

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abbes Laghrou de Khenchela

Faculté des Sciences et Technologies

Département Maths et Informatique



Filière : Maths et Informatique

Spécialité : GLSD (Génie Logiciel et Système Distribué)

Intitulé

L'utilisation de Deep Learning pour la Classification d'objets pour les Besoins de la Police

Présenté Par :

- Messai Anouar
- Moncef Boutebina

Encadrant : Dr. Dalal Bardou

Année universitaire : 2020 / 2021

Remerciement

Mes remerciements vont en premier lieu à mon encadrant **Dalal Bardou**. La réalisation de ce mémoire n'aurait tout simplement pas été possible sans le soutien de **Dalal Bardou**, mon encadrant. Son attention, son aide et sa bonne humeur inaltérable m'ont permis d'avancer dans la bonne direction, du premier au dernier jour. Grâce à lui cette année de master a été à la fois intéressante et agréable, je suis heureux de pouvoir profiter de ces quelques lignes pour lui exprimer toute mon estime et ma gratitude.

J'en profite également pour saluer le soutien moral apporté par mes plus proches connaissances : Zakaria Aouaidjia, Abderaouf Messai.

Enfin, je tiens à remercier profondément ma proche famille, mes frères et surtout mes parents dont la bienveillance et l'appui n'ont cessé de m'encourager depuis mon enfance dans chaque chose que j'entreprends. Ils m'ont permis de surmonter toutes les épreuves, même les plus difficiles. Je souhaite dédier ce travail à ma mère et mon père.

Dédicace

Nous dédions notre travail à : Les deux êtres les plus chères au monde, les bougies qui m'ont toujours guidé vers le bon chemin : Ma mère, la lumière de notre vie, qui a tout fait pour notre réussite et notre bonheur. Mon père, à lui, nous devons offrir tout le respect et l'amour pour son soutien et sa tendresse. A nos sœurs pour leur aide et courage, à nos frères, à mon ami Zakaria, à mon ami Ramzi, à tous les membres de la famille et à toutes les amies.

Résumé :

La vision par ordinateur est un champ d'étude qui permet aux systèmes automatiques à reconnaître les entrées visuelles pour les exploiter dans des tâches de recommandation. Dans ces dernières années, la quantité des images et des vidéos a largement augmenté. L'exploitation des systèmes de vision par ordinateur pour l'analyse de cette quantité d'informations devient importante afin d'extraire de l'information pertinente. Les systèmes de vision par ordinateur sont basés essentiellement sur les méthodes d'apprentissage automatique (ML) et d'apprentissage profond (DL). Avec l'augmentation de la quantité de données et la disponibilité du matériel puissant, les méthodes DL ont connu un grand intérêt en raison de leur bonne performance sur les grands volumes de données et leur capacité d'extraction de caractéristique dans le cadre des données non structurées. Ces techniques étaient exploitées dans différents sous domaines en vision par ordinateur pour effectuer plusieurs tâches : classification, localisation, détection, et segmentation.

Dans le contexte de la présente étude, nous exploitons l'utilisation de Deep Learning, précisément les réseaux de neurones convolutifs (CNN) pour la classification d'objets pour les besoins de la Police. Notre travail est basé sur l'utilisation de transfer learning pour entraîner notre modèle et puis l'employer pour prédire des nouvelles entrées. Les résultats obtenus par notre modèle ont montré son efficacité.

Mots-clés: Deep learning; Transfer learning; CNN; Police; Classification.

Table des matières

Introduction Générale :	4
Chapitre 1 : Classification d'images : Notions, Méthodes... ..	7
1 Introduction	7
2 Intelligence artificielle	7
3 Les différentes méthodes de l'apprentissage.....	9
3.1 Apprentissage supervisé et non supervisé	9
4 La Classification	10
5 Les Techniques de Classification	11
5.1 Les Techniques basée sur La Machine Learning Traditionnelle	11
5.2 Les Techniques basée sur le Deep Learning.....	11
6 L'utilisation de Deep Learning.....	12
7 Implication de l'intelligence artificielle sur le monde du travail	13
8 Conclusion :.....	14
Chapitre 2 : Les Réseaux de Neurones Convolutifs.....	16
1 Introduction	16
1.1 Principe d'architecture d'un CNN.....	17
1.2 L'entraînement d'un nouveau CNN est difficile.....	17
1.3 Les blocs de construction.....	18
1.3.1 La couche de convolution (CONV)	18
1.3.2 La couche de pooling	19
1.3.3 La Couche de Correction ReLU.....	20
1.3.4 Couche entièrement connectée (FC)	20
1.3.5 Couche de perte (LOSS).....	20
2 Transfer Learning	21
2.1 Définition	21
2.2 Sur quelles stratégies les techniques de Transfer Learning reposent ?	21
2.3 Comment le Transfer Learning est mis en œuvre concrètement pour la résolution de problématiques de Deep Learning ?	23
3 Conclusion.....	25
Chapitre 3 : Développement et implémentation.....	27

1	Les outils utilisés	27
2	Data Set	27
3	AlexNet Network.....	30
4	Tester le modèle.....	31
5	Interface graphique :	32
5.1	Composition de l'interface graphique :	32
6	Conclusion :	35
	Conclusion générale :	36
	Bibliographie.....	37

Liste des figures

Figure 1 : Convergence d'un modèle -Apprentissage supervisé.....	9
Figure 2 : Exemple du traitement des entrées dans un réseau de neurone	11
Figure 3: Pourquoi utilisons-nous le Deep Learning ?.....	12
Figure 4: Détection automatique de fracture à l'aide du deep Learning	13
Figure 5: Implication de l'intelligence artificielle sur le monde réel	13
Figure 6 : Transfert de style	14
Figure 7: Un exemple qui montre l'architecture d'un modèle CNN.....	16
Figure 8 : 5-Les réseaux de neurones convolutifs.....	17
Figure 9 : Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution	18
Figure 10: Pooling avec un filtre 2x2 et un pas de 2.....	19
Figure 11: Allure de la fonction Relu.....	20
Figure 12 : Approche traditionnelle vs. Approche de Transfert Learning	21
Figure 13 : Illustration de l'apprentissage par transfert inductif, ou Inductive Transfer Learning	22
Figure 14 : Exemple de modèle de réseau de neurones utilisé pour la classification d'images	23
Figure 15 : Exemple de modèle de réseau de neurones utilisé pour reconnaissance faciale	24
Figure 16 : Code source et exécution du scraper. (Réalisé à l'aide de l'outil : Notepad++)....	28
Figure 17 : Création de dossier d'images pour chaque classe.	29
Figure 18 : Nettoyage manuel des images.....	29
Figure 19 : L'architecture d' AlexNet Network.....	30
Figure 20 : Les différentes couches d' AlexNet Network.	31
Figure 21 : La matrice de confusion.....	32
Figure 22 : Interface graphique au démarrage du programme.....	33
Figure 23 : Interface graphique du programme pendant l'exécution de l'analyse.....	34
Figure 24: Interface graphique du programme après l'analyse.....	34
Figure 25 : Icône de l'application	35

Introduction Générale :

Le deep learning [1] (l'apprentissage profond), une branche de l'apprentissage automatique, est devenu un nouveau domaine d'apprentissage automatique et il a connu un succès incroyable dans de nombreux domaines où la communauté de l'intelligence luttait auparavant depuis de nombreuses années [2]. Le deep learning permet d'extraire un ensemble de caractéristiques auto-adaptatives, correspondant à une tâche spécifique. Ces caractéristiques sont régulées cognitivement par un nombre incalculable de paramètres relatifs au mécanisme du cerveau humain qui fournir des résultats plus précis et efficaces. L'avancée majeure a été réalisée dans l'image et la reconnaissance vocale ainsi que le traitement du langage naturel, et la traduction.

Dans ce mémoire, nous proposons un modèle d'apprentissage basé sur le deep-learning pour aider la police pour faire une automatique classification d'objets. Le type de réseaux de neurones utilisés pour adresser ce problème sont les réseaux de neurones convolutifs (CNN)[3]. Les réseaux de neurones convolutifs sont un type spécifique de réseaux de neurones pour traiter des données qui ont une topologie en forme de grille telles que les images qui peuvent être vues comme un 2-D grille de pixels. Ces réseaux de neurones sont inspirés du cortex visuel du cerveau. Dans le cortex visuel, il est une hiérarchie de deux types de cellules de base : les cellules simples et les cellules complexes. Les cellules simples répondent aux modèles primitifs dans les sous-régions de stimuli visuels, et les cellules complexes synthétisent les informations à partir de cellules simples pour reconnaître des formes plus sophistiquées. Notre travail utilise l'apprentissage par transfert qui permet à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles pour construire notre CNN-modèle. Pour aboutir à notre but, on a utilisé un modèle pré-entraîné nommé AlexNet [4] qui a constitué une percée majeure dans le développement de CNN [3], [5].

Quand un suspect arrive à la police, les policiers vont faire une analyse de son téléphone portable pour analyser les données stockées comme les images, les vidéos...etc. Il est trouvé qu'une personne normale peut avoir plus de 2000 images stockées. Donc, l'analyse manuelle de ces images est une tâche très difficile. Pour cela, il ya une immense nécessité pour automatiser ce travail pour alléger le fardeau de policiers. La technologie actuelle qui permet d'apporter une solution c'est l'automatisation de cette tâche en basant sur le deep learning [8]. Donc pour réaliser ce travail, on avait besoin de collecter le data set pour entraîner le modèle. Le data set contient des images de différentes classes. Chaque classe indique un type d'objet interdit comme la drogue, faux argent, armes...etc. Après, on a fait le développement de l'application et aussi l'a testé.

Le reste de ce mémoire est organisé comme suit : Dans le premier chapitre, on donne une introduction à la classification [6], les notions et les méthodes. Dans le deuxième chapitre, on introduit les réseaux de neurones convolutifs. Finalement, dans le troisième chapitre, on implémente notre modèle proposé et on rapporte les résultats achevés.

CHAPITRE 01 :

Classification d'images : Notions, Méthodes...

Chapitre 1 : Classification d'images : Notions, Méthodes...

1 Introduction

D'un point de vue informatique, une image est simplement un tableau de nombres, dont chacun est la valeur de la luminance du pixel correspondant dans l'image. Il apparaît aisément qu'à partir de tous ces nombres, il est difficile de reconnaître non seulement une forme complexe (comme une personne) mais même des informations de plus bas niveau comme une simple forme géométrique. C'est pourquoi des descripteurs d'images sont couramment utilisés. Ceux-ci permettent de calculer pour une image donnée une information plus riche à partir d'une méthode de calcul adaptée, bien sûr basée sur l'intelligence artificielle.

2 Intelligence artificielle

L'intelligence artificielle est née dans les années 50, quand une poignée de pionniers de l'informatique naissante ont commencé à se demander si les ordinateurs pouvaient être conçus pour « penser » — une question dont nous continuons aujourd'hui d'explorer les ramifications. Une définition précise de ce domaine serait la suivante : c'est l'effort d'automatisation des tâches intellectuelles normalement effectuées par des humains. L'intelligence artificielle est donc un domaine général qui englobe l'apprentissage automatique et l'apprentissage profond, mais qui comprend également de nombreuses autres approches qui n'impliquent aucun apprentissage. Par exemple, les premiers programmes d'échecs ne mettaient en œuvre que des règles codées en dur (hardcoded) élaborées par des programmeurs, et qui ne pouvaient pas être qualifiées d'apprentissage automatique. Pendant très longtemps, de nombreux experts ont cru qu'une intelligence artificielle équivalente à celle de l'homme pouvait être atteinte en faisant en sorte que les programmeurs fabriquent un ensemble suffisamment large de règles explicites pour manipuler les connaissances. Cette approche est connue sous le nom d'intelligence artificielle symbolique (symbolic AI), et ce fut le paradigme dominant de l'IA des années 1950 à la fin des années 1980. Elle a atteint son apogée pendant le boom des systèmes experts (expert system) dans les années 1980[7].

Bien que l'IA symbolique se soit révélée apte à résoudre des problèmes logiques bien définis, tels que jouer aux échecs, elle s'est avérée incapable de définir des règles explicites pour résoudre des problèmes plus complexes et flous, tels que la classification d'images, la reconnaissance de la parole et la traduction linguistique. Une nouvelle approche est apparue, et

elle a pris la place de l'intelligence artificielle symbolique : c'est l'apprentissage automatique (machine Learning) [8]

3 Les différentes méthodes de l'apprentissage

3.1 Apprentissage supervisé et non supervisé

Le Machine Learning est un ensemble de techniques donnant la capacité aux machines d'apprendre, contrairement à la programmation qui consiste en l'exécution de règles prédéterminées [9].

Il existe deux principaux types d'apprentissages en Machine Learning. L'apprentissage supervisé et non supervisé.[9]

En apprentissage supervisé, l'algorithme est guidé avec des connaissances préalables de ce que devraient être les valeurs de sortie du modèle. Par conséquent, le modèle ajuste ses paramètres de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. La marge d'erreur se réduit ainsi au fil des entraînements du modèle, afin d'être capable de l'appliquer à de nouveaux cas.

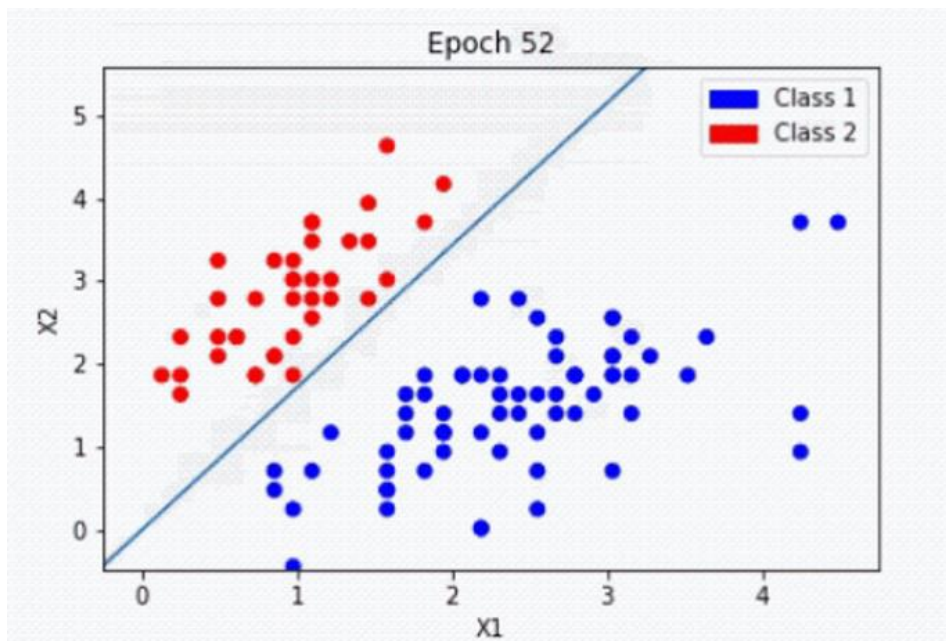


Figure 1 : Convergence d'un modèle -Apprentissage supervisé[9]

En revanche, l'apprentissage non supervisé n'utilise pas de données étiquetées. Il est alors impossible à l'algorithme de calculer de façon certaine un score de réussite. Son objectif est donc de déduire les regroupements présents dans nos données. Prenons l'exemple, d'un jeu de données de fleurs, on recherche à les regrouper en classes. Ici, nous ne connaissons pas l'espèce de la plante, mais nous voulons essayer de les regrouper :

- Exemple : si les formes des fleurs sont similaires alors elles sont en rapport avec une même plante correspondante.

Il existe deux principaux domaines de modèles dans l'apprentissage non-supervisés pour retrouver les regroupements :

4 La Classification

La classification statistique réside dans l'identification de la catégorie à laquelle un nouvel élément appartient, sur la base d'un data set d'entraînement de données contenant des observations (ou instances) dont la catégorie est connue.

Par exemple, attribuer un courrier électronique donné à la classe "spam" ou "non spam" et attribuer un diagnostic à un patient donné en fonction des caractéristiques observées du patient (sexe, tension artérielle, présence ou absence de certains symptômes, etc.). La classification est un exemple de reconnaissance de forme.

Dans la terminologie de l'apprentissage automatique, la classification est considérée comme un exemple d'apprentissage supervisé, c'est-à-dire un apprentissage dans lequel un data set d'entraînement d'observations correctement identifiées est disponible. Le correspondant non supervisé la procédure est connue comme groupe ING, et comprend le regroupement de données en catégories fondées sur une mesure de similarité intrinsèque ou la distance.

Souvent, les observations individuelles sont analysées dans une série de propriétés quantifiables, connues sous différentes appellations de variables ou caractéristiques explicatives. Ces propriétés peuvent diversement être catégorique pour le type de sang), ordinal (par exemple, « grand », « moyen » ou « faible »), à valeurs entières (par exemple, le nombre d'occurrences d'un mot particulier dans un e - mail) ou à valeurs réelles (par exemple, une mesure de sang presse ure). D'autres classificateurs comparent les observations aux observations précédentes au moyen d'une fonction de similarité ou de distance.

Un algorithme qui implémente la classification, en particulier dans une mise en œuvre concrète, est connu comme un classificateur. Le terme "classificateur" désigne parfois aussi la fonction mathématique, mise en œuvre par un algorithme de classification, qui mappe les données d'entrée à une catégorie.[10]

5 Les Techniques de Classification

5.1 Les Techniques basée sur La Machine Learning Traditionnelle

La machine learning traditionnelle est le domaine de l'informatique qui connaît la croissance la plus rapide et de nombreux succès des méthodologies ont été développées pour traiter les tâches de classification. Ces méthodologies consistent généralement en une extraction et une classification des caractéristiques. L'extraction des caractéristiques est l'étape où un ensemble de caractéristiques discriminantes et non redondantes est extrait des données d'entrée pour chaque classe. Ces caractéristiques permettent de distinguer les dissemblances entre les classes. Cependant, cette approche souffre de nombreuses limitations. Tout d'abord, cela dépend des représentations des données (Caractéristiques). Ces algorithmes d'extraction de caractéristiques sont développés par des ingénieurs experts, et identifier le type de fonctionnalités adapté à une tâche spécifique est encore difficile.[11]

5.2 Les Techniques basée sur le Deep Learning

Le Deep learning ou apprentissage profond est l'une des technologies principales du Machine learning. Avec le Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificielles. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.[12]

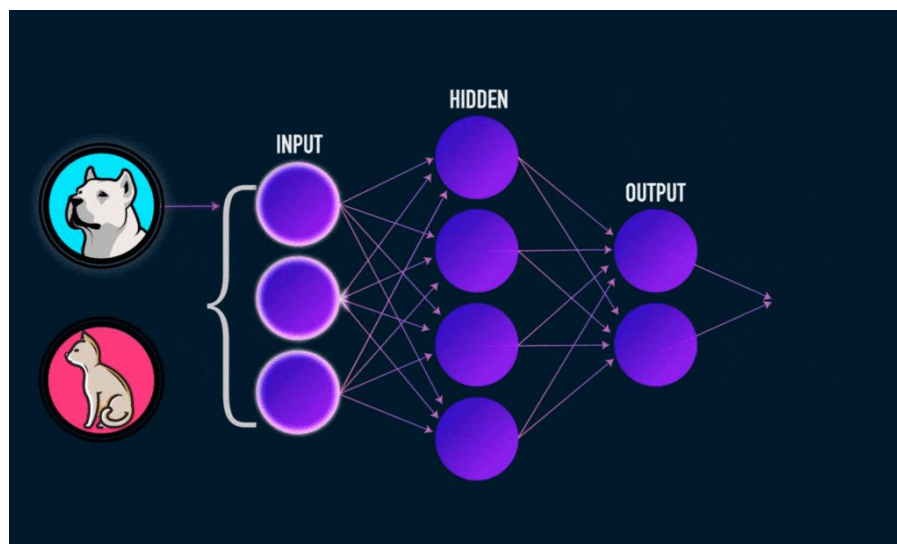


Figure 2 : Exemple du traitement des entrées dans un réseau de neurone[9]

Chaque neurone artificiel représenté dans l'image précédente par un rond, peut être vu comme un modèle linéaire. En interconnectant les neurones sous forme de couche, nous transformons notre réseau de neurones en un modèle non-linéaire très complexe[3].

Pour illustrer le concept, prenons un problème de classification entre chien et chat à partir d'image. Lors de l'apprentissage, l'algorithme va ajuster les poids des neurones de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. Le modèle pourra apprendre à détecter les triangles dans une image puisque les chats ont des oreilles beaucoup plus triangulaires que les chiens.

6 L'utilisation de Deep Learning

Les modèles de Deep Learning ont tendance à bien fonctionner avec une grande quantité de données alors que les modèles d'apprentissage automatique plus classique cessent de s'améliorer après un point de saturation.[13]

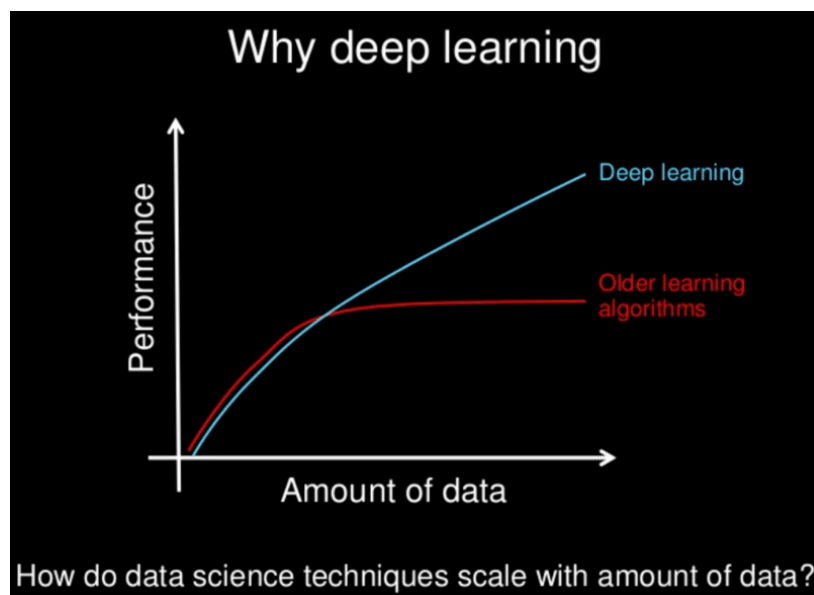


Figure 3: Pourquoi utilisons-nous le Deep Learning ?[9]

Au fil des années, avec l'émergence du big data et de composants informatiques de plus en plus puissant, les algorithmes de Deep Learning gourmands en puissance et en données ont dépassé la plupart des autres méthodes. Ils semblent être prêt à résoudre bien des problèmes : reconnaître des visages, vaincre des joueurs de go ou de poker, permettre la conduite de voitures autonomes ou encore la recherche de cellules cancéreuses.

7 Implication de l'intelligence artificielle sur le monde du travail

Presque toutes les industries sont affectées par l'IA. Le Machine Learning et le Deep Learning y jouent un grand rôle.

Que vous soyez un professionnel de la santé ou un avocat, il est possible qu'un jour un modèle hautement autonome vous assiste ou même vous remplace.

Dans les métiers de la santé, il existe déjà des applications pour automatiquement diagnostiquer un patient.[7]



Figure 4: Détection automatique de fracture à l'aide du deep Learning[9]

Les métiers de l'automobile sont également bousculés avec l'arrivée de la conduite assistée.

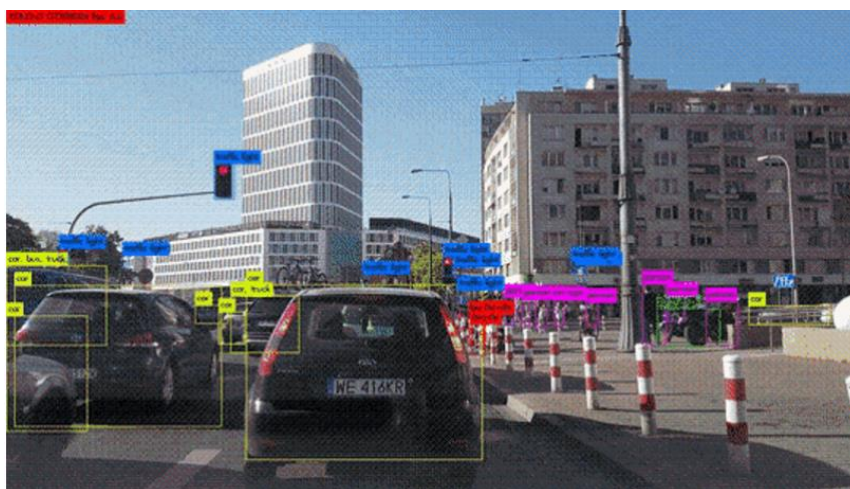


Figure 5: Implication de l'intelligence artificielle sur le monde réel[9]

C'est aussi grâce au deep Learning que le modèle Alpha Go de Google a réussi à battre les meilleurs champions de Go en 2016. Le moteur de recherche du géant américain est lui-même de plus en plus basé sur l'apprentissage par deep Learning plutôt que sur des règles écrites.

Aujourd'hui le deep Learning est même capable de « créer » tout seul des tableaux de peintre. C'est ce qu'on appelle le Style Transfer[9].

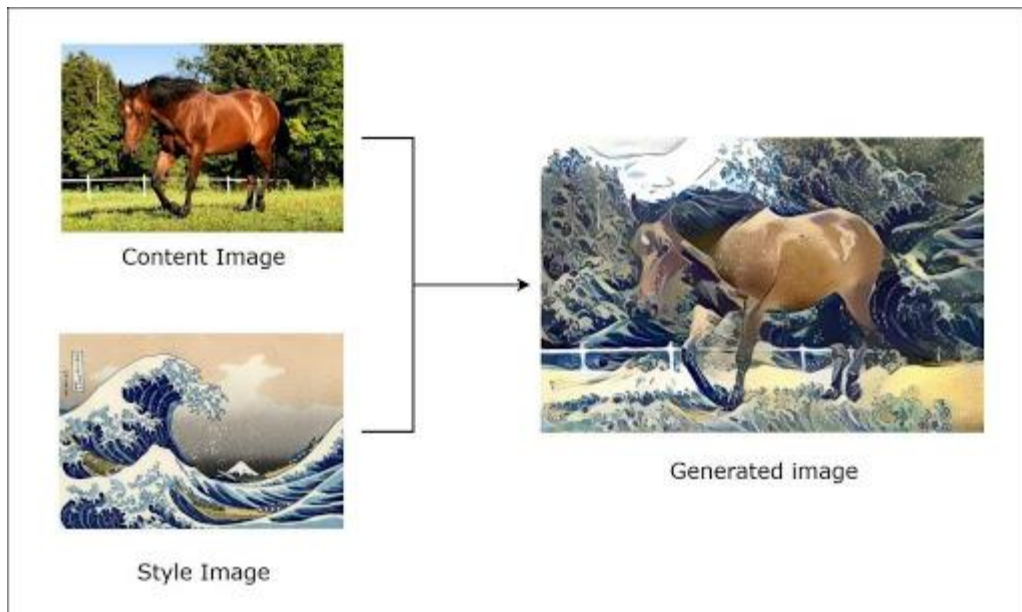


Figure 6 : Transfert de style[9]

8 Conclusion :

Dans ce chapitre, on a introduit l'intelligence artificielle, les différents types d'apprentissage et la classification qui est un type d'apprentissage supervisé.

CHAPITRE 2 :

Les Réseaux de Neurones Convolutifs

Chapitre 2 : Les Réseaux de Neurones Convolutifs

1 Introduction

Les réseaux de neurones convolutifs ou CNN sont un type spécifique de réseaux de neurones pour le traitement de données qui ont une topologie en forme de grille, telles que les données d'image qui peuvent être vues comme une grille de pixels 2D. Le nom « réseau de neurones convolutifs » indique que le réseau utilise une opération mathématique appelée convolution qui est un type d'opération linéaire. Les réseaux de neurones convolutifs sont inspirés du cortex visuel du cerveau. Dans le cortex visuel, il existe une hiérarchie de deux types cellulaires de base : les cellules simples et les cellules complexes. Les cellules simples répondent aux modèles primitifs dans les sous-régions de stimuli visuels, et les cellules complexes synthétisent les informations des cellules simples pour reconnaître des formes plus sophistiquées. Étant donné que le cortex visuel est un système de traitement visuel si puissant et naturel, les CNN sont appliqués pour imiter trois idées clés : la connectivité locale, l'invariance à l'emplacement et l'invariance à la translation locale. Le réseau de neurones de convolution se compose de couches de convolution, de couches non linéaires et de couches de mise en commun. La figure 7 montre une structure simplifiée du réseau de neurones convolutifs.[14]

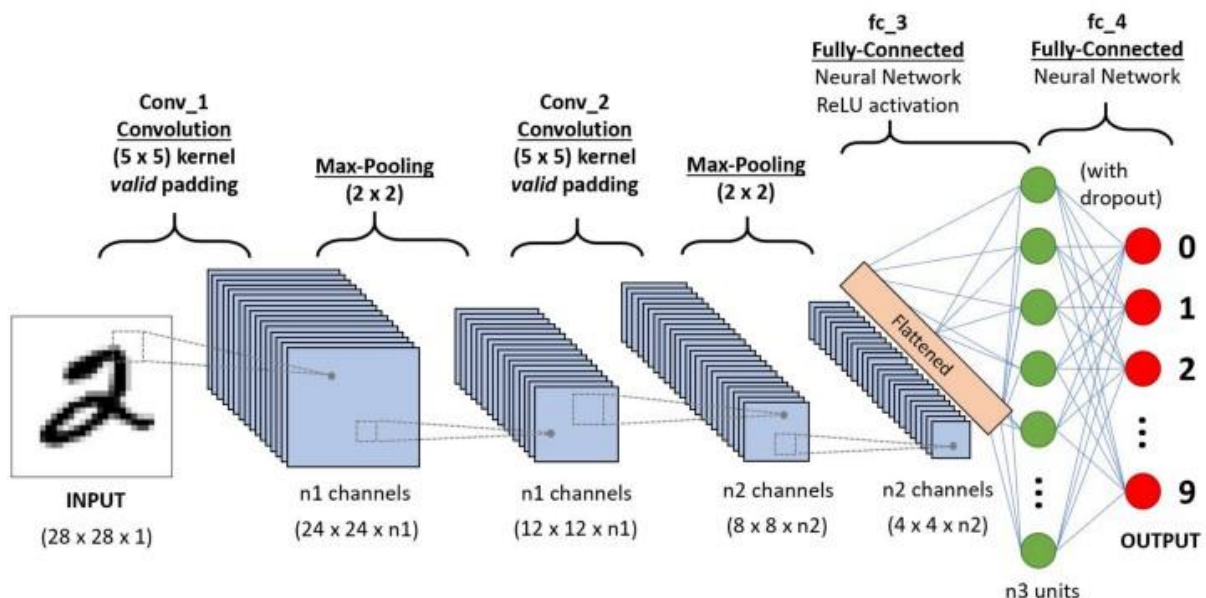


Figure 7: Un exemple qui montre l'architecture d'un modèle CNN.[15]

1.1 Principe d'architecture d'un CNN

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. 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 d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN[14].

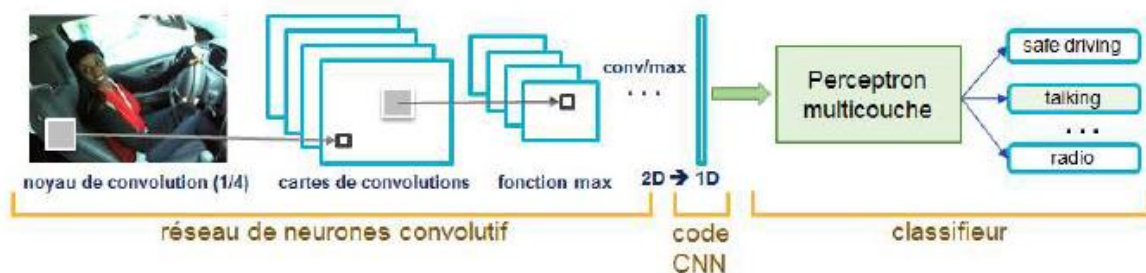


Figure 8 : 5-Les réseaux de neurones convolutifs.[14]

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. 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.

1.2 L'entraînement d'un nouveau CNN est difficile

Créer un nouveau réseau de neurones convolutifs est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires. Il s'agit d'abord de fixer l'architecture du réseau, c'est-à-dire le nombre de couches, leurs tailles et les opérations matricielles qui les

connectent. L'entraînement consiste alors à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées. Des équipes de recherche se spécialisent dans l'amélioration des CNN[16]. Elles publient leurs innovations techniques, ainsi que le détail des réseaux entraînés sur des bases de données de références[17]

1.3 Les blocs de construction

Une architecture CNN est formée par un empilement de couches de traitement indépendantes :

1.3.1 La couche de convolution (CONV)

La couche de convolution est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche. Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée. Pour cela, on réalise un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la feature sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée. Une feature est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte[18].

La couche de convolution reçoit donc en entrée plusieurs images, et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images. On obtient pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature.[18]

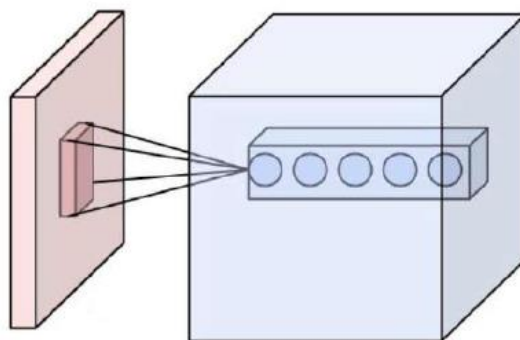


Figure 9 : Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution[9], [18]

1.3.2 La couche de pooling

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2×2 pixels qui ne se chevauchent pas, ou des cellules de taille 3×3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc). On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites.

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage. Les valeurs maximales sont repérées de manière moins exacte dans les feature maps obtenues après pooling que dans celles reçues en entrée – c'est en fait un grand avantage ! En effet, lorsqu'on veut reconnaître un chien par exemple, ses oreilles n'ont pas besoin d'être localisées le plus précisément possible : savoir qu'elles se situent à peu près à côté de la tête suffit ! Ainsi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image[19].

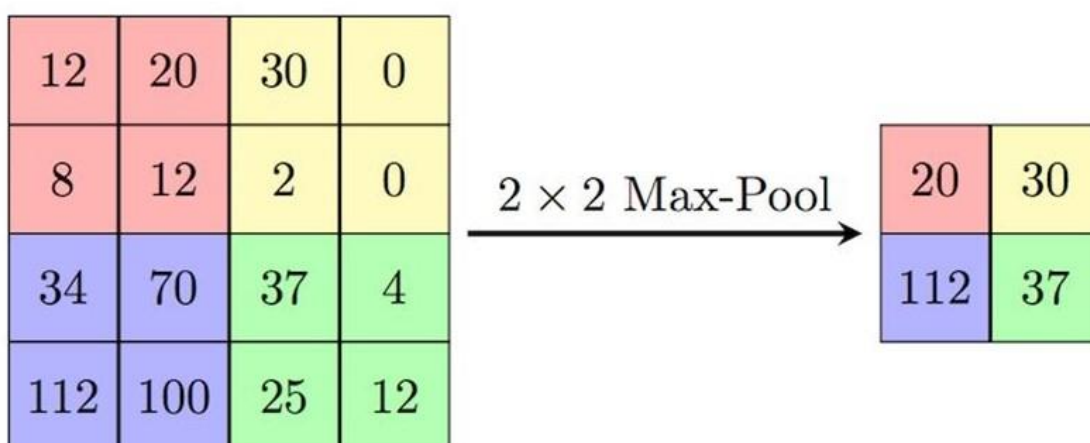


Figure 10: Pooling avec un filtre 2×2 et un pas de 2[20].

1.3.3 La Couche de Correction ReLU

ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie Comme suite :
 $\text{ReLU}(x) = \max(0, x)$

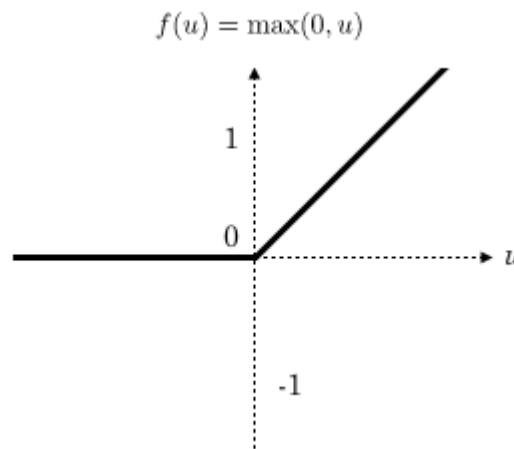


Figure 11: Allure de la fonction Relu [21]

La couche de correction Relu remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation.[22]

1.3.4 Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux réguliers de neurones). Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.[22]

1.3.5 Couche de perte (LOSS)

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte « Soft max » est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans (0,1). La perte euclidienne est utilisée pour régresser vers des valeurs réelles.[23]

2 Transfer Learning

2.1 Définition

Le Transfer Learning, ou apprentissage par transfert en français, désigne l'ensemble des méthodes qui permettent de transférer les connaissances acquises à partir de la résolution de problèmes donnés pour traiter un autre problème. Le Transfer Learning a connu un grand succès avec l'essor du *Deep Learning*. En effet, bien souvent, les modèles utilisés dans ce domaine nécessitent des temps de calcul élevés et des ressources importantes. Or, en utilisant des modèles pré-entraînés comme point de départ, le Transfer Learning permet de développer rapidement des modèles performants et résoudre efficacement des problèmes complexes en Computer Vision ou Natural Language Processing, NLP.[9]

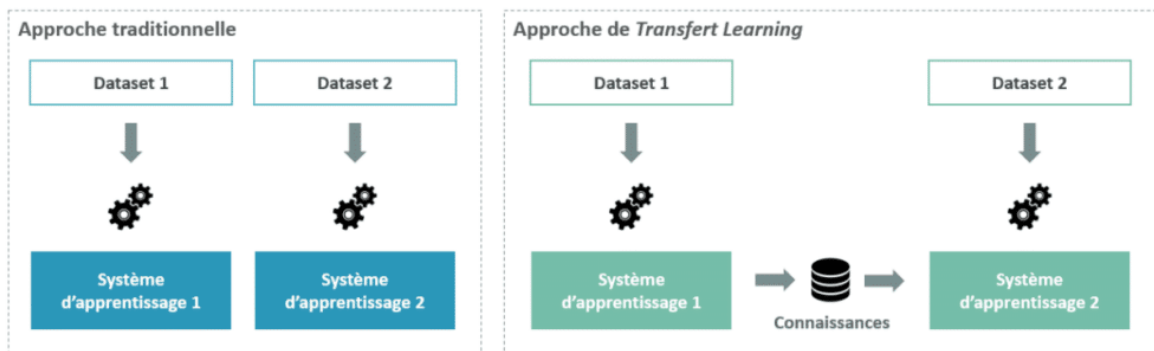


Figure 12 : Approche traditionnelle vs. Approche de Transfert Learning[9]

Intuitivement, le Transfer Learning s'inspire fortement du processus avec lequel nous apprenons. Prenons l'exemple de quelqu'un qui maîtrise la guitare et souhaite apprendre à jouer au piano. Il pourra capitaliser sur ses connaissances en musique pour apprendre à jouer un nouvel instrument. De la même manière, un modèle de reconnaissance de voiture pourra être très rapidement réadapté à la reconnaissance de camions.[9]

2.2 Sur quelles stratégies les techniques de Transfer Learning reposent ?

Le *Transfer Learning* repose sur une idée simple, celle de ré-exploiter les connaissances acquises dans d'autres configurations (sources) pour la résolution d'un problème particulier (cible). Dans ce contexte, on peut distinguer plusieurs approches selon ce que l'on souhaite transférer, quand et comment réaliser le transfert. Globalement, nous pouvons distinguer trois types de *Transfer Learning* [9]:

1. Apprentissage par transfert inductif, ou Inductive Transfer Learning

Dans cette configuration, les domaines source et cible sont les mêmes (même données), mais les tâches source et cible sont différentes mais proches. L'idée consiste alors à utiliser les modèles existants pour réduire de manière avantageuse le champ d'application des modèles possibles (biais de modèle) comme illustré dans la figure ci-dessous.[24]

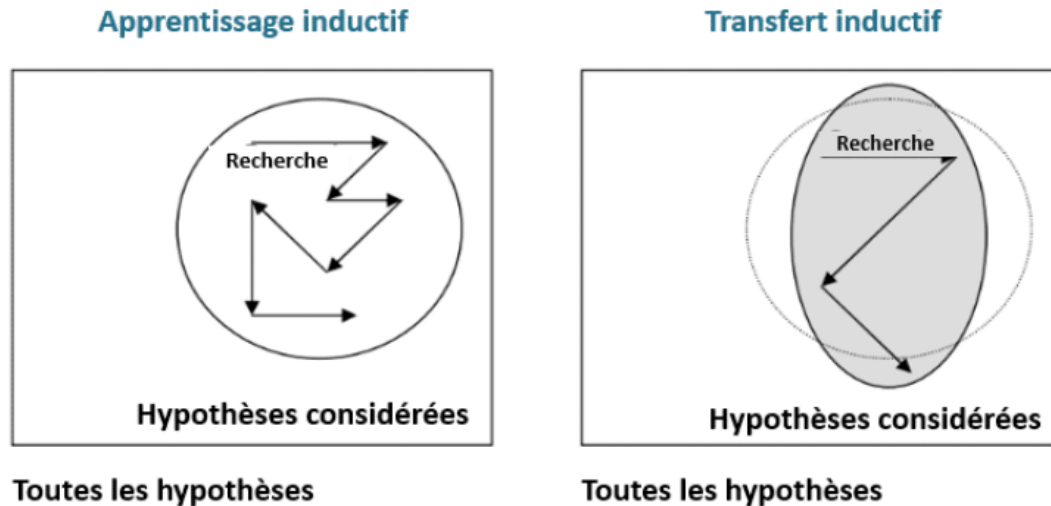


Figure 13 : Illustration de l'apprentissage par transfert inductif, ou Inductive Transfer Learning[16]

Par exemple, il est possible d'utiliser un modèle entraîné pour la détection d'animaux sur des images pour construire un modèle capable d'identifier des chiens.

2. Apprentissage par transfert non supervisé, ou Unsupervised *Transfer Learning*

Comme dans le cas de l'apprentissage par transfert inductif, les domaines source et cible sont similaires, mais les tâches sont différentes. Toutefois, les données des deux domaines ne sont pas labellisées.

Il est souvent plus facile d'obtenir de grandes quantités de données non labellisées, à partir de bases de données et de sources sur le web par exemple, que des données labellisées. C'est pourquoi l'idée d'utiliser l'apprentissage non supervisé en combinaison avec le *Transfer Learning* suscite un grand intérêt.[9]

3. Apprentissage par transfert transductif, ou *Transductive Transfer Learning*

Dans cette configuration, les tâches sources et cibles sont similaires, mais les domaines correspondants sont différents soit en termes de données ou de distributions de probabilités marginales. Par exemple, les modèles de *NLP*, tels que ceux utilisés pour l'étiquetage morpho-syntaxique de mots, Part-Of-Speech Tagger (POS Tagger) en anglais, sont généralement entraînés et testés sur des données d'actualité comme celles du *Wall Street Journal*. Ils peuvent

être adaptés aux données issues des réseaux sociaux dont le contenu est différent mais proche de celles des journaux.[9], [12]

2.3 Comment le Transfer Learning est mis en œuvre concrètement pour la résolution de problématiques de Deep Learning ?

Maintenant que nous avons défini le Transfer Learning, intéressons-nous à son application aux problématiques de Deep Learning, domaine dans lequel il connaît aujourd'hui un grand succès. L'utilisation des méthodes de Transfer Learning en Deep Learning consiste principalement à exploiter des réseaux de neurones pré-entraînés. Généralement, ces modèles correspondent à des algorithmes très performants qui ont été développés et entraînés sur de larges bases de données et qui sont aujourd'hui librement partagés. Dans ce contexte, on peut distinguer 2 types de stratégies :

A. Utilisation de modèles pré-entraînés comme extracteurs de *features*

L'architecture des modèles de *Deep Learning* se présente très souvent sous la forme d'un empilement de couches de neurones. Ces couches apprennent différentes caractéristiques en fonction du niveau où elles se situent. La dernière couche (généralement une couche entièrement connectée, dans le cas de l'apprentissage supervisé) est utilisée pour obtenir la sortie finale. La figure ci-dessous illustre l'architecture d'un modèle de *Deep Learning* utilisé pour la détection de chat/chien. Plus la couche se situe en profondeur, plus elle permet d'extraire des *features* spécifiques[13].

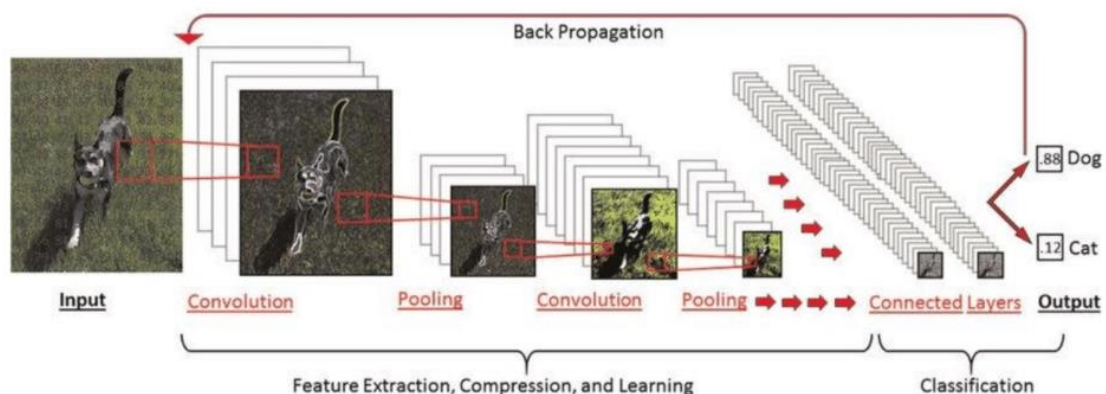


Figure 14 : Exemple de modèle de réseau de neurones utilisé pour la classification d'images[5]

L'idée est donc de réutiliser un réseau pré-entraîné sans sa couche finale. Ce nouveau réseau fonctionne alors comme un extracteur de features fixes pour la réalisation d'autres tâches.

B. Ajustement de modèles pré-entraînés

Il s'agit d'une technique plus complexe, dans laquelle non seulement la dernière couche est remplacée pour réaliser la classification ou la régression, mais d'autres couches sont également ré-entraînés de manière sélective. En effet, les réseaux neuronaux profonds sont des architectures hautement configurables avec divers hyperparamètres. De plus, alors que les premières couches capturent les caractéristiques génériques, les dernières couches se concentrent davantage sur la tâche spécifique à accomplir, comme le montre la figure-ci-dessous[5].

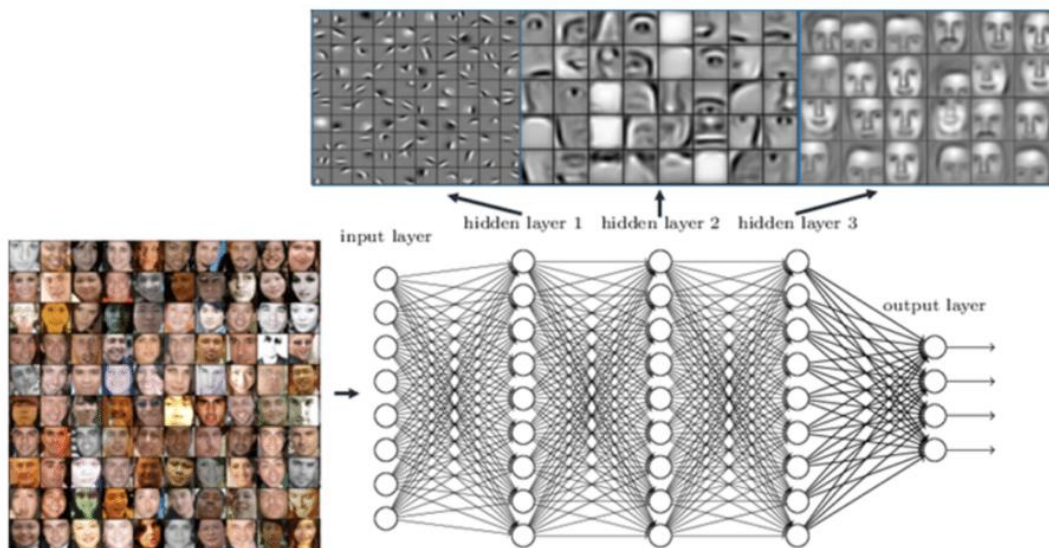


Figure 15 : Exemple de modèle de réseau de neurones utilisé pour reconnaissance faciale[3]

L'idée est donc de geler (c'est-à-dire fixer les poids) de certaines couches pendant l'entraînement et affiner le reste pour répondre à la problématique. Cette stratégie permet de réutiliser les connaissances en termes d'architecture globale du réseau et d'exploiter ses états comme point de départ pour l'entraînement. Elle permet donc d'obtenir de meilleures performances avec un temps d'entraînement plus court.

3 Conclusion

Dans ce chapitre, on a introduit les réseaux de neurones convolutifs, se principales composantes qui sont les layers, la méthode d'apprentissage de CNN et comment appliquer le transfer learning.

CHAPITRE 03 :

Développement et Implémentation

Chapitre 3 : Développement et implémentation

Ce chapitre décrit les différentes phases du développement de la solution. On a détaillé toutes les étapes qu'on a suivi jusqu'à aboutir au final application.

1 Les outils utilisés

On a développé cette application avec le langage Python[25]. Ce langage est utilisé dans le domaine de machine learning car il facilite l'écriture du code complexes avec des simples lignes. Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

Pour les bibliothèques utilisées pour implémenter le CNN modèle, on a utilisé Tensorflow et keras. Tensorflow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels [26].

Pour la création de l'interface graphique, on a utilisé PyQt5. Il s'agit d'un framework applicatif multiplateforme : une même application PyQt fonctionne sans problème sous OS X, Linux et Windows. C'est aussi un gros module de plus de trois cents classes, de six mille fonctions. Il fournit un ensemble complet de classes utiles qui fournissent des interfaces pour l'accès aux fichiers, parser du XML, des structures de données et plus, ainsi que des outils comme Qt Designer, Qt Assistant et Qt Linguist pour concevoir des interfaces, accéder à l'aider et faciliter les traductions. C'est donc un framework complet : il fournit des classes qui gèrent les types basiques de données comme celles qui effectuent des tests unitaires[27].

2 Data Set

Le dataset fait une partie très importante de toute application basée sur l'intelligence artificielle. Le dataset est toujours divisé en en 2 parties : le training set (base d'apprentissage pour réaliser la phase d'entraînement) et le testing set réservé au test ou évaluation du modèle. Le dataset est mélangé de manière aléatoire pour extraire les données pour le training set et le testing set. Dance ce travail, on a positionné 80 % des images dans le training set les 20 % restant dans le testing set[8].

Avant d’entamer la programmation, on avait besoin de collecter notre dataset qui consiste de images de différentes classes. Pour cela, on a utilisé un scraper écrit en python. Ce scraper parcourt la toile en émulant un navigateur web pour se connecter au moteur de recherche Bing (Bing 2020). Le script effectue des recherches par mot clé puis télécharge directement les fichiers sur le poste depuis lequel il est exécuté. Pour utiliser le scraper, il faut définir le dossier cible et la liste de mots clés. Il suffit ensuite de lancer le script à l'aide de l'interpréteur Python.

```

62 def run_scraper(self):
63     for subject in self._LST_SUBJECTS:
64         i = 0 #init
65         while i < self._INDEX_MAX:
66             time.sleep(random.random()) #random sleep to avoid to be blacklist by search engine
67             self._scrape_images_by_page(subject, self._BROWSER_HEADER, str(i), self._DIRECTORY_TARGET)
68             i += 30 #increment by 30 because it's ~ nb images by bing page result
69         print('END of the programme')
70
71 if __name__ == "__main__":
72     directory_target = "armes"
73     lst_subjects = ['pistolet', 'revolver', 'fusil'] #target subject
74     browser_header = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2226.150 Safari/537.36'}
75     index_max = 1000 #~ nb max images we want, warning search engine is limited for number of images
76
77     ScraperImages(directory_target, lst_subjects, browser_header, index_max).run_scraper()
78
79 if __name__ == "__main__":
80     directory_target = "argent"
81     lst_subjects = ['Billets_Banque', 'pieces'] #target subject
82     browser_header = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2226.150 Safari/537.36'}
83     index_max = 1000 #~ nb max images we want, warning search engine is limited for number of images
84
85     ScraperImages(directory_target, lst_subjects, browser_header, index_max).run_scraper()
86
87

```

Figure 16 : Code source et exécution du scraper. (Réalisé à l’aide de l’outil : Notepad++)

Le problème de classification qu’on va résoudre consiste de sept classes. Le nom des classes et le nombre des images utilisés dans la phase de training et testing est donné ci-dessous :

	Nom de la classe	Training set	Testing set
Argent	Billets	200	50
	Pièces	200	50
Armes	Fusil	200	50
	Pistolet	200	50
	Revolver	200	50
Drogue	Amphétamines	200	50
	Cocaïne	200	50

Tab1. Le nom de classe, le nombre de training et testing images par classe.

Notez que durant l'exécution de le scraper, il peut être que certaines images n'ont aucune relation avec les classes. Pour cela, un cleaning (nettoyage) manuel est fait pour garder que les images pertinentes.

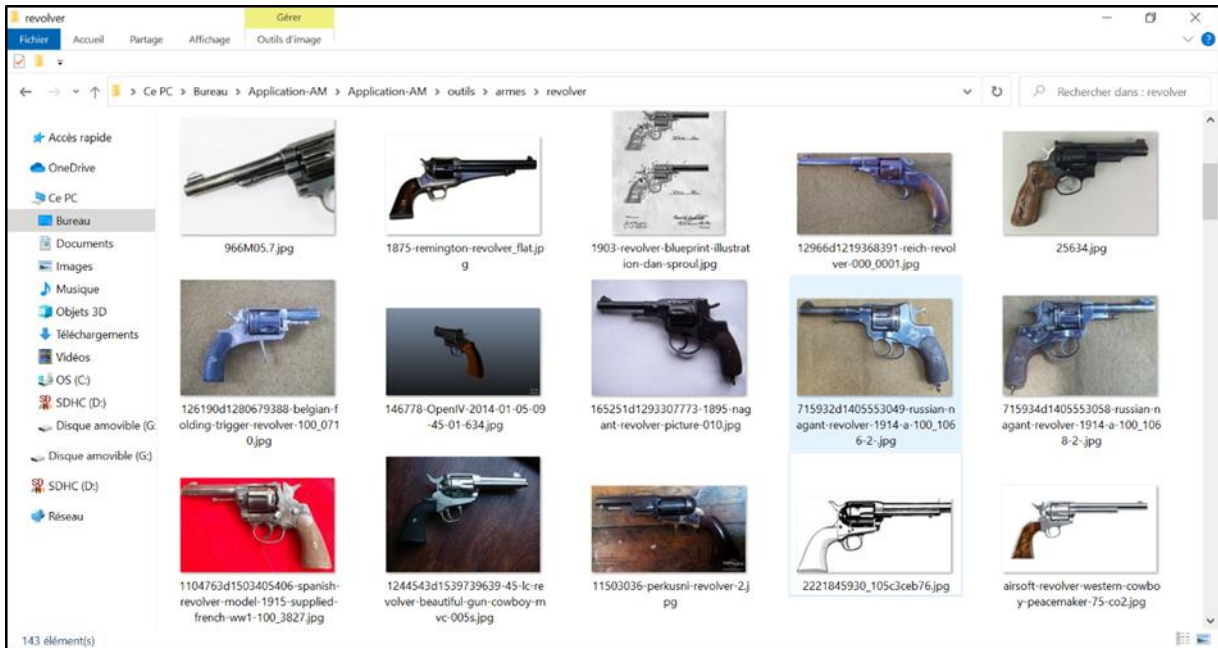


Figure 17 : Création de dossier d'images pour chaque classe.



Figure 18 : Nettoyage manuel des images

3 AlexNet Network

Le développement du modèle de classification constitue la majeure partie du développement de l'application. Il consiste à mettre en place la partie de l'application permettant de détecter des objets dans des images. Pour réaliser cela, on a utilisé le transfer learning[28]. Étant donné qu'un bon nombre des meilleurs modèles utilisent des millions d'instances de training et prennent des semaines à s'exécuter sur des ressources de calcul robustes, il est difficile pour nous de développer notre modèle car le nombre d'images qu'on a est très modeste. Heureusement, nous pouvons incorporer des parties des modèles déjà pré-entraîné comme AlexNet dans un modèle complètement différent et spécifique à un domaine.

AlexNet a été développé en 2012 et a constitué une percée majeure dans le développement de CNN. Non seulement il était le gagnant du concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC) de cette année, mais il avait environ la moitié du taux d'erreur dans ce concours. L'architecture d'AlexNet contient 60 000 paramètres au total dans 8 couches totales : cinq couches convolutives et trois couches entièrement connectées. De plus, le modèle utilisait les fonctions d'activation ReLU.[6]

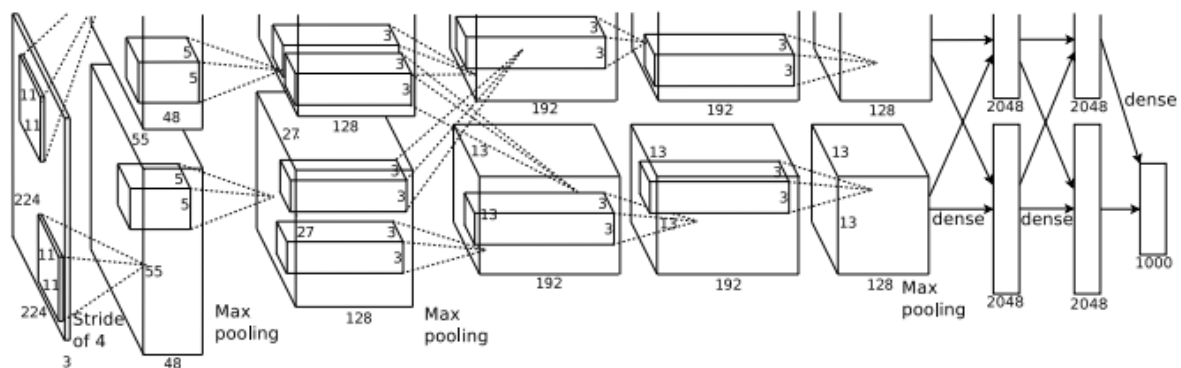


Figure 19 : L'architecture d' AlexNet Network.[4]

L'ensemble de données ImageNet complet contient plus de 15 millions d'images et 22 000 étiquettes de classe, ce qui est bien plus grand qu'un ensemble de données typique utilisé pour l'entraînement. Cela seul peut créer un classificateur assez précis lorsque on travaille avec des images d'objets ordinaires que l'ensemble de données ImageNet a peut-être déjà vu. Mais en conjonction avec l'apprentissage par transfert pour l'adaptation de domaine, on peut augmenter considérablement la performance de notre modèle.

Layer (type)	Output Shape	Param #	Previous Layer
data(null)		0	
alexnet0_conv0_fwd(Convolution)		64	data
alexnet0_conv0_relu_fwd(Activation)		0	alexnet0_conv0_fwd
alexnet0_pool0_fwd(Pooling)		0	alexnet0_conv0_relu_fwd
alexnet0_conv1_fwd(Convolution)		192	alexnet0_pool0_fwd
alexnet0_conv1_relu_fwd(Activation)		0	alexnet0_conv1_fwd
alexnet0_pool1_fwd(Pooling)		0	alexnet0_conv1_relu_fwd
alexnet0_conv2_fwd(Convolution)		384	alexnet0_pool1_fwd
alexnet0_conv2_relu_fwd(Activation)		0	alexnet0_conv2_fwd
alexnet0_conv3_fwd(Convolution)		256	alexnet0_conv2_relu_fwd
alexnet0_conv3_relu_fwd(Activation)		0	alexnet0_conv3_fwd
alexnet0_conv4_fwd(Convolution)		256	alexnet0_conv3_relu_fwd
alexnet0_conv4_relu_fwd(Activation)		0	alexnet0_conv4_fwd
alexnet0_pool2_fwd(Pooling)		0	alexnet0_conv4_relu_fwd
alexnet0_flatten0_flatten0(Flatten)		0	alexnet0_pool2_fwd
alexnet0_dense0_fwd(FullyConnected)		4096	alexnet0_flatten0_flatten0
alexnet0_dense0_relu_fwd(Activation)		0	alexnet0_dense0_fwd
alexnet0_dropout0_fwd(Dropout)		0	alexnet0_dense0_relu_fwd
alexnet0_dense1_fwd(FullyConnected)		4096	alexnet0_dropout0_fwd
alexnet0_dense1_relu_fwd(Activation)		0	alexnet0_dense1_fwd
alexnet0_dropout1_fwd(Dropout)		0	alexnet0_dense1_relu_fwd
alexnet0_dense4_fwd(FullyConnected)		10	alexnet0_dropout1_fwd
Total params: 9354			

Figure 20 : Les différentes couches d' AlexNet Network.

Nous avons entraîné le modèle pour 15 epochs en utilisant l'activation Adam comme optimiseur, un taux d'apprentissage constant de 0,001 et une erreur de perte d'entropie croisée softmax comme fonction de perte. Le nouveau modèle entraîne est sauvegarde après pour faire des nouvelles prédictions. Toutes les images sont redimensionnées en 224×224.

4 Tester le modèle

Après entrainer le modèle pré-entraîne d'AlexNet[4], on l'avait testé sur notre dataset de testing pour estimer sa performance. On a ploté la matrice de confusion pour avoir la performance du modèle pour chaque classe. La performance achevée avec ce modèle est 82.23%.

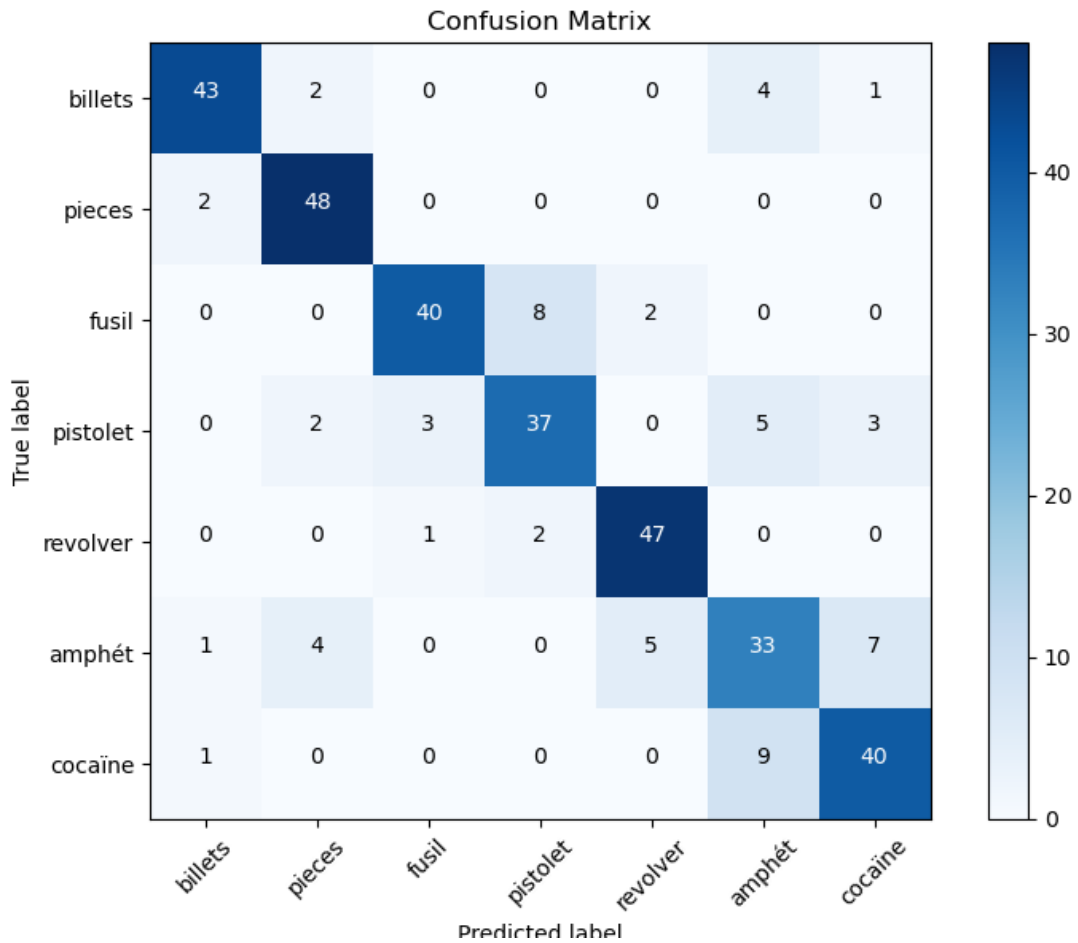


Figure 21 : La matrice de confusion.

5 Interface graphique :

Après avoir développé le modèle qui se charge de classer les images, la dernière étape du développement a été de mettre au point une interface graphique permettant à l'utilisateur final d'utiliser facilement l'application. Cette interface est conçue pour être simple et facile d'utilisation.

5.1 Composition de l'interface graphique :

L'interface graphique est composée d'une simple et unique fenêtre. Les fonctionnalités sont données ci-dessous :

1	Load data for analysis	Charger des fichiers dans le dépôt d'images.
2	Start object detection	Lancer l'analyse (affiche la progression durant l'analyse) ou le modèle entraîné est utilisé pour faire les prédictions.
3	Show statistics	Afficher les statistiques relatives aux résultats
4	Access processed images	Afficher et accéder aux résultats
5	Quit	Quitter l'application

Tab2. Les fonctionnalités de l'interface graphique

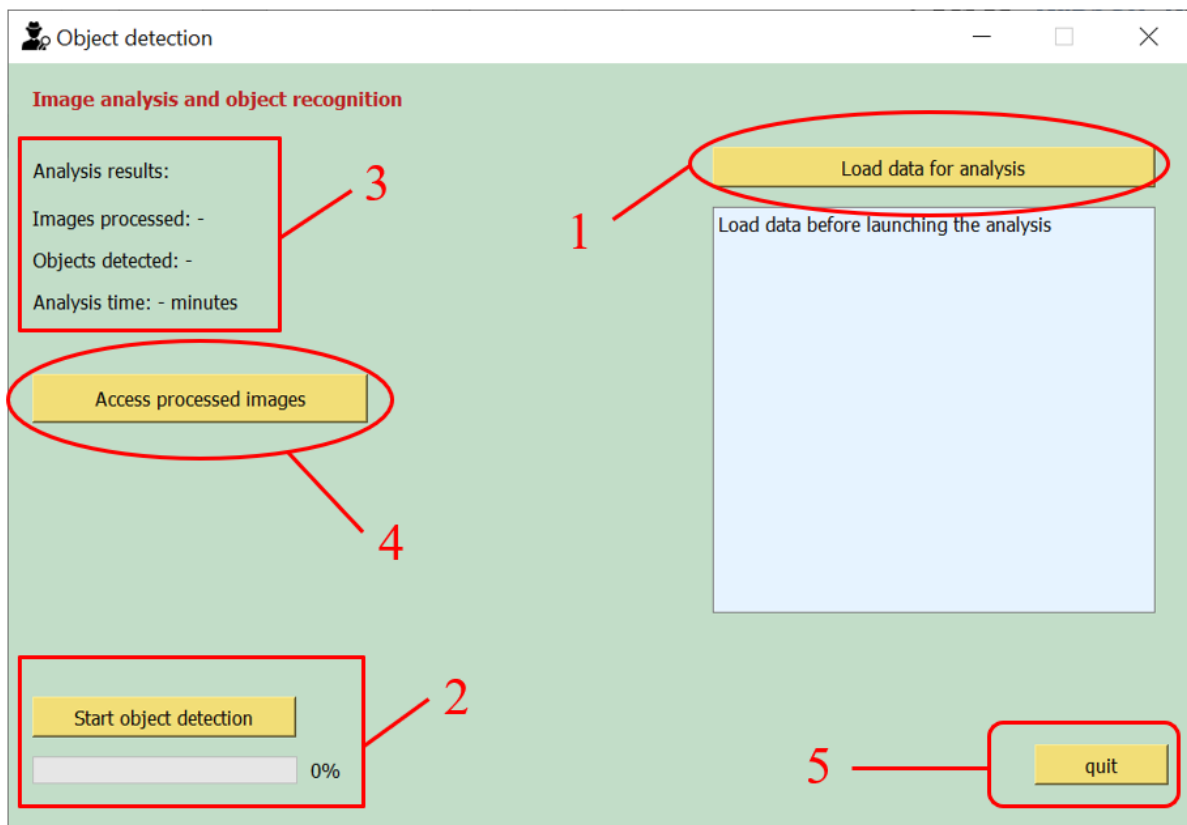


Figure 22 : Interface graphique au démarrage du programme

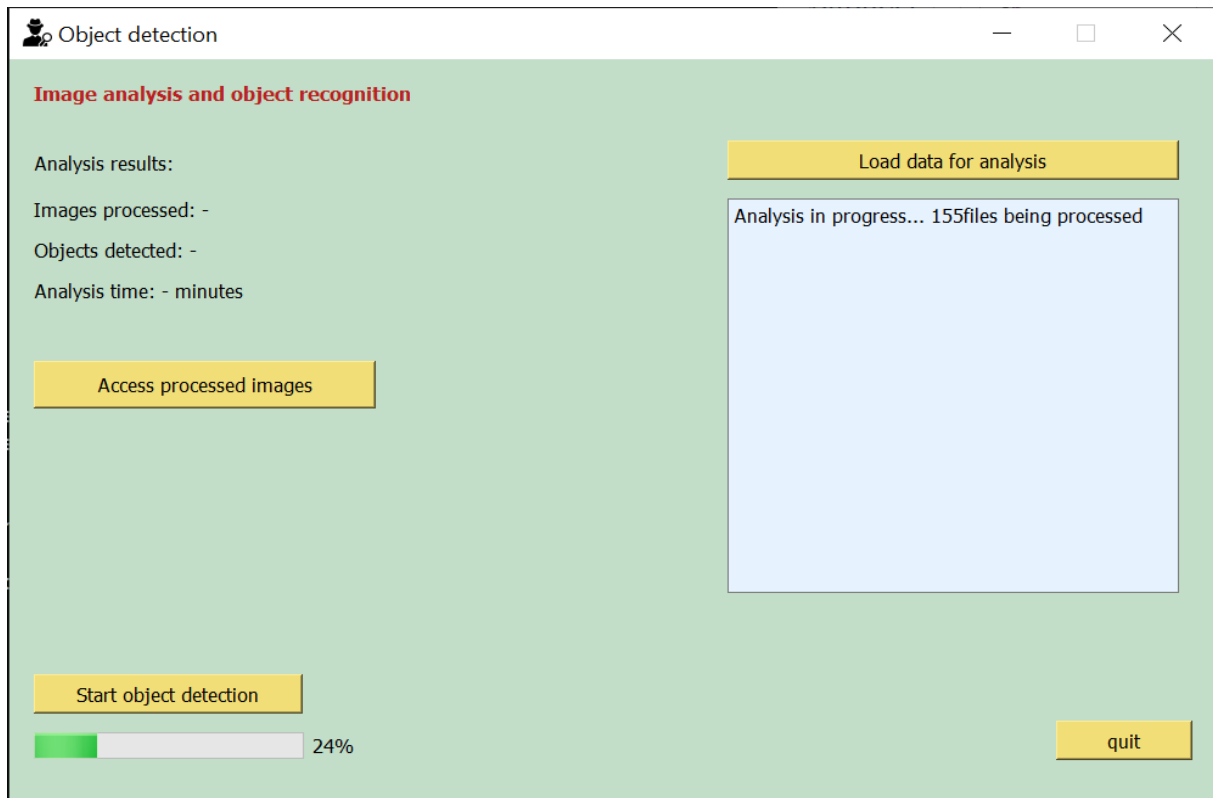


Figure 23 : Interface graphique du programme pendant l'exécution de l'analyse

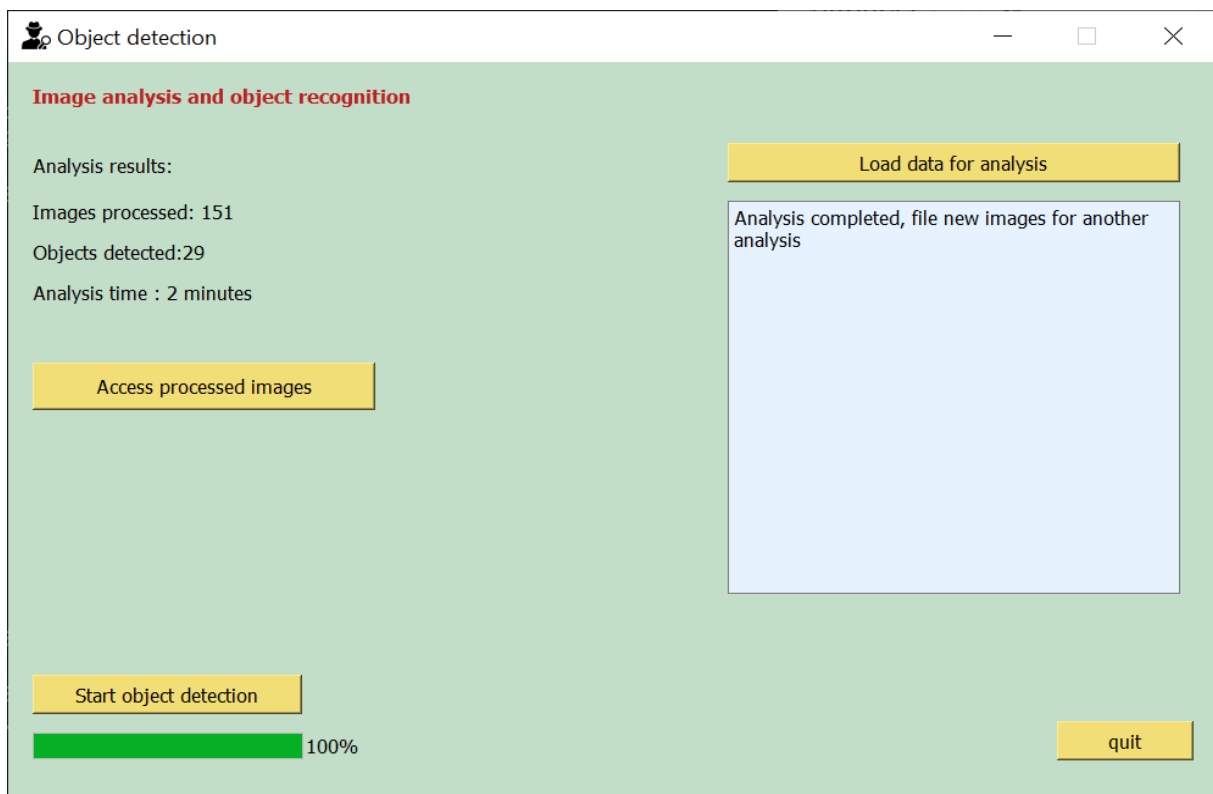


Figure 24: Interface graphique du programme après l'analyse

L'icône de l'application (figure ci-dessous) a été réalisée à l'aide du logiciel Adobe Illustrator CC 2019



Figure 25 : Icône de l'application

Après l'exécution du programme, pour chaque type de classe, un dossier est créé et les images classées sont mises dedans. Dans le cas où l'image n'appartient à aucune classe, un dossier spécial est créé pour ce type d'images.

6 Conclusion :

Dans ce chapitre, on a introduit les outils utilisés pour le développement de cette application. La première partie consiste à développer le modèle qui va faire les prédictions des images fournies. La deuxième partie consiste à développer une simple interface graphique pour faciliter le travail de l'utilisateur. Nous avons présenté notre application en donnant quelques captures d'écran qui expliquent le déroulement et le fonctionnement de notre travail, ainsi que les résultats obtenus.

Conclusion générale :

La classification d'images est une tâche importante dans le domaine de la vision par ordinateur, la reconnaissance d'objets et l'apprentissage automatique. Bien que les capacités des activités réalisées dans le domaine de classification des images soient nombreuses, aucune méthode n'est jugée fiable à 100%, mais au fur et à mesure les nouveaux travaux essaient d'améliorer la performance pour des meilleurs résultats. C'est dans ce cadre que s'inscrit notre travail, qui a pour objectif de proposer une application qui réalise une classification d'une base d'images en différentes classes pour aider la police à faire une classification initiale des images trouvées dans les téléphones portables des suspects. On a adressé ce problème de classification en utilisant le deep Learning, la méthode d'apprentissage qui a montré ses performances ces dernières années et nous avons choisi la méthode CNNs comme méthode de classification pour sa simplicité et son efficacité. Le résultat obtenu après l'application de transfer learning pour faire apprendre notre modèle de distinguer les différentes classes prouve l'efficacité de cette approche.

Bibliographie

- [1] I. Goodfellow, Y. Bengio, A. Courville, et Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [2] J. D. Kelleher, *Deep learning*. MIT press, 2019.
- [3] R. Idoui, A. Himeur, et S. E. Bouatmane, « Reconnaissance d'images par les réseaux de neurones convolutionnels (CNN) », PhD Thesis, Université de Jijel, 2020.
- [4] A. Abd Almisreb, N. Jamil, et N. M. Din, « Utilizing AlexNet deep transfer learning for ear recognition », in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, 2018, p. 1-5.
- [5] D. Monnin, « Réseaux de neurones cellulaires et traitement d'images: une approche analytique de la conception d'opérateurs », PhD Thesis, Grenoble 1, 2003.
- [6] L. Zhu, Z. Li, C. Li, J. Wu, et J. Yue, « High performance vegetable classification from images based on alexnet deep learning model », *Int. J. Agric. Biol. Eng.*, vol. 11, n° 4, p. 217-223, 2018.
- [7] la rédaction de Futura, « Intelligence artificielle », *Futura*. <https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/> (consulté le juin 13, 2021).
- [8] « Machine Learning et Big Data : définition et explications de la combinaison ». <https://www.lebigdata.fr/machine-learning-et-big-data> (consulté le juin 13, 2021).
- [9] « Transfer Learning | Formation Data Science | DataScientest.com ». <https://datascientest.com/transfer-learning> (consulté le juin 13, 2021).
- [10] G. Bochra et S. Safia, « Les réseaux de neurone convolutionnel (CNN) pour la classification des images associées aux places de stationnement d'un parc de véhicule », 2019.
- [11] J. Creusier et F. Biétry, « Analyse comparative des méthodes de classifications », *RIMHE Rev. Interdiscip. Manag. Homme Entrep.*, vol. n° 10, 3, n° 1, p. 105-123, janv. 2014.
- [12] C. Belloni, « Deep learning and featured-based classification techniques for radar imagery », phdthesis, Ecole nationale supérieure Mines-Télécom Atlantique, 2019. Consulté le: juin 13, 2021. [En ligne]. Disponible sur: <https://tel.archives-ouvertes.fr/tel-02945414>
- [13] L. Deng et D. Yu, « Deep learning: methods and applications », *Found. Trends Signal Process.*, vol. 7, n° 3-4, p. 197-387, 2014.
- [14] « TP CNN architecture (1) », *Pensée Artificielle*. <https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/tp-cnn-architecture-1/> (consulté le juin 13, 2021).
- [15] « Réseau neuronal convolutif », *Wikipédia*. avr. 15, 2021. Consulté le: juin 13, 2021. [En ligne]. Disponible sur:

https://fr.wikipedia.org/w/index.php?title=R%C3%A9seau_neuronal_convolutif&oldid=181914105

[16] « Qu'est ce qu'un réseau de neurones convolutif (ou CNN) ? », *OpenClassrooms*. <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn> (consulté le juin 13, 2021).

[17] « Apprenez à construire un CNN et gagnez du temps avec le Transfer Learning », *OpenClassrooms*. <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning> (consulté le juin 13, 2021).

[18] « DataScienceToday - Les réseaux de neurones convolutifs », <https://datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs> (consulté le juin 13, 2021).

[19] « Découvrez les différentes couches d'un CNN - Classez et segmentez des données visuelles - OpenClassrooms », <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn> (consulté le juin 13, 2021).

[20] « Découvrez les différentes couches d'un CNN », *OpenClassrooms*. <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn> (consulté le juin 13, 2021).

[21] « Figure 3.13 -Allure de la fonction ReLU », *ResearchGate*. https://www.researchgate.net/figure/Allure-de-la-fonction-ReLU_fig21_344266838 (consulté le juin 13, 2021).

[22] « Découvrez les différentes couches d'un CNN », *OpenClassrooms*. <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn> (consulté le juin 13, 2021).

[23] « Deep learning : les réseaux de neurones convolutifs pour la classification d'images médicales », *IMAIOS*. <https://www.imaios.com/fr/Societe/blog/Classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-CNN> (consulté le juin 13, 2021).

[24] « Transfer Learning », *Formation Data Science / DataScientest.com*, juill. 21, 2020. <https://datascientest.com/transfer-learning> (consulté le juin 13, 2021).

[25] B. Jiang *et al.*, « Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues », *Artif. Intell. Agric.*, vol. 1, p. 1-8, 2019.

[26] A. Gambardella, A. G. Baydin, et P. H. S. Torr, « Transflow Learning: Repurposing Flow Models Without Retraining », *ArXiv191113270 Cs Stat*, déc. 2019, Consulté le: juin 13, 2021. [En ligne]. Disponible sur: <http://arxiv.org/abs/1911.13270>

[27] « Introduction aux interfaces graphiques en Python avec Qt 5 et PyQt5 — Cours Python ». <https://courspython.com/interfaces.html> (consulté le juin 13, 2021).

[28] S. Lu, Z. Lu, et Y.-D. Zhang, « Pathological brain detection based on AlexNet and transfer learning », *J. Comput. Sci.*, vol. 30, p. 41-47, 2019.