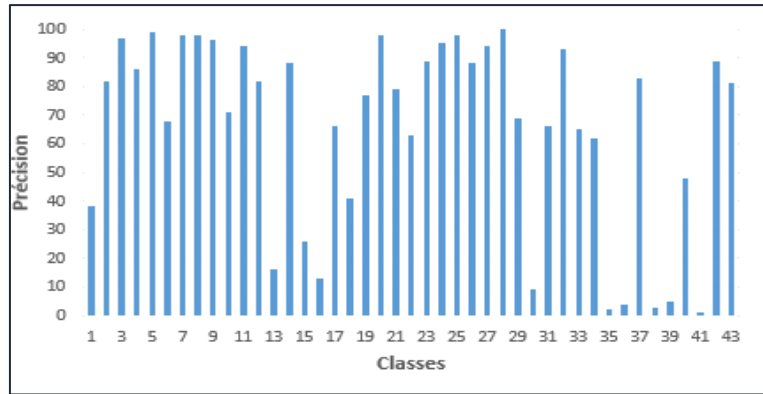


Figure 28 : Histogramme de précision pour 43 classes

7.4. Recall

C'est une mesure des observations réelles qui sont prédites correctement, c'est-à-dire combien d'observations de classe positive sont réellement prédites comme positives. On remarque que le Recall de notre modèle est très élevés pour la plupart de différents classes ce qui nous donne, un modèle excellent pour les classes TP (True positif), la Figure 29 suivant exprime l'évaluation de Recall score de 43 classes de notre modèle CNN proposé.

Tableau 5: Résultat de Metrics de Recall de quelques classes du modèle CNN proposé

Classe	Recall	Classe	Recall	Classe	Recall
1	0.98	4	0.84	7	0.68
2	0.85	5	0.79	8	0.70
3	0.92	6	0.89	9	0.68
Recall score moyenne des classes (43)			0.63		

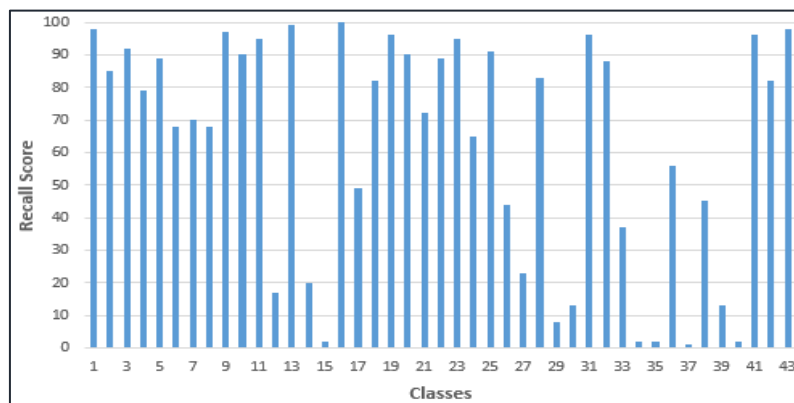


Figure 29 : Histogramme de Recall score pour 43 classes

7.5. F1-Score

Le F1-score maintient en quelque sorte un équilibre entre la précision et le rappel pour votre classificateur ce qui nous donne le résultat dans le tableau suivant pour des classes différents d'où on remarque que les valeurs d'évaluation de f1-score par rapport au nombres des classes est élevé comme il est indiqué dans la Figure 30 pour la plupart des classes ça veut dire que notre modèle est parfait.

Tableau 6 : Résultat de Metrics de F1-score de quelques classes du modèle CNN proposé

Classe	F1-Score	Classe	F1-Score	Classe	F1-Score
1	0.55	4	0.85	7	0.80
2	0.83	5	0.88	8	0.82
3	0.95	6	0.77	9	0.79
F1-score moyenne des classes (43)			0.64		

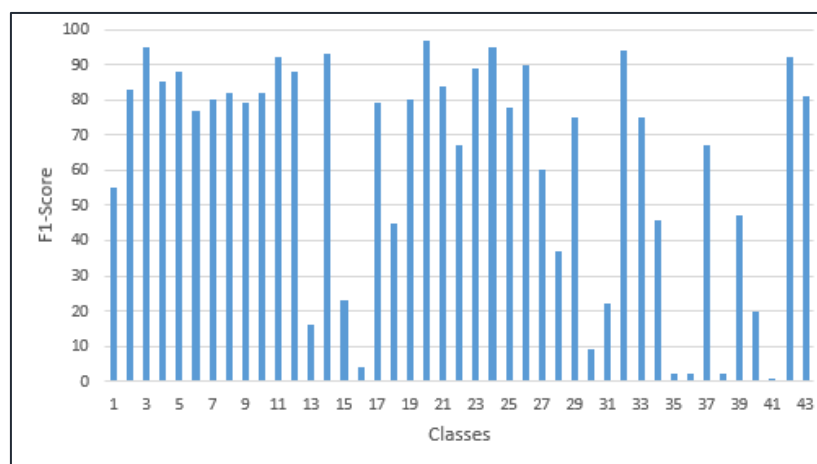


Figure 30 : Histogramme de F1-score pour 43 classes

8. Phase de Test

Finalement, pour vérifier la fonctionnalité de notre model CNN et ce précision en va le tester sur notre petite application qui est basé sur la bibliothèque Tkinter.

Tkinter est la bibliothèque la plus couramment utilisée pour développer une interface utilisateur graphique (Graphical User Interface) en Python. Il s'agit d'une interface Python standard pour la boîte à outils Tk GUI livrée avec Python. Graphical User Interface(GUI) est une forme d'interface utilisateur qui permet aux utilisateurs d'interagir avec les ordinateurs via des indicateurs visuels à l'aide d'éléments tels que des icônes, des menus, des fenêtres, etc. Il présente des avantages par rapport à

l' Command Line Interface(CLI) où les utilisateurs interagissent avec les ordinateurs en écrivant des commandes à l'aide du clavier uniquement et dont l'utilisation est plus difficile que l'interface graphique. Alors dans notre application en va porter une image qu'éloquente puis nous appuyons sur le bouton Classify Image et on obtient le résultat comme indique dans la Figure 31.

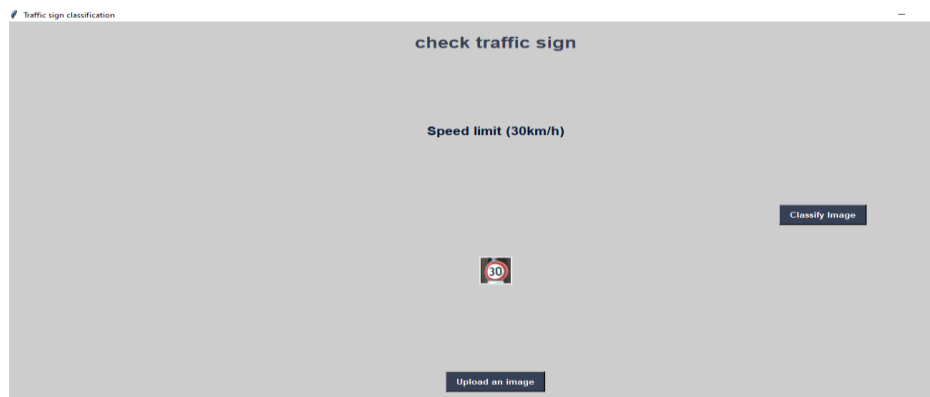


Figure 31 : Interface graphique pour la classification de notre model

Après en va tester en temps réel avec le camera webcam comme l'indique Figure 32.



Figure 32 : Tester en temps réel avec camera webcam

9. Conclusion

Dans ce chapitre, nous introduisons une méthode de classification basée sur un réseau de neurones convolutifs, pour cela nous utilisons un modèle avec une architecture CNN et certains paramètres qui nous permettent d'identifier les images de panneaux de signalisation, puis nous expliquons les résultats obtenus en termes de précision, et enfin nous utilisons Tkinter, une bibliothèque graphique gratuite en langage Python qui permet de créer des interfaces graphiques pour tester notre modèle. Dans le chapitre suivant, nous attribuons l'utilisation du modèle proposé à l'implémentation d'un robot utilisant un Raspberry pi.

Chapitre3 : Réalisation d'un Robot Intelligent

1. Introduction

En se basant sur l'étude le modèle CNN proposé faite dans le chapitre précédent, nous allons montrer dans ce chapitre les étapes d'implémentation du notre projet. Premièrement, nous avons justifié notre choix des cartes électroniques puis nous aborderons la présentation de l'environnement de travail ou la conception et le langage de programmation. En deuxième partie, nous allons montrer comment la communication entre l'ordinateur et la carte Raspberry Pi puis entre Raspberry Pi et l'Arduino est établie.

2. Choix du matériel

2.1. Raspberry Pi 3 (Model B)

La Raspberry Pi 3 est équipé d'un processeur SoC 64 bits Broadcom BCM2837 ARM Cortex A53 avec quatre cœurs à 1,2 GHz, comme l'indique la figure 33, pour une augmentation de la performance de l'ordre de 50% par rapport à la Pi2. La puissance de ce nouveau processeur permettra notamment à la Raspberry Pi 3 d'être utilisée pour des usages bureautiques ou de navigation web. L'innovation majeure apportée à cette troisième version est sans conteste l'ajout d'une puce Wifi et du Bluetooth basse consommation. Gain de place donc (plus besoin de brancher des d'ongles Wifi et Bluetooth), et plus de ports USB pour raccorder différents périphériques.

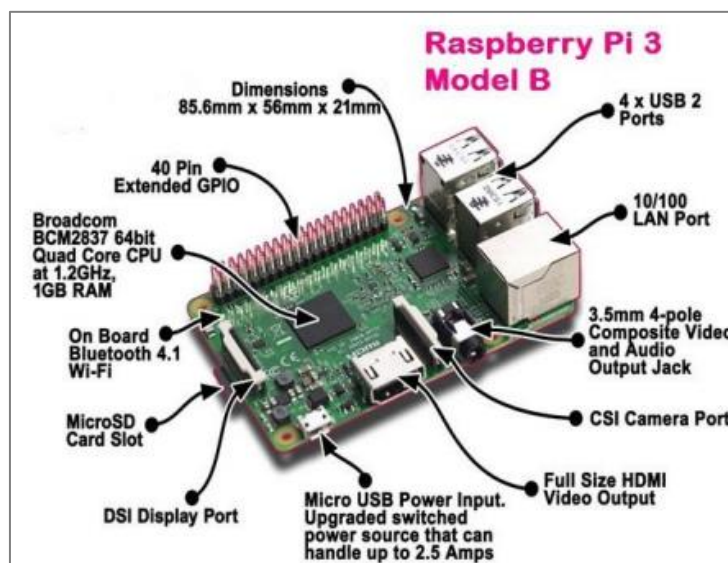


Figure 33 : Composition générale d'une Raspberry Pi 3

2.2. Arduino Uno

La carte arduino Uno est le choix préféré pour les débutants elle peu coûteuse et facile à utiliser pour ceux qui vont se démarrer la programmation arduino. Le Uno dispose de 14 entrées/sorties (E/S anglaises), il est capable de fournir jusqu'à 20mA (via des broches numériques) et 6 E/S analogiques, comme il est montré dans la Figure 34.

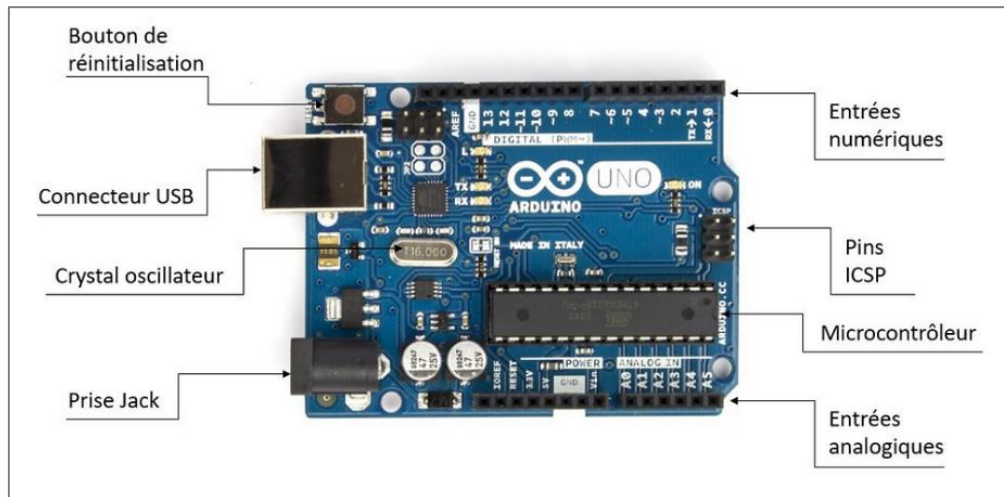


Figure 34 : Chemin d'un carte Arduino Uno et ses composants

2.3. Raspberry Pi Camera V1

La caméra Raspberry Pi comme indique la figure 35, a une résolution de 5 mégapixels et est entièrement compatible avec les Raspberry Pi 2B, 3, 3B, 3B+ et 4. Le pi caméra dispose d'une bibliothèque Python dédiée qui permet aux utilisateurs de contrôler l'appareil photo, de prendre des photos et d'enregistrer des vidéos. La bibliothèque permet également aux utilisateurs de contrôler la balance des blancs, l'obturateur et de lire chaque pixel RVB directement depuis l'appareil photo.

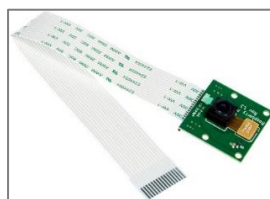


Figure 35 : Raspberry Pi Camera

2.4. Module L298

L298N est un composant couramment utilisé pour les prototypes et les projets amateurs, car il est facile à utiliser et à interfacier le L298 avec un Raspberry Pi ou un Arduino. Outre sa conception minimale,

il fournit également un régulateur 5V intégré que vous pouvez utiliser pour alimenter vos circuits 5V de manière très pratique, comme indique la figure 36 [14].

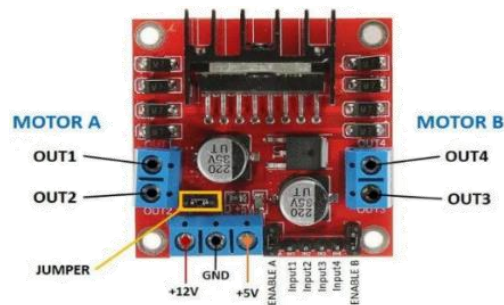


Figure 36 : Module L298

2.5. Module suiveur de ligne

Les capteurs suiveurs de ligne que nous avons utilisée sont des modules basés sur le réflecteur optique TCRT5000 et un amplificateur, comme indique la figure 37.

Alimentation: 5 Vcc

Sortie:

- état bas: ligne noire
- état haut: ligne blanche

Nous utiliserons 2 capteurs qui seront situés à l'avant de notre robot qui doit se déplacer en fonction des marquages (lignes noires) au sol et sera autonome.

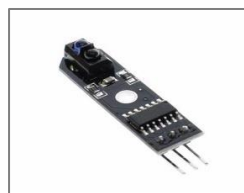


Figure 37 : Module suiveur de ligne

3. Premier démarrage de la carte Raspberry

3.1. Création carte SD

Pour le premier démarrage on ait obligé tout d'abord de créer notre carte SD pour la carte qui va contient l'os est-elle fonctionnée comme un disque dur, pour la faciliter de cette tache les constructeurs on poser une application « Raspberry pi imager ». À l'aide de ce dernier on installer la version d'os que nous souhaite utilisée facilement et plus rapide. Raspberry pi OS x64 ou nommé Raspbian x64 est

la version qu'on a utilisée dans pour notre système embarqué. Raspbian est un système d'exploitation libre et gratuit basé sur Debian optimisé pour fonctionner sur un Raspberry Pi. - Raspbian est un mot-valise formé par la fusion des mots « Raspberry Pi » et « Debian. », Comme indique la figure 38.



Figure 38 : Raspberry pi imager

3.2. Configuration et démarrage de la carte

Pour accéder et afficher la carte sur le pc via **VNC VIEWER** on doit passer par les étapes suivantes :
Au niveau de carte SD il faut d'abord attribuer une adresse IP fixe au Raspberry Pi pour faire ça on peut commencer par :

- ✓ Insérer la carte SD sur l'ordinateur
- ✓ Ouvrir le fichier texte **cmdline.txt**
- ✓ On ajoute à la dernière ligne la construction « ip=192.168.1.xxx »
- ✓ Appuyer su **ctrl+s** pour enregistrer la modification
- ✓ Créer le fichier **ssh**

Au niveau de la Raspberry :

- ✓ On va insérer la carte SD sur au Raspberry pi
- ✓ Lier la carte Raspberry avec l'ordinateur via un câble Ethernet
- ✓ En utilisant putty on va accéder à l'interface de commande de la carte via la commande suivante « sudo raspi-config »
- ✓ On va activer VNC VIEWER
- ✓ Enregistrer les modifications et quitter putty.
- ✓ Ouvrir l'application VNC VIEWER
- ✓ Enter l'adresse qu'on a fixée
- ✓ Enter le login et le mot de passe

Finalement on succès à afficher la carte sur notre ordinateur via VNC VIEWER comme afficher à la figure 39.

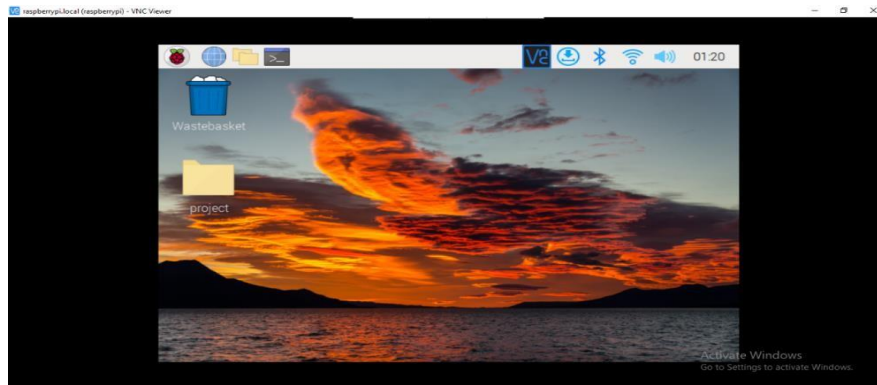


Figure 39 : Affichage du Raspberry pi au VNC VIEWER

4. Relation entre Raspberry et Arduino

Dans notre projets, il est intéressant d'établir une communication série entre Raspberry Pi et Arduino, comme indique la figure 40. C'est pourquoi La première idée qui nous vient en tête est l'utilisation de ce système dans la robotique dans lequel le Raspberry Pi va héberger l'interface de contrôle et l'intelligence et les Arduino vont servir d'automate programmable agissant sur les composants en bout de chaîne (lumière, ventilateur, capteurs, etc.) [15].

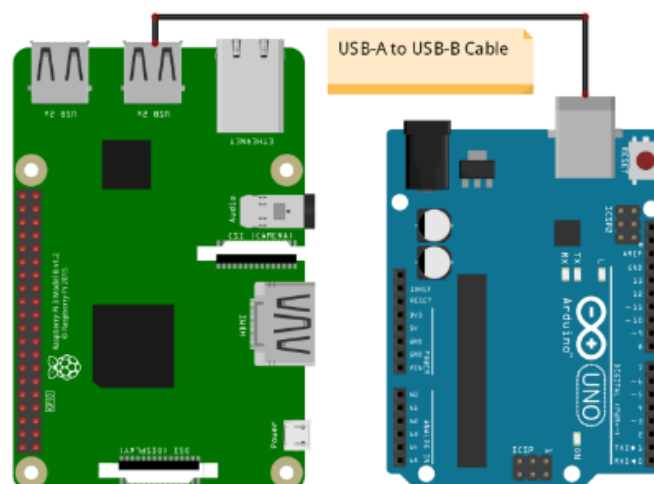


Figure 40 : Relation entre Raspberry Pi3 et Arduino Uno

5. Implémentation du modèle au carte et simulation réel du système

5.1. Conception du système

Le but de notre travail est d'avoir un robot intelligent capable de faire la détection des panneaux routiers de manière automatique en utilisant le modèle CNN proposé dans le chapitre 2, Selon le

matériel que nous avons utilisé, nous avons réalisé un schéma qui explique le fonctionnement global du système, qui est globalement composé de trois étapes, Tout d'abord le robot va observer les panneaux avec la caméra du raspberry pi ,puis il va faire la détection et la reconnaissance des panneaux routier grâce au modèle CNN , Et enfin la carte Arduino Uno va prendre une décision qui va commander les moteurs de robot pour faire l'action selon la classe détecté par le Raspberry Pi comme il est expliqué dans la figure 41.

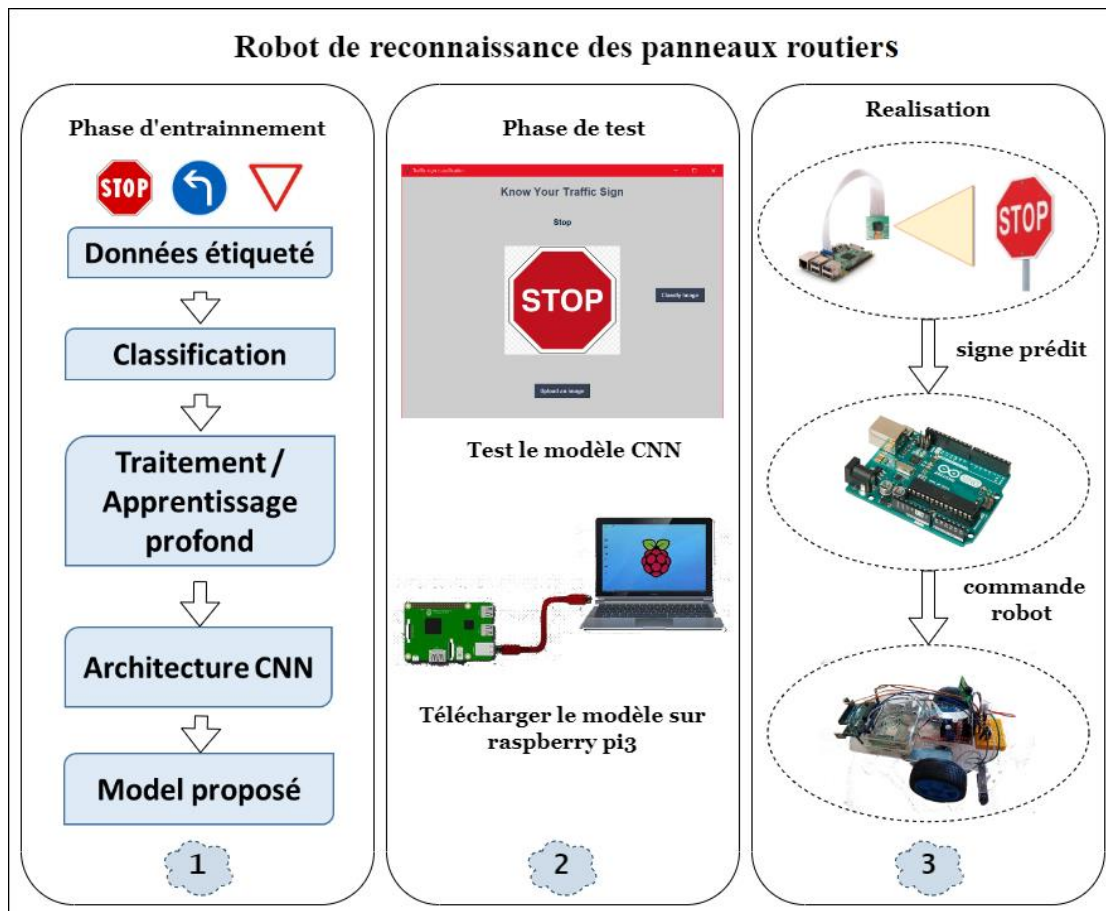


Figure 41 : Schéma de système

5.2. Implémentation du modèle sur la carte

Maintenant on va implémenter notre modèle sur la carte Raspberry Pi 3 et voir réellement le résultat obtenue dans la Figure 42.

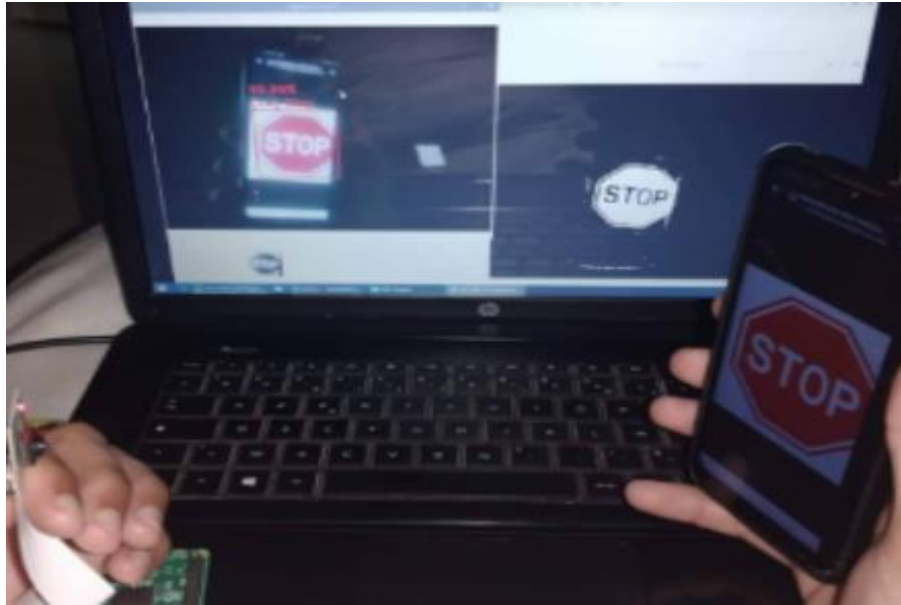


Figure 42 : Simulation de système sur Raspberry pi 3

6. Assemblage du Robot

6.1. Réalisation en ISIS

Pour contrôler des moteurs CC à l'aide d'une Arduino UNO, il faut :

- ✓ Arduino UNO
- ✓ Un double pont H - L298N.
- ✓ 2 Moteur CC 12 V.
- ✓ Des câbles.
- ✓ Une alimentation power Bank avec un courant 2.4 A

Une fois tous les éléments réunis, on peut passer au branchement de la carte Raspberry Pi avec le camera pi v1 pour la détection des panneaux 5. Et finalement nous avons fait la relation entre les deux cartes en utilisant un câble USB, comme indique la figure 43.

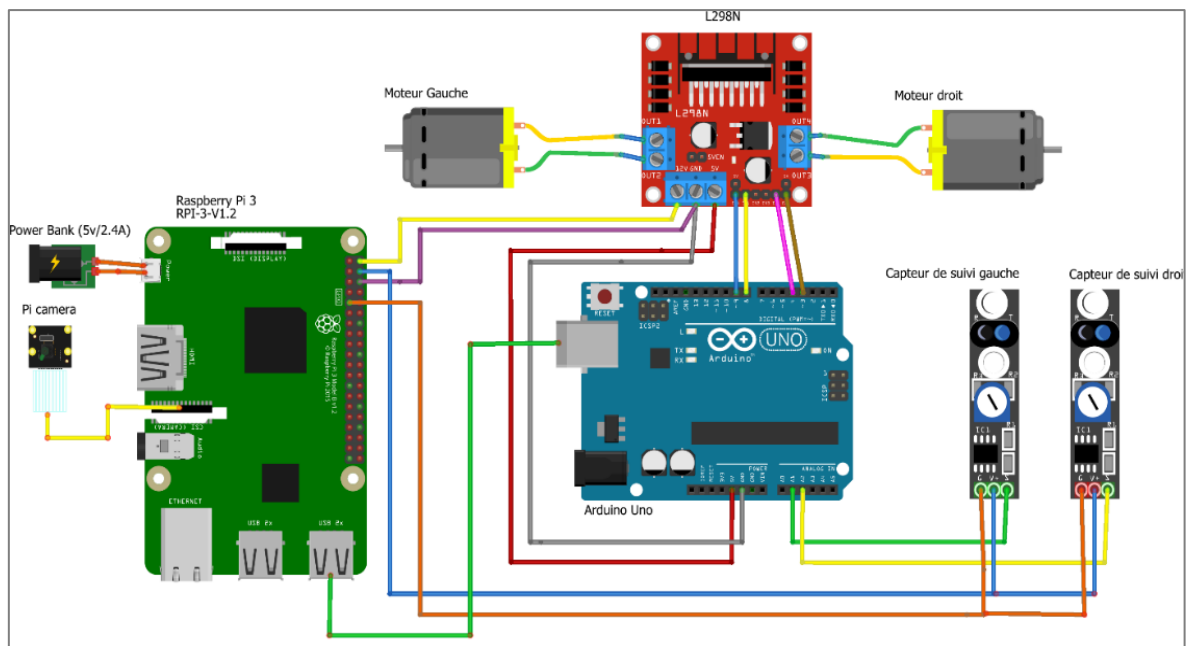


Figure 43 : Réalisation en ISIS

6.2. Réalisation réelle

Pour réaliser ou fabriquer un robot, il faut assembler tous les éléments qui le composent, Le processus d'assemblage ressemble étroitement à la construction d'un ensemble Lego complexe, elle nécessite beaucoup de coordination œil-main, comme indique la figure 44.

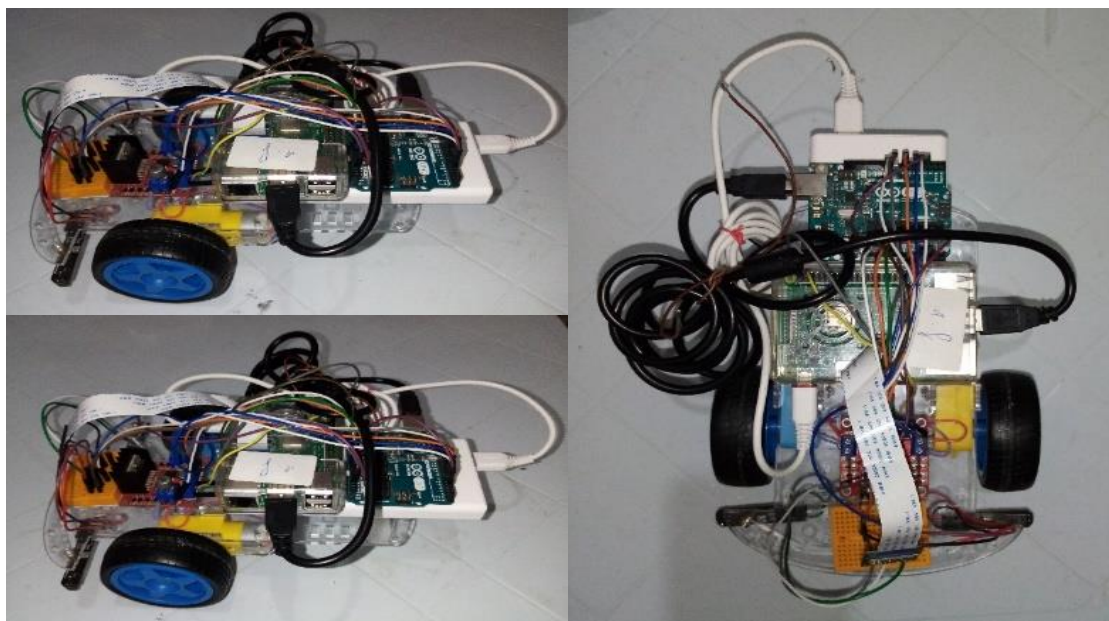


Figure 44 : Robot Intelligent

7. Conclusion

Dans ce chapitre, nous introduisons les matériels utilisés pour faire le câblage de notre robot, la configuration de la carte Raspberry Pi et l'implémentation du système ou de modèle CNN proposé, par la suite nous expliquons la relation effectuée entre la carte Arduino UNO et Raspberry Pi 3, Enfin nous constatons quelques problèmes lorsque le robot se met à traverser. Il y a un grand décalage de temps entre la vision par la caméra et la réponse des moteurs. Ce décalage est causé par la faible fréquence de CPU de la carte. Ce décalage provoque un problème de précision et le robot devient incapable de changer sa direction ou faire l'action rapidement lorsqu'il détecte les panneaux.

Conclusion Générale

Ce projet de fin d'études a pour objectif d'étudier et de réaliser un robot Autonome basé sur l'intelligence artificielle qui va faire la reconnaissance des panneaux routiers d'une façon intelligente, précise et rapide pour cela nous avons touché deux domaines qui s'accomplissent d'une manière ou d'autres qui sont indiqués dans le premier chapitre : l'intelligence artificielle qui se présente pour faire le calculs mathématiques et informatique complexe pour la reconnaissance des panneaux routiers et la robotique qui manifeste dans l'utilisation de la technologie Arduino , Raspberry et ses modules.

Tout d'abord on a commencé par parler à l'évolution de l'intelligence artificielle dont on a mentionné les différents types d'apprentissage tels que l'apprentissage automatique (supervisé, par renforcement, Semi-supervisé...).

Plus loin que ça on a parlé sur l'apprentissage profond dont on a mentionné les différents modèles utilisés. On a concentré sur le modèle CNN que l'on a utilisé pour réaliser notre système puis on a passé au deuxième chapitre où on a expliqué comment on a entraîné notre modèle et comment on a fait mesurer les performances du modèle après l'entraînement et on a changé chaque fois les paramètres jusqu'à obtenir le modèle idéal qui va faire les tâches que l'on va demander.

Ensuite, dès que le modèle est prêt à utiliser on a passé à la réalisation de robot dans le chapitre trois, au début on a précisé notre choix de matériel, puis on a implémenté notre modèle sur carte raspberry et on a réussi à réaliser le fonctionnement du modèle en temps réel, par suite nous avons fait le câblage de deux cartes électroniques, le raspberry Pi 3, l'arduino Uno et les autres différents composants, pour assurer la réalisation du robot.

Ce projet a été à la fois intéressant et formateur, nous avons pu concevoir et développer un robot autonome, mettre en application nos connaissances acquies durant nos études tout le long de nos années universitaires, malgré la qualité de matérielle qui nous tracasse, prenons par exemple la faible fréquence de CPU de la carte Raspberry Pi 3 qui nous a obligé d'utiliser une carte complémentaire (carte Arduino Uno), et la mauvaise version et qualité de résolution de notre pi caméra.

Bibliographies

- [1] Damien Trentesaux, R. R. (2020). Ethique des robots intelligents dans la société humaine. 1Université Polytechnique Hauts-de-France: HAL.
- [2] (2018, juin 01). Intelligence artificielle. Retrieved from <https://www.clubic.com/>: <https://www.clubic.com/technologies-d-avenir/intelligence-artificielle/article-835684-1-intelligence-artificielle.html?fbclid=IwAR2w5WRoWjxY33rgOKtPWaZXJGBRveKKi7V9-AavED40Y7V7YHGKim8LtF8>
- [3] LAROCHELLE, Hugo. Étude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes. 2009.
- [4] (2022). 3-types-of-machine-learning. Retrieved from <https://fr.blender3dtutorials.com/>: <https://fr.blender3dtutorials.com/3-types-of-machine-learning>
- [5] TRANSDUCTIF, Apprentissage et REVEL, Arnaud. Apprentissage Semi-Supervisé.
- [6] Deluzarche, C. (n.d.). /tech/definitions/intelligence-artificielle-deep-learning-17262/.
- [7] Retrieved from <https://www.futura-sciences.com/>: <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>
- [8] (1994-2022). Réseau de neurones. Retrieved from <https://fr.mathworks.com/>: <https://fr.mathworks.com/discovery/neural-network.html%5b7%5d.html>
- [9] article/Machine-learning-scikit-learn-tutoriel. (2019, 11 29). Retrieved from <https://www.kongakura.fr/>: <https://www.kongakura.fr/article/Machine-learning-scikit-learn-tutoriel>
- [10] G. B, «Convolutional neural network,» Datascientest, 25 Juin 2020. [En ligne]. Available: <https://datascientest.com/convolutional-neural-network#architecturecnn>.
- [11] https://fr.wikipedia.org/wiki/R%C3%A9tropropagation_du_gradient

[12] EL KHALIFI, Hicham, CHIADLI, Abdellatif, et BELHAJ, Rachid Saif. Elaboration d'un modèle d'aide à la prise de décision dans un projet de référentialisation basé sur l'intelligence artificielle pour la prédiction de compétences des personnes ressources.

[13] <https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>

[14] «components101,» 13 April 2021. [En ligne]. Available:
<https://components101.com/modules/1293n-motor-driver-module>

[15] <https://www.aranacorp.com/fr/communication-serie-entre-raspberry-pi-et-arduino/>