

UNIVERSITÉ KASDI MERBAH OUARGLA

Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Informatique et des Technologies de l'Information



Mémoire

Master Professionnel

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Administration et Sécurité des Réseaux

Présenté par :

✎ Mr MEDJOUDJA Abdelouahed

✎ Mr CHIBANE Mourad

Thème

**La Reconnaissance de la Langue des Signes Algérienne :
« La Mise en Place D'un Système De Traduction SIGNES/MOTS
En Temps Réel »**

Soutenu publiquement

Le: 24/06/2018

Devant le jury :

✎ Mme.CHAMA.Wafa	Président	UKM OUARGLA
✎ Mme.MARREF.Nadia	Examineur	UKM OUARGLA
✎ Mme.TOUMI Chahrazad	Rapporteur	UKM OUARGLA
✎ Mr.ZITOUNI Farouq	Co-Rapporteur	UKM OUARGLA

Année universitaire : 2017/2018.

Remerciements

Nous tenons tous d'abord à remercier le bon dieu tout puissant qui nous a donné la force, le courage, la patience, et l'espoir nécessaire pour accomplir ce modeste travail,

*Nous tenons à remercier aussi nos chers **parents** pour leur soutien et leurs encouragements durant toutes les années d'études,*

*On exprime nos profonds remerciements à notre encadreur, Mme **TOUMI Chahrazad** pour son aide, ses conseils et pour les orientations qu'elle nous a fourni durant tout le travail,*

*Un grand merci à notre Co-encadreur, Mr **ZITOUNI Farouq**, Nous voudrions également lui témoigner notre gratitude pour sa patience et son soutien qui nous a été précieux afin de mener notre travail à bon port,*

Nous tenons à remercier les membres de jury et tous les enseignants de département d'INFORMATIQUE,

Enfin, on remercie tous nos professeurs, amis et collègues et toute personne ayant participé de près ou de loin la réalisation de ce Modest travail.

***CHIBANE Mourad**
&
MEDJOUDDJA Abdelouahed*

Dédicaces

Je dédie ce modeste travail à :

Mes chers parents, pour leurs soutiens durant toutes mes études,

Mes amis KHERFI Abdelhak, Belhadj Nadir et tous mes collègues de travail,

Tous les enseignants de département d'INFORMATIQUE,

A tous mes collègues de la promotion 2017/2018.

MEDJOUDJA Abdelouahed

Dédicaces

Je dédie ce mémoire à :

A mes chers parents, pour tout leur sacrifice, amour, tendresse, soutien tout au long de mes années d'étude,

A mon petit frère et ma sœur et ma fiancée pour leurs encouragements permanents, et leur soutien moral,

A tous mes chères amis sans exception,

A mes collègues de travail,

A Mon encadreur et co-encadreur Mme.TOUMI et Mr. ZITOUNI qui doivent voir dans ce travail la fierté d'un savoir bien acquis,

A Tous mes professeurs de département Informatique pour leurs disponibilité et conseils,

Merci d'être toujours là pour moi.

CHIBANE Mourad

Table des matières

Table des matières	V
Liste des figures	VIII
Introduction Générale	1

Chapitre I : Concepts de Base en Reconnaissances Des Formes

I.1 Introduction	3
I.2 C'est quoi la RDF	3
I.3 Domaines d'application de la RDF	3
I.4 Chaîne de reconnaissance des formes	3
I.4.1 Le monde physique	4
I.4.2 Acquisition	5
I.4.3 Le prétraitement des données	5
I.4.4 Extraction des caractéristiques	5
I.4.5 L'apprentissage automatique	6
I.4.6 Principe et types de l'apprentissage	6
I.4.7 La décision	6
I.5 Principales approches pour la classification	7
I.5.1 Approches statistique	7
I.5.2 Approches structurelle	8
I.5.3 Approche connexionniste (Les Réseaux de Neurones)	8
I.6 Mesure de performances	9
I.7 Conclusion	10

Chapitre II : Apprentissage en Profondeur et TensorFlow

II.1 Introduction	11
II.2 L'apprentissage en profondeur	11
II.2.1 Quelques algorithmes de Deep Learning	11
II.2.2 Domaine d'application	12
II.2.3 L'architecture de l'apprentissage en profondeur	13
II.3 Tensorflow	13
II.4 Les composants de TensorFlow	14

II.5	L'architecture générale de TensorFlow	14
II.5.1	Client	15
II.5.2	Distributed Master	16
II.5.3	Worker service.....	18
II.5.4	Kernel implementations.....	19
II.6	Conclusion.....	19

Chapitre III : La Langue des Signes Algérienne LSA

III.1	Introduction	20
III.2	La langue des signes	20
III.3	La Langue des Signes Algérienne	21
III.3.1	Définition	21
III.3.2	Types des Signes	22
III.3.2.1	Les Gestes Impliquant Tout le Corps	22
III.3.2.2	Les Gestes de la Tête et du Visage	22
III.3.2.3	Les Gestes de la Main et du Bras.....	22
III.3.3	Structure de la langue des signes	22
III.3.3.1	La configuration.....	22
III.3.3.2	Le mouvement	23
III.3.3.3	L'orientation	23
III.3.3.4	L'emplacement	24
III.3.3.5	La mimique faciale	24
III.3.4	Alphabet de la Langue des Signes Algérienne (ALSA).....	25
III.4	Reconnaissance automatique des signes	26
III.4.1	Définition	26
III.4.2	Avantages de la reconnaissance automatique des signes	26
III.5	Reconnaissance des gestes	27
III.5.1	Reconnaissance des gestes corporels	27
III.5.2	Reconnaissance des gestes d'expression faciales	27
III.5.3	Reconnaissance des gestes manuels.....	28
III.6	Conclusion.....	28

Chapitre IV : Conception et Implémentation

IV.1	Introduction	29
------	--------------------	----

IV.2	Objectif et motivation.....	29
IV.3	Architecture générale de l'application.....	31
IV.3.1	Source (caméra)	32
IV.3.2	Prétraitement	32
IV.3.2.1	Image en niveau de gris	32
IV.3.2.2	Lissage (filtrage Gaussien)	32
IV.3.2.3	Segmentation par seuillage et Détection de contours	33
IV.3.2.4	Filtrage morphologique.....	33
IV.4	Bibliothèques et logiciels utilisé.....	35
IV.5	Tensorflow.....	36
IV.5.1	Les réseaux de neurones convolutifs	37
IV.5.2	L'Entraînement d'un CNN	38
IV.6	Le Modèle.....	42
IV.7	Optimisation	43
IV.8	Expérimentations	44
IV.9	Conclusion.....	44
	Conclusion Générale.....	45
	Bibliographie	46

Liste des figures

Figure I.1 : Schéma général d'un système de reconnaissance de formes [3].	4
Figure II.1 : L'architecture de l'apprentissage en profondeur [31].	13
Figure II.2 : L'architecture de système [26].	14
Figure II.3 : Un graphe construit par le client [26].	15
Figure II.4 : Sous-graphes optimisé par le maître distribué [26].	16
Figure II.5 : Regroupement des paramètres du modèle [26].	17
Figure II.6 : Transmission des informations entre les tâches distribuées [26].	17
Figure II.7 : L'envoi des éléments graphiques aux tâches distribuées [26].	18
Figure II.8 : La distribution des noyaux aux périphériques locaux [26].	19
Figure III.1 : Signe du chiffre « 5 » [3].	23
Figure III.2 : Signe de la lettre « و » [3].	23
Figure III.3 : Signe de la lettre « ب » [3].	24
Figure III.4 : Signe du mot « يأكل » [27].	24
Figure III.5 : Le signe du mot « جو حار » [27].	25
Figure III.6 : Alphabet de la Langue des Signes Algérienne (ASLA) [3].	25
Figure IV.1 : L'analyse de StatCounter sur les parts de marché des systèmes d'exploitation mobiles pour la navigation sur internet [16].	30
Figure IV.2 : Système de reconnaissance de la L.S.A.	31
Figure IV.3 : Représentation du contour de la main en vert.	33
Figure IV.4 : Représentation de l'enveloppe convexe du contour de la main en rouge.	33
Figure IV.5 : Image segmentée du signe « ي ».	34
Figure IV.6 : Image obtenue après filtrage morphologique du signe « ي ».	34
Figure IV.7 : Une capture d'écran de l'interface OpenCV Manager.	36
Figure IV.8 : La page d'accueil de Tensorflow[26].	36
Figure IV.9 : Architecture standard d'un CNN [23].	37
Figure IV.10 : Image du signe « ب ».	40
Figure IV.11 : Interface d'apprentissage de réseau de neurones.	41
Figure IV.12 : Le taux de reconnaissance finale.	41
Figure IV.13 Le taux de reconnaissance ou taux de succès (en anglais : accuracy).	42
Figure IV.14 : Visualisation de graph à l'aide de Tensorboard.	43
Figure IV.15 : Un exemple de traduction.	44

Introduction Générale

Comme nous le savons, la technologie basée sur la vision artificielle pour la reconnaissance des gestes de la main est une partie très importante de l'interaction homme-machine. Cependant, en raison du développement rapide au niveau matériel, et logiciel, la mise en place de nouvelles méthodes plus efficaces d'interaction homme-machine, devient une nécessité primordiale. En particulier, des technologies telles que la reconnaissance vocale, et la reconnaissance des gestes, reçoivent une très grande attention ces derniers temps. De nombreux chercheurs se sont efforcés d'améliorer la technologie de la reconnaissance des gestes de la main qui a une grande valeur dans de nombreuses applications telles que la reconnaissance de la langue des signes, la réalité augmentée (réalité virtuelle), l'interprétation gestuelle des personnes handicapées, et le contrôle des robots.

D'une manière assez intuitive, la langue des signes est une langue basée sur des gestes. En fait, il s'agit d'une langue à part entière au même titre que les langues parlées. Entre autres, il n'y a pas de langues des signes universelles. La langue des signes algérienne (LSA) est reconnue officiellement par la loi du 8 mai 2002 en tant que première langue de la communauté des sourds-muets en Algérie qui est le seul pays du monde arabe et d'Afrique à reconnaître officiellement la langue des signes. En Algérie la communauté des sourds-muets compte environ 80 000 personnes, qui éprouvent généralement des difficultés d'adaptation vis-à-vis la quasi-totalité des supports d'information, que ce soit dans le domaine éducatif ou dans les autres domaines de la vie courante, qui ne prennent pas en charge leurs spécificités [17]. Ainsi que, les limites de certains programmes de tradition qu'il ne se traduit pas en temps réel et qui sont uniquement disponible au niveau des ordinateurs bureautique.

D'une manière générale, notre but est de proposer un système qui aide cette tranche de la communauté (sourds-muets) à bien s'adapter et les facilite la communication avec d'autre personnes (non sourds-muets) même si ces derniers ne connaissent pas LSA. A cet effet, nous allons présenter à travers ce mémoire la conception et l'implémentation d'une application mobile pour la reconnaissance de la langue des signes algérienne. L'objectif d'un système de reconnaissance des signes d'une langue donnée est de classer chaque signe de la main vers un nombre fini de classes de signes prédéfinies, en utilisant un certain nombre de caractéristiques. Donc, notre système doit détecter l'existence ou non d'un objet main dans une scène donnée, et surtout interpréter sa posture.

Ce mémoire est organisé comme suit :

Dans le premier chapitre, Nous commençons tout d'abord par la présentation du domaine de reconnaissance des formes. Ainsi que ses champs d'application, Nous détaillons les étapes qui composent une chaîne de reconnaissance d'une forme donnée, c'est-à-dire de l'acquisition jusqu'à la décision. Ensuite, nous allons présenter le principe général des méthodes de reconnaissance des formes et ces Mesures de performance.

Dans le deuxième chapitre nous commençons par une introduction sur l'apprentissage en profondeur (Deep Learning). Ainsi que ses domaines d'application. Ensuite, Nous présentons la structure de la bibliothèque Tensorflow.

Dans le troisième chapitre, nous présentons un état de l'art sur la langue des signes. D'abord, nous commençons par une brève définition de la langue des signes, Ensuite nous présentons la structure de la langue des signes, les Types des signes, et l'alphabet de la langue des signes algérienne. Enfin, nous présentons la reconnaissance automatique des signes, et les différents types des systèmes de reconnaissance des gestes.

Dans le quatrième chapitre, Nous présentons le système de reconnaissance d'alphabet statique de la langue des signes algérienne en temps réel, et expliquer en détail, les principes de fonctionnement de notre approche utilisée pour la reconnaissance en temps réel des gestes de la main, Nous utilisons une méthode de classification Réseau de Neurones Convolutifs(CNN).

Enfin, dans la conclusion générale, nous faisons le bilan de notre travail. Puis nous proposons des perspectives pour améliorer notre système dans des futurs travaux.

Chapitre I

Concepts de Base en Reconnaissances Des Formes

1.1 Introduction

La reconnaissance des formes est à la fois une discipline historique de l'intelligence artificielle et un domaine de recherches extrêmement dynamique qui a subi de multiples influences de domaines scientifiques variés depuis une trentaine d'années. Elle est aujourd'hui partie intégrante de l'apprentissage automatique, domaine auquel elle a apporté de nombreuses idées et méthodes, en outre elle couvre un spectre large d'applications.

Dans notre chapitre, Nous commençons par une brève introduction sur le domaine de la reconnaissance des formes(RDF). Puis nous donnons quelques définitions de la RDF, ses domaines d'application, et par la suite, nous exposons le processus de fonctionnement d'un système de RDF en détaillant ses étapes. Enfin, nous clôturons par une conclusion introduisant au chapitre deux.

1.2 C'est quoi la RDF

La Reconnaissance Des Formes (en anglais pattern recognition) ou reconnaissance des motifs est un domaine de l'intelligence artificielle qui s'intéresse à la conception et à la réalisation des systèmes (matériels ou logiciels) capables de percevoir et, dans une certaine mesure, d'interpréter des signaux captés dans le monde physique.

1.3 Domaines d'application de la RDF

La RDF contribue dans des domaines d'application variés. On peut citer :
Traitement de signal, Reconnaissance de parole,

- La reconnaissance de caractères à partir d'un texte imprimé (ROC : reconnaissance optique de caractères),
- Détection de route et suivi de véhicules d'aide de l'interprétation des signaux et des images,
- Identification et localisation d'objets,
- L'interprétation et analyse des images satellite et aérien,
- La reconnaissance des empreintes digitales, visages et la reconnaissance des gestes en temps réel, etc.

1.4 Chaîne de reconnaissance des formes

Le processus de reconnaissance des formes est composé de plusieurs modules, on retrouve principalement cinq (5) étapes dans l'élaboration du processus complet de reconnaissance de formes [3], soit :

- ☞ Monde physique.
- ☞ Acquisition.
- ☞ Extraction de caractéristiques.
- ☞ Apprentissage.
- ☞ Décision.

Ces étapes sont visualisées sur la figure I.1

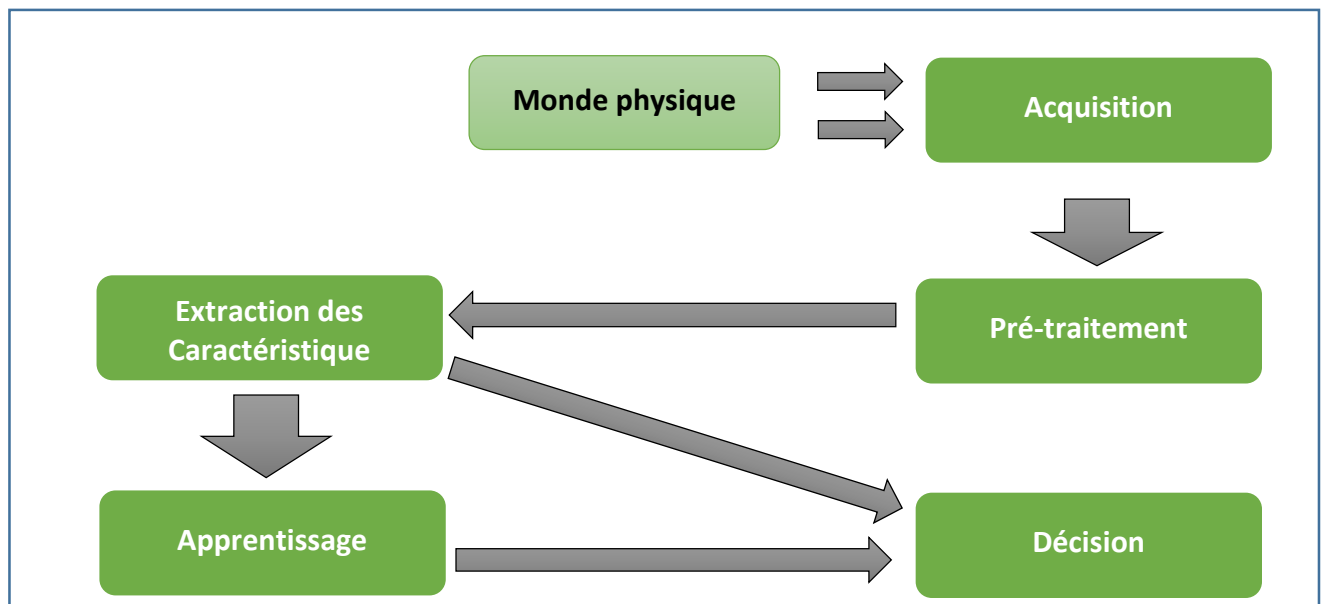


Figure I.1 : Schéma général d'un système de reconnaissance de formes [3].

1.4.1 Le monde physique

C'est celui qui est présenté dans sa forme la plus primaire, c'est-à-dire dont nous devons déterminer les caractéristiques les plus apparentes avant l'étape du codage. Le monde physique qui nous entoure est considéré comme un espace analogique de dimension n appelé l'espace des formes F . Le nombre de dimensions n est très grand pour cet espace. Il est particulièrement difficile de distinguer parmi toutes ces dimensions lesquelles représentent les particularités générales de l'objet d'étude. La loi de passage au monde discret nécessite alors une sélection et une simplification des paramètres qui sont nécessaires pour la classification de l'objet. Imaginons un instant que l'on veuille classer différentes espèces de champignons. Il apparaît clairement que les caractéristiques visuelles de longueur du pied, grosseur du pied, forme du chapeau, aplatissement du chapeau et couleur représentent des caractéristiques distinctives de

l'identification des champignons. Nous pourrions également prendre d'autres caractéristiques (et il en existe de nombreuses) qui permettraient une classification plus fine. De toutes ces caractéristiques, il appert qu'il est possible de se limiter à quelques-unes seulement pour une classification présentant peu de risques d'être inexacte.

1.4.2 Acquisition

L'acquisition des données est un processus par lequel un capteur (scanner, caméra, microphone, ...etc.) permet d'acquérir les objets du monde réel, de les enregistrer dans un support de données et de les convertir en données numériques pour pouvoir être utilisés par un traitement informatique. On passe alors du monde réel au monde des données ou des formes.

1.4.3 Le prétraitement des données

Un module de prétraitement permet de sélectionner l'information nécessaire à l'application. Cette sélection passe par la réduction des bruits des capteurs à cause des conditions d'acquisition telles que l'orientation, la position des capteurs et la distance entre les objets et les capteurs. Le prétraitement comporte une ou plusieurs des techniques suivantes : la normalisation des données, la suppression de la redondance, le filtrage et autres techniques. Le résultat de ce module est une nouvelle représentation des données à une dimension plus petite et plus utile.

1.4.4 Extraction des caractéristiques

Cette phase correspond au troisième module dans un système de RDF. Dans ce module, le système calcule les caractéristiques communes les plus utiles, qui identifient la forme prétraitée. Ces caractéristiques correspondent à des mesures de nature géométrique, topologique ou statistique. Le but de l'extraction des caractéristiques consiste à trouver les informations les plus discriminatoires dans les données enregistrées pour réduire la quantité de données, éviter l'explosion combinatoire et accélérer les calculs. Pour calculer les caractéristiques, il y a plusieurs indices visuels utilisés comme : la couleur, la texture, la forme, l'information spatiale et le mouvement dans la vidéo. Une condition très importante est que ces caractéristiques devraient être invariantes à la rotation, au changement d'échelle et à la translation. Les caractéristiques sont utilisées dans l'espace de représentation, pour déterminer l'appartenance d'une forme inconnue à l'une des k classes correspondantes. Elles peuvent être continues ou discrètes, et entachées de bruit.

L'ensemble des caractéristiques ou l'espace de représentation X est défini par une fonction f qui a une forme w de Ω (l'ensemble de toutes les formes possible des objets à analyser), associe sa représentation, tel que [8] :

$$f : \Omega \rightarrow X$$

1.4.5 L'apprentissage automatique

L'apprentissage ou modélisation est une étape importante dans le processus de reconnaissance de formes. Son rôle est d'éclairer la décision à l'aide de connaissances à priori sur les formes. À partir de critères spécifiques aux formes, l'apprentissage tente de définir des classes de décision ou d'appartenance. La dimension n de l'espace des noms correspond globalement au nombre de modèle ou de classes existantes.

1.4.6 Principe et types de l'apprentissage

Dans le cas de l'apprentissage, il s'agit en fait de fournir un ensemble de formes. 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 les formes de classes inconnues. Il y a plusieurs cas : on distingue au moins trois scénarios possibles en matière d'apprentissage, et des stratégies différentes pour les aborder selon le problème sous-jacent.

a) Apprentissage Supervisé :

On connaît les classes auxquelles appartiennent les données d'apprentissage.

b) Apprentissage Non Supervisé :

Le système ne connaît pas les classes auxquelles appartiennent les exemples mais on connaît le nombre de classes [3].

c) Apprentissage semi-supervisé :

Ce type d'apprentissage est une hybridation entre l'apprentissage supervisé et non supervisé. Il améliore les performances en combinant de l'ensemble de données étiquetées et l'ensemble de données non-étiquetés. Dans l'apprentissage semi-supervisé ne nécessite que un ensemble de données étiquettes réduit. [4].

1.4.7 La décision

La décision ou classement est l'étape de reconnaissance proprement dite. Son rôle est de classer la forme ciblée à partir de l'apprentissage réalisé. Les critères utilisés pour la décision sont habituellement les mêmes que ceux utilisés pour l'apprentissage.

1.5 Principales approches pour la classification

Le principe de RDF est de classer les nouvelles formes en utilisant un classifieur qui génère une fonction d'appartenance pour chaque classe.

- Caractériser les classes de forme : apprentissage.
- Assigner automatiquement des données à ces classes : reconnaissance.

D'un point de vue géométrique, on peut dire qu'un classificateur est une fonction qui divise l'espace des formes en plusieurs régions, chaque région contient les formes d'une certaine classe. La classe d'une forme inconnue dépend alors de la région dans laquelle elle se trouve.

Dans ce qui suit, nous rappelons les principales approches de classification.

1.5.1 Approches statistique

En reconnaissance de formes statistique, une forme est caractérisée par n attribut numérique extraits des données, et représentée par un vecteur d'un espace numérique de dimension n . La construction du classifieur consiste donc en partition de l'espace des attribut en différentes régions, chacune affectée à une classe.

Les approches statistiques bénéficient des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques fondées, telles que la théorie de la décision bayésienne, les méthodes de séparation linéaire, les méthodes de classification non supervisée. En reconnaissance, le problème revient à affecter une forme inconnue à l'une des classes obtenues pendant l'apprentissage.

Parmi les méthodes statistiques les plus couramment utilisées, nous décrivons les trois suivantes :

- **Classification bayésienne** : La théorie bayésienne de décision a été élaborée dans le cadre des statistiques mathématiques, son application à la reconnaissance des formes a été formalisée par CHOW en 1965. [5]
- **Méthode des K plus proches voisins (KPPV)** : Cette méthode consiste, étant donné un point $x \in R^n$ représentant la forme à reconnaître, à déterminer la classe de chacun des k points les plus proches de x parmi les formes d'apprentissage et à retenir pour la décision la classe la plus représentée. [6]

- **Discrimination fonctionnelle** : Ces méthodes sont basées sur la définition des fonctions permettant de séparer des classes représentées par les vecteurs de leurs échantillons. Ainsi, on peut distinguer deux catégories de fonctions :
 - a) Fonctions de discrimination linéaires.
 - b) Fonctions de discrimination linéaires par morceaux.

1.5.2 Approches structurelle

Si l'approche statistique permet de se placer dans un cadre mathématique fondé, elle présente cependant le défaut de ne pas tenir compte de la nature physique des formes et de leurs mesures : en particulier une métrique de type mathématique oblige à considérer les coordonnées caractérisant un point de façon indifférenciée. En utilisant de tel modèle, on ne peut pas exprimer certains types de contraintes auxquelles pourtant obéissent les formes étudiées [7]

En reconnaissance de formes structurelle, les formes sont représentées par des structures complexes composées de primitives simples et de relations topologiques entre les primitives.

1.5.3 Approche connexionniste (Les Réseaux de Neurones)

Les réseaux de neurones ont connu un essor important grâce l'algorithme de retro-propagation du gradient. Ce classifieur a trouvé application dans beaucoup de domaines tels que la reconnaissance de caractères, reconnaissances de visages...etc. La démarche connexionniste est très intéressante pour la classification car les réseaux neuronaux sont dotés de capacités d'apprentissage. Les algorithmes mis au point pour les entraîner permettent de modéliser des fonctions de discrimination très complexes à partir d'exemples. Même si chaque neurone effectue une fonction de discrimination très simple, les connexions entre neurones permettent de combiner les fonctions entre elles pour aboutir à des fonctions complexes difficiles à analyser et à spécifier explicitement. La capacité d'apprentissage, à partir des exemples, est importante surtout dans le cas où la constitution de recueils d'expertise pour un système expert devient difficile (reconnaissance intuitive ou implicite).

Il existe beaucoup types de réseaux de neurones, chaque type étant développé pour un objectif particulier :

- Réseau à un seul neurone (le neurone formel ou artificiel).
- Réseau de neurones multi-couches [28].
- Réseau de neurones convolutifs (CNN) [29].

1.6 Mesure de performances

L'évaluation est une composante essentielle de la RDF, il existe des mesures qui permettent de comparer les performances des systèmes entre eux, pour autant que les données soient les mêmes (d'où l'importance des bases de données publique), les hypothèses de travail soient les mêmes.

L'évolution ne doit pas porter sur les données d'apprentissage : il faut séparer les données d'apprentissage et les données de test pour mesurer l'efficacité d'un système de reconnaissance univoque on utilise :

- ❖ **Le taux de reconnaissance** : le nombre de décisions correctes divisé par nombre de décisions total.
- ❖ **Le taux de rejet** : le nombre de décisions non univoques divisé par le nombre de décisions total.
- ❖ **Le taux d'erreur** : le nombre de décisions univoques erronées divisé par le nombre de décisions total.

$$\text{Taux de reconnaissance} = 1 - \text{taux de rejet} - \text{taux d'erreur}$$

Il y a des cas où ne dispose pas d'assez de données pour estimer correctement la bonne détection. On utilise alors les mesures de précision et rappel [9]:

$$\text{Précision} = \frac{VP}{FP + VP} * 100$$

$$\text{Rappel} = \frac{VP}{FN + VP} * 100$$

VP : nombre de Vrais Positifs c-à-d le nombre d'éléments correctement détectés par le système.

FP : nombre de Faux Positifs, le nombre d'éléments détectés mais qui n'auraient pas dû l'être.

FN : nombre de Faux Négatifs, le nombre d'éléments non détectés mais qui n'auraient dû l'être.

La F-mesure est une mesure qui combine rappel et précision en une unique mesure.

La F-mesure est la moyenne géométrique entre le rappel et la précision :

$$F - \text{mesure} = \frac{2 * \text{Rappel} * \text{Précision}}{\text{Rappel} + \text{Précision}}$$

1.7 Conclusion

Nous avons présenté dans ce chapitre des généralités sur le domaine de la RDF, les techniques pouvant être utilisées, les problèmes à résoudre et les différentes étapes dans un système de RDF, à savoir, les différents méthodes de prétraitement, extraction des caractéristiques et d'apprentissage pouvant être utilisées, et aboutissant à l'étape ultime, qui es appelée "la décision" .

Dans le prochain chapitre, Nous présentons une introduction générale sur l'apprentissage en profondeur (Deep Learning) et la nouvelle bibliothèque TensorFlow.

Chapitre II

Apprentissage en Profondeur et TensorFlow

II.1 Introduction

L'apprentissage en profondeur (deep learning) est une technique d'apprentissage automatique (machine learning) qui a considérablement amélioré les résultats dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole et la traduction automatique. Les techniques d'apprentissage profond permettent, à l'aide de données, de résoudre de nombreux problèmes dans de nombreux domaines de l'économie tels que la santé, le transport, le commerce, la finance ainsi que l'énergie. [30].

Dans ce chapitre, Nous présentons une introduction sur l'apprentissage en profondeur (Deep Learning), ses domaines d'applications, son architecture générale, Ensuite nous présentons la bibliothèque Tensorflow, ses composants et par la suite, nous exposons leur processus de fonctionnement en détaillant ses étapes. Enfin, nous clôturons par une conclusion introduisant au chapitre trois.

II.2 L'apprentissage en profondeur

L'apprentissage en profondeur (en anglais : Deep Learning) est plutôt un nouveau domaine de recherche sur l'apprentissage automatique qui a été introduit avec l'objectif de rapprocher l'apprentissage automatique des objectifs de l'intelligence artificielle. Cela signifie que les algorithmes d'apprentissage en profondeur essaient de fonctionner comme le cerveau humain.

Cette famille d'algorithmes a permis de faire des progrès importants dans les domaines de la classification des images et du traitement du langage par exemple [31].

II.2.1 Quelques algorithmes de Deep Learning

Il existe différents algorithmes de Deep Learning. Nous pouvons ainsi citer [29]:

- **Les réseaux de neurones profonds** (Deep Neural Networks). Ces réseaux sont similaires aux réseaux MLP mais avec plus de couches cachées. L'augmentation du nombre de couches, permet à un réseau de neurones de détecter de légères variations du modèle d'apprentissage, favorisant le sur-apprentissage ou sur-ajustement (« *overfitting* »).
- **Les réseaux de neurones convolutionnels** (CNN ou Convolutional Neural Networks). Le problème est divisé en sous parties, et pour chaque partie, un «cluster» de neurones sera créer afin d'étudier cette portion spécifique. Par exemple, pour une image en couleur, il est possible de diviser l'image sur la largeur, la hauteur et la profondeur (les couleurs).
- **La machine de Boltzmann profonde** (Deep Belief Network): Ces algorithmes fonctionnent suivant une première phase non supervisée, suivi de l'entraînement

classique supervisé. Cette étape d'apprentissage non-supervisée, permet, en outre, de faciliter l'apprentissage supervisé.

II.2.2 Domaine d'application

L'apprentissage profond s'applique à divers secteurs des NTIC, notamment :

- la reconnaissance visuelle par exemple, D'un panneau de signalisation par un robot ou une voiture autonome et vocale) ;
- la robotique ;
- la bio-informatique, p. ex., pour l'étude de l'ADN et des segments non codants du génome ;
- la reconnaissance ou la comparaison de formes ;
- la sécurité ;
- la santé ;
- la pédagogie assistée par l'informatique ;
- l'intelligence artificielle en général.

L'apprentissage profond peut, par exemple, aider à :

- mieux reconnaître des objets hautement déformables ;
- analyser les émotions révélées par un visage photographié ou filmé ;
- analyser les mouvements et position des doigts d'une main, ce qui peut être utile pour traduire le langage des signes ;
- améliorer le positionnement automatique d'une caméra, etc. ;
- poser, dans certains cas, un diagnostic médical (ex. : reconnaissance automatique d'un cancer en imagerie médicale), ou de prospective ou de prédiction (ex. : prédiction des propriétés d'un sol filmé par un robot).

II.2.3 L'architecture de l'apprentissage en profondeur

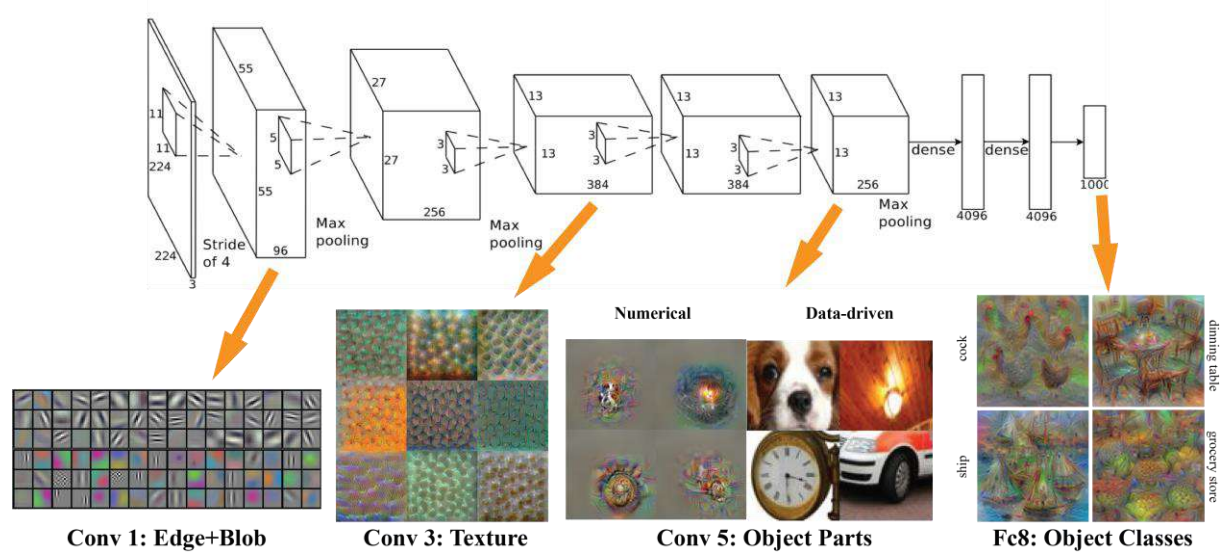


Figure II.1 : L'architecture de l'apprentissage en profondeur [31].

Il existe plusieurs bibliothèques de recherche de l'apprentissage en profondeur (Deep Learning) utilisables sous python afin d'utiliser des fonctions déjà faites et les adapter à notre problème. Dans notre cas nous avons choisi la bibliothèque **Tensorflow**.

II.3 Tensorflow

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le système d'apprentissage en profondeur et donc les réseaux de neurones. Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (en anglais : Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow : Gmail, Google Photos, Reconnaissance de voix [26].

Le fonctionnement interne de Tensorflow est la clé de son succès. Un calcul TensorFlow est décrit par un graphe, qui se compose d'un ensemble de nœuds. Le graphe représente un flux de données et ses transformations, avec des extensions permettant à certains types de nœuds de maintenir et de mettre à jour des états persistants. Les différents clients (Python, C++, autres) permettent de générer ce type de graphe de calcul.

II.4 Les composants de TensorFlow

- Le graphe de calcul, qui contient l'information de toutes les opérations qu'il est possible d'effectuer.
- Chaque nœuds correspond à une opération possédant zéro ou plusieurs entrées et de même pour la sortie.
- Une arête définit le sens suivant lequel circulent les tenseurs dans le graphe.
- Un tenseur est un tableau de dimensionnalité non définie mais de type connu ou inféré depuis le graphe.
- Une opération représente un calcul abstrait (par exemple, multiplication matricielle, ou addition). Elle possède un nom et des attributs éventuellement pour son exécution (exemple type des tenseurs).
- Un noyau est une implémentation particulière d'une opération exécutée sur un type particulier de périphérique (par exemple CPU ou GPU).
- Un binaire TensorFlow définit les opérations du graphe et leurs noyaux associés et disponibles.

II.5 L'architecture générale de TensorFlow

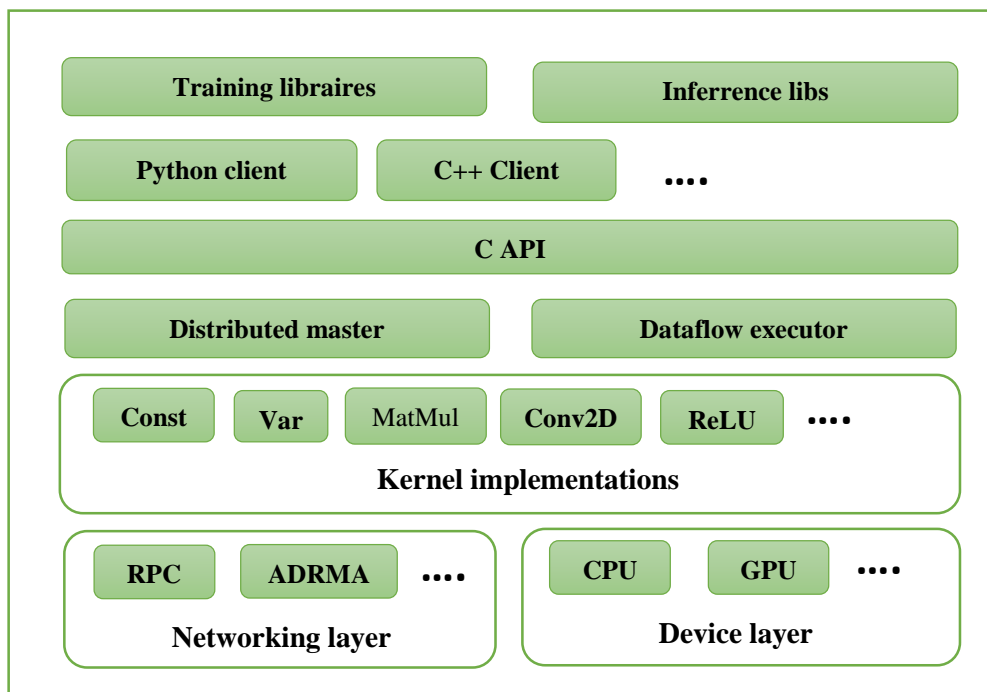


Figure II.2 : L'architecture de système [26].

II.5.1 Client

Les utilisateurs écrivent le programme client TensorFlow qui construit le graphique de calcul. Ce programme peut soit directement composer des opérations individuelles ou utiliser une bibliothèque de commodité comme « API Estimators » pour composer des couches de réseau neuronal et d'autres abstractions de plus haut niveau. TensorFlow prend en charge plusieurs langages client, et il a priorisé Python et C ++, car les utilisateurs internes sont les plus familiers avec ces langages. À mesure que les fonctionnalités deviennent plus établies, Tensorflow habituellement porté en C ++, de sorte que les utilisateurs peuvent accéder à une implémentation optimisée à partir de toutes les langues client. La plupart des bibliothèques d'apprentissage sont toujours en Python, mais C ++ prend en charge une inférence efficace.

Le client crée une session qui envoie la définition de graphe au maître distribué en tant que tampon de protocole « `tf.GraphDef` ». Lorsque le client évalue un ou plusieurs nœuds dans le graphique, l'évaluation déclenche un appel au maître réparti pour lancer le calcul.

Le client a construit un graphique qui applique des poids (w) à un vecteur de caractéristiques (x), ajoute un terme de biais (b) et enregistre le résultat dans une variable (s).

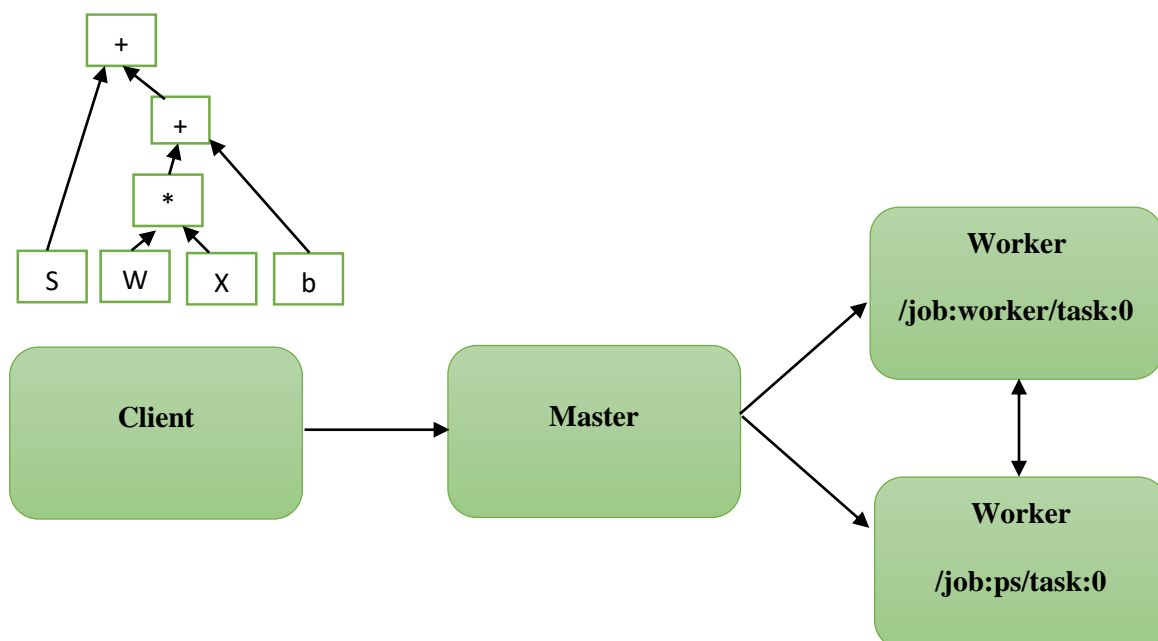


Figure II.3 : Un graphe construit par le client [26].

Notez que : `"/job: worker/task:0"` et `"/job: ps / task: 0"` sont les deux tâches avec les services de travail. "PS" signifie "serveur de paramètres" une tâche chargée de stocker et de mettre à jour les paramètres du modèle. L'autre tâche envoie des mises à jour à ces paramètres pendant qu'ils travaillent sur l'optimisation des paramètres.

II.5.2 Distributed Master

Le maître distribué :

- Elague le graphique pour obtenir le sous-graphe requis pour évaluer les nœuds demandés par le client,
- Partitionne le graphique pour obtenir des morceaux de graphique pour chaque périphérique participant, et
- Met ces pièces en cache afin qu'elles puissent être réutilisées dans les étapes suivantes.

Puisque le maître voit le calcul global pour une étape, il applique des optimisations standard telles que l'élimination de sous-expression commune et le repliement constant. Il coordonne ensuite l'exécution des sous-graphes optimisés sur un ensemble de tâches.

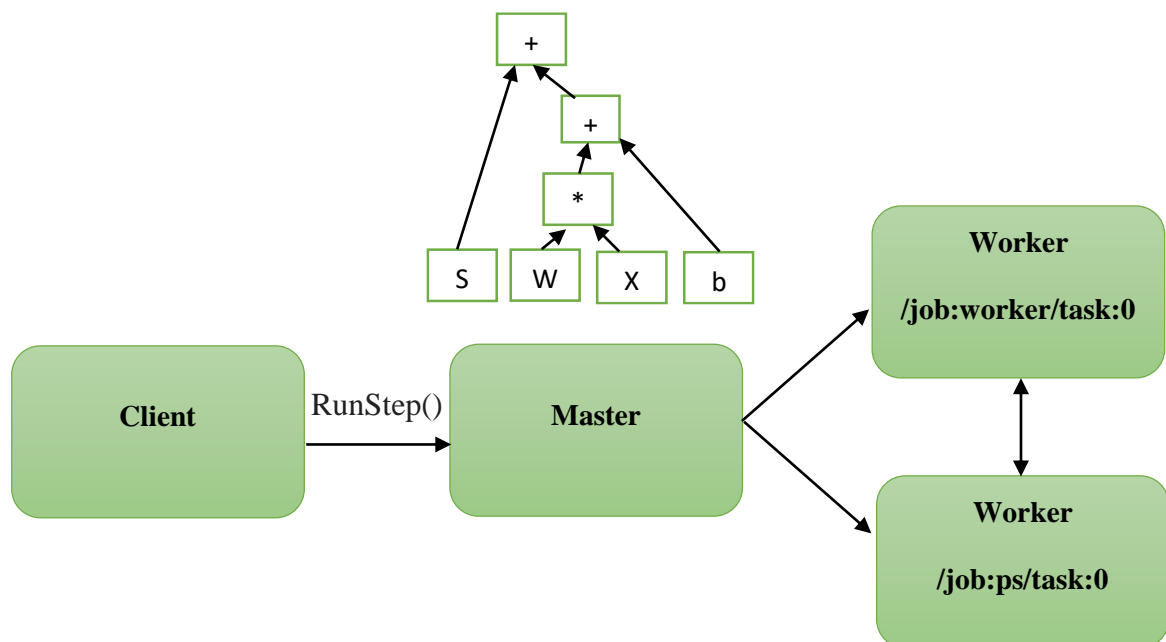


Figure II.4 : Sous-graphes optimisé par le maître distribué [26].

(La figure II.5) montre une partition possible de notre graphe d'exemple. Le maître distribué a regroupé les paramètres du modèle afin de les placer ensemble sur le serveur de paramètres.

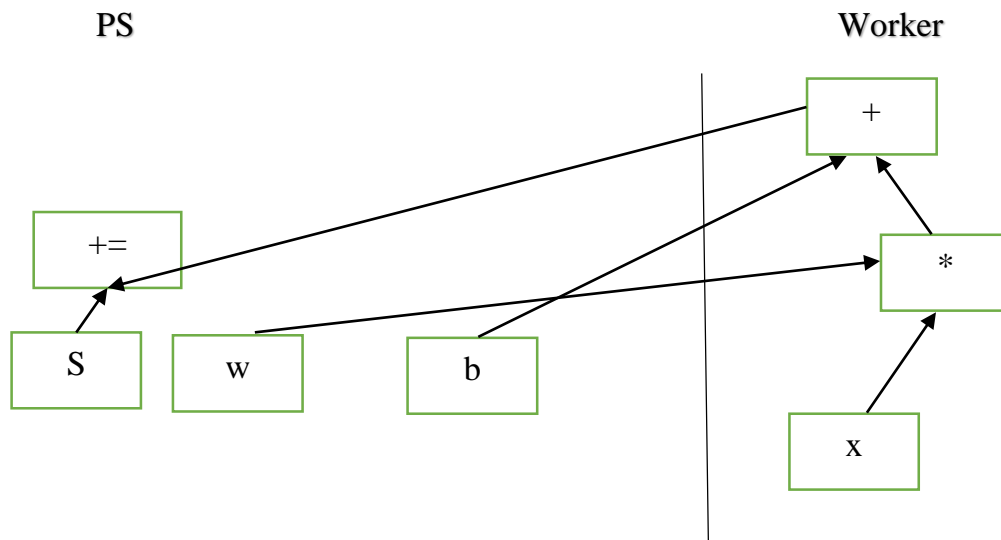


Figure II.5 : Regroupement des paramètres du modèle [26].

Lorsque les arêtes de graphique sont coupées par la partition, le maître distribué insère des nœuds d'envoi et de réception pour transmettre des informations entre les tâches distribuées (figure II.6).

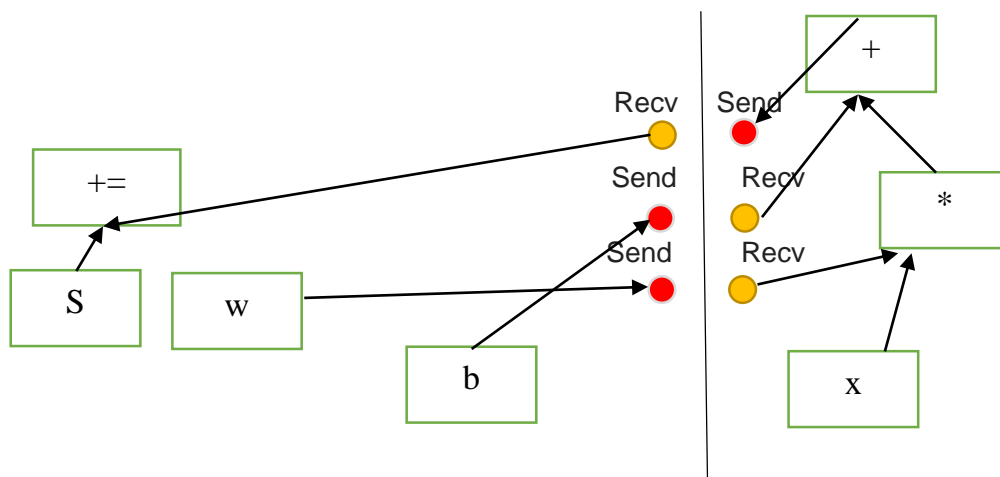


Figure II.6 : Transmission des informations entre les tâches distribuées [26].

Le maître distribué envoie ensuite les éléments graphiques aux tâches distribuées.

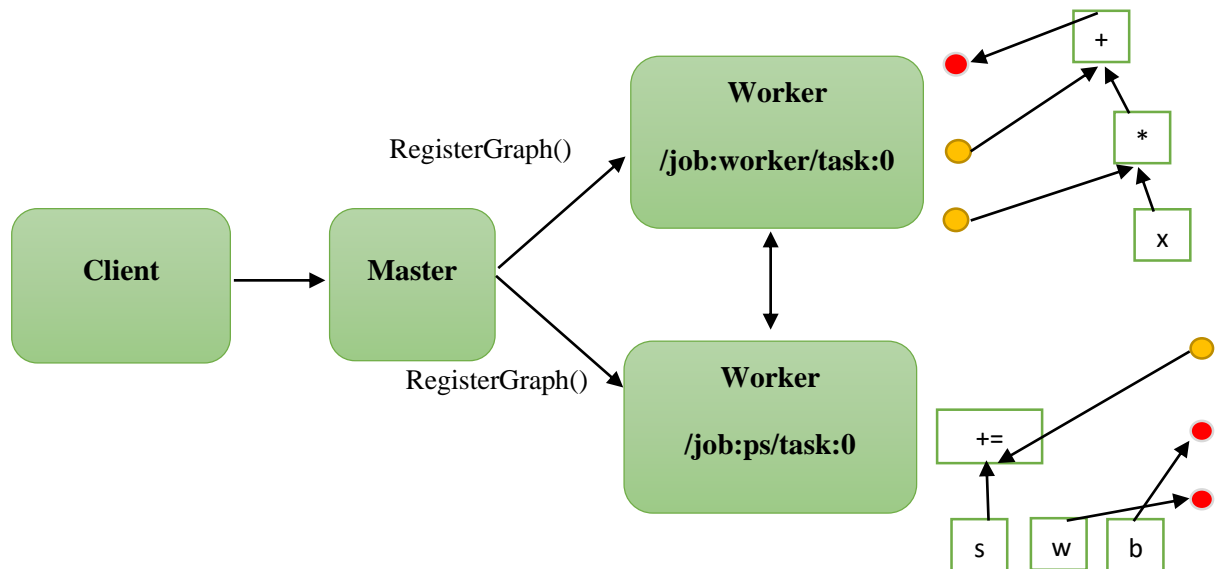


Figure II.7 : L'envoi des éléments graphiques aux tâches distribuées [26].

II.5.3 Worker service

Le service de travail dans chaque tâche :

- Traite les demandes du maître,
- Planifie l'exécution des noyaux pour les opérations qui comprennent un sous-graphe local, et
- Médiatise la communication directe entre les tâches.

Tensorflow optimise le service de travail pour l'exécution de grands graphiques avec un faible coût. L'implémentation actuelle peut exécuter des dizaines de milliers de sous-graphes par seconde, ce qui permet à un grand nombre de répliques d'effectuer des étapes d'apprentissage rapides et précises. Le service de travail distribue les noyaux aux périphériques locaux et l'exécute en parallèle lorsque cela est possible, par exemple en utilisant plusieurs cœurs de CPU ou flux de GPU.

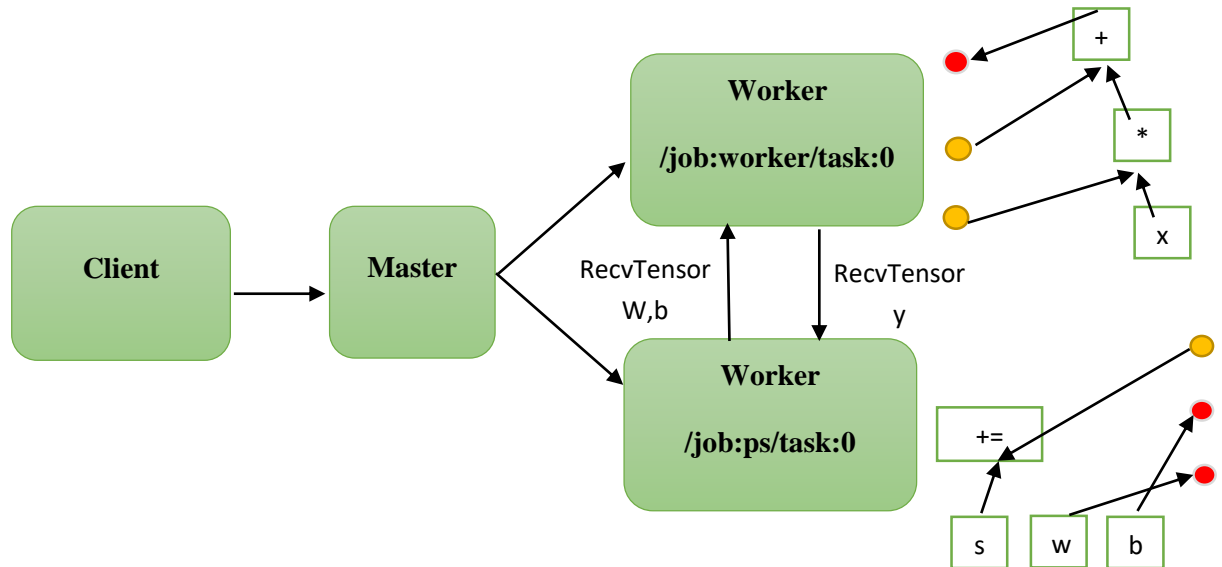


Figure II.8 : La distribution des noyaux aux périphériques locaux [26].

II.5.4 Kernel implementations

Le moteur d'exécution contient plus de 200 opérations standard, notamment des opérations mathématiques, de manipulation des tableaux, de flux de contrôle et de gestion d'état. Chacune de ces opérations peut avoir des implémentations de noyau optimisées pour une variété de périphériques. La plupart des noyaux d'opération sont implémentés en utilisant Eigen : Tensor, qui utilise des templates C++ pour générer un code parallèle efficace pour les processeurs multi-cœurs et les GPU ; Cependant, Tensorflow utilise des bibliothèques comme CuDNN où une implémentation plus efficace du noyau est possible. Et Aussi la quantification, qui permet une inférence plus rapide dans des environnements tels que les périphériques mobiles et les applications de centre de données à haut débit, et utilise la bibliothèque matricielle de précision basse `gemmlowp` pour accélérer le calcul quantifié.

II.6 Conclusion

Nous avons présenté dans ce chapitre une introduction générale sur l'apprentissage en profondeur (Deep Learning). Ainsi les domaines d'application et l'architecture. Ensuite, Les composants et la structure de la bibliothèque TensorFlow.

Dans le prochain chapitre, nous présentons un état de l'art sur la langue des signes algérienne, et les différents systèmes de reconnaissance des gestes.

Chapitre III

La Langue des Signes Algérienne LSA

III.1 Introduction

Les langues des signes (LS) sont des langues naturelles, visuelles, et non verbales. Par ailleurs, elles diffèrent des langues parlées. Toutefois, les deux langues, de signes et parlées, ont les mêmes fonctions. Les LS sont utilisées par des personnes sourdes, muettes, et malentendantes. Ce qui leur permet de communiquer par des gestes avec d'autres personnes.

Dans ce chapitre, nous débutons par une brève introduction sur les différentes langues des signes universels, leurs caractéristiques principales. Puis, nous donnons une définition de la langue des signes algérienne, ainsi que son alphabet. Ensuite, nous donnons une définition de la reconnaissance automatique des signes, et leurs avantages. Enfin, la dernière partie de ce chapitre est consacrée à la présentation de quelques systèmes de reconnaissance des gestes humains, les plus utilisés. En conclusion, nous introduisons le quatrième chapitre.

III.2 La langue des signes

La langue des signes est actuellement pratiquée par plusieurs centaines de milliers de personnes sourdes. Par ailleurs, les LS sont des langues gestuelles, et qui dépendent essentiellement de l'utilisation des deux mains pour exprimer des idées, et des propos. De ce fait, chaque pays ou région possède sa propre langue des signes. Donc, la langue des signes n'est pas forcément une langue universelle, et chacune d'elles possède ses propres de signes. En réalité, il existe environ 130 langues des signes différentes dans le monde entier [10]. A titre d'exemple nous pouvons citer :

- La langue des signes australienne "AUSLAN" (en anglais : Australian Sign Language), qui a été inventée dans les années 80, par un certain Trevor Johnston. Bien que la langue soit elle-même plus ancienne. Toutefois, celle-ci n'a été reconnue officiellement qu'en 1987, et en 1991 par le gouvernement de l'Australie.
- La langue des signes brésilienne (LIBRAS) est reconnue officiellement en 2002, dans le domaine de l'éducation. En fait, au Brésil, il est statué que chaque enfant sourd possède le droit absolu d'apprendre sa propre langue, et d'avoir le portugais comme 2ème langue.
- La langue des signes belge francophone est reconnue officiellement le 21 octobre 2003 par le parlement de la communauté française de Belgique.

- La langue des signes française « LSF » est reconnue à part entière dans le domaine de l'enseignement sous la loi du 11 février 2005.
- La langue des signes américaine (en anglais : American Sign Language, AmeSLan ou ASL) est la langue des signes principale aux États-Unis, et reconnue dans plusieurs états en tant que langue étrangère.
- La langue des signes britannique (en anglais : British Sign Language, BSL), est utilisée par une partie des personnes sourdes, et leurs proches au Royaume-Uni. En fait, la BSL est reconnue officiellement comme une langue à part entière par la loi au Royaume-Uni depuis le mois de février **2003**.
- En Ouganda, la langue des signes d'Ouganda est reconnue dans la constitution.- En Thaïlande, la langue des signes thaïlandaise est reconnue le 17 août 1999.
- Au Canada, la province du Manitoba est la première à reconnaître officiellement la langue des signes américaine comme celle des communautés sourdes en milieu anglophone (1988), suivi de l'Alberta qui reconnaît l'ASL comme langue optionnelle dans l'enseignement (1990); l'Ontario reconnaît l'ASL et la LSQ comme langues d'enseignement (1993).

III.3 La Langue des Signes Algérienne

III.3.1 Définition

La langue des signes algérienne (LSA) est la langue gestuelle utilisée par les personnes sourdes et leurs proches en Algérie pour pouvoir se comprendre par le biais de signes. La langue des signes algérienne est reconnue officiellement par la loi du 8 mai 2002 en tant que première langue de la communauté des sourds-muets en Algérie qui est le seul pays du monde arabe et d'Afrique à reconnaître officiellement la langue des signes [11].

La LSA est entièrement basée sur des gestes (signes), chaque signe est réalisé au moyen de différentes parties du corps, la ou les main(s), le visage, l'épaule,... ou par le corps entier. Intuitivement parlant, la LSA est une langue comme toutes les autres langues. En effet, elle dispose d'un vocabulaire, et d'une syntaxe organisée tout comme les langues parlées. Donc, pour apprendre, et comprendre la LSA, alors tout simplement il faut connaître son alphabet. En fait, tout signe de l'alphabet de la LSA est généré par une ou deux posture(s) particulière(s) de la main [3].

III.3.2 Types des Signes

Un signe est une classe de gestes qui dépend ou non d'une certaine durée dans le temps [3]. On peut tout d'abord classer les gestes en fonction des parties du corps impliquées. On distingue généralement trois types de gestes [12] :

III.3.2.1 Les Gestes Impliquant Tout le Corps

Les recherches dans ce domaine s'intéressent à tout le corps en interaction avec son environnement (analyse des gestes d'un athlète pour améliorer ces performances).

On différencie également les gestes dynamiques et les gestes statiques. Un geste statique, également appelé posture, concerne la configuration du corps ou d'une partie du corps à un moment fixe dans le temps alors que le geste dynamique désigne une succession continue de postures.

III.3.2.2 Les Gestes de la Tête et du Visage

Peu de gestes de la tête ont une signification spécifique ; l'orientation de la tête est quant à elle très utile pour la détection du champ de vision.

III.3.2.3 Les Gestes de la Main et du Bras

Ils forment la principale Catégorie de gestes interactifs. La main permet de réaliser des gestes précis et complexes. Les recherches autour de ces gestes concernent principalement la reconnaissance de positions de la main, l'interprétation du langage des signes et permettant la manipulation et l'interaction avec des données ou des éléments d'un environnement.

III.3 Structure de la langue des signes

Chaque geste d'une main peut être décomposé en cinq paramètres qui sont indépendants et peuvent être aussi bien dynamiques qu'invariants durant l'émission du signe. Ces paramètres sont définis comme suite [13] :

III.3.3.1 La configuration

Une configuration est la forme prise par une main à un instant donné, et elle est définie par les postures des doigts et de la paume. En réalité, la forme d'une main varie d'une personne à une autre. Donc, elle n'est jamais parfaitement stable. La figure ci-dessous, montre un exemple de configuration dans la langue des signes algérienne (une main avec les cinq doigts écartés).



Figure III.1 : Signe du chiffre « 5 » [3].

III.3.3.2 Le mouvement

Le mouvement est la trajectoire (une ligne, un arc, un cercle, etc.) réalisée soit par une seule main (simple), soit par les deux mains qui se rapprochent (complexes). En revanche, les mouvements de la tête peuvent aussi entrer en jeu. En effet, les différents éléments qui peuvent intervenir durant un mouvement sont, le trajet (ligne, arc, etc.), la direction (vers le haut, vers le bas, etc.), et la vitesse (très vite, moyenne, lente, etc.). Parfois, ces mouvements sont répétitifs ou plutôt périodiques. À titre d'exemple, le signe « ٥ » dans la langue des signes algérienne peut être défini par deux paramètres, à savoir : la configuration et le mouvement (l'exemple ci-dessous montre une main en mouvement, qui désigne une trajectoire sous la forme d'un arc, et qui représente le paramètre mouvement du signe « ٥ » de l'alphabet algérienne).



Figure III.2 : Signe de la lettre « ٥ » [3].

III.3.3.3 L'orientation

Les directions vers lesquelles sont dirigés les mains et les doigts, [14] (Figure III.3). L'orientation désigne le sens de la main par rapport au signeur. De ce fait, la direction de la main est déterminée par la position de la paume de la main par rapport au signeur [3].



Figure III.3 : Signe de la lettre « ب » [3].

III.3.3.4 L'emplacement

L'emplacement désigne l'endroit de la main dans l'espace par rapport au corps du signeur, ou à un objet particulier, et qui peut entièrement changer la signification d'un geste. Par exemple, la main peut se trouver en générale devant le signeur, dans la zone d'à côté de la tête du signeur, sur la bouche, les yeux, le bras, la paume, etc. À l'instar du mot algérien « يأكل » qui est représenté en langue des signes algérienne par le signe de la figure III.4 (la main est devant le visage, et près de la bouche).



Figure III.4 : Signe du mot « يأكل » [27].

III.3.3.5 La mimique faciale

En réalité, quelques signes se composent d'un geste de main, ou des mains, et une expression faciale, ou plutôt **Mimique faciale**. En fait, cette dernière donne un sens à un signe isolé, et est fondamentale dans la construction d'une phrase. Par exemple, une mimique faciale est utilisée dans le mot « جو حار » qui est représentée en langue des signes algérienne par la figure III.5.



Figure III.5 : Le signe du mot « جو حار » [27].

III.3.4 Alphabet de la Langue des Signes Algérienne (ALSA)

En langue de signes algérienne, il y a 42 signes de l'alphabet algérien, parmi lesquels il y a 37 signes statiques et 5 signes dynamiques. En fait, ces signes sont représentés par une seule main (voir la figure III.6). Par ailleurs, chacun des signes statiques est déterminé au moyen de deux paramètres qui sont : la configuration et l'orientation.



Figure III.6 : Alphabet de la Langue des Signes Algérienne (ASLA) [3].

À titre d'exemple, le signe de l'alphabet « ب » est décrit par deux paramètres [3] :

- **Une configuration** : serrer le poing avec l'index tendu.
- **Une orientation** : la paume est orientée vers le haut (ou la position du poignet en bas).

III.4 Reconnaissance automatique des signes

III.4.1 Définition

La reconnaissance automatique des gestes en langues des signes (ARGSL : **A**utomatic **R**ecognition **G**esture of **S**ign **L**anguages) tente de faire reconnaître les signes, et de les traduire en une langue écrite, parlée, etc. [15]. De ce fait, l'ARGSL est une tâche difficile et complexe, car elle implique un processus de reconnaissance qui comprend en générale les phases suivantes (voir le chapitre de reconnaissance des formes) :

- Acquisition des données ;
- Prétraitement ;
- Extraction des caractéristiques .

Cependant, il faut noter que les langues des signes ne se limitent pas à des gestes des mains. Elles font également intervenir des mimiques faciales, et des postures du corps entier [3].

III.4.2 Avantages de la reconnaissance automatique des signes

Les systèmes de reconnaissance automatique des langues des signes possèdent de nombreux avantages, parmi lesquels nous pouvons citer [3] :

- Aider les enfants sourds à apprendre tôt, et rapidement la langue des signes.
- Très importants pour assurer une bonne communication entre les personnes sourdes et malentendantes.
- Ne nécessite pas d'avoir une surdité pour apprendre, ou communiquer en langue des signes.
- Permet à deux sourds de régions différentes, et de niveaux dissemblables, de se comprendre, et de se transmettre des informations.
- Les personnes sourdes peuvent écrire des phrases à partir des signes d'alphabet de la langue des signes, et de les traduire par un système de reconnaissance de langue des signes en alphabet écrits, et vice à versa.
- Simplifie l'enseignement, et augmente la capacité d'apprentissage chez les enfants sourds, puisque l'éducation est un droit pour chaque enfant.

- Les différents handicapés auditifs qui utilisent la langue des signes peuvent communiquer avec des personnes qui ne connaissent pas la langue des signes.
- Facilite l'élucidation pour apprendre la langue des signes.
- Permet de communiquer avec l'ordinateur facilement au moyen des gestes (signes), sans utiliser le clavier ou la souris.

III.5 Reconnaissance des gestes

Apparu depuis une dizaine d'années, la reconnaissance des gestes est l'un des domaines de recherche actuels. Les gestes humains peuvent être partiels ou complets, produits respectivement par des mouvements de/des main(s), de/des bras, de tête et du visage ou du corps entier. Dans notre vie, nous avons de nombreuses utilisations des gestes, selon le système de reconnaissance, nous trouvons la reconnaissance des gestes corporels, la reconnaissance des gestes du visage, et la reconnaissance des gestes de la main, etc.

III.5.1 Reconnaissance des gestes corporels

La reconnaissance des gestes corporel s'intéresse particulièrement à étudier l'interaction du corps complètement avec son environnement. Dans ces systèmes de reconnaissance de gestes, nous cherchons à identifier et reconnaître les actions des différentes parties du corps. De ce fait, de nombreux travaux ont été effectués dans cette perspective, tels que les interfaces gestuelles, suivi des différentes parties colorées du corps (les sportifs, les piétons, etc.).

III.5.2 Reconnaissance des gestes d'expression faciales

La reconnaissance des gestes de visage dépend en général des informations des expressions faciales. De ce fait, ces expressions se composent de trois composants du visage qui sont les yeux, les sourcils, et la bouche. Les caractéristiques des trois composants permettent au système de reconnaissance des gestes du visage de déterminer l'existence d'un objet visage ou non dans l'image (phase de détection), et de retrouver à qui appartient un visage particulier (phase de classification). Parmi les travaux effectués nous avons : la vidéo surveillance, la biométrie, la robotique, l'indexation d'images et de vidéos, la recherche d'images par le contenu (CBIR), etc.

III.5.3 Reconnaissance des gestes manuels

La/les main(s) permettent de réaliser des gestes précis et complexes. Le geste de la main est à la fois un moyen d'action, de perception et de communication. Donc, le geste est l'une des méthodes importante de communication non verbale entre les êtres humains, ou entre l'homme et la machine. La seconde méthode de communication est assurée par une interface homme-machine basée sur un système de reconnaissance des gestes de la main.

III.6 Conclusion

Dans ce chapitre, nous avons présenté la langue des signes algérienne, ainsi que les trois types de signes utilisés dans l'alphabet de cette langue, la composition de signe (la configuration, l'orientation, le mouvement, l'emplacement, mimique faciale). De ce fait, les gestes, ou plutôt les signes, sont des formes de communication, chaque forme est représentée par un ou plusieurs objets. Ces gestes sont reconnus par un système de reconnaissance des gestes, qui prend la même architecture qu'un système de reconnaissance des formes.

Dans le prochain chapitre de reconnaissance de gestes en langues des signes, nous allons présenter notre Système de Reconnaissance de l'Alphabet Statique de la Langue des Signes Algérienne (SRASLSA).

Chapitre IV

Conception

Et

Implémentation

IV.1 Introduction

Dans ce chapitre, nous présentons notre Système de Reconnaissance de l'Alphabet Statique de la Langue des Signes Algérienne (SRASLSA). En premier temps, nous débutons par une introduction qui constitue une brève définition de notre système de reconnaissance automatique, ensuite nous présenterons l'architecture générale du système proposé. Puis nous détaillerons les différents modules qui le composent. Nous présentons les résultats expérimentaux obtenus par l'application de la méthode de reconnaissance les réseaux de neurones convolutifs.

IV.2 Objectif et motivation

L'objectif d'un système de reconnaissance des signes d'une langue donnée est de classer chaque signe de la main vers un nombre fini de classes de signes prédéfinies, en utilisant un certain nombre de caractéristiques. Donc, notre système doit détecter l'existence ou non d'un objet main dans une scène donnée, et surtout interpréter sa posture.

En Algérie la communauté des sourds-muets compte environ 80 000 personnes, qui éprouvent généralement des difficultés d'adaptation vis-à-vis la quasi-totalité des supports d'information, que ce soit dans le domaine éducatif ou dans les autres domaines de la vie courante, qui ne prennent pas en charge leurs spécificités. Ainsi que, les limites de certains programmes de tradition qu'il ne se traduit pas en temps réel et qui sont uniquement disponible au niveau des ordinateurs bureautique. Notre but est de proposer un système qui aide cette tranche de la communauté (sourds-muets) à bien s'adapter et les facilite la communication avec d'autre personnes (non sourds-muets) même si ces derniers ne connaissent pas LSA. A cet effet, nous allons présenter à travers ce mémoire la conception et l'implémentation d'une application mobile pour la reconnaissance de la langue des signes algérienne.

Nous avons choisi d'implémenter notre approche sous Android, notre choix s'explique par une utilisation accrue de ce dernier partout à travers la planète, comme il a réussi quelque chose d'historique en surpassant le système d'exploitation Windows dans une période très courte, et en devenant l'OS le plus populaire pour surfer sur le net en 2018[16], (Figure IV.1).

De plus la portabilité des appareils mobiles, l'intégrité de leurs cameras puissantes, et la souplesse de leur manipulation, nous ont beaucoup aidé, de mener à bien la mise en œuvre, et le test de notre application.

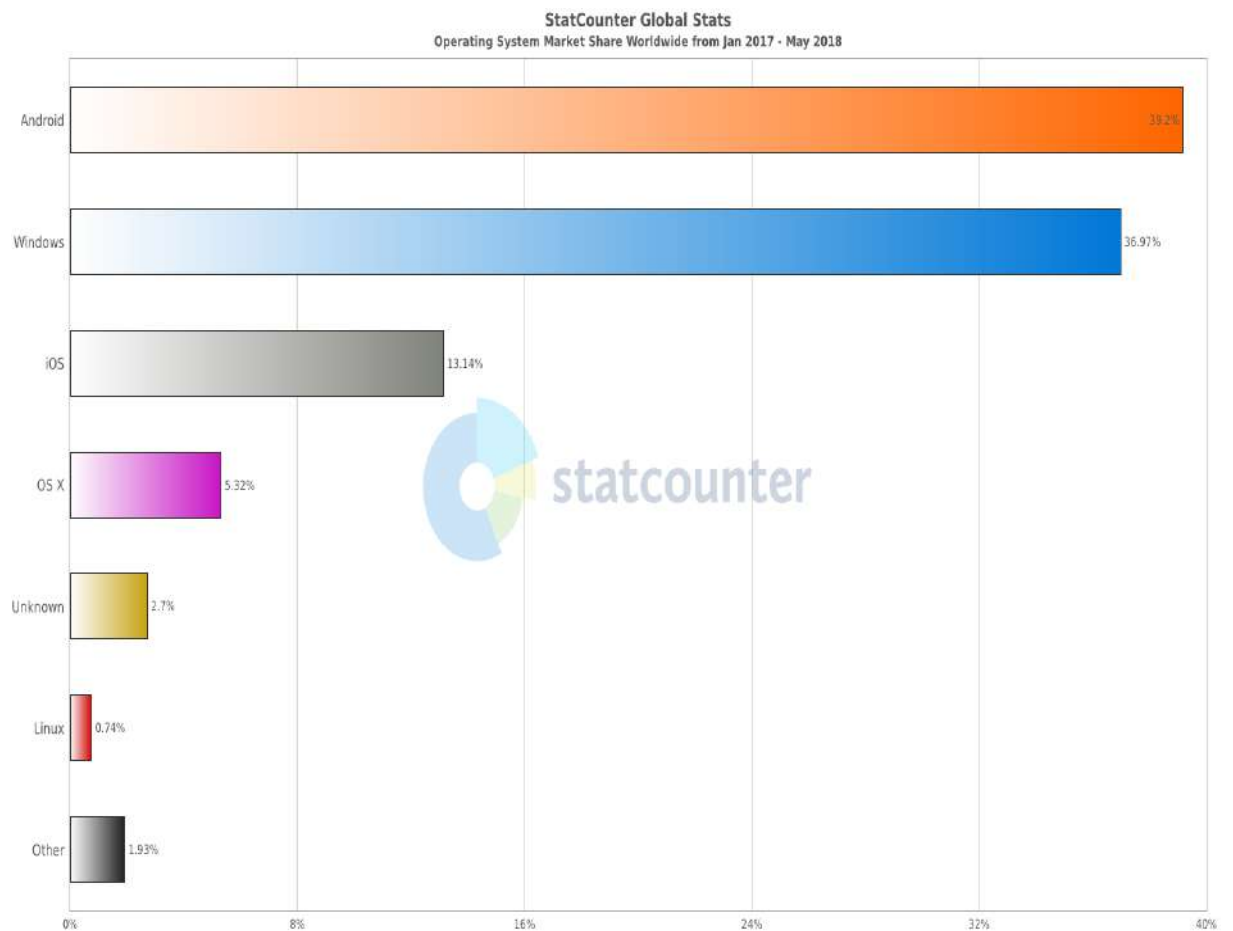


Figure IV.1 : L'analyse de StatCounter sur les parts de marché des systèmes d'exploitation mobiles pour la navigation sur internet [16].

IV.3 Architecture générale de l'application

Dans cette section, nous proposons un **Système de Reconnaissance de l'Alphabet Statique de la Langue des Signes Algérienne (SRASLSA)** en temps réel. La figure IV.2 présente l'architecture modulaire de notre système de reconnaissance proposé.

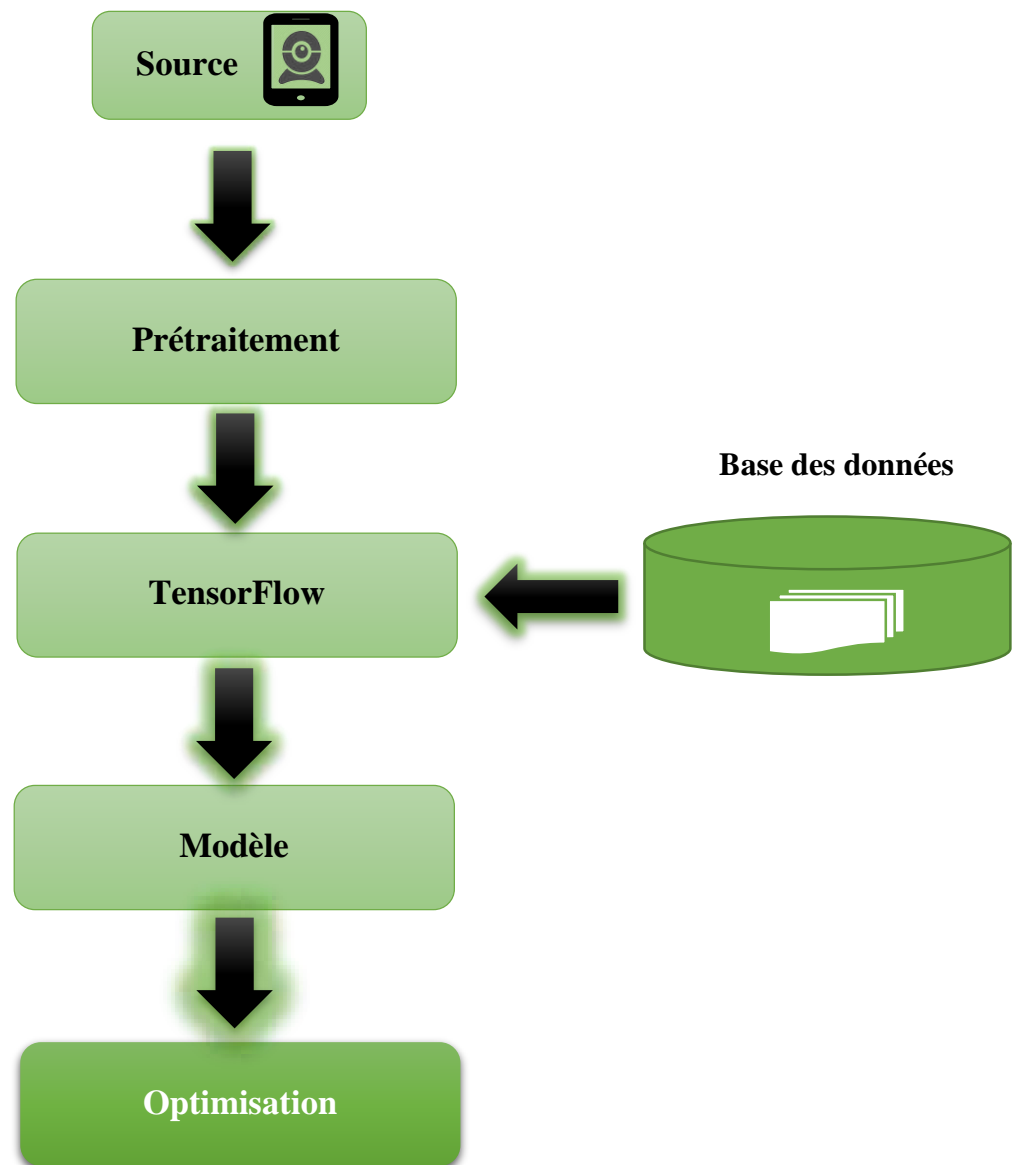


Figure IV.2 : Système de reconnaissance de la L.S.A.

IV.3.1 Source (caméra)

Dans ce module, on utilise l'appareil photo intégré dans les smartphone en appuyant sur la main à travers l'écran tactile pour avoir un bon échantillonnage de la main, Afin que la prise de la couleur cible juste la couleur de la peau de la main.

Une fois la représentation finale de la main est faite, la main sera traitée suivant plusieurs étapes. Le but de ce traitement est la préparation des éléments nécessaires pour le calcul des points caractéristiques spécifiques à chaque signe.

IV.3.2 Prétraitement

Les images (frames) capturées par la caméra sont très floues et contiennent du bruit, c'est pourquoi, on utilise des opérations de prétraitement pour améliorer la qualité des images. Le prétraitement comporte plusieurs étapes : l'obtention de l'image en niveaux de gris, le lissage...

IV.3.2.1 Image en niveau de gris

Le niveau de gris appelé aussi intensité d'une image $I(RGB)$ est une image I' bidimensionnelle, de même taille que l'image originale, mais dont chaque pixel est codé par 8 bits, ce qui limite les valeurs des pixels à l'intervalle $[0,255]$, c'est-à-dire il y a 256 valeurs de niveau de gris possibles. La conversion de l'image I en image I' est définie par la transformation [19] :

$$Gray: I \rightarrow I'$$

$$\hat{I} = 0.298 * R + 0.5870 * G + 0.1140 * B$$

IV.3.2.2 Lissage (filtrage Gaussien)

L'image en niveau de gris est pleine de bruits invisible à l'oeil. Le lissage est une opération de filtrage de ce type d'images. Parmi les filtres 2D on trouve les filtres moyen, médian et gaussien. Dans notre cas, nous utilisons le filtre gaussien qui permet de supprimer ces bruits, améliore la qualité de l'image et donne un bon résultat pour le module de segmentation par seuillage. Le filtre gaussien est défini par un écart type δ , une moyenne μ ainsi que des dimensions.

$$I_{lissée} = I' \otimes H.$$

IV.3.2.3 Segmentation par seuillage et Détection de contours

La segmentation des images permet de séparer chacun des objets présents dans une image. Les images de notre système ne contiennent que deux objets, un objet arrière-plan et un objet avant plan (la main). L'image segmentée par l'algorithme de seuillage contient des données non désirées (des trous et parfois des îles). Pour obtenir une bonne segmentation on ajoute les améliorations suivantes [20]:

- **Le contour de la main :** Afin de trouver le contour de la main dans l'image finale résultante, il suffit juste de chercher le plus grand contour, la **Figure IV.3** illustre le contour de la main en vert.



Figure IV.3 : Représentation du contour de la main en vert.

- **L'enveloppe convexe du contour (ou Convex Hull) :** Le but de l'utilisation de l'enveloppe convexe, c'est la détection des points convexes, et les points défauts de convexité par la suite. La Figure IV.4 illustre l'enveloppe convexe de la main en rouge.



Figure IV.4 : Représentation de l'enveloppe convexe du contour de la main en rouge.

IV.3.2.4 Filtrage morphologique

Les opérations de morphologie mathématique permettent de filtrer l'image en binaire ou en niveaux de gris. Dans un premier temps, nous appliquons une ouverture sur l'image segmentée de la figure IV.5, puis nous appliquons une opération de fermeture.

L'ouverture peut être caractérisée géométriquement : elle donne l'union de tous les B_x inclus dans X . Ainsi, la forme de l'élément structurant permet de choisir les structures qui peuvent le contenir.

La fermeture est le dual de l'ouverture : la fermeture du complémentaire d'un ensemble est égale au complémentaire de l'ouverture de cet ensemble.

La fermeture et l'ouverture sont des opérations croissantes et idempotentes, deux propriétés qui définissent les **filtres morphologiques**. La fermeture est extensive ($X \subset \phi(X)$) , et l'ouverture est anti-extensive ($\psi(X) \subset X$) .

- **L'ouverture** : l'opération d'ouverture filtre l'image, en supprimant les points et les petites îles blanches dans l'arrière-plan.

$$r_B(X) = X \circ B = \delta_B(\epsilon_B(X))$$

- **La fermeture** : l'opération de fermeture filtre l'image, en fermant les points et les petites îles noires de l'objet main. La figure IV.6 illustre le résultat du filtrage morphologique.

$$\phi_B(X) = X \bullet B = \epsilon_B(\delta_B(X))$$



Figure IV.5 : Image segmentée du signe « ٥ ».



Figure IV.6 : Image obtenue après filtrage morphologique du signe « ٥ ».

IV.4 Bibliothèques et logiciels utilisés

Pour assurer la bonne réalisation, et le bon déroulement de notre projet, nous avons utilisé un ensemble de logiciels, dont les plus importants sont les suivants :

- **Niveau Android ciblés :** Nous avons ciblé le niveau 19 d'Android 4.4, KitKat (2013), puis qu'il est encore largement utilisé, comme il contient tous les packages nécessaires pour le développement, et l'utilisation de notre application.
- **IDE utilisé :** Nous avons utilisé Android Studio 3.1.2 pour le développement de notre application, c'est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.
- **La base de données :** Nous avons repris la base d'image de Mr.Foudil Nekkaa qui est présenté un mémoire intitulé : Détection automatique de la main : Application à la reconnaissance de la langue des signes arabe.
Cette base d'images scindée en deux parties (apprentissage + test), la partie apprentissage contient 980 images en raison de 35 images pour chaque lettre, et la base de test contient 420 images en raison de 15 images pour chaque lettre.
- **OpenCV :** Une grande partie de notre application repose sur l'utilisation de la bibliothèque spécialisée dans le Computer Vision, OpenCV version 3.4.0 pour Android. L'utilisation de notre application nécessite aussi l'installation de OpenCV Manager, c'est un service Android pour la gestion des bibliothèques binaires de l'OpenCV, il permet aussi de partager les bibliothèques dynamiques entre les différentes applications dans le même appareil, en même temps. Comme il offre les avantages suivants [25] :
 - Moins d'utilisation de la mémoire.
 - Optimisations spécifiques aux ressources matérielles pour toutes les plates-formes prises en charge.
 - Mises à jour régulières et corrections de bugs.

Dans notre projet, nous avons utilisé la version 3.0 de OpenCV Manager.

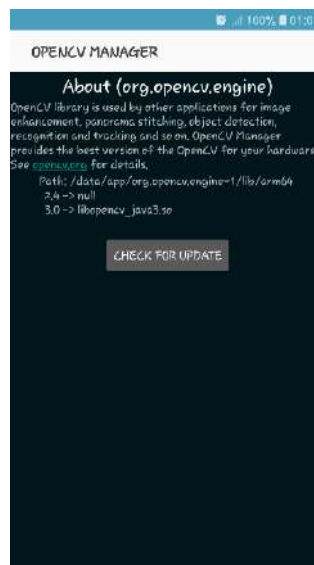


Figure IV.7 : Une capture d'écran de l'interface OpenCV Manager.

IV.5 Tensorflow

TensorFlow est disponible avec le support Python et C ++, et nous utiliserons Python 3.6 pour l'apprentissage, car en effet l'API Python est mieux supportée et beaucoup plus facile à apprendre.

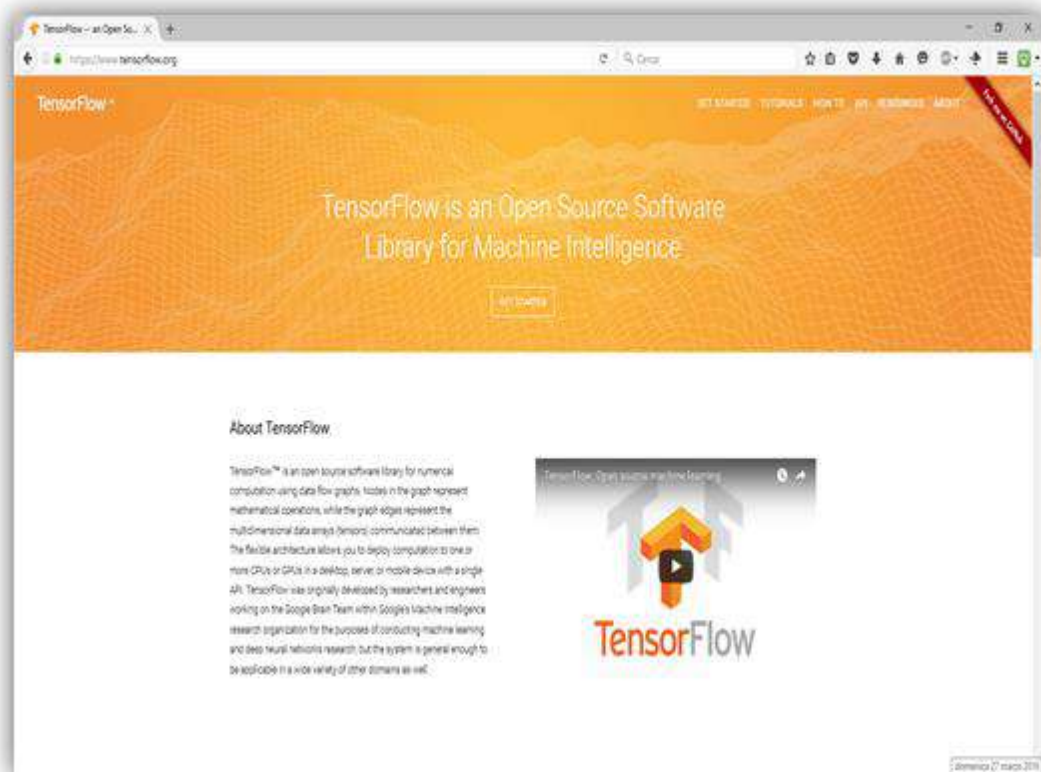


Figure IV.8 : La page d'accueil de Tensorflow[26].

TensorFlow propose une série de modèles pré-entraînés et prêts à l'emploi. Ces modèles permettent d'effectuer des tâches diverses et variées (classification d'images, extraction de texte d'images, ...) Nous avons cloné le dépôt Git pour télécharger le script « retrain.py » [21] pour que nous puissions commencer l'apprentissage et classifier les images de la base de données. Nous avons repris la base d'image de Mr.Foudil Nekkaa qui est présenté un mémoire intitulé : Détection automatique de la main : Application à la reconnaissance de la langue des signes arabe.

Il y a plusieurs méthodes de réseau de neurones (voir le chapitre I). Dans notre cas nous avons choisis les réseaux de neurones convolutifs (CNN).

IV.5.1 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Souvent désignés par l'acronyme CNN, (en anglais : Convolutional Neural Network), ils comportent deux parties bien distinctes. En entrée du CNN, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu] [23].

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution (taille) de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

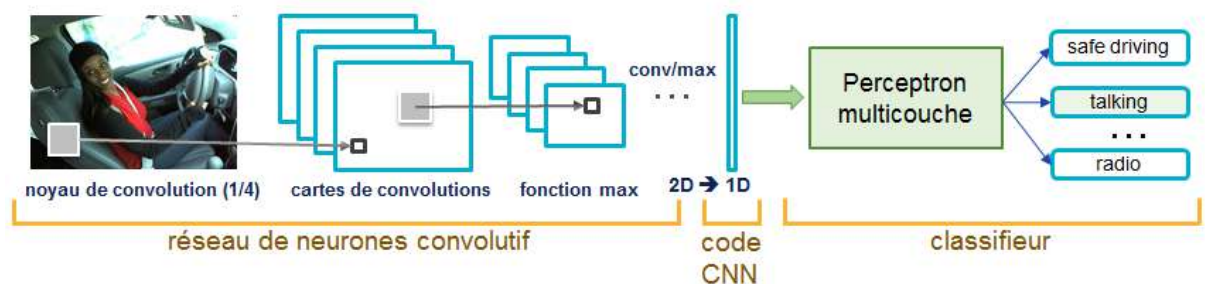


Figure IV.9 : Architecture standard d'un CNN [23].

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

IV.5.2 L'Entraînement d'un CNN

Il existe 2 méthodes pour entraîner un réseau de neurones convolutifs [22]:

- a) **L'entraînement d'un nouveau CNN** : Créer un nouveau réseau de neurones convolutif est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires. Il faut d'abord de fixer l'architecture du réseau : le nombre de couches, leurs tailles et les opérations matricielles qui les connectent. L'entraînement consiste alors à en optimiser les coefficients pour minimiser l'erreur de classification en sortie des exemples connus. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées [22].
- b) **Transfert learning : Adapter des CNN existants** : Toute la complexité de création de CNN peut être évitée en adaptant des réseaux pré-entraînés disponibles publiquement. Ces techniques sont appelées transfert learning, car on exploite la connaissance acquise sur un problème de classification général pour l'appliquer de nouveau à un problème particulier.

La "connaissance" sur la classification d'images contenue dans un tel réseau peut être exploitée de deux façons :

- Comme un **extracteur automatique de caractéristiques** des images, matérialisé par le code CNN. Une méthode simple et performante !
- En **Fine Tuning**, comme une initialisation du modèle, qui est ensuite ré-entraîné plus finement pour traiter le nouveau problème de classification. Une méthode qui demande plus d'investissement (technique, GPU) mais permet de faire monter la performance [22].

Les réseaux de neurones par convolutions sont basés sur trois hypothèses principales :

- **Associations locales** : tous les pixels ne sont pas connectés à tous les neurones cachés. Les connexions sont faites dans des petites zones localisées de l'image.
- **Mêmes poids et biais pour tous les neurones d'une même couche** : un seul type de filtre est appliqué dans une même couche, afin de détecter un pattern particulier (un coin, un angle, etc.). Tous les neurones d'une même couche détectent la présence ou non du même pattern.
- **Pooling** : condensation de l'information. On ne cherche pas à connaître l'emplacement exact d'un pattern, sa localisation approximative suffit. Il est donc courant de faire suivre une couche de convolution par une phase de pooling qui va condenser l'information et réduire la dimension des phases intermédiaires (on va par exemple garder la valeur maximale de 4 neurones d'une même zone).

Plusieurs couches de convolution peuvent alors s'enchaîner, et la dernière couche est suivie d'une couche dense où tous les neurones sont liés entre eux, pour enfin appliquer une fonction d'activation *softmax*.

Les fonctions d'activation, également appelées fonction de transfert, sont utilisées pour mapper les nœuds d'entrée aux nœuds de sortie d'une certaine manière. Il existe de nombreuses fonctions d'activation utilisées dans Machine Learning, Nous avons utilisé des réseaux de neurones pour la classification binaire. Souvent, on trouve que nous voulons prédire si une entrée appartient à l'une des nombreuses classes. C'est un problème de classification, Nous avons donc trouvé que le meilleur choix c'est d'utiliser la fonction **SOFTMAX**.

La fonction Softmax écrase les sorties de chaque unité entre 0 et 1, Il divise également chaque sortie de sorte que la somme totale des sorties soit égale à 1. La sortie de la fonction softmax est équivalente à une distribution de probabilité catégorique, elle vous indique la probabilité que l'une des classes soit vraie. Donc la fonction softmax normalise les sorties de sorte qu'elle somme à un, vous pouvez mettre un vecteur et sortir un vecteur où les sorties sont un vecteur de la même taille [24].

Par exemple, si vous avez trois entrées pour une fonction softmax, disons pour un réseau avec trois unités de sortie, cela ressemblerait à ceci :

$$\begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \rightarrow \text{Softmax} \rightarrow \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

Mathématiquement, la **fonction softmax**, ou **fonction exponentielle normalisée**, est une généralisation de la fonction logistique qui prend en entrée un vecteur de nombres réels et qui en sort un vecteur de nombres réels strictement positifs et de somme. La fonction est définie par [24] :

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ pour tout } j \in \{1, \dots, K\}$$

C'est-à-dire que la composante **j** du $\sigma(z)$ vecteur est égale à l'exponentielle de la composante **j** du vecteur **z** divisée par la somme des exponentielles de toutes les composantes de **z**.

En théorie des probabilités, la sortie de la fonction softmax peut être utilisée pour représenter une loi catégorielle – c'est-à-dire une loi de probabilité sur **K** différents résultats possibles.

La fonction softmax est également connue pour être utilisée dans diverses méthodes de classification en classes multiples, par exemple dans le cas de réseaux de neurones artificiels.

- Exemple de calcul pour l'image de la figure IV.9 :

$V = (0.8351391, 0.013643971, 0.17997506, 0.041772407, 0.3660802, 0.25601467, \dots, 0.3635863, 0.458322, 0.22507721, 0.10338175, 0.47627804).$



Figure IV.10 : Image du signe « ب ».

Après avoir exécuté le script « retrain.py » appliquera un traitement du modèle Inception sur les images de la base de données et suggérera des tags classés par score décroissant de probabilité.

```

C:\WINDOWS\system32\cmd.exe - python retrain.py --graph=D:\lsa\PFE --labels=D:\lsa\PFE --input_layer=Mul output_layer=final_result --input_meas...
INFO:tensorflow:2018-05-15 19:36:48.182299: Step 200: Cross entropy = 1.378010
INFO:tensorflow:2018-05-15 19:36:48.879792: Step 200: Validation accuracy = 78.0% (N=100)
INFO:tensorflow:2018-05-15 19:36:55.090683: Step 210: Train accuracy = 87.0%
INFO:tensorflow:2018-05-15 19:36:55.091683: Step 210: Cross entropy = 1.425211
INFO:tensorflow:2018-05-15 19:36:55.689109: Step 210: Validation accuracy = 75.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:01.368452: Step 220: Train accuracy = 93.0%
INFO:tensorflow:2018-05-15 19:37:01.368452: Step 220: Cross entropy = 1.266308
INFO:tensorflow:2018-05-15 19:37:01.928852: Step 220: Validation accuracy = 81.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:07.908104: Step 230: Train accuracy = 89.0%
INFO:tensorflow:2018-05-15 19:37:07.908104: Step 230: Cross entropy = 1.239494
INFO:tensorflow:2018-05-15 19:37:08.623634: Step 230: Validation accuracy = 85.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:14.554820: Step 240: Train accuracy = 81.0%
INFO:tensorflow:2018-05-15 19:37:14.555820: Step 240: Cross entropy = 1.234282
INFO:tensorflow:2018-05-15 19:37:15.121229: Step 240: Validation accuracy = 83.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:21.951020: Step 250: Train accuracy = 92.0%
INFO:tensorflow:2018-05-15 19:37:21.966031: Step 250: Cross entropy = 1.214925
INFO:tensorflow:2018-05-15 19:37:22.712562: Step 250: Validation accuracy = 81.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:28.784876: Step 260: Train accuracy = 90.0%
INFO:tensorflow:2018-05-15 19:37:28.785877: Step 260: Cross entropy = 1.121592
INFO:tensorflow:2018-05-15 19:37:29.811554: Step 260: Validation accuracy = 80.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:35.214699: Step 270: Train accuracy = 88.0%
INFO:tensorflow:2018-05-15 19:37:35.215695: Step 270: Cross entropy = 1.123283
INFO:tensorflow:2018-05-15 19:37:35.638992: Step 270: Validation accuracy = 87.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:39.846985: Step 280: Train accuracy = 92.0%
INFO:tensorflow:2018-05-15 19:37:39.846985: Step 280: Cross entropy = 1.006623
INFO:tensorflow:2018-05-15 19:37:40.185223: Step 280: Validation accuracy = 88.0% (N=100)
INFO:tensorflow:2018-05-15 19:37:45.134769: Step 290: Train accuracy = 90.0%
INFO:tensorflow:2018-05-15 19:37:45.135771: Step 290: Cross entropy = 1.052392
INFO:tensorflow:2018-05-15 19:37:45.756183: Step 290: Validation accuracy = 87.0% (N=100)

```

Figure IV.11 : Interface d'apprentissage de réseau de neurones.

Après l'apprentissage, nous avons obtenus un taux de reconnaissance (86% environ).

```

C:\WINDOWS\system32\cmd.exe
INFO:tensorflow:2018-05-23 18:32:24.589917: Step 3990: Validation accuracy = 84.0% (N=100)
INFO:tensorflow:2018-05-23 18:32:30.010675: Step 3999: Train accuracy = 93.0%
INFO:tensorflow:2018-05-23 18:32:30.011678: Step 3999: Cross entropy = 0.202102
INFO:tensorflow:2018-05-23 18:32:30.359925: Step 3999: Validation accuracy = 83.0% (N=100)
INFO:tensorflow:2018-05-23 18:32:30.359925: Step 3999: Validation accuracy = 85.6% (N=208)
INFO:tensorflow:Froze 2 variables.
Converted 2 variables to const ops.
Model path: /tmp/imagenet/classify_image_graph_def.pb
INFO:tensorflow:Restoring parameters from D:\lsa\PFE\_retrain_checkpoint
INFO:tensorflow:No assets to save.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: b'/tmp/saved_models/1/saved_model.pb'

```

Figure IV.12 : Le taux de reconnaissance finale.

Un petit bonus avant de passer à la suite : TensorFlow nous permet de voir les courbes d'entraînement de notre algorithme utiliser.

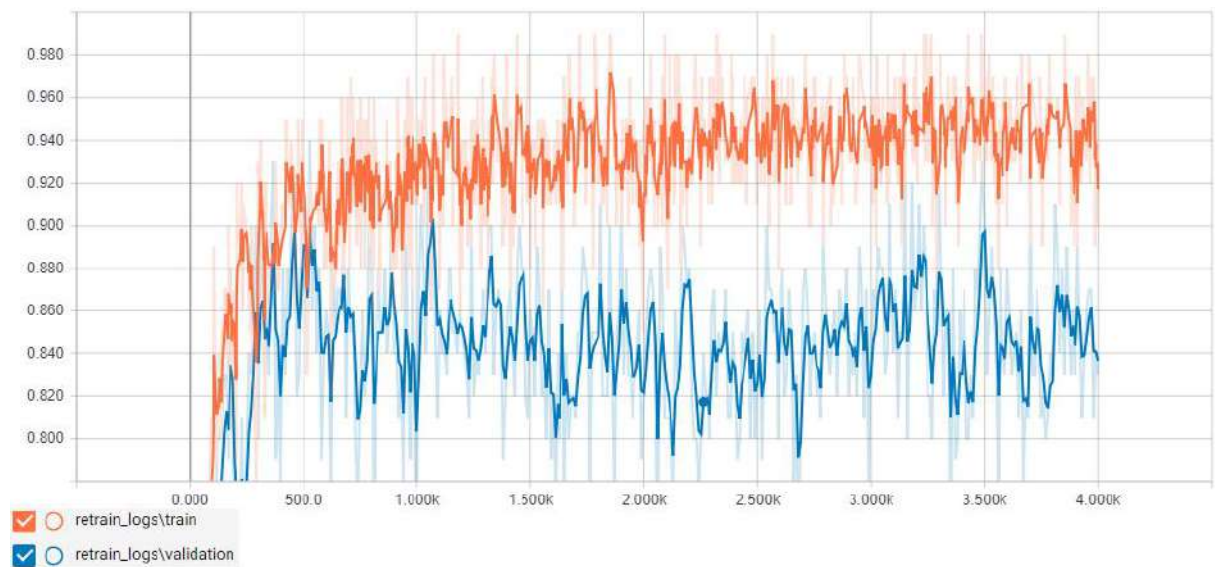


Figure IV.13 Le taux de reconnaissance ou taux de succès (en anglais : accuracy).

IV.6 Le Modèle

Dans ce module, Le script écrira le nouveau modèle ré-entraîner « output_graph.pb » contient le fichier de définition du graph du modèle TensorFlow. C'est le cœur puisqu'il contient le réseau de nœuds qui traitent les données en entrée et « output_labels.txt » Contient les informations de mapping pointant vers les différentes étiquettes (labels).

Nous nous en rendrons compte très rapidement lorsque nous allons créer des réseaux de neurones, l'enchaînement des opérations peut vite devenir complexe. Nous allons très vite ressentir le besoin de visualiser le graphe créé, ainsi que de contrôler l'évolution de nos phases d'apprentissage (évolution du taux de prédiction, activités des neurones, etc.). Heureusement pour nous, TensorFlow met à disposition un outil, **TensorBoard**, qui répond à ces besoins.

TensorBoard est une réelle force et constitue un vrai élément différenciant de TensorFlow par rapport aux autres frameworks de Deep Learning.

Pour utiliser **TensorBoard**, Il suffit de lancer la commande adéquate avec le chemin d'accès vers les logs pour que l'interface graphique associée se lance.

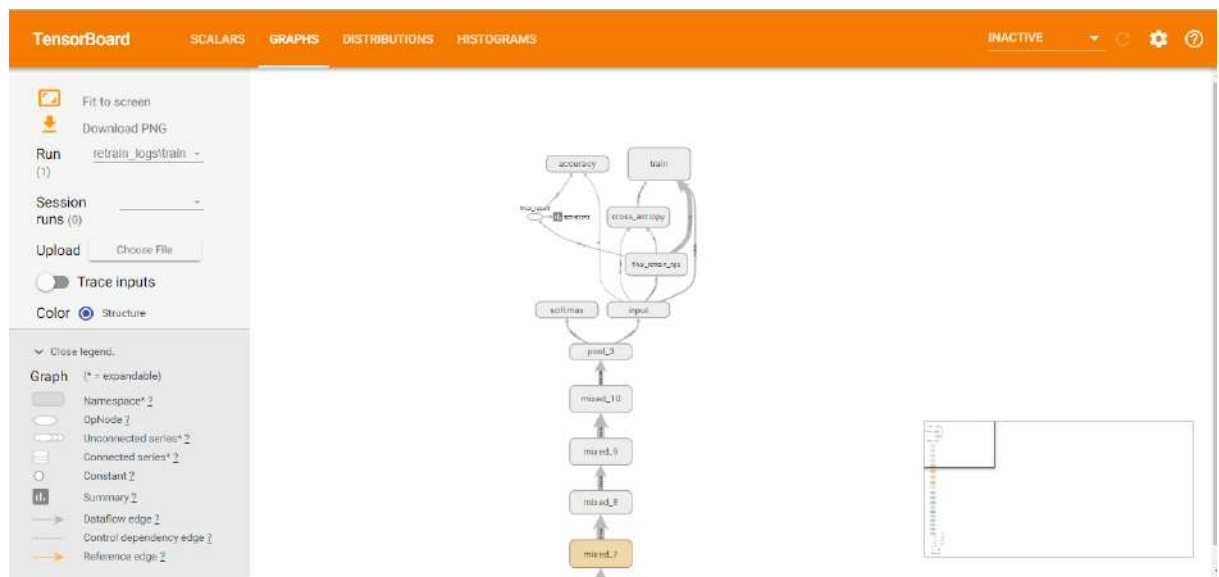


Figure IV.14 : Visualisation de graph à l'aide de Tensorboard.

IV.7 Optimisation

Dans cette étape, Les résultats fournis par le modèle « output_graph.pb » n'est pas supporté par l'appareil Android qu'on utilise. Pour y remédier, On utilise un script python «Optimize_for_inference.py » ce dernier permet d'optimiser le graph et le rend utilisable sur notre appareil.

Le script effectue également quelques autres optimisations qui accélèrent la vitesse, telles que la fusion d'opérations de normalisation par lots explicites dans les poids convolutionnels afin de réduire le nombre de calculs.

IV.8 Expérimentations

Dans cette étape, Une fois la base d'apprentissage est prête, nous allons tester la performance finale de notre système pour la reconnaissance d'un exemple de mot « التخرج ». On utilise les meilleurs paramètres que nous avons obtenus précédemment (L'évaluation).

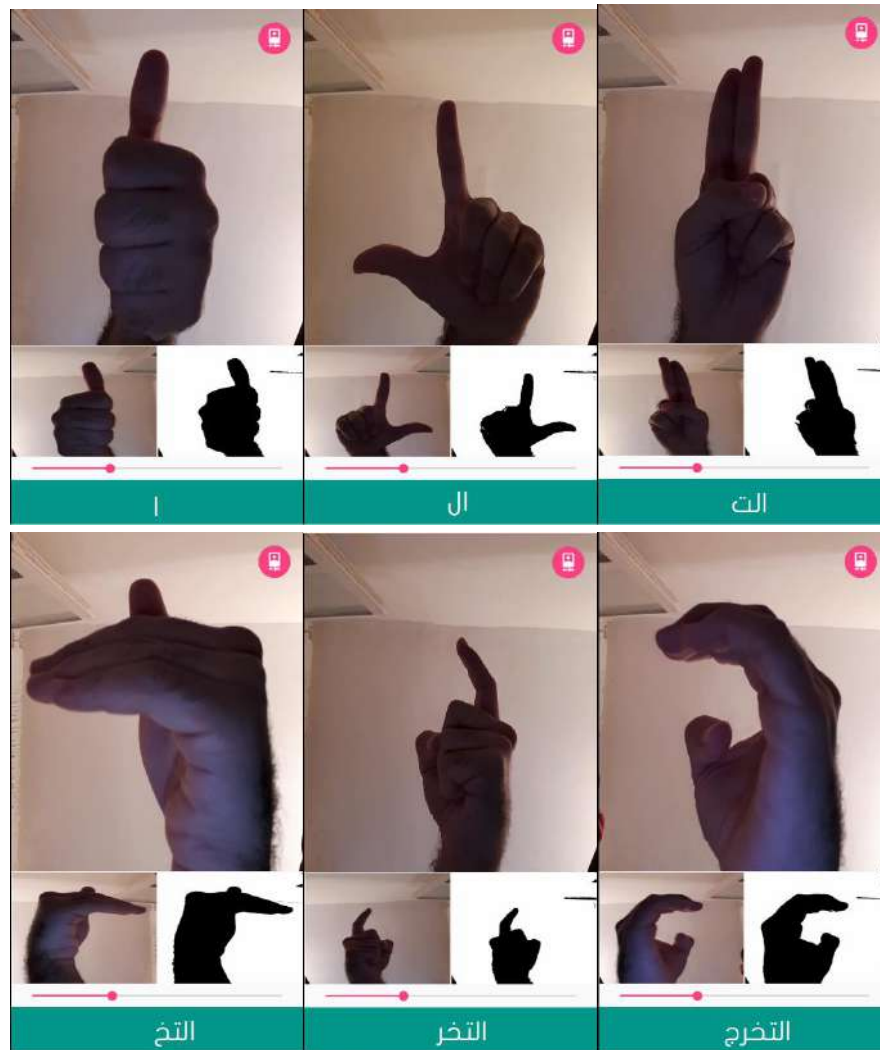


Figure IV.15 : Un exemple de traduction.

IV.9 Conclusion

Nous avons présenté dans ce chapitre l'architecture de notre système de d'alphabet statique de la langue des signes algérienne en temps réel, ainsi que une approche de classification basée sur les réseaux de neurones convolutionnels(CNN), pour cela on a utilisé un modèle d'architecture connue sous le nom Tensorflow et on a montré les résultats obtenus. Les résultats que nous avons trouvés ont montré que la profondeur des réseaux, le nombre d'époque et la taille de la base sont des facteurs important d'obtenir de meilleurs résultats.

Conclusion Générale

De nos jours, la vision par ordinateur joue un rôle vital dans notre vie quotidienne, ce qui nous a motivés à présenter à travers ce mémoire la conception, et l'implémentation d'une approche utilisée pour la reconnaissance automatique des signes statiques de la langue des signes algérienne en temps réel.

La première étape de notre travail a été consacrée à l'acquisition des images et leur prétraitement afin d'améliorer leur qualité et faciliter l'extraction des caractéristiques. Nous avons opté pour certains algorithmes de prétraitement, tels que : le filtrage pour la réduction de bruits, la segmentation pour rendre l'image binaire (noir et blanc), la morphologie mathématique pour le remplissage des trous, le cadrage de la main et la normalisation des images.

La deuxième étape, Concerne l'utilisation de la nouvelle bibliothèque TensorFlow et l'implémentation de la méthode de reconnaissance à savoir : Un réseau de neurones convolutifs (CNN) qui donne un meilleur taux de reconnaissance (86%) environ.

Notre système de reconnaissance est capable de « lire » les signes statiques de la main de d'alphabet de la langue des signes algérienne en temps réel, de les reconnaître puis les traduire automatiquement en lettres écrites.

Comme future perspective pour notre système, Nous pensons que l'amélioration des performances peut se faire soit par l'amélioration des méthodes de prétraitement, soit par le choix des méthodes de reconnaissances de la main automatique en temps réel, Nous pensons également d'étendre notre système à la reconnaissance des signes dynamiques afin de pouvoir traduire tous les signes de la langue des signes algérienne.

Bibliographie

- [1] J.-P. Asselin de Beauville, F. Kettaf, Bases théoriques pour l'apprentissage et la décision en reconnaissance des formes , Éditions Cépaduès, 2005.
- [2] Dietrich W. R. Paulus, Joachim Hornegger ,Applied Pattern Recognition, Fourth Edition: Algorithms and Implementation in C++ (December 02,2003)
- [3] Foudil Nekkaa, « Détection automatique de la main : Application à la reconnaissance de la langue des signes arabe », spécialité : Systèmes Distribués et Méthodes Formelles (SDMF), Université Abdelhamid Mehri-Constantine 2,2014/2015.
- [4] Nicholas Journet. *Introduction au traitement d'images Reconnaissance des formes*, 12 janvier 2011.
- [5] C.K.Chow, "Statistical Independence and Threshold Functions", IEEETransactions of Electrical Computer", pp 66-68, 1965 .
- [6] «Méthode Des K Plus Proches Voisins», disponible sur le lien:
http://fr.wikipedia.org/wiki/M%C3%A9thode_des_k_plus_proches_voisins
- [7] Mehennaoui Zahra, “RECONNAISSANCE DE L’ECRITURE ARABEMANUSCRITE A BASE DES MACHINES AVEC TEURS DE SUPPORTS”. Spécialité : Intelligence Artificielle Distribuée (IAD). Mémoire de Magistère en informatique. annaba : UNIVERSITE BADJI –MOKHTAR – ANNABA, 2006.
- [8] Laurence Likforman-Sulem, Elisa Barney Smith, « Reconnaissance des formes », Edition Ellipses, paris 2013.
- [9] LIKFORMAN-SULEM, Elisa BARNEY SMITH. Reconnaissance des formes théorie et pratique sous Matlab Cours et exercice corrigés.paris.ellipse.2013.222p.
- [10] http://fr.wikipedia.org/w/index.php?title=Langue_des_signes. [consulté le 11/04/2018].
- [11] <http://www.francosourd.com> [consulté le 20/04/2018].
- [12] Julien, Thomet. *Une vue d'ensemble de la reconnaissance de gestes*, Département d’informatique Université de Fribourg, vol.8.
- [13] Bossard, Bruno. *Problèmes posés par la reconnaissance de gestes en Langue des Signes*. LIMSI-CNRS - Université Paris XI, Orsay. Soutenu 24–27 juin 2002.10 p.
- [14] LAKHFIF, Abdelaziz. *Un Environnement de Traduction Automatique du Texte Arabe vers la Langue des Signes Algérienne (LSA)*. Magistère en Informatique. Université Badji Mokhtar-Annaba. Soutenu le Juillet 2009.134p.
- [15] Mauro dos Santos Anjo, Ednaldo Brigante Pizzolato, Sebastian Feuerstack «An evaluation of real-time requirements for automatic sign language recognition using ANNs and HMMs - The LIBRAS use case», Computer Science and Engineering, Department Universidade Federal de Sao Carlos – UFSCar, vol 4.
- [16] Parts de marché selon StatCounter, sur gs.statcounter.com , mai 2018

- [17] Pablo MUSÉ. Frédéric SUR. Jean-Michel MOREL. *Les seuils de reconnaissance des formes On shape recognition thresholds*, Centre de Mathématiques et de Leurs Applications. École Normale Supérieure de Cachan, 2003,294p.
- [18] https://fr.wikipedia.org/wiki/Langue_des_signes_alg%C3%A9rienne [consulté le 15/03/2018]
- [19] Rachid Belaroussi, « *Traitement De L'image Et De La Vidéo* », Edition Ellipses, paris 2010.
- [20] Z.Chen, J.Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan, “Real-Time Hand Gesture Recognition Using Finger Segmentation,” The Scientific World Journal, vol. 2014, Article ID 267872, 9 pages, 2014. doi:10.1155/2014/267872
- [21] https://github.com/tensorflow/hub/blob/master/examples/image_retraining/retrain.py [consulté le 29/04/2018].
- [22] Classification d’images : les réseaux de neurones convolutifs en toute simplicité. Posté le 25/10/2016 par Julien Krywyk, Pierre-Alain Jachiet Disponible sur : <http://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/> [consulté le 11/05/2018].
- [23] Mr Mokri Mohammed Zakaria. *Classification des images avec les réseaux de neurones Convolutionnels*. master en Informatique. Université Abou Bakr Belkaid Tlemcen. Soutenu le 03/07/2017.62p.
- [24] Deep learning concepts—PART 1 ,Disponible sur : <https://towardsdatascience.com/deep-learning-concepts-part-1-ea0b14b234c8> [consulté le 15/05/2018].
- [25] <https://opencv.org/> [consulté le 20/04/2018].
- [26] <https://www.tensorflow.org/> [consulté le 10/04/2018].
- [27] <https://www.slideshare.net/ALMOTAZBELLAH/ss-36969315> [consulté le 03/05/2018].
- [28] Fabrice Rossi. *Réseaux de neurones : le perceptron multi-couches*. Université Paris-IX Dauphine.52p.
- [29] Mokri Mohammed Zakaria « Classification des images avec les réseaux de neurones Convolutionnels », Option : Modèle Intelligent et Décision, Université Abou Bakr Belkaid Tlemcen, 03/07/2017.
- [30] Valentin BARRIERE, *Approches « deep learning » appliquées aux signaux audio : parole et musique*, Université Paris Saclay,04/09/2015.
- [31] Kaz sato , Staff developer advocate at Cloud Platform team, Google Inc, machine intelligence at google scale , Presentation sur : <https://www.slideshare.net/mediahanbit/track2-02-machine-intelligence-at-google-scale-google-kaz-sato-staff-developer-advocate> [consulté le 03/05/2018].

Résumé : Notre travail concerne le domaine de la vision par ordinateur, et plus particulièrement la reconnaissance des formes. Nous proposons un système de reconnaissance des signes statiques de l'alphabet de la langue des signes algérienne en temps réel.

La langue des signes algérienne (LSA) est la langue gestuelle utilisée par les personnes sourdes et leurs proches en Algérie pour pouvoir se comprendre par le biais de signes. Le but de ce travail est d'implémenter un système qui est capable de reconnaître et interpréter ces signes, afin de les traduire après en mots compris par des personnes ayant une faible connaissance en cette langue. Notre système est basé sur le développement de la méthode de classification les réseaux de neurones convolutifs (CNN) et l'apprentissage en profondeur (Deep Learning) en utilisant la nouvelle bibliothèque TensorFlow.

Mots clés : reconnaissance des formes, langue des signes algérienne, classification, Réseau de neurones, apprentissage en profondeur, bibliothèque Tensorflow.

Abstract: Our work is concerned with the domain of computer vision, and more particularly pattern recognition. We propose a system of recognition of the static signs of Algerian sign language in real time.

The Algerian sign language (ASL) is the gestural language used by deaf people and their relatives in Algeria to be able to understand through signs. The purpose of this work is to implement a system which is capable of recognizing and interpreting these signs, in order to translate them into words understood by persons having a low knowledge in that language. Our system is based on the development of the classification method convolutional neural networks (CNN) and deep learning using the new TensorFlow library.

Key words: pattern recognition, the Algerian sign language, classification, neural networks, deep learning, Tensorflow library.

ملخص: يتعلق هذا العمل بمجال الرؤية الاصطناعية والتعرف على الرموز عن طريق الحاسب. وبالأخص التعرف على الأنماط. نقترح من خلال هذا العمل نظام مباشر للتعرف الفوري على رموز لغة الإشارات الجزائرية.

لغة الإشارات الجزائرية هي اللغة الإيمائية المستخدمة من قبل الأشخاص الصم والبكم وأقاربهم في الجزائر للتمكن من التفاهم من خلال الإشارات. الهدف من هذا العمل هو تنفيذ نظام قادر على معرفة وتفسير هذه الإشارات، لكي نترجم بعد ذلك إلى كلمات مفهومة بواسطة أشخاص لديهم القليل من المعرفة في هذه اللغة. يستند نظامنا على تطوير طريقة للتصنيف وهي الشبكات العصبونية الالتفافية (CNN) وكذلك على التدريب العميق باستخدام المكتبة الحديثة (Tensorflow).

الكلمات المفتاحية : التعرف على الأنماط , لغة الإشارات الجزائرية , التصنيف , الشبكات العصبونية , التدريب العميق , المكتبة الحديثة Tensorflow.