



Mémoire de Projet de fin d'étude

Préparé par

BOUFETTAL BRAHIM

Pour l'obtention du diplôme

Master Sciences et Techniques

Electronique, Signaux et Systèmes Automatisés

(E.S.S.A)

Intitulé

**Développement et implémentation d'un
système de reconnaissance d'objets en
utilisant le Deep Learning**

Encadré par :

Pr Najia ES-SBAI

Mr Mohammed SERBOUTI(Raynov)

Soutenu le **15 Juin 2016**, devant le jury composé de :

Pr Najia ES-SBAI.....: Encadrant

Mr Mohammed SERBOUTI: Encadrant

Pr Ali AHAITOUF..... : Examineur

Pr Mohammed JORIO..... : Examineur

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَقْرَأُ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ١ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ٢ أَقْرَأُ وَرَبُّكَ الْأَكْرَمُ ٣
الَّذِي عَلَّمَ بِالْقَلَمِ ٤ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ

الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذَا وَمَا كُنَّا لِنَهْتَدِيَ لَوْلَا أَنْ هَدَانَا اللَّهُ

Dédicaces

Je dédie mon modeste travail

A mes chers parents

*Pour leur grand soutien, leurs sacrifices que rien au monde ne peut les récompenser et sans
lesquels je n'aurais jamais pu parcourir ce chemin.*

Que Dieu vous protège.

A mes chers frères et sœurs

Pour leur amour et leur incontestable appui.

A mes chers ami(e)s.

A toutes les personnes chères à mon cœur je dédie ce travail.

Remerciements

Avant tout remerciement, louange à Dieu.

*J'exprime toute ma reconnaissance à **M. Mohammed Serbouti** le directeur de la société Raynov, de m'avoir accueilli au sein de sa société pendant cette période de stage et d'avoir créé les conditions nécessaires à l'adaptation et à l'évolution de mon projet de fin d'études. Aussi je le remercie pour son encadrement sa disponibilité et ses conseils judicieux.*

*Je tiens à présenter mes reconnaissances à mon professeur encadrante **M. Najia Es-sbai**, pour ses conseils, ses orientations, et la disponibilité qu'il m'a accordée pour faire réussir ce travail, tout au long de la période de stage.*

Je remercie mes professeurs de la FST de Fès qui ont fait beaucoup d'efforts pour nous transmettre leurs connaissances. Vos compétences incontestables ainsi que vos qualités humaines vous valent l'admiration et le respect de tous. Je vous adresse mes sincères remerciements pour votre patience et votre encadrement durant toutes ces années.

Merci aux membres du jury pour l'honneur qu'ils me font en jugeant ce travail. Et à vous tous, honorables lecteurs.

Résumé

Le présent rapport synthétise le travail effectué dans le cadre de mon stage de fin d'études au sein de la société RAYNOV. Il s'agit du développement et de l'implémentation d'un système de reconnaissance d'objets en utilisant le Deep Learning.

En premier lieu nous avons développé une application avec une interface graphique en C++ destinée à la reconnaissance des objets et des visages, elle se base sur les réseaux de neurones artificiels, la méthode de Viola et Jones et celle de SURF, cette application donne des résultats intéressants au niveau de la reconnaissance même pour une base d'images petite.

En deuxième lieu nous avons développé une autre application avec une interface graphique en python destinée pour la reconnaissance des objets, des émotions, des places publiques, de l'âge et de genre ainsi que des célébrités. Cette application est basée sur le Deep Learning, en utilisant des modèles inclus dans le Framework Caffe qui ont montré des résultats intéressants au niveau de la reconnaissance ainsi qu'un taux d'erreur petit.

Finalement nous avons conçu un système capable de reconnaître des objets en utilisant un ordinateur puissant pour servir le Raspberry Pi afin d'exécuter les algorithmes de Deep Learning.

Mots clés:

Reconnaissance des objets, Deep Learning, Apprentissage profond, Réseaux de neurones, Surf, Haar features, python, Opencv, Raspberry Pi.

Abstract

This report summarizes the work done as part of my end of studies within the RAYNOV Company is the development and implementation of an object recognition system using Deep Learning.

Firstly we developed an application with a GUI in C ++ for objects and faces recognition, using neural network, viola and Jones, and SURF algorithm. This application has interesting results in terms of recognition when we use a small datasets.

Secondly we have developed another application with GUI in python for the recognition of objects, emotions, public place, age and gender, and famous, it is based on Deep Learning and we used models include in the Framework Caffe they have interesting results in terms of recognition and they have small error rate.

Finally we designed a system that can recognize objects in real time using a powerful computer to use the Raspberry Pi to run Deep Learning algorithms.

Keywords:

Object recognition, Deep Learning, neural networks, Surf, Haar features, Python, OpenCV, Raspberry Pi.

Organisme d'accueil

RAYNOV est une société Marocaine d'ingénierie fondée en : 2012, son cœur de métier est la conception et la réalisation des systèmes de haute technologie, depuis l'étude de faisabilité du projet jusqu'à la réalisation et la mise en service des installations. Son domaine de spécialisation est l'industrie technologique. Ses ressources humaines comptent environ 10 employés.

Parmi ses domaines d'expertise on trouve :

- La conception électronique
- Le développement logiciel
- Les systèmes Robotiques
- Les bancs de test
- La conception système-Produit.

Liste des Figures

Figure 1 : Le résultat de la reconnaissance d'une rose en utilisant le modèle de correspondance.....	14
Figure 2 : la reconnaissance des membres de la famille Simpsons en utilisant les couleurs ...	15
Figure 3 : La méthode de reconnaissance de multi objets dans une image.....	16
Figure 4 : l'extension du caractéristique pseudo haar par Lienhart et Maydt	17
Figure 5 : Le résultat de détection de visage en utilisant le Haar Features	18
Figure 6 : Fonctionnement de la méthode surf.....	19
Figure 7 : le résultat de la reconnaissance de logo de la FST en utilisant la méthode SURF ..	19
Figure 8 : Modèle simplifié d'un réseau de neurones artificiels	23
Figure 9 : la tangente hyperbolique.....	25
Figure 10 : Perceptron multicouche	25
Figure 11 : La descente du gradient : Minimum global et local	26
Figure 12: Interface d'accueil de l'application Raynov Learning.....	29
Figure 13: Interface graphique de l'apprentissage (Raynov Learning).....	29
Figure 14 : La fenêtre d'aide (Raynov Learning).....	30
Figure 15 :L'interface graphique de la Reconnaissance (Raynov Learning).....	30
Figure 16 : Bases des images pour le test de fonctionnement des visages (Raynov Learning)	31
Figure 17: Test d'apprentissage pour les visages (Raynov Learning)	31
Figure 18 : Renommer une catégorie (Raynov learning).....	32
Figure 19: Résultat de la reconnaissance du visage	32
Figure 20: Exemple d'un système de Deep Learning	36
Figure 21: Principe de DeepFace (de Facebook)	37
Figure 22: Nvidia Drive	38
Figure 23 : interface graphique de l'application Raynov Deep Learning	41
Figure 24 : Télécharger une image (Raynov Deep Learning).....	41
Figure 25 : la classification d'une image (Raynov Deep Learning).....	42
Figure 26 : La classification en temps réel (Raynov Deep Learning).....	42
Figure 27 : Le Help de l'application Raynov Deep Learning	42
Figure 28 : avertissement d'erreur (Raynov Deep Learning).....	43
Figure 29 : la base d'images pour le test de fonctionnement (Raynov Deep Learning)	43
Figure 30 : le résultat de classification de l'image d'une télécommande	43
Figure 31 : L'image de deux ordinateurs utilisée pour le test de fonctionnement (Raynov Deep Learning).....	44
Figure 32 : Résultat de la classification d'un ordinateur fixe (Raynov Deep Learning).....	44
Figure 33 : Résultat de la classification d'un ordinateur portable (Raynov Deep Learning)...	44
Figure 34 : le système Raynov artificiel.....	49
Figure 35 : Interface graphique de l'application du serveur (Raynov artificiel).....	50
Figure 36 : Interface graphique de l'application du serveur (Raynov artificiel).....	50
Figure 37 : Résultat textuel en anglais de la reconnaissance (Raynov artificiel).....	51

Liste des acronymes

Abréviations	Désignations
SIFT	scale-invariant feature transform
SURF	Speeded Up Robust Features
OpenCV	Open Source Computer Vision
API	Application Programming Interface
CPU	central processing unit
GPU	Graphics Processing Unit
PMC	Perceptron multi couche
XML	eXtensible Markup Language

Sommaire :

Dédicaces	2
Remerciements	3
Résumé	4
Abstract	5
Organisme d'accueil.....	6
Introduction générale	11
Chapitre1 : la reconnaissance des objets	12
I. Introduction.....	13
II. L'état de l'art de la reconnaissance des objets :	14
1. Le Modèle de correspondance (Template matching) :.....	14
2. Reconnaissance par couleurs :	15
3. Reconnaissance par formes :.....	15
4. La Reconnaissance de plusieurs objets dans une image :.....	16
III. La détection des Objets par la méthode de Viola et Jones :	16
1. L'algorithme de Viola et Michael Jones :.....	16
2. Application de la détection faciale en utilisant L'Opencv :	17
IV. La reconnaissance des objets par la méthode de SURF :.....	18
1. La méthode de SURF :	18
2. Application de La méthode de SURF en utilisant Opencv:.....	19
V. Conclusion.....	20
Chapitre2 : Le développement de l'application Raynov Learning en utilisant les Réseaux de neurones artificiels	20
I. Introduction.....	22
II. Les Réseaux de neurones artificiels :	22
1. Concept :.....	23
2. Historique :	23
3. Perceptron multicouche :.....	24
a. Architecture :.....	24
b. La Rétropropagation du gradient :	25
III. Environnement de développement :	27
1. Environnement hardware :	27
2. Environnement software :.....	27
IV. Méthodologie de développement :.....	27

V.	Description de l'application :.....	29
VI.	Test de fonctionnement :.....	31
VII.	Conclusion	32
	Chapitre3 : Le développement de l'application Raynov Deep Learning	33
I.	Introduction.....	34
II.	Le Deep Learning :	34
1.	Concept et techniques:.....	34
2.	Les champs d'application :.....	36
a.	DeepFace de Facebook :.....	36
b.	Skype Translator de Microsoft :	37
c.	NVIDIA Drive :	37
d.	DreamQuark :.....	38
3.	Le Framework Caffe :.....	38
III.	Environnement de développement :	39
1.	Environnement hardware :	39
2.	Environnement software :	39
IV.	Méthodologie de développement :.....	40
V.	Description de l'application :.....	41
VI.	Test de Fonctionnement :	43
VII.	Conclusion :.....	44
	Chapitre4 : développement et implémentation du système Raynov artificiel	46
I.	Introduction.....	47
II.	Environnement de développement :	47
1.	Environnement Hardware :	47
2.	Environnement software :	48
III.	Méthodologie du travail :.....	48
IV.	Conclusion.....	51
	Conclusion générale.....	52
	Bibliographie et Webographie	53

Introduction générale

La reconnaissance des objets est parmi les sujets les plus importants en vision par ordinateur, il a subi une immense évolution pendant des siècles, en revanche les dernières années l'intelligence artificielle a connu l'apparition de Deep Learning, et grâce aux efforts des chercheurs le Deep Learning connaît un grand succès, ses applications ont touché des domaines différents, comme la robotique, l'industrie, l'automobile

Dans ce cadre, Raynov a pris l'initiative de développer un système de reconnaissance d'objets en utilisant le Deep Learning qui est l'objet de ce projet de fin d'études.

En fait, il s'agit du développement et de l'implémentation d'un système de reconnaissance d'objets en utilisant le Deep Learning.

Ce document comporte quatre chapitres principaux :

-La reconnaissance des objets : dans ce chapitre nous donnons l'état de l'art de la reconnaissance pour découvrir les techniques existantes, nous intéressons aux deux méthodes les plus utilisées dans ce domaine à savoir : la méthode Surf dédiée pour la reconnaissance et la méthode de Viola et Jones pour la détection.

-Le développement de l'application Raynov Learning en utilisant les réseaux de neurones artificiels : Dans ce chapitre on abordera les démarches suivies durant le développement de cette application, et on finira par tester l'application sur des bases d'images.

- Le développement de l'application Raynov Deep Learning: au début de ce chapitre nous donnons des généralités sur la méthode de Deep Learning puis nous explicitons la méthodologie que nous avons suivie pour appliquer cette méthode pour la reconnaissance des objets, des émotions ..., enfin nous testons notre application pour vérifier son fonctionnement.

-Le développement et l'implémentation d'un système de reconnaissance d'objets en utilisant le Deep Learning et le Raspberry Pi : dans ce chapitre nous développons et nous implémentons un système de reconnaissance d'objets basé sur le Deep Learning et le Raspberry Pi puis nous le testons pour la validation.

Chapitre1 : la reconnaissance des objets

I. Introduction

En vision par ordinateur, on désigne par la reconnaissance des objets (ou la classification d'objet) une méthode permettant de détecter la présence d'une instance ou d'une classe d'objets dans une image numérique. Une attention particulière est portée à la détection du visage et la détection de personnes. Ces méthodes font souvent appel à l'apprentissage supervisé [1] et ont des applications dans de multiples domaines tels que:

- La reconnaissance biométrique : la technologie biométrique utilise la physique ou le comportement humain pour reconnaître tout individu pour la sécurité et l'authentification [2]. La biométrie est l'identification d'un individu sur la base de caractéristiques biologiques telles que les empreintes digitales, la géométrie de la main, les motifs de la rétine et de l'iris, l'ADN, etc. Pour l'analyse biométrique, les techniques de reconnaissance d'objets peuvent être utilisées.
- La surveillance: les objets peuvent être suivis et reconnus pour différents systèmes de vidéosurveillance.
- L'inspection industrielle: les machines peuvent être reconnues en utilisant la reconnaissance d'objets et peuvent être surveillées pour un dysfonctionnement ou des dommages.
- La robotique: La recherche de robots autonomes est l'une des questions les plus importantes de ces dernières années. Par exemple dans le cas du robot humanoïde du football. Les robots joueurs comptent sur leurs systèmes de vision très fortement quand ils sont dans l'imprévisible et des environnements dynamiques. Le système de vision peut aider le robot à recueillir diverses informations sur son environnement (localisation de robot, tactique de robot, barrière évitant, etc) [3].
- L'analyse médicale: la détection des tumeurs dans les images IRM, la détection du cancer de la peau peut être quelques exemples de la reconnaissance d'objets.
- L'optique caractère / chiffre / reconnaissance de documents: les caractères dans les documents numérisés peuvent être reconnus par des techniques de reconnaissance.
- Les systèmes de véhicules intelligents: systèmes de véhicules intelligents sont nécessaires pour la détection de signe de trafic, ainsi que pour la détection de véhicules et de suivi.

II. L'état de l'art de la reconnaissance des objets :

L'approche générale pour détecter un objet dans une image (sans connaissance a priori des conditions de prise de vue) consiste en trois étapes :

- Déterminer les localisations potentielles de l'objet, soit au moyen d'une fenêtre glissante sur l'image, soit directement en détectant les points d'intérêt de l'image.
- Extraire des caractéristiques dans l'image permettant d'identifier la classe d'objets visée.
- Classifier automatiquement chaque sous fenêtre comme contenant l'objet recherché ou pas ou bien apparier les caractéristiques extraites avec celles d'un modèle de référence [4].

Plusieurs techniques de reconnaissance d'objets existent depuis des décennies. Mais, dans ce rapport on citera que celles les plus utilisées, comme :

1. Le Modèle de correspondance (Template matching) :

Le modèle de correspondance est une technique simple permettant de trouver de petites pièces dans une image qui correspond à une image modèle. Dans cette technique, les images modèles de différents objets sont stockées dans un espace. Lorsqu'une image est mise à l'entrée du système, elle est comparée avec tous les modèles stockés dans l'espace, pour déterminer ou identifier la classe d'appartenance de l'image d'entrée.

Les modèles sont souvent utilisés pour la reconnaissance de caractères, des chiffres, des objets, etc. Les images modèles peuvent être des images en couleur afin de profiter des informations de couleur de l'objet à détecter ou des images en niveaux de gris.

Le Modèle de correspondance peut être basé sur la correspondance pixel à pixel ou la correspondance entre les caractéristiques de l'image. Les caractéristiques des images du modèle sont comparées aux caractéristiques de l'image d'entrée, pour déterminer si l'objet de modèle est présent dans l'image d'entrée [5].

La figure 1, illustre le résultat de la reconnaissance d'une rose dans la robe d'une petite fille.

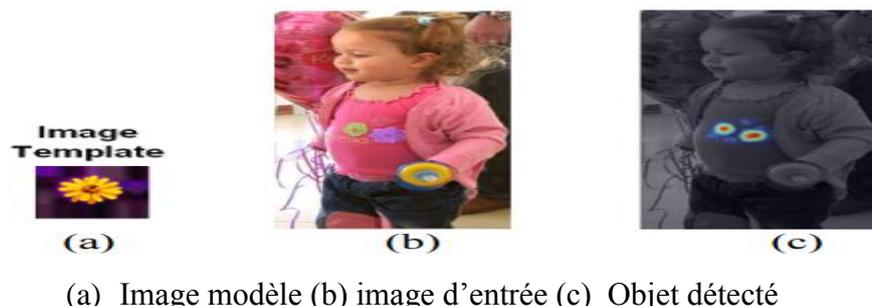


Figure 1 : Le résultat de la reconnaissance d'une rose en utilisant le modèle de correspondance

2. Reconnaissance par couleurs :

Dans ce cas, la classification des objets se fait en utilisant les histogrammes de couleur. Surtout, lorsqu'on cherche à détecter un objet qui se caractérise par une couleur unique dans l'environnement.

Dans des travaux antérieurs [6], les auteurs ont proposées l'utilisation des attributs de couleur comme une représentation de couleur explicite pour la détection d'objet. À titre d'exemple la figure 2 illustre une bonne détection des membres de la famille Simpson en utilisant les couleurs ainsi que les formes.



Figure 2 : la reconnaissance des membres de la famille Simpsons en utilisant les couleurs

3. Reconnaissance par formes :

La reconnaissance de formes (ou parfois reconnaissance de motifs) est un ensemble de techniques et méthodes visant à identifier des motifs informatiques à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif. On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et statistiques [7].

Le mot forme est à comprendre dans un sens très général, pas seulement celui de « forme géométrique » mais plutôt de motifs qui peuvent être de natures très variées. Il peut s'agir de contenu visuel (code barre, visage, empreinte digitale...) ou sonore (reconnaissance de parole), d'images médicales (rayon X, EEG, IRM...) ou multi spectrales (images satellitaires) et bien d'autres.

Par ailleurs, on peut reconnaître les objets ayant des formes particulières.

La reconnaissance de motifs peut être effectuée au moyen de divers algorithmes d'apprentissage automatique tels que :

- les réseaux de neurones
- l'analyse statistique
- l'utilisation de modèles de Markov cachés

- la recherche d'isomorphisme de graphes ou sous-graphes

Les formes recherchées peuvent être des formes géométriques, descriptibles par une formule mathématique, telles que :

- un cercle ou ellipse
- une droite

Elles peuvent aussi être de nature plus complexe :

- Une lettre
- Un chiffre
- Une empreinte digitale

4. La Reconnaissance de plusieurs objets dans une image :

Une seule image peut être constituée d'un ou plusieurs objets [2]. Dans le cas où on désire détecter plusieurs objets dans une même image, on peut utiliser le procédé de reconnaissance multi objets représenté sur la figure 3 ci- dessous.

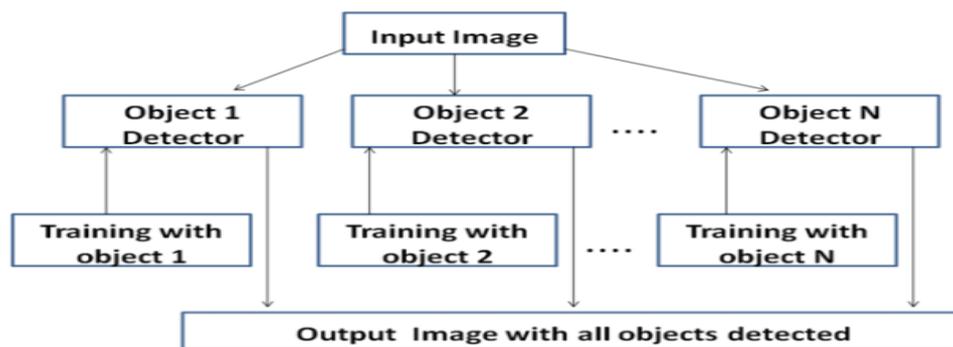


Figure 3 : La méthode de reconnaissance de multi objets dans une image

III. La détection des Objets par la méthode de Viola et Jones :

C'est une méthode efficace de détection d'objets qui a été proposée par Paul Viola et Michael Jones [8]. C'est une méthode basée sur l'apprentissage de la machine où la fonction cascade est générée à partir d'un grand nombre d'images positives et négatives. La fonction cascade est ensuite utilisée pour détecter des objets pertinents dans d'autres images.

1. L'algorithme de Viola et Michael Jones :

Les caractéristiques pseudo-Haar sont décrites pour la première fois dans un article de Paul Viola et Michael Jones paru en 2001 dans la revue scientifique « International Journal of Computer Vision », dans laquelle ils décrivent une nouvelle méthode de détection du visage.

Il est possible de calculer très rapidement les caractéristiques pseudo-Haar à l'aide des images intégrales. Une image intégrale est une table de correspondance 2D, construite à partir

de l'image d'origine, elle est de même taille que l'image d'entrée. Elle contient en chacun de ses points la somme des pixels situés au-dessus et à gauche du pixel courant.

Une fois le calcul de l'image intégrale effectué, la somme des pixels dans n'importe quel rectangle situé dans l'image peut se calculer en seulement 3 opérations et 4 accès à l'image intégrale. Une caractéristique pseudo-Haar à deux rectangles peut alors être déterminée en seulement 6 accès (2 points sont partagés) à l'image, et une caractéristique à 3 rectangles en seulement 8 accès.

Lienhart et Maydt [9] ont introduit en 2002 le concept de caractéristiques pseudo-Haar orientées à 45° afin de mieux prendre en compte dans une image les structures inclinées (par exemple un contour à 45°, voir figure 4). Ils ont montré que ce concept apportait une amélioration de près de 10 % des performances par rapport à l'ensemble de caractéristiques utilisées par Viola et Jones. Ils ont également proposé une extension des images intégrales pour permettre le calcul rapide de ces caractéristiques à 45°.

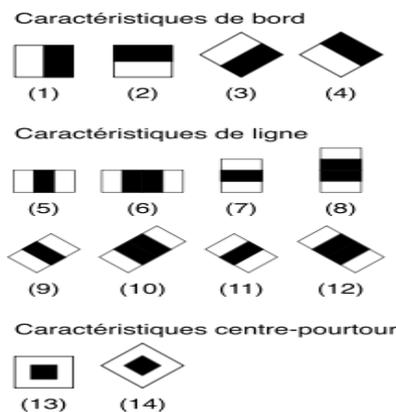


Figure 4 : l'extension du caractéristique pseudo haar par Lienhart et Maydt

2. Application de la détection faciale en utilisant L'Opencv :

Nous avons appliqué la méthode de Viola et Jones pour la détection faciale, cette détection a été effectuée en deux étapes :

Dans la première étape, il s'agit de l'apprentissage et la génération du fichier xml : Cette étape est indispensable car, la machine doit apprendre à différencier entre les visages et les autres objets, pour cela on a utilisé deux bases : une base des images qui contiennent des visages (images positives), et une autre base qui contiennent des objets (qui ne sont pas des visages (des images négatives)).

Les images subissent un processus d'apprentissage, la méthodologie de la génération du fichier XML, a été illustré dans un fichier [10].

La détection des visages : L'opencv intègre des fonctions prédéfinies qui permettent d'exploiter le fichier xml qui a été généré dans l'étape précédente. Dans un premier temps

l'algorithme extrait les caractéristiques des visages, puis il les compare avec les caractéristiques incluses dans le fichier Xml afin de déterminer si l'image contient un visage ou non.

On peut travailler aussi sur des vidéo en temps réel, la vidéo est une séquence d'images, il suffit de travailler sur chaque image pour détecter la présence d'un visage.

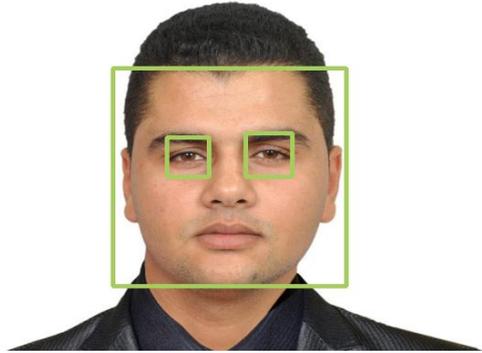


Figure 5 : Le résultat de détection de visage en utilisant le Haar Features

La méthode de Viola et Jones est une méthode de détection des objets très fiable et rapide, ses applications sont énormes et on les retrouve dans notre vie quotidienne notamment dans les Smartphones.

IV. La reconnaissance des objets par la méthode de SURF :

1. La méthode de SURF :

Speeded Up Robust Features (SURF), que l'on peut traduire par caractéristiques robustes accélérées, est un algorithme de détection de caractéristiques et un descripteur, présenté par des chercheurs de l'ETH Zurich et de l'université catholique de Leuven pour la première fois en 2006 [11], puis dans une version révisée en 2008[12]. Cet algorithme est utilisé dans le domaine de vision par ordinateur, pour des tâches de reconnaissance d'objet [13].

L'algorithme SURF (Speeded Up Robust Feature) est l'algorithme le plus puissant existant et parmi les algorithmes de détection d'objet dans une scène par détection de concordances de points (Feature Description). Le taux de détection atteint 85% de réussite avec un bon paramétrage. Parmi ses avantages on peut citer les suivants :

- L'algorithme est annoncé invariant d'échelle et de rotation.
- Sa simplicité et sa rapidité au niveau d'extraction des caractéristiques.
- Suivi/reconnaissance d'objet sur un flux vidéo continu issu d'une webcam

La mise en œuvre de l'algorithme SURF passe par plusieurs étapes :

- La détection de points-clés dans l'image objet et dans l'image scène.
- Le calcul d'un descripteur pour chaque point-clé de l'image objet et de l'image scène.

- La détection des concordances entre les descripteurs des points-clé de l'image objet et de l'image scène. Chaque concordance (appelée DMatch) se caractérise par l'index du point objet.

La figure 7 illustre ses étapes de fonctionnement :

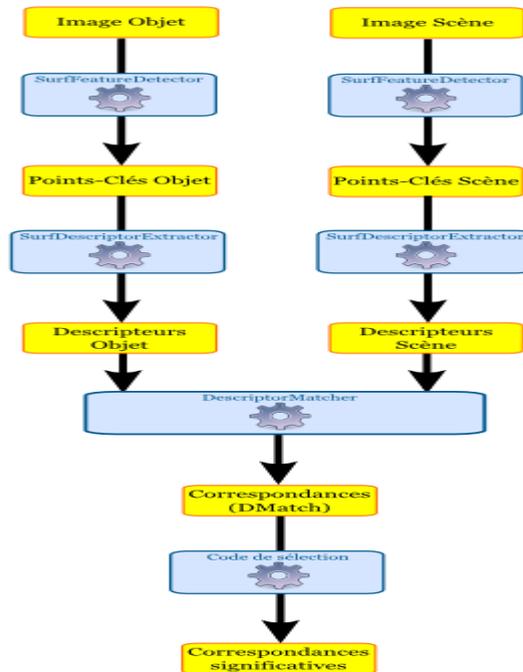


Figure 6 : Fonctionnement de la méthode surf

2. Application de La méthode de SURF en utilisant Opencv:

A titre d'exemple de l'utilisation de l'algorithme SURF, la distinction du logo de la fst entre deux logos (voir figure 7). La première étape, consiste à extraire les points d'intérêt des deux logos puis calculer le descripteur pour chaque point.

Finalement, il reste à déterminer s'il y a une correspondance entre les deux images.

Pour se faire, on se base sur les descripteurs des points d'intérêt qu'on a déjà calculés, finalement l'algorithme recadre le logo de la FST pour différencier les deux logos et pour assurer la reconnaissance.

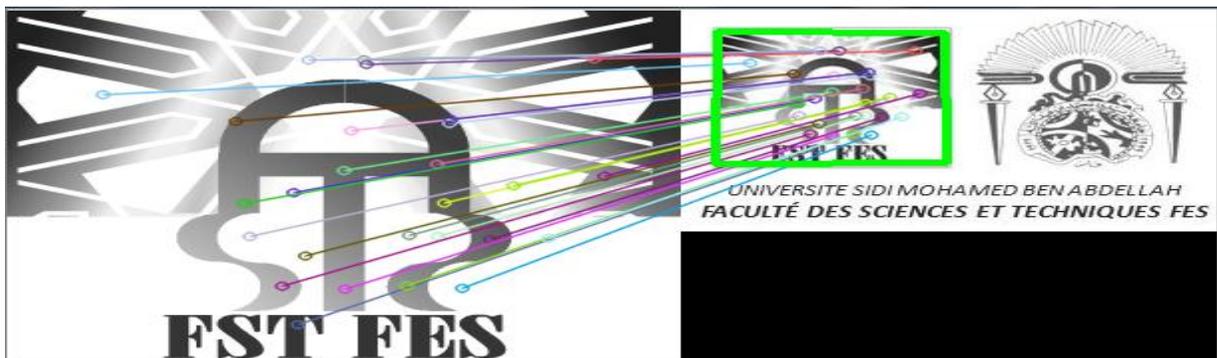


Figure 7 : le résultat de la reconnaissance de logo de la FST en utilisant la méthode SURF

V. Conclusion

Dans ce chapitre nous avons donné l'état d'art de la reconnaissance des objets pour découvrir les techniques existantes, puis nous avons abordé la méthode de Viola et Jones et celle de SURF. Ensuite nous avons utilisé ces deux méthodes pour la détection du visage ainsi que pour la reconnaissance du logo de la FST.

*Chapitre2 : Développement de l'application Raynov
Learning en utilisant les Réseaux de neurones artificiels*

I. Introduction

L'idée de ce chapitre est de développer une application qui exploitera la méthode de Viola et Jones et la méthode de SURF, pour pouvoir reconnaître des objets.

En vision par ordinateur, les chercheurs utilisent des méthodes qui permettent à la machine l'apprentissage de son environnement afin d'être capable de décider toute seule, ces méthodes sont basées sur les réseaux de neurones artificiels. L'idée des réseaux de neurones est inspirée de la biologie et du système nerveux humain.

Nous commençons ce chapitre par une présentation succincte des réseaux de neurones pour comprendre leur principe.

Ensuite, nous développons en langage C++ une application de reconnaissance d'objets et du visage, que nous avons nommée « Raynov Learning », avec une interface graphique pour faciliter son utilisation. Puis nous donnons la méthodologie suivie au cours du développement, finalement on teste cette application sur des bases d'images.

II. Les Réseaux de neurones artificiels :

Les ordinateurs ont des unités de traitement et de mémoire. Ils permettent à l'ordinateur d'effectuer le calcul numérique le plus complexe durant un temps très court.

Si on compare l'ordinateur et le cerveau, on notera que, théoriquement, l'ordinateur devrait être plus puissant que notre cerveau: Il comprend 10^9 transistors avec un temps de commutation de 10^{-9} secondes. Le cerveau contient 10^{11} neurones, mais ceux-ci ont seulement un temps de commutation d'environ 10^{-3} secondes.

La plus grande partie du cerveau travaille en permanence, tandis que la plus grande partie de l'ordinateur est réservée seulement pour le stockage de données passives. En revanche Un ordinateur est statique, le cerveau comme un réseau neuronal biologique peut se réorganiser au cours de sa durée de vie, et est donc en mesure d'apprendre, pour compenser les erreurs et ainsi de suite [14].

En fait la caractéristique que le cerveau humain peut se réorganiser, est la clé derrière sa capacité d'apprentissage qu'on ne trouve pas dans les ordinateurs. Le but de l'utilisation des réseaux de neurones est de s'inspirer de la biologie pour établir un système artificiel capable d'apprendre et de compenser ses erreurs.

Dans le tableau 1 on donne une comparaison entre les capacités du cerveau humain et l'ordinateur.

	Le cerveau	L'ordinateur
Nombre des unités de traitements	10^{11}	10^9
Type des unités de traitement	Neurones	Transistors
Type de calcul	En parallèle	En série
Stockage de données	Associative	Les adresses basées
Temps de commutation	10^{-3} s	10^{-9} s
les opérations possibles pendant la commutation	10^{13} par s	10^{18} par s
Les opérations réelles pendant la commutation	10^{12} par s	10^{10} par s

Tableau 1 : une comparaison entre le cerveau et l'ordinateur

1. Concept :

Un réseau de neurones artificiels, où réseau neuronal artificiel, est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques [15]. Ces réseaux ont été développés pour résoudre des problèmes:

- de contrôle.
- de reconnaissance de formes.
- de décision.
- de mémorisation.

2. Historique :

Le domaine des réseaux de neurones , comme dans tous les autres domaines de la science, a connu une longue histoire de développement, il a apparu pour la première fois en 1943 ,lorsque Les neurologues Warren McCulloch et Walter Pitts [16], ont créé un modèle des réseaux de neurones, simple et capable de calculer toutes arithmétiques et logiques fonctions

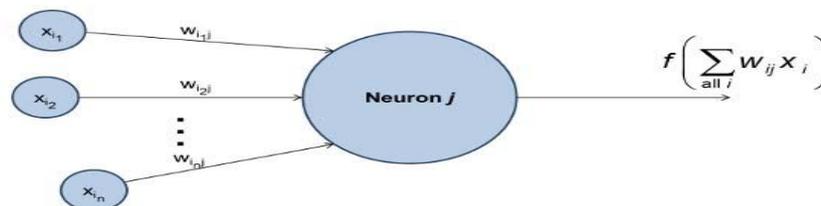


Figure 8 : Modèle simplifié d'un réseau de neurones artificiels

✓ En 1949 :

Donald Hebb a formulé la loi de Hebb classique, qui représente la base de toutes les procédures d'apprentissage neural. La règle implique que la connexion entre deux neurones est renforcée lorsque les deux neurones sont actifs en même temps [17].

✓ En 1957-1958 :

Au laboratoire MIT, Frank Rosenblatt et ses collègues. Ont développé le premier neurocomputer, Mark Perceptron, réussi qui était capable de reconnaître des chiffres simples au moyen d'un capteur d'image 20 x 20 pixels.

✓ En 1969 :

Marvin Minsky et Seymour ont fait un analyse de Perceptron, pour montrer que le modèle de perceptron n'a pas été capable de résoudre de nombreux problèmes importants tel que le problème XOR et la séparabilité non linéaire [18].

✓ En 1986 :

La rétropropagation de l'erreur d'apprentissage a été séparément développé et largement publié par le Parallel Distributed Processing Group. Les problèmes non linéairement séparables pourraient être résolus par perceptrons multicouches. Ce résultat a été vu comme une renaissance dans ce domaine [19].

Depuis 1943 jusqu'à nos jours ce domaine connaît une évolution rapide car beaucoup des chercheurs travaillent le dessus.

3. Perceptron multicouche :

Le perceptron multicouche (PMC) est la deuxième grande famille de réseaux de neurones. Après qu'on va décrire l'architecture de ces réseaux on va aborder leur principe, et le concept de rétropropagation de l'erreur.

a. Architecture :

Un neurone de perceptron réalise un produit scalaire entre son vecteur d'entrées et un vecteur de paramètres appelé poids, à chaque produit on y ajoute un biais et on utilise une fonction d'activation pour déterminer sa sortie :

$$(1) \mathbf{y} = \mathbf{f}(\mathbf{x} * \mathbf{w} + \mathbf{b})$$

Il existe trois types de fonctions d'activation. Mais, seulement deux fonctions d'activation sont souvent utilisées dans la littérature. ses fonctions sont la tangente hyperbolique $f_1()$ et la sigmoïde standard $f_2()$:

$$(2) f_1 = \mathbf{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$(3) f_2 = \frac{\mathbf{tanh}(x)+1}{2}$$

La différence entre ces deux dernières fonctions est le domaine des valeurs de la sortie, qui varient entre $[-1: +1]$ pour la tangente hyperbolique et de $[0: 1]$ pour la sigmoïde standard. La figure 2.12 montre la courbe de la tangente hyperbolique :

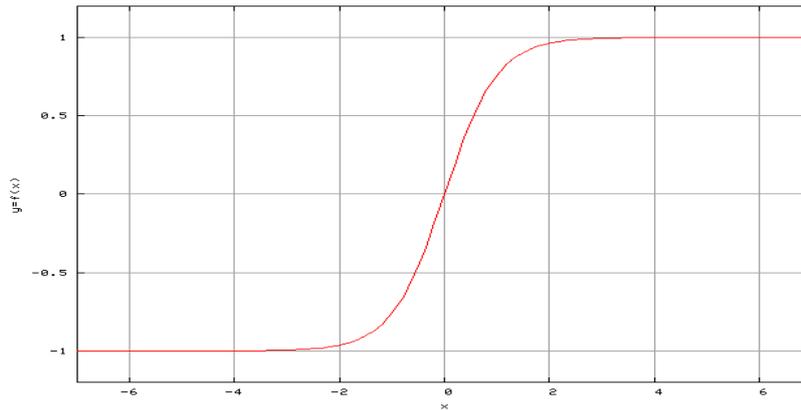


Figure 9 : la tangente hyperbolique

Le perceptron est organisé en plusieurs couches : la première couche est reliée aux entrées, puis ensuite chaque couche est reliée à la couche précédente. C'est la dernière couche qui produit les sorties du PMC. Les sorties des autres couches ne sont pas visibles à l'extérieur du réseau, et elles sont appelées pour cette raison couches cachées.

Dans la figure 10 on donne l'architecture de perceptron multicouche.

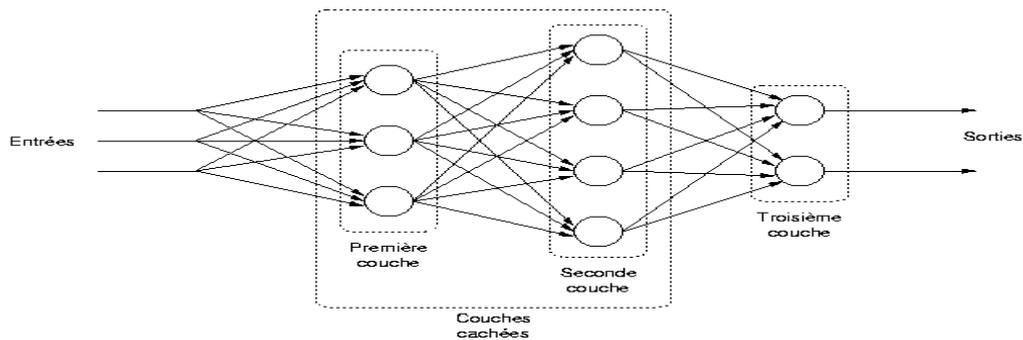


Figure 10 : Perceptron multicouche

b. La Rétropropagation du gradient :

La technique de rétropropagation du gradient (*backpropagation* en anglais) est une méthode qui permet de calculer l'erreur du gradient pour chaque neurone, de la dernière couche vers la première. De façon abusive, on appelle souvent technique de rétropropagation du gradient, l'algorithme classique de correction des erreurs basé sur le calcul du gradient.

Cette technique consiste à corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs. Dans le cas des réseaux de neurones, les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur petite [20].

Les poids dans le réseau de neurones sont aux préalables initialisés avec des valeurs aléatoires. On considère ensuite un ensemble de données qui va servir à l'apprentissage. Chaque échantillon possède ses valeurs cibles qui sont celles que le réseau de neurones doit à

terme prédire lorsqu'on lui présente le même échantillon. L'algorithme se présente comme ceci :

Soit un échantillon \vec{x} que l'on met à l'entrée du réseau de neurones et la sortie recherchée pour cet échantillon \vec{t} .

- ❖ On propage le signal en avant dans les couches du réseau de neurones : $x_k^{n-1} \rightarrow x_j^n$
- ❖ La propagation vers l'avant se calcule à l'aide de la fonction d'activation $g()$, w_{jk} indique bien un poids de k vers j

$$(4) x_j^n = g^n(\sum_k (w_{jk}^n * x_k^{n-1}))$$

- ❖ Lorsque la propagation vers l'avant est terminée, on obtient à la sortie le résultat \vec{y}
- ❖ On calcule alors l'erreur entre la sortie \vec{y} donnée par le réseau et le vecteur \vec{t} désiré à la sortie pour cet échantillon. Pour chaque neurone i dans la couche de sortie, on calcule :

$$(5) e_i^{sortie} = g'(\sum_k w_{jk}^n * x_k^{n-1}) [t_i - y_i]$$

- ❖ On propage l'erreur vers l'arrière $e_i^n \rightarrow e_j^{n-1}$ grâce à la formule suivante :

$$(6) e_i^{n-1} = g'^{(n-1)}(\sum_k w_{jk}^{n-1} * x_k^{n-2}) \sum_i w_{ij} e_i^n$$

- ❖ On met à jour les poids dans toutes les couches :

$$(7) \Delta w_{ij}^n = \Delta w_{ij}^n + \gamma * e_i^n x_j^{n-1}$$

Où γ représente le taux d'apprentissage (de faible magnitude et inférieur à 1.0)

Pour éviter les problèmes liés à une stabilisation dans un minimum local voir figure, on ajoute un terme d'inertie (momentum). Celui-ci permet de sortir des minimums locaux dans la mesure de poursuivre la descente de la fonction d'erreur. À chaque itération, l'algorithme mémorise les informations du changement de poids. Cet effet de mémoire permet d'éviter les oscillations et accélère l'optimisation du réseau. Alors, la nouvelle formule de modification des poids se traduit par :

$$(8) \Delta w_{ij}^n(t) = \gamma e_i^n x_j^{n-1} + \alpha \Delta w_{ij}^n(t-1)$$

Avec α un paramètre compris entre 0 et 1.0

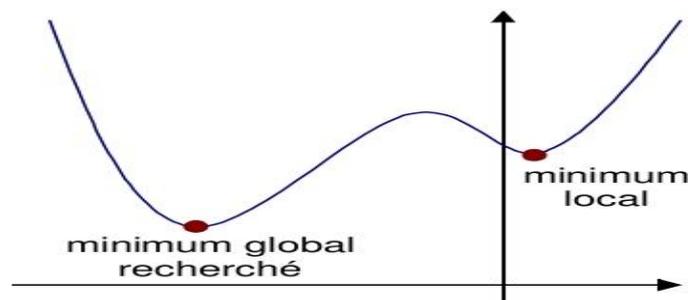


Figure 11 : La descente du gradient : Minimum global et local

III. Environnement de développement :

Cette partie est dédiée au développement de l'application Raynov Learning pour se faire nous avons utilisé des ressources hardware et software dont les caractéristiques sont les suivantes :

1. Environnement hardware :

Nous avons utilisé un Ordinateur Sony Vaio VPC-F22S1E/B qui a la configuration ci-dessous :

Configuration matériel : Processor Intel Core i7-2630QM 2.8 GHz

Carte graphique : NVIDIA GeForce GT 540M - 1024 MB, Core: 672 MHz, Memory: 900 MHz, DDR3, ForceWare 267.80

Mémoire : 8 giga Ram

2. Environnement software :

Pour la partie logicielle nous avons utilisé les logiciels suivants :

Framework Qt : est un Framework qui permet le développement en C++ [21], il permet aussi de créer des interfaces graphiques, il est donc constitué d'un ensemble de bibliothèques, appelées « modules ».

Ubuntu 14.04 : Ubuntu 14.04 LTS c'est un système d'exploitation linux, c'est la vingtième version d'Ubuntu. Son développement a débuté en octobre 2013. La version stable, finale d'Ubuntu 14.04.1 LTS (long terme support = support disponible pendant une longue durée).elle a été sortie le 17 avril 2014 et sera supportée jusqu'en avril 2019

La bibliothèque OPENCV :Opencv est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est développée en C et C++ proposée par Intel et peut-être utilisée sous Linux, Windows et Mac OS [22].

Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage. Opencv est orienté vers des applications en temps réel.

Un des buts d'Opencv est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur.

La bibliothèque d'Opencv contient près de 500 fonctions. Il est possible grâce à la licence de code ouvert de réaliser un produit commercial en utilisant tout ou partie d'Opencv.

IV. Méthodologie de développement :

Pour développer notre application de reconnaissance d'objets et des visages par réseaux de neurones artificiels, nous avons suivi les trois étapes suivantes

- La détection faciale par la méthode de viola et Jones (dans le cas des visages seulement) et l'extraction des caractéristiques par la méthode de SURF

- L'apprentissage des réseaux de neurones
- La reconnaissance par réseaux de neurones

i. La détection faciale par la méthode de Viola et Jones et l'extraction des caractéristiques par la méthode de SURF :

Dans l'application Raynov Learning on utilise la méthode SURF pour l'extraction des caractéristiques, ainsi qu'un perceptron avec une seule couche cachée. Le nombre de neurones à l'entrée du perceptron est égale à 225, en effet, nous avons fixé la taille du descripteur des points d'intérêts extrais de l'objet par la méthode SURF à 225. Les valeurs d'entrée du réseau varient entre 0 et 1, car la méthode surf génère automatiquement des caractéristiques de valeur inférieure à 1. Le nombre des neurones dans la couche de sortie est égale au nombre de classe d'objets.

Dans le cas du visage nous avons utilisé la méthode de Viola et Jones pour la détection faciale. En fait, la détection du visage avant l'extraction des caractéristiques augmente le taux de reconnaissance.

ii. L'apprentissage des réseaux de neurones :

Durant l'apprentissage l'algorithme utilise une base d'images dédiée pour l'apprentissage, chaque catégorie d'image doit contenir des images du même objet (ou la même personne dans le cas des visages). L'algorithme traite chaque image à la fois, l'image de chaque catégorie active un neurone de la couche de sortie, les poids et les biais au début de chaque processus d'apprentissage vont être initialisés par des valeurs aléatoires entre -1 et 1. Cependant on utilise la tangente hyperbolique comme une fonction d'activation car elle a une marge de sortie entre -1 et 1, par contre le sigmoïde entre 0 et 1. Par la suite les valeurs seront réglées en utilisant la rétropropagation du gradient et si l'erreur atteint la valeur minimale fixée par l'utilisateur, l'algorithme enregistrera les poids et les biais dans un fichier. Cette phase d'apprentissage se fait seulement une fois.

iii. La reconnaissance par les réseaux de neurones :

Durant la phase de la reconnaissance, l'utilisateur peut exploiter les données de la phase d'apprentissage pour effectuer la reconnaissance. Sur l'interface graphique développée l'utilisateur doit choisir l'objet de classification. Si l'objet à reconnaître est un visage ou une personne, l'algorithme fait appel à la méthode de Viola et Jones afin de chercher le visage sur l'image. Ensuite, il extrait les points d'intérêts qui vont être classifiés via les réseaux de neurones entraînés.

Pour les objets l'algorithme extrait directement les points d'intérêts sans utiliser la méthode de Viola et Jones.

V. Description de l'application :

L'application Raynov Learning est une application de la reconnaissance des objets et des visages qui se base sur les réseaux de neurones artificiels, l'interface d'accueil de l'application est illustrée dans la figure 12 ci-dessous.

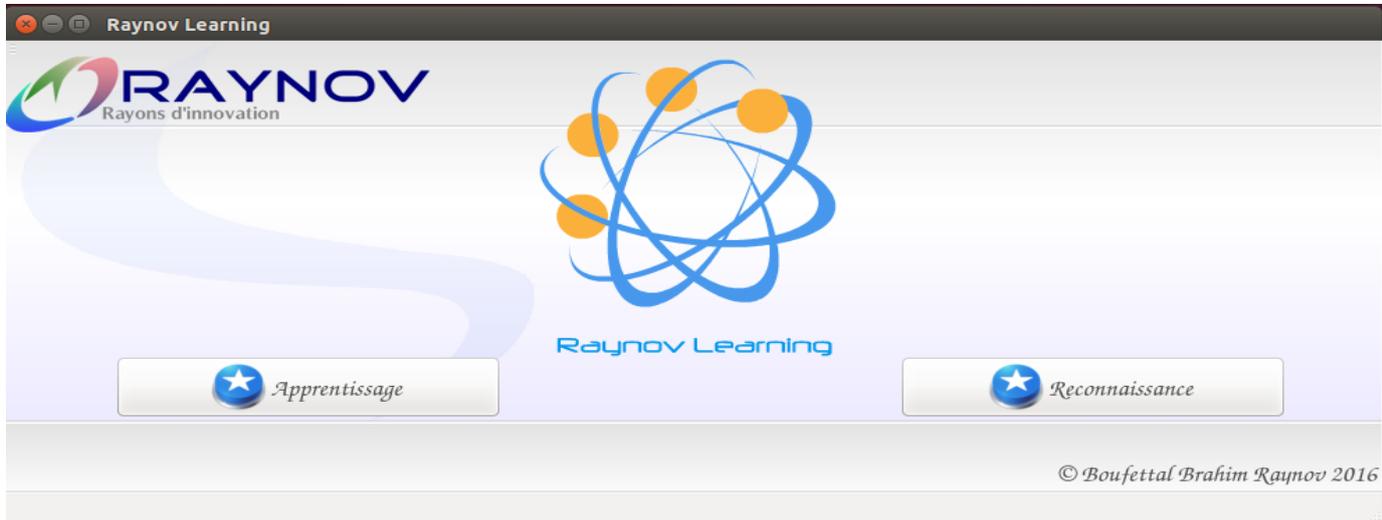


Figure 12: Interface d'accueil de l'application Raynov Learning

L'utilisateur a deux choix: le premier c'est l'apprentissage et le deuxième c'est la reconnaissance, la première étape c'est d'effectuer une phase d'apprentissage, en cliquant sur Apprentissage.

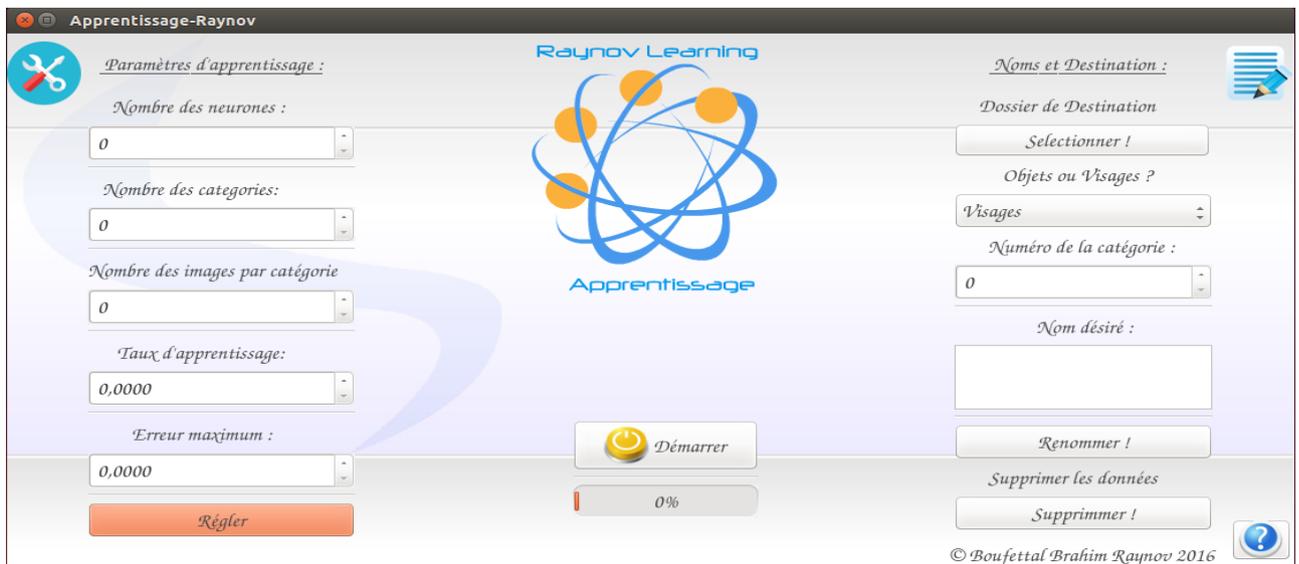


Figure 13: Interface graphique de l'apprentissage (Raynov Learning)

Cette interface facilite la phase d'apprentissage à l'utilisateur. Il suffit de sélectionner un dossier de destination pour y stocker les bases et les résultats d'Apprentissage, l'utilisateur doit aussi choisir si l'objet à reconnaître est un visage ou un objet. Après il faut fixer les

paramètres d'apprentissages : le nombre de neurones de la couche cachée, le nombre de catégories, le nombre des images par catégorie, le taux d'apprentissage et l'erreur minimum. Après un clic sur régler, des dossiers seront créés pour chaque catégorie. Ensuite on donne à chaque catégorie le nom correspond de l'objet à reconnaître. Après avoir démarré l'application, cette dernière commence l'apprentissage du réseau de neurones en visualisant l'état d'avancement de l'apprentissage. Quand l'apprentissage se termine les poids de chaque catégorie ou objet sont stockés dans un fichier « .txt » afin d'être utilisés durant la phase de reconnaissance. Une icône de help a été développée afin de visualiser le mode d'emploi de l'application.



Figure 14 : La fenêtre d'aide (Raynov Learning)

A la fin de l'apprentissage, l'utilisateur rentre dans l'interface d'accueil, et ouvre la fenêtre de reconnaissance en cliquant sur le bouton Reconnaissance.



Figure 15 :L'interface graphique de la Reconnaissance (Raynov Learning)

Pour pouvoir déterminer la classe d'appartenance de chaque objet, l'utilisateur peut choisir ou télécharger une image. Avec un simple clic sur le bouton classifieur, l'application identifie ou reconnaît efficacement la catégorie de l'objet choisi.

VI. Test de fonctionnement :

Après avoir donné une description du mode d'emploi de l'application, nous testons son fonctionnement. Tout d'abord nous commençons par la reconnaissance des visages, pour se faire on utilise deux bases d'images des deux célèbres joueurs de football qui sont Ronaldo et Messi. Pour chaque joueur nous avons téléchargé dix images sous différentes conditions comme le montre la figure 16.



Figure 16 : Bases des images pour le test de fonctionnement des visages (Raynov Learning)

On fixe les paramètres d'apprentissage tels que le nombre des neurones, le nombre de catégories, le nombre d'images par catégorie, le taux d'apprentissage ainsi que l'erreur minimum.



Figure 17: Test d'apprentissage pour les visages (Raynov Learning)

Puis on attribue à chaque catégorie un nom significatif.



Figure 18 : Renommer une catégorie (Raynov learning)

Afin de vérifier si l'application peut reconnaître les deux joueurs, on a utilisé une image qui contient les visages de Messi et Ronaldo qui n'appartiennent pas aux bases d'images utilisées durant la phase d'apprentissage.



Figure 19: Résultat de la reconnaissance du visage

D'après la figure 19, on peut conclure que le test est réussi puisque les deux joueurs ont été identifiés même si nous avons utilisé de petites bases d'images.

VII. Conclusion

L'application Raynov Learning a été développée en utilisant les réseaux de neurones artificiels. Elle est dédiée à la reconnaissance des objets et des visages.

L'application se base essentiellement sur la méthode SURF qui permet l'extraction des caractéristiques ou les points d'intérêts. Ses caractéristiques sont ensuite considérées comme des entrées des réseaux de neurones. À l'avenir pour améliorer les performances de l'application Raynov Learning, il faut penser à ajouter la possibilité de la reconnaissance en temps réel. Aussi pour généraliser l'application, il faut pouvoir identifier d'autres classes comme les émotions par exemple.

*Chapitre3 : Développement de l'application Raynov
Deep Learning*

I. Introduction

Le Deep Learning ou l'apprentissage profond est une branche de l'intelligence artificielle et du Big Data, ce domaine connaît un immense développement notamment ces dernières années. Les grandes sociétés comme Google, Facebook, Microsoft et amazon, travaillent sur ce sujet.

Notre mission consiste à développer une application capable de reconnaître les objets, les émotions, l'âge et le genre, les places publiques, et aussi les gens célèbres.

Dans ce chapitre nous donnons des généralités sur la technologie Deep Learning, puis nous présentons la méthodologie de développement de l'application que nous proposons. Finalement, nous testons notre application pour la validation.

II. Le Deep Learning :

1. Concept et techniques:

Le Deep Learning ou l'apprentissage profond est un ensemble de méthodes d'apprentissage automatique tentant de modéliser, avec un haut niveau d'abstraction, des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur ainsi que du traitement automatisé du langage.

Dans les années 2000, ces progrès ont suscité des investissements privés, universitaires et publics importants, notamment de la part du GAFA (Google, Apple, Facebook, Amazon) [23].

En 1989, le chercheur français Yan LeCun a développé un système de reconnaissance automatique de chèques pour les banques basé sur des algorithmes de Deep Learning. Malgré un franc succès, l'application qui nécessitait trois jours de traitement prenait trop de temps. C'était tout le problème, même avec la bonne technique, le Deep Learning ne permettait pas d'obtenir des résultats satisfaisants dans un temps raisonnable.

À partir de 2012, l'accès à de quantités de données de plus en plus importantes, les nouveaux algorithmes issus de l'intelligence artificielle, et l'utilisation de GPU (cartes graphiques du type Titan X de Nvidia) pour effectuer du calcul très intensif ont complètement changé la donne. Il y a eu vraiment un saut technologique très important qui a transformé les réseaux de neurones en de véritables machines à reconnaître.

Les techniques de Deep Learning constituent une classe d'algorithmes d'apprentissage automatique dont les caractéristiques sont les suivantes :

- Elles utilisent différentes couches d'unité de traitement non linéaire pour l'extraction et la transformation des caractéristiques. Chaque couche prend en entrée la sortie de la précédente. Les algorithmes peuvent être supervisés ou non supervisés et leurs applications comprennent la reconnaissance de modèles ou la classification statistique.
- Elles fonctionnent avec un apprentissage à plusieurs niveaux de détails ou de représentations des données. A travers les différentes couches on passe de paramètres de bas niveau à des paramètres de plus haut niveau.
- Ces différents niveaux correspondent à différents niveaux d'abstraction des données.
- Ce nouveau champ d'étude a pour objectif d'avancer davantage vers les capacités d'intelligence artificielle. Ses architectures permettent aujourd'hui de donner du sens à des données sous forme d'image, de son ou de texte.

Par ailleurs la figure 20 illustre un système de Deep Learning pour la reconnaissance du visage. Ce système est basé sur les réseaux de neurones artificiels, qui est constitué d'un ensemble de couches cachées, le mot profond (apprentissage profond) vient de ce nombre des couches cachées, on peut trouver des systèmes qui contiennent un nombre important de couches et aussi de neurones. Ce type d'algorithmes nécessite une capacité de calcul importante, pour cela il est nécessaire d'utiliser les GPU (processeur de la carte graphique) afin d'être capable d'exécuter des opérations complexes en un temps court.

La différence entre un perceptron classique (comme le cas de Raynov Learning), et un système de Deep Learning (Raynov Deep Learning) est que pour le premier les entrées des réseaux sont les caractéristiques de l'image, mais pour le deuxième ce sont les pixels bruts de l'image d'entrée. En effet dans un système de Deep Learning chaque couche est considérée comme un niveau d'abstraction de l'image.

Sur la figure 20, la couche 1 est capable d'extraire des caractéristiques de niveau d'abstraction inférieure à celui de la couche 2, tandis que la couche 3 a une qualité plus importante. À partir de de ces caractéristiques le système est capable de reconnaître les objets avec un taux d'erreur petit.

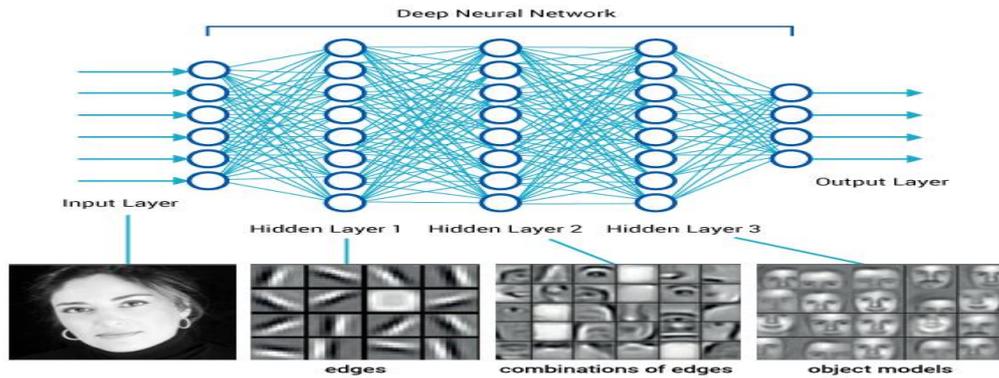


Figure 20: Exemple d'un système de Deep Learning

2. Les champs d'application :

Des millions de personnes profitent déjà des progrès réalisés grâce au Deep Learning. Parmi les champs d'action de cette technologie on trouve: le traitement d'image, de la vidéo, de la voix, et plus spécifiquement de la détection d'objets dans le domaine automobile (piétons, panneaux de signalisation, voitures, bus, marquage au sol...). De nombreux secteurs d'activité exploitent des programmes de reconnaissance d'image: les sites marchands pour classifier, identifier et trouver des similitudes entre des produits ou des familles de produits , la robotique pour permettre par exemple à un robot-aspirateur de se déplacer en toute autonomie, les réseaux sociaux et autres sites de partage de photos pour la reconnaissance faciale, les moteurs de recherche pour détecter les similitudes entre des images , l'industrie automobile pour le développement des voitures autonomes, la médecine pour la recherche de cellules cancéreuses, etc....[24].

a. DeepFace de Facebook :

Dévoilé au mois de juin 2014, le système de reconnaissance faciale DeepFace de Facebook se révèle très impressionnant. Le réseau social, qui possède la plus importante galerie photos en ligne du monde (250 milliards de photos !), a développé un réseau de neurones capable de reconnaître le visage d'une personne sur une photo avec un taux de réussite de 97,25%. Pour réussir un tel exploit, Facebook n'a pas fait les choses à moitié et s'est notamment offert les services du chercheur français Yan LeCun pour diriger son laboratoire d'intelligence artificielle basé à Paris. L'équipe de chercheurs a utilisé des techniques d'apprentissage profond et a traité 4 millions de photos représentant 4000 personnes issues de sa base de données [24]. Chaque visage a été modélisé en 3D pour pouvoir être positionné de face puis analysé à l'aide de 120 millions de paramètres variables. Une fois les données récoltées, le système effectue des comparaisons entre les photos pour trouver des correspondances (voir figure 21). Cette fonction a, depuis, été implémentée dans

le réseau social partout dans le monde, sauf en Europe où elle a été jugée trop intrusive par les défenseurs du droit à la vie privée.

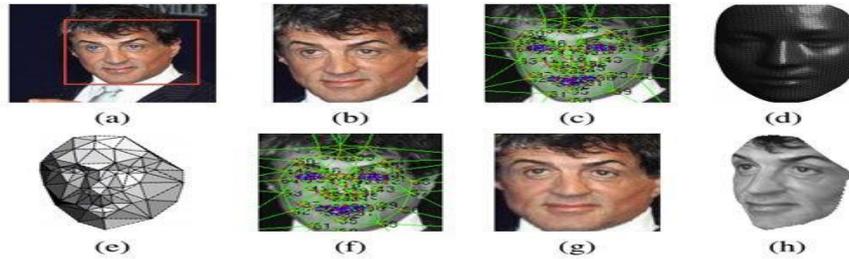


Figure 21: Principe de DeepFace (de Facebook)

b. Skype Translator de Microsoft :

La firme de Redmond a développé l'une des applications basées sur le Deep Learning et la machine Learning la plus impressionnante. Lancée en version d'essai au mois de décembre 2014, Skype Translator est un système de traduction à la volée de conversations. L'application traduit les appels audio et vidéo quasiment en temps réel dans six langues : anglais, espagnol, français, allemand, italien, et mandarin. Pour les messages écrits, Skype Translator prend en charge 50 langues. Pour l'heure, le système ne fonctionne pas sur les terminaux mobiles, mais uniquement sur PC. Les premiers utilisateurs ne tarissent pas d'éloges à son propos sur les réseaux sociaux [24].

c. NVIDIA Drive :

Outre les applications grand public précitées, il existe d'innombrables systèmes basés en partie sur le Deep Learning. Le fabricant NVIDIA est impliqué dans de nombreux secteurs d'activité, dont celui de l'automobile, pour lequel il fournit des outils d'aide à la conduite assistée et autonome. Le fondateur développe à la fois des SOC embarqués (NVIDIA Tegra K1 et X1) et une plate-forme logicielle dédiée NVIDIA Drive exploitée par de nombreux constructeurs: Audi, Tesla, Honda, etc.

Véritable cerveau de l'automobile, le Nvidia Drive PX est capable de gérer, non seulement les écrans de bord à l'intérieur de l'habitacle, mais aussi une douzaine de caméras. Comme on peut le voir sur la figure 22 ci-dessous, le système analyse et interprète en temps réel les données captées par les caméras : piétons, vélos, voitures, bus, camions, couleurs des feux de signalisation, sens de la route, etc. Grâce à des algorithmes de deep learning et la puissance du calcul embarquée, le NVIDIA Drive PX apprend et se perfectionne à mesure que le véhicule roule. Outre la reconnaissance des éléments extérieurs, NVIDIA développe également Drive CX pour la gestion de l'habitacle (traitement du langage parlé et assistance au conducteur), et Drive NX pour l'assistance à la conduite. Les premières démonstrations

réalisées à l'occasion du CES de Las Vegas en 2015 sur un véhicule Audi montre l'énorme potentiel du Deep Learning dans le domaine de l'automobile [24].

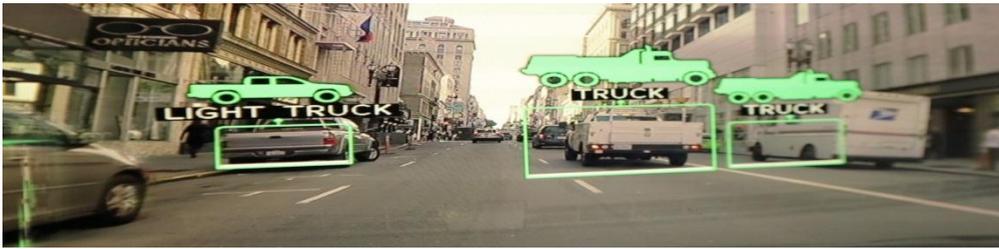


Figure 22: Nvidia Drive

d. DreamQuark :

Détecter des cellules cancéreuses ou un risque de cécité, voici les autres prouesses possibles grâce au Deep Learning. DreamQuark est une start-up française qui exploite cette technologie pour proposer des applications dans le domaine de la santé. Elle a développé notamment un outil permettant aux ophtalmologistes de prévenir les risques de rétinopathie chez leurs patients diabétiques. Cette maladie qui est la première cause de cécité chez les diabétiques peut être évitée si elle est détectée à temps. Grâce aux technologies du Deep Learning, la start-up a développé un programme pour les médecins capables de détecter les cellules défaillantes sur des photographies en haute définition de rétines. Les débuts de l'intelligence artificielle dans la médecine sont particulièrement prometteurs [24].

3. Le Framework Caffe :

Caffe est un Open source Framework d'apprentissage profond. Il est développé par le Berkeley Vision and Learning Center (BVLC) et par les contributeurs communautaires. Yangqing Jia a créé le projet au cours de son doctorat à l'Université de Berkeley [25].

Parmi les avantages de ce Framework, il y a :

- L'architecture expressive : encourage l'innovation, les modèles et l'optimisation sont définis dans la configuration, on peut basculer entre le GPU ou le CPU selon les besoins.
- Un code extensible favorisant le développement actif : Dans la première année de Caffe, il a été utilisé par plus de 1000 développeurs et a eu de nombreux changements importants.
- La rapidité fait de Caffe un outil parfait pour des expériences de recherche et pour le déploiement de l'industrie. Caffe peut traiter plus de 60 millions images par jour avec une seule carte graphique NVIDIA GPU K40. Le temps d'exécution est de 1 ms par image pour l'inférence et de 4 ms par image pour l'apprentissage. Ce qui fait de Caffe le plus rapide Framework disponible jusqu'à maintenant.

- **Communauté:** Caffe offre la possibilité aux chercheurs d'apporter des améliorations aux modèles pré entraînés, et puis d'intégrer ses modèles dans le site de Caffe pour les mettre à la disposition des utilisateurs.

En revanche Caffe offre un ensemble de modèles pré-entraînés. Dans ce travail nous avons utilisé les modèles suivants :

- **BVLC GoogLeNet :** GoogLeNet est un modèle pour la reconnaissance des objets, décrit dans la référence[26], c'est un réseau profond de 22 couches, capable de reconnaître mille catégories d'objets, dont la qualité est évaluée dans le contexte de la classification et de détection.
- **Emotion Recognition :** ce modèle est dédié à la reconnaissance des émotions en utilisant le Deep Learning, il a été entraîné avec une base de visages très grande qui dépasse un million d'images [27].
- **VGG Face CNN descriptor :** ce modèle est capable de reconnaître les gens célèbres (les artistes, les joueurs ...), il a été entraîné avec une base d'image de 2.3 millions [28].
- **Places-CNN :** ce modèle destiné pour la reconnaissance des places publiques il est capable de différencier entre 205 places, ce modèle a été entraîné avec une base de 2.5 millions image [29].
- **Age and Gender Classification :** c'est un modèle pour la classification de l'âge [30], il est destiné à la reconnaissance de l'âge et le genre.

III. Environnement de développement :

Nous avons utilisé comme environnement de développement le hardware et le software suivant :

1. ENVIRONNEMENT HARDWARE :

Ordinateur : Sony Vaio VPC-F22S1E/B

Configuration matériel : Processor Intel Core i7-2630QM 2.8 GHz

Carte graphique : NVIDIA GeForce GT 540M - 1024 MB, Core: 672 MHz, Memory: 900 MHz, DDR3, ForceWare 267.80

Mémoire : 8 giga Ram

2. ENVIRONNEMENT SOFTWARE :

- **Framework pyqt :** PyQt [31] est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribuée sous deux licences : une commerciale et une version open source. Il permet ainsi de créer des interfaces graphiques en Python. Une

extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python des interfaces graphiques.

➤ **Ubuntu 14.04 :** Ubuntu 14.04 LTS c'est un système d'exploitation linux, c'est la vingtième version d'Ubuntu. Son développement a débuté en octobre 2013 et s'est échelonné sur une période de six mois. La version stable, finale d'Ubuntu 14.04.1 LTS (long terme support = support disponible pendant une longue durée) sortit le 17 avril 2014 et sera supportée jusqu'en avril 2019.

➤ **OpenCV :** déjà décrit dans le chapitre 2.

IV. Méthodologie de développement :

L'application Raynov Deep Learning, est une application de la reconnaissance d'objets, elle a été développée en utilisant le langage de programmation python. Ce dernier connaît une immense évolution. Il a été amélioré pour qu'il soit le langage le plus utilisé dans les années à venir. Les grandes sociétés du monde comme Google, Nasa et Microsoft utilisent le python comme un langage de programmation. L'intérêt porté à ce langage vient de fait que python intègre plus que 6000 bibliothèques. Outre, c'est un langage interprété ce qui veut dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter.

Dans cette partie, nous avons utilisé le Framework Caffe pour le Deep Learning, ce dernier offre un ensemble de modèles pré-entraînés et aussi il permet d'utiliser les performances du GPU de la carte graphique Nvidia. Nous avons aussi installé différentes ressources comme : Pyqt, QtDesigner, et les packages de python.

Dans cette application nous offrons à l'utilisateur la possibilité de télécharger une image à partir d'un endroit sur son ordinateur, ou à partir d'une caméra pour effectuer la reconnaissance des objets en temps réel.

L'application est capable de reconnaître des différentes catégories comme : les objets en utilisant le modèle GoogLeNet, les places en utilisant le modèle Places CNN, les gens célèbres en utilisant le modèle VGG Face CNN descriptor, et finalement les émotions en utilisant le modèle des émotions et l'âge et le genre en utilisant le modèle Age and genre.

Nous offrons la possibilité à l'utilisateur de choisir le sujet de la reconnaissance, il peut basculer facilement entre les différentes catégories.

Une importante amélioration, que nous avons ajoutée à l'application, est l'utilisation de la notion de multithread pour que la caméra fonctionne en parallèle avec la classification.

Le résultat de la classification apparaît sous forme de probabilités de reconnaissance. Dans ce cas, l'application affichera les trois plus grandes probabilités.

V. Description de l'application :

L'interface graphique de l'application Raynov Deep Learning que nous avons créé est illustrée dans la figure 23

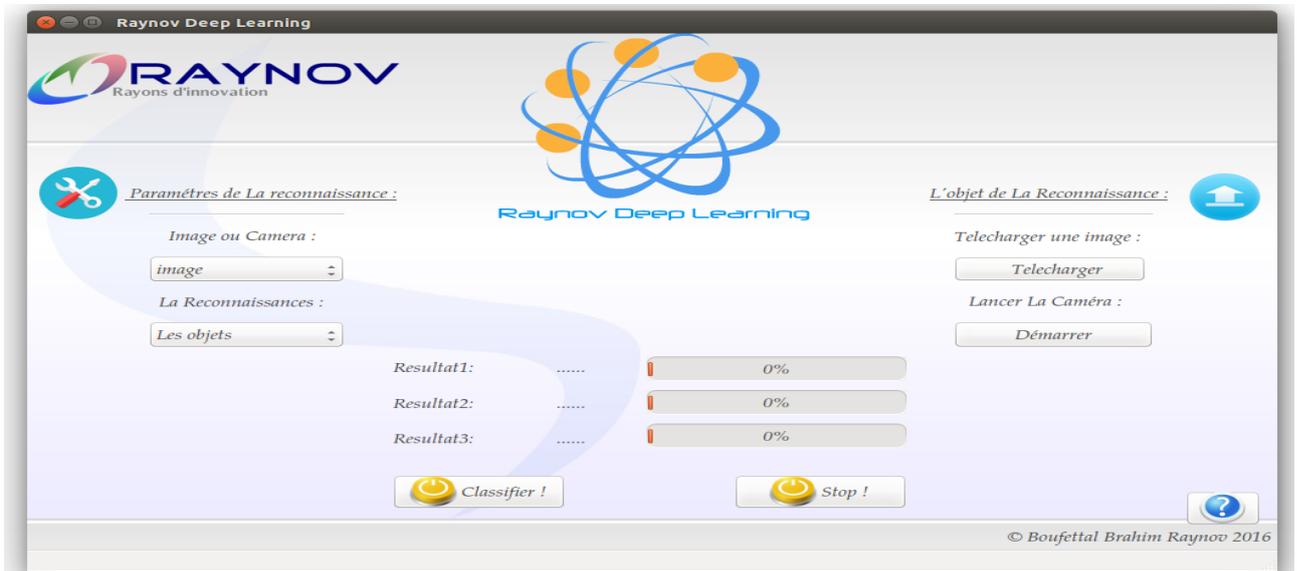


Figure 23 : interface graphique de l'application Raynov Deep Learning

La première étape consiste à télécharger une image à partir d'un endroit dans l'ordinateur de l'utilisateur en cliquant sur le bouton Télécharger.

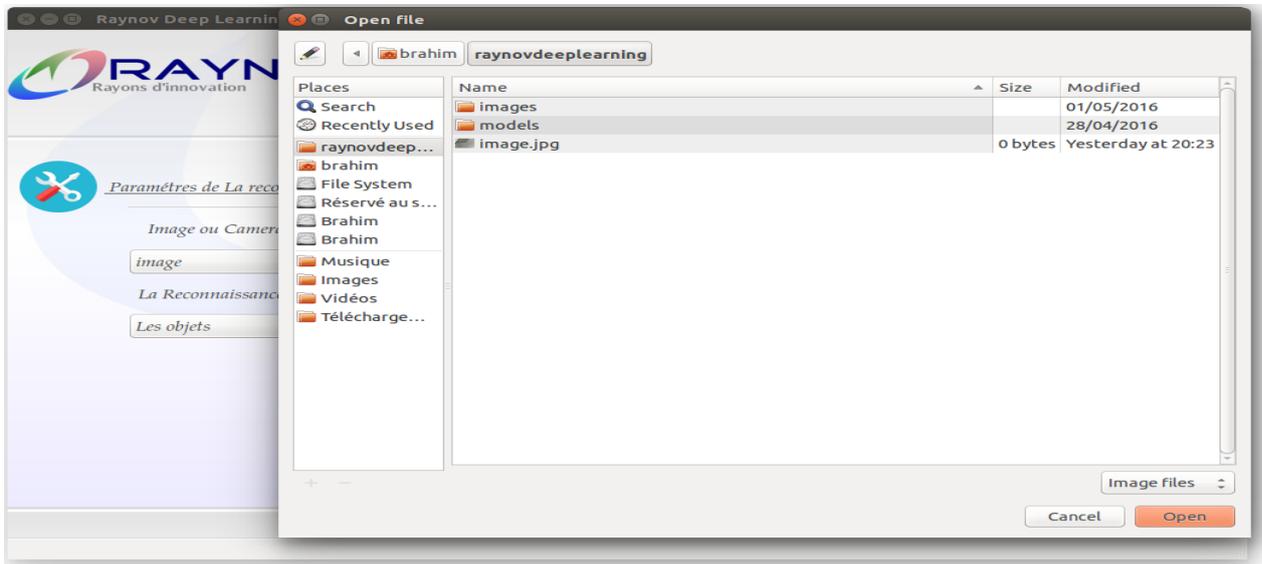


Figure 24 : Télécharger une image (Raynov Deep Learning)

On choisit une image et puis on cliquera sur Open.

Après cela on choisit le sujet de la reconnaissance : les objets, les émotions, les places publiques, les célèbres ou l'âge et le genre, puis on démarre la reconnaissance des objets en utilisant le bouton Classifier.

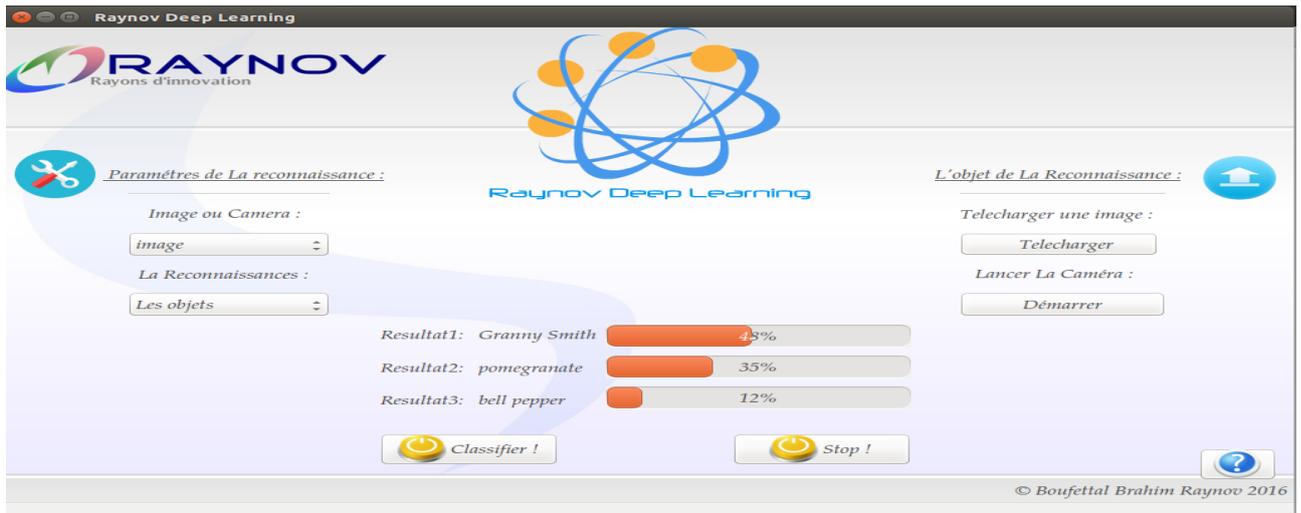


Figure 25 : la classification d'une image (Raynov Deep Learning)

Nous avons trois résultats qui correspondent aux objets qui ont la grande probabilité.

Pour une classification en temps réel, il suffit de choisir le sujet de reconnaissance et de lancer la visualisation de la caméra en cliquant sur démarrer, puis on lance la classification de la même manière par un clic sur Classifier !

La figure 26 illustre le résultat de la classification

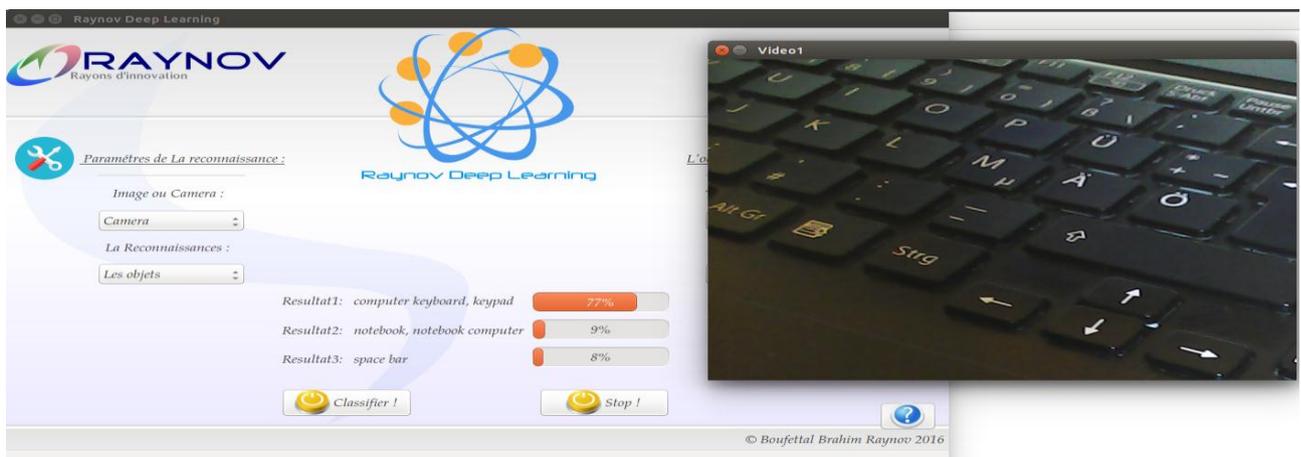


Figure 26 : La classification en temps réel (Raynov Deep Learning)

Si l'utilisateur a besoin du Help, il doit cliquer sur le bouton Help et il va lire les indications à suivre.

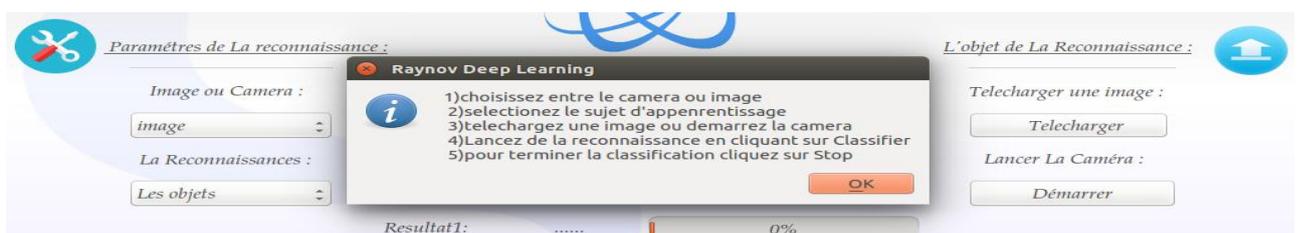


Figure 27 : Le Help de l'application Raynov Deep Learning

Par ailleurs pour éviter les problèmes de dysfonctionnement de l'application qui peuvent causer le blocage de l'application il faut donner des indications afin de régler le problème et éviter le blocage de l'application. Par exemple si l'utilisateur a choisi une caméra, et puis il a cliqué sur Télécharger pour télécharger une image, dans ce cas l'application va afficher un message lui rappelant qu'il a choisi la Camera (voir figure 28).



Figure 28 : avertissement d'erreur (Raynov Deep Learning)

VI. Test de Fonctionnement :

Dans cette partie nous avons testé le fonctionnement de l'application Raynov Deep Learning sur des images, dans ce test nous avons utilisé une base d'images téléchargée à partir du Google images (voir figure 29).



Figure 29 : la base d'images pour le test de fonctionnement (Raynov Deep Learning)

Le test est porté sur l'image d'une télécommande (remote controle) qui se trouve dans la base utilisée. Le résultat obtenu est indiqué sur la figure 30, nous avons obtenu une identification avec un taux de reconnaissance de 99%.



Figure 30 : le résultat de classification de l'image d'une télécommande

Pour tester la fiabilité de l'application sur des objets qui ont les mêmes caractéristiques, on a utilisé deux images : la première contient un ordinateur portable et l'autre un ordinateur fixe (voir figure 31).



Figure 31 : L'image de deux ordinateurs utilisée pour le test de fonctionnement (Raynov Deep Learning)

Pour la classification d'ordinateur fixe, nous avons comme résultat (voir figure 32):

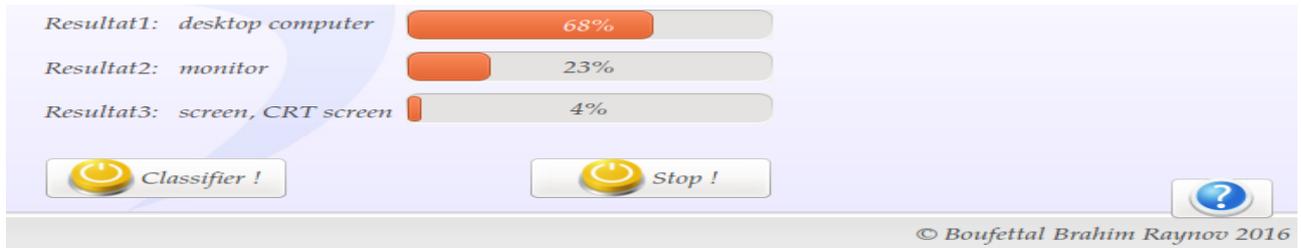


Figure 32 : Résultat de la classification d'un ordinateur fixe (Raynov Deep Learning)

L'application a donné un taux de reconnaissance de 68%.



Figure 33 : Résultat de la classification d'un ordinateur portable (Raynov Deep Learning)

Pour la classification d'ordinateur portable, l'application a donné un taux de 91% pour le notebook.

VII. Conclusion :

L'application Raynov Deep Learning est une application de la reconnaissance basée sur le Deep Learning et destinée à la classification des objets, des émotions, des gens célèbres, des places, d'âge et de genre.

Dans ce chapitre, nous avons abordé le concept de Deep Learning, et ses applications. Puis nous avons présenté en détails notre application et nous l'avons testé sur une base d'images.

Les résultats de reconnaissance que nous avons trouvée montrent la fiabilité de l'application que nous avons développée.

Dans le chapitre suivant nous allons utiliser le Deep Learning pour le développement d'un système de la reconnaissance des objets en utilisant le raspberry Pi.

*Chapitre4 : Développement et implémentation du
système Raynov artificiel*

I. Introduction

Dans les chapitres précédents nous avons travaillé sur le côté software du projet, car nous avons développé des applications utilisées sur un ordinateur. En revanche dans ce dernier chapitre nous travaillons sur le côté hardware. En effet, nous proposons une application de reconnaissance d'objets embarquée dans un microordinateur à l'aide d'un raspberry Pi.

En effet un microordinateur comme le raspberry Pi n'a pas les performances suffisantes pour exécuter des algorithmes de type Deep Learning, notre idée et d'utiliser la notion du serveur, dans notre cas nous allons utiliser un ordinateur portable, qui a les performances nécessaires d'exécuter les algorithmes de Deep Learning pour servir de microordinateur.

II. Environnement de développement :

Dans cette partie nous avons utilisé les ressources software et hardware suivantes :

1. Environnement Hardware :

-Ordinateur : Sony Vaio VPC-F22S1E/B

Configuration matériel : Processor Intel Core i7-2630QM 2.8 GHz

Carte graphique : NVIDIA GeForce GT 540M - 1024 MB, Core: 672 MHz,

Memory: 900 MHz, DDR3, ForceWare 267.80

Mémoire : 8 giga Ram

-Raspberry Pi : Le Raspberry Pi est un micro-ordinateur monocarte à processeur ARM conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi2.

Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique, il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Dans notre cas nous allons utiliser le raspberry Pi B+ qui a la configuration suivante :

-Quad Core Broadcom BCM2836 CPU avec 1GB de Ram

-40 Pin externe GPIO

-Micro SD externe comme une mémoire

-4 ports USB, et Full Size HDMI, 4 pole Stéréo output et un Composite vidéo Port

-Micro USB pour l'alimentation.

-Un écran tactile 7inch HDMI LCD : qui a une résolution de 800×480 et supporte le raspberry Pi.

-Une Camera Pi : c'est une caméra vidéo haute définition pour Raspberry Pi Modèle A ou B. elle a une qualité de 5mp comme appareil photo et enregistrement vidéo de 1080p Full HD.

-W311M wifi usb : Compatible avec les appareils 802.11g / b, W311M est un adaptateur USB sans fil à gain élevé avec un taux de transmission sans fil 3 fois plus rapide que les appareils 802.11g.

-Deux hautparleurs

2. Environnement software :

-PyQt :Le Framework pyqt déjà décrit dans le chapitre 3.

-Raspbian : est un système d'exploitation libre et gratuit fondé sur GNU/Linux/Debian et optimisé pour fonctionner sur un Raspberry Pi.

-Python : nous avons utilisé différents packages du python (socket par exemple)

-Le Framework Caffe.

III. Méthodologie du travail :

Les algorithmes de Deep Learning nécessitent des performances importantes. Malgré l'évolution des ordinateurs, ces derniers restent incapables d'exécuter les algorithmes du Deep Learning en un temps très court. L'utilisation d'un GPU est la seule solution pour booster les processus d'exécution du Deep Learning.

Le Raspberry est un micro-ordinateur de petite taille donc on peut facilement l'intégrer dans des grands systèmes (les robots par exemple). Malheureusement les micro-ordinateurs disponibles comme Raspberry Pi ou banana n'ont pas les performances nécessaires, donc ne peuvent pas être utilisés pour exécuter les algorithmes de Deep Learning. Pour remédier à ce problème nous avons proposé une méthode que nous décrivons dans ce qui suit.

Notre méthode consiste à utiliser la communication réseau entre le Raspberry Pi et un serveur, dans notre cas c'est un ordinateur, le python inclut un package nommé socket et qui facilite la communication entre deux machines de même réseau. Pour le Raspberry leur rôle est de recevoir une image et de l'envoyer au serveur (ordinateur sony vaio), le serveur reconnaît les objets inclus puis il envoie la réponse au raspberry, donc nous avons un échange d'informations entre le Raspberry et le serveur.

Une amélioration que nous avons apporté au système c'est sa capacité d'annoncer l'objet reconnu, pour cela nous avons utilisé un autre package de python, nommé speech et qui nous permet d'envoyer le résultat sous forme de texte à l'API de google, qui va le convertir en parole dans cette étape le système requiert Internet pour communiquer en temps réel avec l'API de google.

Nous avons donné la possibilité à l'utilisateur de choisir la langue désirée, et aussi de choisir entre la communication sous forme de texte ou sous forme de parole.

Nous a développé deux applications en python, une pour le serveur et l'autre pour le raspberry. Dans la partie suivante nous donnons une description du système ainsi que les deux applications développées.

VIII. Description et test du système :

Le système Raynov artificiel que nous avons développé, est un système intelligent qui permet la reconnaissance des objets en utilisant le Deep Learning et le raspberry pi. Ce système est montré dans la figure 34.



Figure 34 : le système Raynov artificiel

Ce système est constitué d'un Raspberry Pi, de deux hautparleurs, d'un écran HDMI tactile et d'une clé wifi. Nous avons rassemblé tous les composants dans une plaque afin de faciliter son utilisation. Le système communique avec un ordinateur sony vaio qui a une carte graphique Nvidia GeForce GT 540M, il est capable d'exécuter des algorithmes de Deep Learning en temps court, inférieure à une seconde. Pour assurer la communication réseau nous avons développé deux applications, d'une part l'application destinée pour le serveur est illustrée dans la figure 35 :



Figure 35 : Interface graphique de l'application du serveur (Raynov artificiel)

Dans cette application, nous avons réalisé une interface nommée Raynov artificiel. L'utilisateur peut changer facilement la langue en utilisant le bouton <<Language choice >>, aussi il est capable de choisir l'adresse IP de son Raspberry avant de lancer l'application.

Alors que la deuxième application que nous avons développée est destinée à le raspberry, la figure 36 illustre son interface graphique :



Figure 36 : Interface graphique de l'application du serveur (Raynov artificiel)

L'interface de cette application est similaire à l'interface graphique de l'application du serveur, sauf qu'on y a ajouté une fonctionnalité qui permet de faire le choix du type de la communication (texte ou speech). Si l'utilisateur choisit « texte » le résultat apparaîtra sous forme du texte devant <<i think this is >>, par contre s'il choisit <<Speech>> le résultat de

reconnaissance sera prononcé par la machine .Sur la figure 37 nous donnons le résultat d'un cas de reconnaissance.



Figure 37 : Résultat textuel en anglais de la reconnaissance (Raynov artificiel)

IV. Conclusion

Dans ce chapitre nous avons réalisé le système Raynov artificiel dédié à la reconnaissance des objets. Notre idée est d'utiliser une machine puissante pour booster la reconnaissance qui repose sur les algorithmes du Deep Learning et servir le raspberry.

Le système Raynov artificiel développé peut être embarqué facilement dans d'autres systèmes comme les robots.

Conclusion générale

Cet humble rapport présente le résultat d'un travail soutenu et assidu qui s'inscrit dans le cadre de la réalisation d'un projet de fin d'études pour l'obtention du diplôme du master sciences et techniques en « Electronique Signaux et Systèmes automatisés ». Ce stage d'une durée de 4 mois, a été réalisé au sein de la société Raynov

Dans ce projet de fin d'études nous avons réalisé un système de reconnaissance d'objets en utilisant le Deep Learning.

Nous avons développé, dans un premier temps, une application dédiée pour la reconnaissance des objets et des visages en se basant sur les réseaux de neurones. Ensuite, nous avons développé une application de la reconnaissance des objets, des émotions, de l'âge et le genre..., en utilisant le Deep Learning.

Dans un deuxième temps nous avons conçu un système de reconnaissance d'objets en utilisant un serveur (sony vaio) afin de booster les algorithmes du Deep Learning et de pouvoir effectuer une classification des objets en temps réel.

En perspective, nous pensons à embarquer ce système de reconnaissance dans un robot autonome qui soit capable de prendre des décisions tout seul.

Bibliographie et Webographie

- [1] : https://fr.wikipedia.org/wiki/D%C3%A9tection_d%27objet
- [2] : V. Bjorn, "One Finger at a Time: Best Practices for Biometric Security," Banking Information Source (Document ID: 1697301411), April, 2009 .
- [3] : W. Chang, C. Hsia. Y. Tai, et.al, "An efficient object recognition system for human and robot vision," Pervasive Computing (JCPC), IEEE, December, 2009
- [4] : Techniques for Object Recognition in Images and Multi-Object Detection : International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 4, April 2013
- [5] : K. Murphy, A. Torralba , D. Eaton and W. Freeman, "Object detection and localization using local and global features," Towards Category-Level Object Recognition, 2005
- [6] : F. Khan, R. Muhammad , et.al., "Color Attributes for Object Detection," In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3306 – 3313, 2012
- [7] : http://docs.opencv.org/2.4/modules/imgproc/doc/object_detection.html?highlight=matchtemplate#matchtemplate
- [8] Paul Viola et Michael Jones, « Rapid Object Detection using a Boosted Cascade of Simple Features », IEEE CVPR, 2001
- [9] : Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002
- [10] : <http://www.mediafire.com/download/049s8I967btghz6/g%C3%A9n%C3%A9ration+xml.pdf>
- [11] : Herbert Bay, Tinne Tuytelaars et Luc Van Gool, « SURF: Speeded Up Robust Features », dans 9th European Conference on Computer Vision, Graz, Autriche, 7-13 mai 2006
- [12] : Herbert Bay, Andreas Ess, Tinne Tuytelaars et Luc Van Gool, « SURF: Speeded Up Robust Features », Computer Vision and Image Understanding, vol. 110, no 3, 2008
- [13] : https://en.wikipedia.org/wiki/Speeded_up_robust_features
- [14] : A Brief Introduction to Neural Networks : David Kriesel
- [15] : https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels
- [16] : W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 5(4):115–133, 1943

- [17] : Donald O. Hebb. The Organization of Behavior: A Neuropsychological Theory. Wiley, New York, 1949
- [18] : M. Minsky and S. Papert. Perceptrons. MIT Press, Cambridge, Mass, 1969
- [19] : D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. Nature, 323:533–536, October 1986.
- [20] : https://fr.wikipedia.org/wiki/R%C3%A9tropropagation_du_gradient
- [21] : <https://www.qt.io/>
- [22] : <http://docs.opencv.org>
- [23] : https://fr.wikipedia.org/wiki/Deep_learning
- [24] : Deep learning : quand l'intelligence artificielle tente d'imiter le cerveau humain Par Jérôme Cartegini
- [25] : Going Deeper with Convolutions ,Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich ,17 Sep 2014.
- [26] : "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.
- [27] : Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns , Gil Levi and Tal Hassner ,Proc. ACM International Conference on Multimodal Interaction (ICMI), Seattle, Nov. 2015
- [28] : Deep Face Recognition Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman BMVC 2015
- [29] : B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva :Learning Deep Features for Scene Recognition using Places Database. , Advances in Neural Information Processing Systems 27 (NIPS) spotlight, 2014.
- [30] : Age and Gender Classification using Convolutional Neural Networks Gil Levi and Tal Hassner ,IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG),at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015
- [31] : <https://wiki.python.org/moin/PyQt>