Recherche de l'information dans les réseaux de neurones convolutifs pré-entraînés

par

Mohsen BEN LAZREG

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M. Sc. A.

MONTRÉAL, LE 02 OCTOBRE 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Matthew Toews, directeur de mémoire
Département de génie des systèmes, École de technologie supérieure

M. Simon Drouin, président du jury
Département de génie logiciel et des technologies de l'information, École de technologie supérieure

M. Alessandro Lameiras Koerich, Examinateur Externe
Département de génie logiciel et des technologies de l'information, École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE "09 SEPTEMBRE 2020"

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

C’est avec grand plaisir que je réserve ces quelques lignes en reconnaissance de tous ceux qui ont contribué à la réalisation de ce mémoire. Tout d’abord, je voudrais exprimer ma gratitude à mon directeur de recherche, le professeur Matthew Toews, pour sa disponibilité et ses conseils judicieux, qui ont contribué à ma réflexion. Je le remercie pour son attention aux détails lors de son encadrement et la vérification de mes progrès ainsi que pour ses encouragements tout au long de la période de mes études.

Je voudrais exprimer ma gratitude aux collègues de Laboratoire d’imagerie, de vision et d’intelligence artificielle (LIVIA).

Je remercie mon collègue Laurent Chauvin pour ses collaborations, contributions de recherche et les moments de partage que nous avons vécu.

Je voudrais également exprimer ma reconnaissance à ma famille qui m’ont permis d’arriver à ce niveau et mes soeurs pour leur patience et leurs encouragements. Je dédie ce travail à l’âme de mon père qui m’a quitté le 14 octobre 2019.

À tous les membres du jury, j’offre mes remerciements, mon respect et ma gratitude.

Recherche de l'information dans les réseaux de neurones convolutifs pré-entraînés

Mohsen BEN LAZREG

RÉSUMÉ

Cette thèse évalue l'utilisation de réseaux neuronaux convolutifs (CNN) pré-entraînés comme extracteurs de caractéristiques génériques pour de nouveaux contextes de classification d'images, une stratégie connue sous le nom d'apprentissage par transfert. Un certain nombre de questions de recherche ouvertes sont abordées, étant donné la disponibilité croissante de diverses architectures de CNN pré-entraînés de haute performance pour l'apprentissage par transfert : Quels sont les réseaux et les couches d'activation de réseau les plus efficaces pour l'apprentissage par transfert ? Est-il possible de combiner différents réseaux pour améliorer la classification ? Comment l'efficacité de la classification diffère-t-elle selon les contextes de données, c'est-à-dire selon les grandes catégories visuelles (par exemple, bâtiments, voitures) et les contextes d'imagerie spécifiques (par exemple, photos du visage ou du cerveau de la même personne ou des membres de la même famille) ?

Un modèle de classification générique basé sur la mémoire est proposé afin d'évaluer et de comparer la précision des architectures CNN, où des cartes d'activation génériques provenant de réseaux arbitraires servent de caractéristiques d'image et où la classification est réalisée par l'indexation du plus proche voisin.

Un certain nombre de modèles de mise en commun et de normalisation des caractéristiques sont évalués, notamment la mise en commun (pooling) maximale, moyenne et moyenne généralisée, et les modèles de normalisation comprennent des cartes d'activation brutes et la normalisation L2. Enfin, un schéma de codage binaire des caractéristiques est proposé pour comprimer les données et améliorer la précision de la classification, où les caractéristiques d'activation individuelles sont binarisées afin de maximiser le gain d'informations. Comme base de référence supplémentaire, la classification est également évaluée à l'aide de caractéristiques d'images traditionnelles extraites via la transformation de caractéristiques invariantes à l'échelle (SIFT).

L'évaluation compare une liste importante et complète d'architectures CNN existantes, toutes pré-entrainées sur l'ensemble de données standard ImageNet (1000 catégories x 1000 images), y compris VGG, Inception, ResNet, Xception, DenseNet, MobileNet,NasLarge, NasMobile etc. Afin d'éviter tout biais potentiel vers les données utilisées dans la préformation des CNN, les expériences de classification sont basées sur des ensembles de données d'images indépendants et des catégories sans rapport avec les données ImageNet. Celles-ci comprennent de larges catégories visuelles issues de l'ensemble de données Caltech101 (Fei-Fei *et al.* (2004)), et des contextes spécifiques comprenant des images de visages humains issues de l'ensemble de données FERET (Phillips *et al.* (1998)) et des images de résonance magnétique du cerveau humain issues du projet Human Connectome (HCP) (Van Essen *et al.* (2013)).

Pour les catégories générales (données Caltech101), les précisions les plus élevées pour chaque réseau vont de 73,29% à 93,02% pour les réseaux (NasMobile couche 739) et (DenseNet201

couche 704), respectivement, ce qui est cohérent aux résultats de l'état de l'art. La concaténation de couches de haute précision provenant de différents réseaux augmente généralement la précision, la précision la plus élevée (94,01%) a été obtenue en combinant (Xception, Resnet, DenseNet, et InceptionResNetV2).

Pour les cas de visages spécifiques (données FERET), la plus grande précision de reconnaissance (parfaite à 100 %) est obtenue à partir des réponses des filtres dans les couches du réseau et pour la correspondance SIFT 2D. Les réponses des filtres à la sortie du réseau sont moins précises (98%).

La binarisation améliore la précision de la classification par sexe (InceptionV3 avec binarisation, AUC=0,981) par rapport à (InceptionV3 sans binarisation, AUC=0,966), ce qui est supérieur à la SIFT 2D (AUC=0,926). Pour les cas de cerveau humain (données HCP), les fonctions CNN préformées combinées aux fonctions SIFT 3D permettent d'obtenir une précision de pointe pour la classification binaire des sexes (DenseNet201 avec binarisation, AUC=0,987), et la classification des membres de famille pour 1010 sujets et 400 familles est (DenseNet201 avec binarisation, AUC=0,925).

Mots-clés: modèles pré-entraînés ,apprentissage par transfert, classification des images, caractéristiques binaires, CNN, SIFT

Searching for information in pre-trained convolutional neural networks

Mohsen BEN LAZREG

ABSTRACT

This thesis evaluates the use of pre-trained convolutional neural networks (CNNs) as generic feature extractors for new image classification contexts, a strategy known as transfer learning. A number of open research questions are addressed, given the increasing availability of diverse, high performance, pre-trained CNN architectures for transfer learning : Which networks and network activation layers are most effective for transfer learning ? Can different networks be combined to improve classification ? How does the effectiveness of classification differ across data contexts, i.e. broad visual categories (e.g. buildings, cars) vs. specific imaging contexts (e.g. face or brain scans of the same person or family members) ?

A generic memory-based classification model is proposed in order to evaluate and compare the accuracy of CNN architectures, where generic activation maps from arbitrary networks serve as image features and classification is achieved via nearest neighbor indexing. A number of feature pooling and normalization schemes are evaluated, including maximum, average and generalised mean pooling, and normalisation schemes include raw activation maps, L2 normalization. Finally, a binary feature encoding scheme is proposed to compress data and improve classification accuracy, where individual activation features are binarized in order to maximize information gain. As an additional baseline, classification is also evaluated using traditional hand-crafted image features extracted via the scale-invariant feature transform (SIFT).

Evaluation compares a large, comprehensive list of existing CNN architectures, all pre-trained on the standard ImageNet dataset (1000 categories x 1000 images), including VGG, Inception, ResNet, Xception, DenseNet, MobileNet, NasLarge, NasMobile etc. In order to avoid potential bias towards data used in CNN pre-training, classification experiments are based upon independent image datasets and categories unrelated to ImageNet data. These include broad visual categories from Caltech101 dataset (Fei-Fei *et al.* (2004)), and of specific contexts including human face images from the FERET dataset (Phillips *et al.* (1998)) and brain magnetic resonance images of the human brain from the Human Connectome Project (HCP) (Van Essen *et al.* (2013)).

For general categories (Caltech101 data), the highest accuracies for each network range from (73.29% to 93.02%) for networks (NasMobile layer 739) and (DenseNet201 layer 704), respectively, consistent with state-of-the-art performance. Concatenating high-accuracy layers from different networks generally increases accuracy, the highest accuracy (94.01%) was achieved combining (Xnception-Resnet-DenseNet-InceptionResNetV2).

For specific face instances (FERET data), the highest recognition accuracy (perfect 100%) is achieved from filter responses within network layers and for 2D SIFT matching. Filter responses at the network output are less accurate (98%). Binarization improves the accuracy of gender classification (InceptionV3 with binarization, AUC=0.981) vs. (InceptionV3 without

binarization, AUC=0.966), improving upon 2D SIFT (AUC=0.926). For human brain instances (HCP data), pre-trained CNN features combined with 3D SIFT features achieves state-of-the-art accuracy for binary gender classification (DenseNet201 with binarization, AUC=0.987), and family member classification for 1010 subjects and 400 families is (DenseNet201 with binarization, AUC=0.925).

Keywords: pre-trained networks, transfert learning, image classification, binary features, CNN, SIFT

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ETS	École de Technologie Supérieure
CNN	Réseau de neurones convolutifs (Convolutional Neural Networks)
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
ReLU	Rectified Linear Unit
ANN	Réseau de neurones artificielles (Artificial Neural network)
GPU	Graphics Processing Unit
IA	Intelligence Artificielle
GMAX	Global MAX pooling
GAVG	Global Average pooling
GeM	Generalized Mean pooling
VGG	Oxford Vision Geometry Group
SIFT	Scale-Invariant Feature Transform
PCA	Principal Component Analysis
SURF	Speeded up robust features
ORB	Oriented FAST and Rotated BRIEF
BoW	Bag of words
FV	Feature Vector
MAC	Maximum Activation
IG	Information Gain
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
FERET	Facial Recognition Technology
ID	Identifiant
SPP	Spatial Pyramid Pooling

INTRODUCTION

L'apprentissage et la classification basés sur les images utilisent le plus souvent des filtres de convolution pour identifier les informations pertinentes sur les images, par exemple les réseaux de neurones convolutifs (CNN) LeCun *et al.* (1989), où les filtres sont ajustés au contenu d'intérêt. À l'époque, il était impossible de calculer l'algorithme de rétro-propagation Rosenblatt (1958), Rumelhart *et al.* (1986) sur une grande base de données d'images, car les performances des ordinateurs étaient très limitées donc plusieurs recherches se sont intéressées à l'utilisation des filtres conçus manuellement, comme par exemple, Harris *et al.* (1988) , Lindeberg (1994), SIFT (Lowe (2004)), etc. Depuis 2012 cependant, la méthodologie la plus utilisée consiste à former des séries de filtres de réseaux neuronaux convolutifs (CNN) via l'algorithme de rétro-propagation (Rumelhart *et al.* (1986)), où la mise en œuvre d'unités de traitement graphique (GPU) hautement parallélisées permet de dériver des filtres de convolution à partir de grands ensembles de données d'images, par exemple ImageNet (Deng *et al.* (2009)).

Malgré l'utilisation des GPU, l'entraînement de CNN est difficile pour plusieurs raisons. La rétropropagation Rumelhart *et al.* (1986) reste un processus à forte intensité de calcul, où de nombreuses itérations et des hyperparamètres soigneusement choisis sont nécessaires pour assurer la convergence du modèle. En outre, un grand nombre d'images étiquetées sont nécessaires pour l'entraînement, et dans de nombreux cas, les données sont insuffisantes pour estimer les paramètres d'un grand réseau.

En raison de ces défis, un grand nombre de méthodes (Shin *et al.* (2016), Deniz *et al.* (2018), Kaur & Gandhi (2020)) évitent l'entraînement de CNN à partir de zéro et utilisent plutôt des réseaux existants et préformés comme extracteurs de caractéristiques générales d'images, une méthodologie connue sous le nom de transfert d'apprentissage. De cette manière, les nouvelles données d'images peuvent être encodées en termes de réponses génériques d'activations de réseau, et classées via des modèles d'apprentissage standard basés sur des données en mémoire,

c'est-à-dire basés sur l'indexation du plus proche voisin, l'esimation de la densité par noyau, etc. L'hypothèse est que les réponses d'activation des filtres entraînés sur un grand ensemble de données diverses (par exemple, ImageNet (Deng *et al.* (2009)) 1000 catégories d'objets x 1000 exemples par catégories) sont en fait accordées sur des caractéristiques génériques, qui peuvent être utilisées pour classer de nouvelles catégories d'images jamais vues auparavant.

Un certain nombre de questions de recherche restent encore sans réponses. Quels sont les CNNs pré-entraînés les mieux adaptés à la classification ? Quelles sont les meilleures couches ? Est-il possible de combiner différents CNN de manière complémentaire et efficace pour améliorer la classification ? Cette thèse examine ces questions en utilisant des CNN courants populaires pré-entraînés sur l'ensemble de données ImageNet dans deux contextes distincts : incluant la reconnaissance d'instance spécifique (reconnaissance de visage, base de données FERET), et la catégorisation générique d'images (base de données Caltech101).

Les contributions scientifiques de cette thèse sont les suivantes :

- **Contribution 1 :** Nous proposons un modèle bayésien général basé sur la mémoire pour le transfert d'apprentissage, où une nouvelle image peut être classée à partir de couches d'activation génériques de multiples CNN pré-entraînés ;
- **Contribution 2 :** Nous proposons une combinaison des caractéristiques de plusieurs CNNs pré-entraînés ;
- **Contribution 3 :** Une nouvelle méthode de codage de caractéristiques basée sur la théorie de l'information est proposée dans le cadre d'une contribution de recherche avec mon collègue Laurent Chauvin ;

Le reste de cette thèse est organisé en une revue de la littérature sur les réseaux neuronaux convolutifs, notre méthodologie, les résultats expérimentaux et les conclusions finales.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

Ce premier chapitre présente une revue de la littérature sur les architectures des CNNs, la combinaison des caractéristique globales de CNNs avec les descripteurs locales et l'extraction des caractéristiques à partir d'un modèle pré-entrainé.

1.1 Apprentissage profond

Les systèmes d'apprentissage automatique, avec des architectures peu profondes ou profondes, ont la capacité d'apprendre et de s'améliorer avec l'expérience. Le processus d'apprentissage automatique commence par les données brutes qui sont utilisées pour extraire des informations utiles qui aident à la prise de décision. L'objectif principal est de permettre à une machine d'apprendre des informations utiles, tout comme les humains.

Les architectures non profondes sont bien comprises et fonctionnent bien sur de nombreux problèmes d'apprentissage automatique courants, et elles sont toujours utilisées dans la grande majorité des applications d'apprentissage machine d'aujourd'hui. Cependant, il y a eu récemment un intérêt accru sur les architectures profondes, dans l'espoir de trouver des moyens de résoudre des problèmes plus complexes du monde réel (par exemple, l'analyse d'images ou la compréhension du langage naturel) pour lesquels les architectures peu/non profondes ne sont pas en mesure de construire des modèles pertinents.

Les architectures profondes fait référence aux architectures qui contiennent plusieurs couches cachées pour apprendre différentes fonctionnalités avec plusieurs niveaux d'abstraction. Les algorithmes d'apprentissage profond cherchent à exploiter la structure inconnue dans la distribution d'entrée afin de découvrir de bonnes représentations, souvent à plusieurs niveaux, avec des caractéristiques apprises de niveau supérieur définies en termes de caractéristiques de niveau inférieur.

Les techniques classiques d'apprentissage machine sont limitées dans la façon dont elles traitent les données naturelles sous leur forme brute. Pendant des décennies, la construction d'un système de reconnaissance de formes ou d'apprentissage automatique a nécessité une expertise considérable dans le domaine et une ingénierie manuelle minutieuse pour trouver un extracteur de caractéristiques qui transforme les données en entrée (telles que les valeurs de pixels d'une image) en une représentation interne appropriée ou un vecteur de caractéristiques à partir duquel le système d'apprentissage, tel qu'un classificateur, pourrait détecter ou classer des motifs en entrée.

Les algorithmes d'apprentissage profond peuvent apprendre les bonnes caractéristiques et ils sont de loin mieux que d'extraire ces caractéristiques manuellement, c'est à dire, au lieu de construire un ensemble d'algorithmes pour encoder les données en des vecteurs de caractéristiques, les architectures profondes impliquent l'apprentissage automatique de ces caractéristiques au cours du processus d'apprentissage.

Le mot «profond» fait référence à l'apprentissage de couches successives de représentations de plus en plus significatives des données d'entrée. Le nombre de couches utilisées pour modéliser les données détermine la profondeur du modèle. L'apprentissage en profondeur actuel implique souvent l'apprentissage automatique de dizaines, voire de centaines de couches successives de représentation à partir des données d'entraînement.

Bien que l'apprentissage profond existe depuis l'année 1986 par Dechter (1986), il était relativement impopulaire pendant plusieurs années car l'infrastructure informatique (à la fois matérielle et logicielle) n'était pas adéquate et les bases de données disponibles étaient assez petites. Ce n'est que récemment que les réseaux profonds ont fait une grande réapparition en obtenant des résultats spectaculaires dans les tâches de reconnaissance vocale et de vision par ordinateur grâce aux réseaux de neurones convolutifs LeCun *et al.* (1990, 1998).

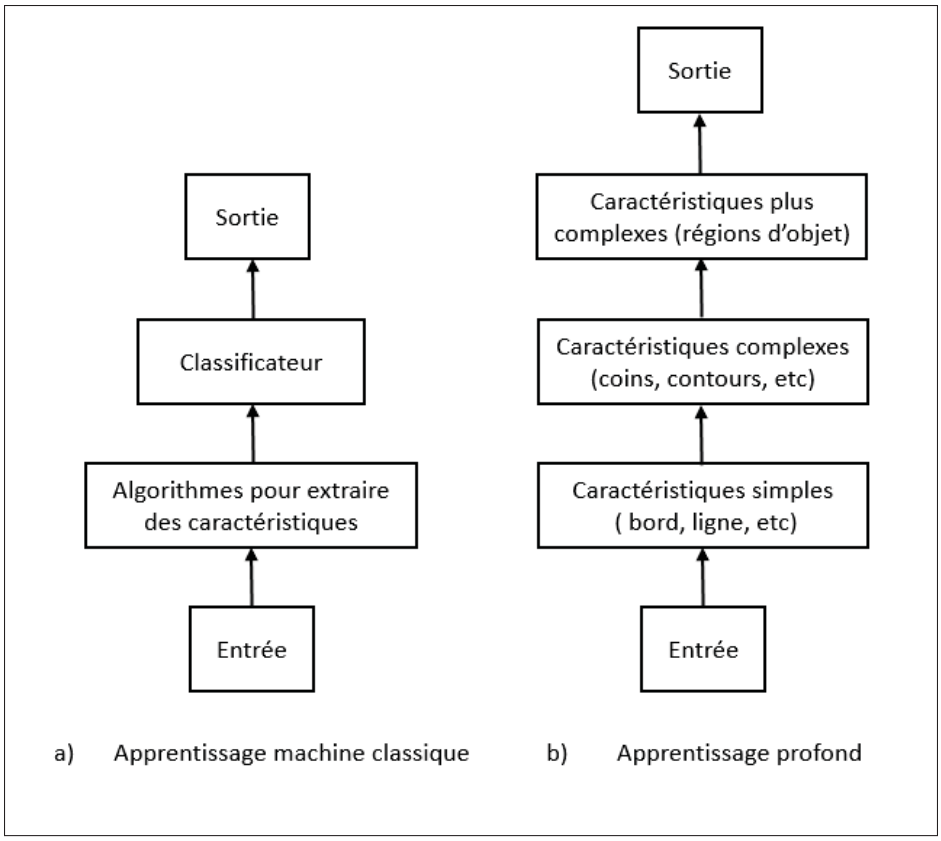


Figure 1.1 La différence entre une architecture d'apprentissage machine classique et une architecture d'apprentissage profond

1.2 Réseaux de neurones convolutifs (CNN) : Concepts de base

Les CNNs LeCun *et al.* (1989) s'inspirent biologiquement du cortex visuel. Ce dernier comprend de petites régions de cellules sensibles à des régions spécifiques du champ visuel. Cette idée a été développée par une expérience guidée par Hubel & Wiesel (1962), qui ont montré que certaines cellules neuronales individuelles du cerveau ne réagissaient (ou n'étaient déclenchées) qu'en présence des contours d'une certaine orientation. Par exemple, certains neurones sont activés lorsqu'ils sont exposés à des bords verticaux et d'autres lorsqu'ils sont représentés par des bords horizontaux ou diagonaux. Hubel et Wiesel ont découvert que tous ces neurones étaient organisés dans une architecture en colonnes et ils étaient capables de produire une perception visuelle. Cette idée de composants spécialisés à l'intérieur d'un système ayant des tâches spécifiques (les cellules neuronales du cortex visuel recherchant des caractéristiques

spécifiques) est également utilisée par les machines et constitue la base des CNNs. Un aperçu plus détaillé de ce que font les CNNs serait de prendre l'image, de la passer à travers une série de couches convolutives, non linéaires (ReLU), de sous-échantillonnage (Pooling) et entièrement connectées. Le résultat peut être une classe unique ou une probabilité de classes décrivant la catégorie de l'image.

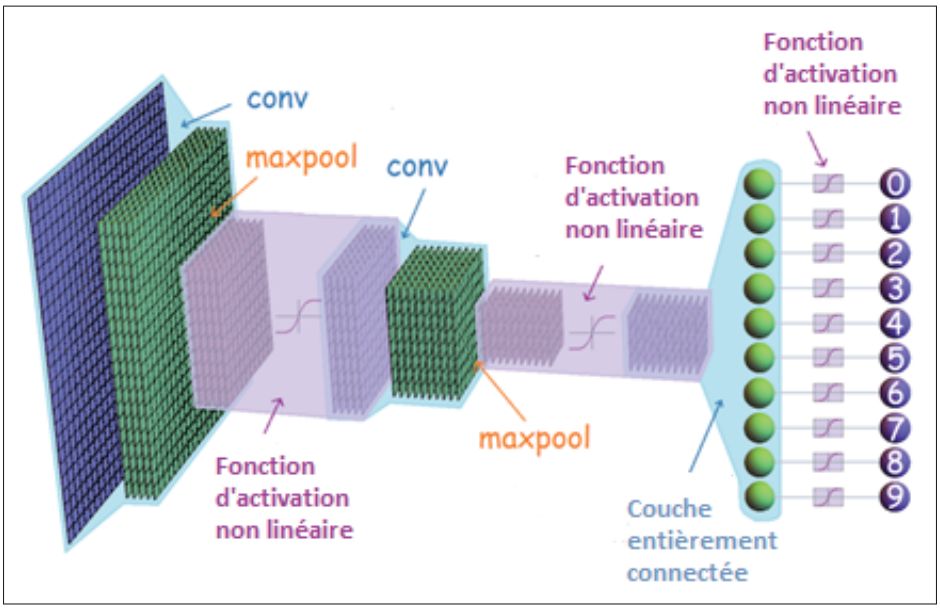


Figure 1.2 Structure générale d'un réseau de neurones convolutifs modifiée de https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/convolutional_neural_networks.html

Un réseau de neurones convolutifs classique commence toujours par une couche de convolution. Cette couche détecte les caractéristiques de bas niveau telles que les arêtes et les courbes etc. Comme on pourrait l'imaginer, pour prédire si une image est un type d'objet, il est nécessaire que le réseau soit capable de reconnaître les caractéristiques de niveau supérieur telles que les mains, les pattes ou les oreilles. Lorsque nous traversons une autre couche de convolution, la sortie de la première couche de convolution devient l'entrée de la deuxième couche. Ainsi, chaque couche de l'entrée décrit en gros les emplacements de l'image d'origine à l'endroit où certaines entités de bas niveau apparaissent. Désormais, lorsque nous appliquons un ensemble de filtres par-dessus (la deuxième couche de convolution), les activations générées

représentent des entités de niveau supérieur. Les types de ces caractéristiques peuvent être des demi-cercles (combinaison d'une courbe et d'un bord) ou des carrés (combinaison de plusieurs bords droits) etc. Au fur et à mesure que nous passons sur le réseau et que nous passons à travers plus de couches de convolution, nous obtenons des cartes d'activation qui représentent des fonctionnalités de plus en plus complexes (Zeiler & Fergus, 2014).

1.2.1 CNN vs ANN

Dans un réseau neuronal artificiel traditionnel Rosenblatt (1962), Rumelhart *et al.* (1986), les neurones sont entièrement connectés entre différentes couches. Les neurones situés entre la couche d'entrée et la couche de sortie construisent les couches cachées. Chaque couche cachée est composée d'un certain nombre de neurones, où chaque neurone est entièrement connecté à tous les neurones de la couche précédente. Le problème avec le réseau neuronal entièrement connecté est que son architecture de réseau densement connectée ne s'adapte pas bien aux grandes images. Pour les images de grande taille, l'approche la plus privilégiée consiste à utiliser un réseau neuronal convolutif (CNN). Les CNNs sont concus pour traiter des données qui ont une topologie de type grille connue. Ils ont trois caractéristiques clés : champ récepteur local, partage de poids et sous-échantillonnage (Pooling).

- **Champ récepteur local :** Dans un réseau neuronal traditionnel, chaque neurone cachée est connecté à chaque neurone de la couche précédente. Les réseaux de neurones convolutifs, cependant, ont une architecture de champ récepteur local, c'est-à-dire que chaque neurone cachée ne peut se connecter qu'à une petite région de l'entrée appelée champ récepteur local. Pour cela, le filtre (poids) est plus petit que l'entrée. Avec le champ récepteur local, les neurones peuvent extraire des caractéristiques visuelles élémentaires comme les bords, les coins etc ;
- **Partage de poids :** Le partage de poids fait référence à l'utilisation du même filtre pour tous les champs récepteurs d'une couche. Dans un CNN, puisque les filtres sont plus petits que l'entrée, chaque filtre est appliqué à chaque position de l'entrée ;

- **Sous-échantillonnage (Pooling) :** Le sous-échantillonnage réduit la taille spatiale des cartes de caractéristiques, réduisant ainsi les paramètres du réseau. Il existe peu de techniques de sous-échantillonnage disponibles, et les technique de sous-échantillonnage les plus courantes sont Max-Pooling et Average-Pooling ;

1.2.2 Opération de convolution

La convolution est l'une des opérations les plus importantes dans le domaine de traitement de signal et elle est utilisée dans plusieurs applications en traitement naturel des langages, vision par ordinateur et traitement d'images. L'opération de convolution peut également être appliquée à des fonctions multidimensionnelles. Comme indique la figure 1.3 la convolution peut être appliquée aux images pour effectuer diverses transformations ; ici, les images sont traitées comme des fonctions bidimensionnelles.

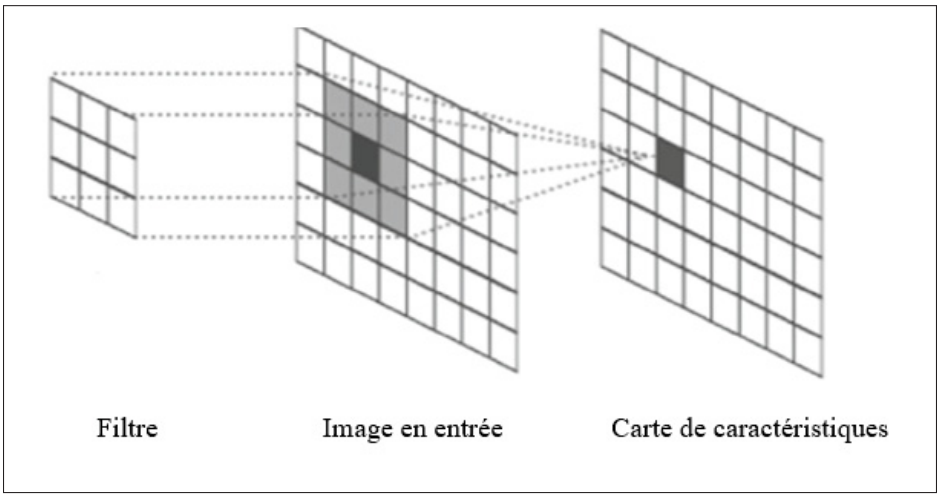


Figure 1.3 Opération de convolution

On note I l'image en entrée, K le filtre 2D dont les dimensions m x n et F la carte de caractéristiques qui représente le résultat de la convolution de l'image avec le filtre. On peut exprimer

mathématiquement cette opération comme suit :

$$\boldsymbol{F(i,j)} = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{1.1}$$

Dans la convolution le filtre K doit être inversé par rapport à l'image en entrée. Dans le cas où K n'est pas inversé, l'opération de convolution sera la même formule que la corrélation croisée, et cela peut être formulé mathématiquement par :

$$\boldsymbol{F(i,j)} = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{1.2}$$

Plusieurs bibliothèques d'apprentissage profond utilisent la corrélation croisé au lieu de la convolution car son implémentation est plus pratique.

Les CNNs ont beaucoup de hyper-paramètres qui définissent le modèle, spécialement dans la couche de convolution on doit définir :

- **La taille de filtre :** La taille de filtre peut être n'importe quelle taille n x n avec n>2 et n doit être impaire (3x3, 5x5, 7x7, etc) ;
- **Nombre de filtre par couche :** Il peut être n'importe quel numéro raisonnable (32, 64, 128, 256, etc) ;
- **Le pas :** C'est le nombre de pixels à déplacer en appliquant le filtre sur l'image en entrée.
- **Zero Padding :** C'est le nombre de pixels (de valeur zéro) à ajouter à l'image en entrée pour contrôler la taille de sortie de convolution ;

1.2.3 Les fonctions d'activations

La sortie de chaque couche convolutionnelle est envoyée à une couche de fonction d'activation. La couche de fonction d'activation se compose d'une fonction d'activation qui prend la carte de caractéristiques produite par la couche de convolution et génère la carte d'activation comme sortie. La fonction d'activation est utilisée pour transformer le niveau d'activation d'un neurone en un signal de sortie. Il spécifie la sortie d'un neurone vers une entrée donnée. Une fonction

d'activation a généralement un effet d'écrasement qui prend une entrée (un nombre), effectue une opération mathématique sur elle et génère le niveau d'activation d'un neurone dans une plage donnée, par exemple, 0 à 1 ou -1 à 1. En général, 3 propriétés sont attendues de la fonction d'activation :

- **Non linéarité :** c'est la propriété cruciale de la fonction d'activation. Grâce à cette propriété le réseau de neurones peut être utilisé pour résoudre des problèmes non linéaires ;
- **Différentiable** - ce qui signifie que nous avons une dérivée continue du premier ordre. C'est une propriété souhaitable pour activer des méthodes d'optimisation basées sur un gradient comme la rétropropagation Rumelhart *et al.* (1986) ;
- **Monotone** Cette caractéristique aide le réseau de neurones à converger plus facilement vers un modèle plus précis. De nombreuses fonctions d'activation sont utilisées comme l'indique la figure 1.4.

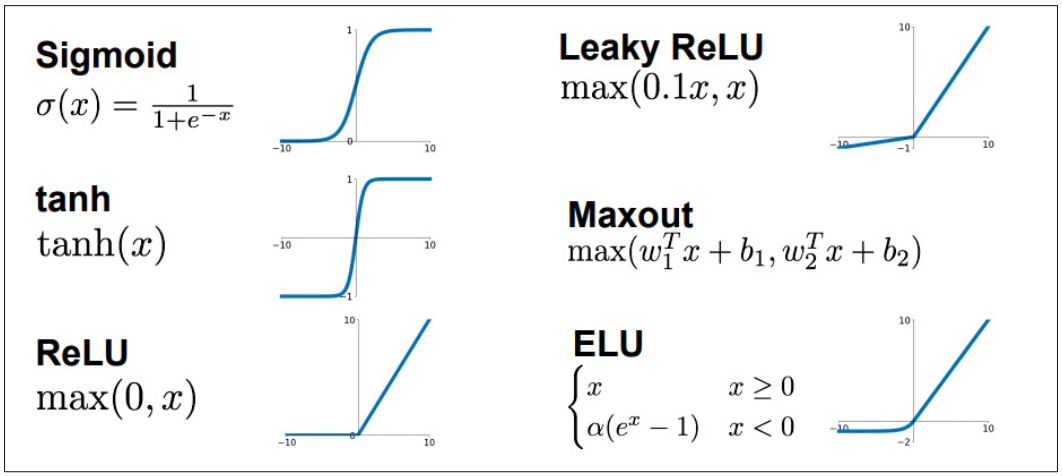


Figure 1.4 Quelques fonctions d'activation.
Tirée de http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture4.pdf

En pratique, on utilise le ReLU comme activation pour les couches cachées car cette dernière accélère le processus d'entraînement. Le calcul du gradient est très simple (0 ou 1 selon le signe de x). De plus, le calcul d'un ReLU est facile : tous les éléments négatifs sont mis à 0, donc pas d'exponentiels, pas d'opérations de multiplication ou de division. Pour la couche de sortie, on utilise Sigmoid pour un problème de classification binaire. Par contre, pour la classification

multiclasse, on définit la fonction Softmax qui est représentée mathématiquement par :

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{1.3}$$

1.2.4 La couche de Pooling

Une couche de Pooling (regroupement) prend chaque sortie des cartes de caractéristiques résultantes de la couche de convolution et la sous-échantillonne. Les techniques de regroupement les plus courantes sont MaxPooling (la figure 1.5) et AveragePooling.

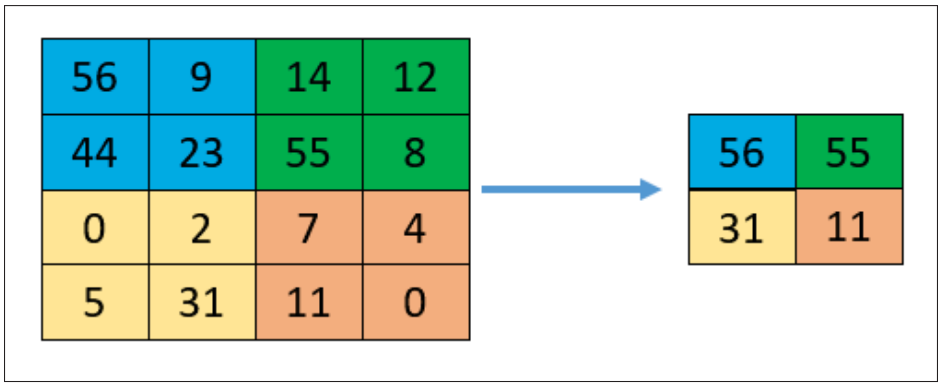


Figure 1.5 MaxPooling

Le raisonnement intuitif derrière l'opération de sous-échantionnage est que la détection des caractéristiques est plus importante que l'emplacement exact des caractéristiques. Cette stratégie fonctionne bien pour des problèmes simples et basiques mais elle a ses propres limites.

1.2.5 Couches entièrement connectées

Les réseaux de neurones convolutifs sont composés de deux étapes : l'étape d'extraction des caractéristiques et l'étape de classification (figure 1.6). Les couches entièrement connectées représentent la partie de classfication où chaque neurone est connecté à chaque neurone de la

couche suivante et chaque valeur contribue à prédire à quel point une valeur correspond à une classe particulière.

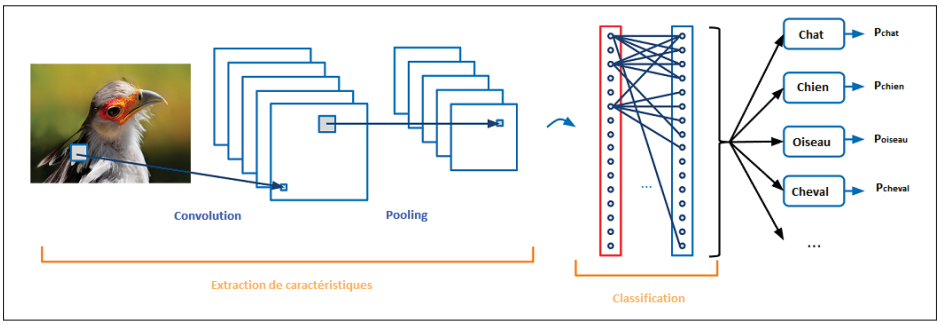


Figure 1.6 Architecture CNN

1.2.6 Les architectures populaires des CNNs pour la classification d'images

Le premier CNN a été introduit par le chercheur Lecun [1] dans son article scientifique LeCun *et al.* (1989) puis il a proposé l'architecture LeNet LeCun *et al.* (1998) et finalement, après près de 15 ans, a conduit à des modèles révolutionnaires remportant le défi ILSVRC (ImageNet Large Scale Visual Recognition Challenge) Krizhevsky *et al.* (2012) (figure 1.7).

ILSVRC Russakovsky *et al.* (2015) comme son nom l'indique est une compétition de vision par ordinateur organisée entre 2010 et 2017 qui a présenté trois différents défis :

- **Classification d'images :** Prédiction de la classe de l'image (une image représente un seul objet) ;
- **Localisation d'un seul objet :** Localisation d'un objet par un cadre dans une image après la tache de classification ;
- **Détection d'objets :** Localisation de multi-objets dans une seule image ;

ImageNet Deng *et al.* (2009) est une base de données de plus de 15 millions d'images de haute résolution étiquetées appartenant à environ 22 000 catégories. Pour la compétition ILSVRC, un sous-ensemble d'ImageNet est utilisé, avec environ 1000 images dans chacune des 1000

1. http ://yann.lecun.com/

catégories. Au total, il y a environ 1,2 million d'images d'entraînement, 50 000 images de validation et 150 000 images de test.

Depuis septembre 2012, toutes les architectures gagnantes de ILSVRC sont des architectures CNNs (figure 1.7).

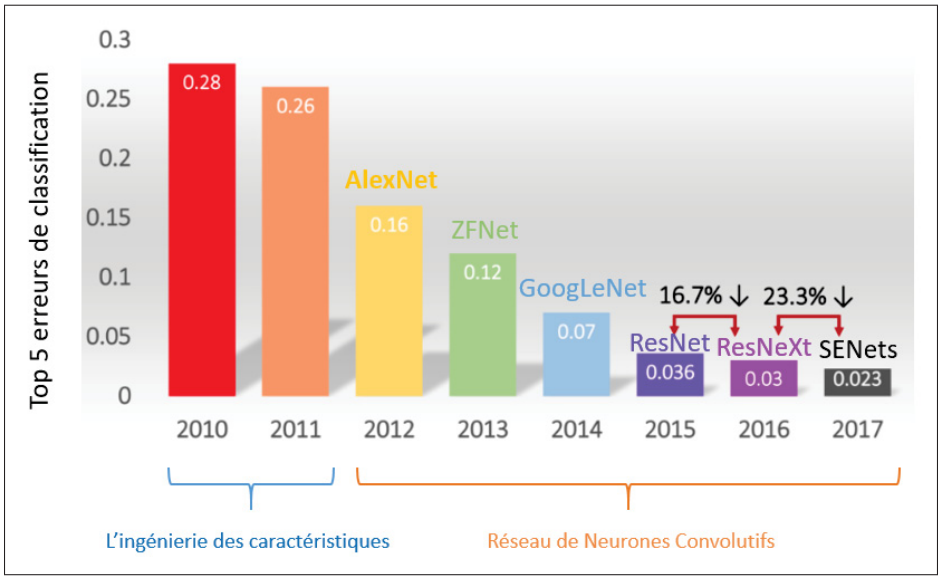


Figure 1.7 L'erreur top-5 en % des architectures gagnantes de la compétition ILSVRC

Le rythme d'amélioration d'une année à une autre a été spectaculaire et on peut dire choquant pour le domaine de la vision par ordinateur. Les architectures suivantes sont les plus populaires :

- **AlexNet** Krizhevsky *et al.* (2012)

  AlexNet a réalisé une erreur Top-5 de 15,3% dans le défi ImageNet 2012, soit plus de 10,8% de moins que celle du finaliste de l'année précédente. Cela a été rendu possible grâce à l'utilisation d'unités de traitement graphique (GPU) pendant l'entraînement qui est un outil essentiel de la révolution de l'apprentissage profond. Les chercheurs ont commencé à donner attention à ce modèle, non seulement au sein de la communauté de l'IA, mais à travers l'industrie technologique aussi ;

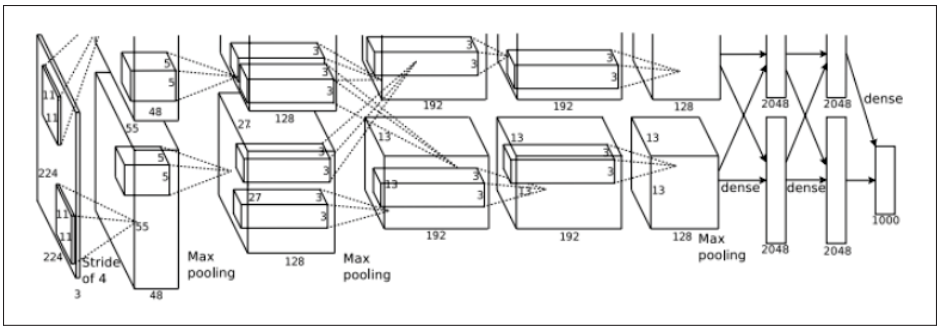


Figure 1.8 Architecture CNN proposée par Krizhevsky *et al.* (2012)

- **ZFNet** Zeiler & Fergus (2014)

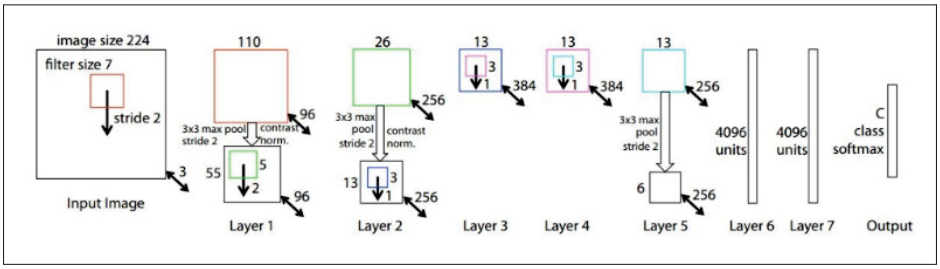


Figure 1.9 Architecture CNN proposée par Zeiler & Fergus (2014)

ZFNet est une version modifiée d'AlexNet qui donne une meilleure précision. Une différence majeure dans les deux modèles était que ZFNet utilisait des filtres de taille 7x7 tandis qu'AlexNet utilisait des filtres 11x11. L'intuition derrière cela est qu'en utilisant des filtres plus gros, nous perdions beaucoup d'informations sur les pixels, que nous pouvons conserver en utilisant des filtres de taille plus petite ;

- **VGG16** Simonyan & Zisserman (2014)

  Karen Simonyan et Andrew Zisserman de Oxford Vision Geometry Group (VGG) ont proposé le VGG-16 qui a 13 couches de convolution et 3 couches entièrement connectée. VGGNet utilise des filtres de taille 3x3 comparant de 11x11 de AlexNet et 7x7 de ZFNet. Les auteurs donnent l'intuition derrière cela qu'avoir deux filtres de taille 3x3 consécutifs donne un champ récepteur efficace de 5x5, et trois séries de filtres de taille 3x3 donnent un champ

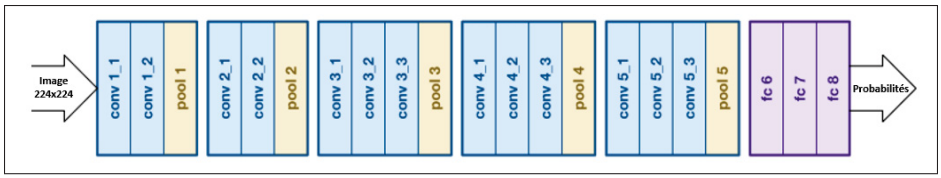


Figure 1.10 Architecture VGG-16

récepteur de filtres 7x7, mais en utilisant cela, nous pouvons utiliser un nombre beaucoup moins élevé d'hyper-paramètres. Ils ont aussi proposé une variante plus profonde VGG-19 ;

- **GoogleNet/Inception** Szegedy *et al.* (2015)

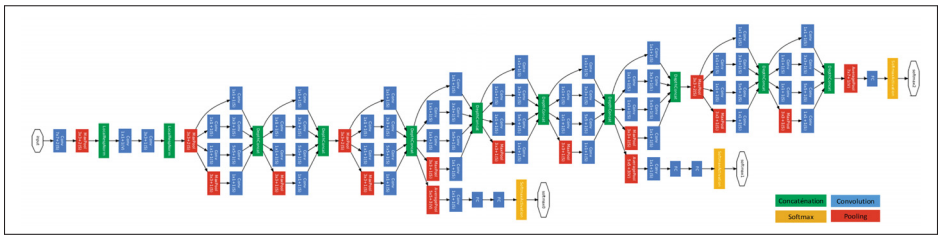


Figure 1.11 GoogleNet proposée par Szegedy *et al.* (2015)

GoogleNet est la première architecture qui utilise des multi-connections entre les couches dans un module appelé «Inception» et élimine les couches entièrement connectées et les remplacer une couche de Softmax. GoogleNet contient 9 modules «Inception» et propose plusieurs sortie de softmax ;

- **ResNet** He *et al.* (2016)

  ResNet est proposé par une équipe de recherche de Microsoft. Ils ont montré que l'ajout des connections entre les couches (les liens résiduels) peut représenter des chemins simples pour la propagation de gradient, ce qui rend la mise à jour des poids plus efficaces. Il a fallu deux à trois semaines pour l'entraîner sur une machine à 8 GPU. Ce modèle arrive à une erreur Top-5 de 3.57%, qui dépasse l'erreur humaine qui est estimée à 5% ;

- **ResNeXt** Xie *et al.* (2017)

  ResNext comme son nom l'indique, c'est le «Next» ResNet. Dans cette architecture les chercheurs ont proposé une combinaison entre le bloc «inception» de GoogleNet et les

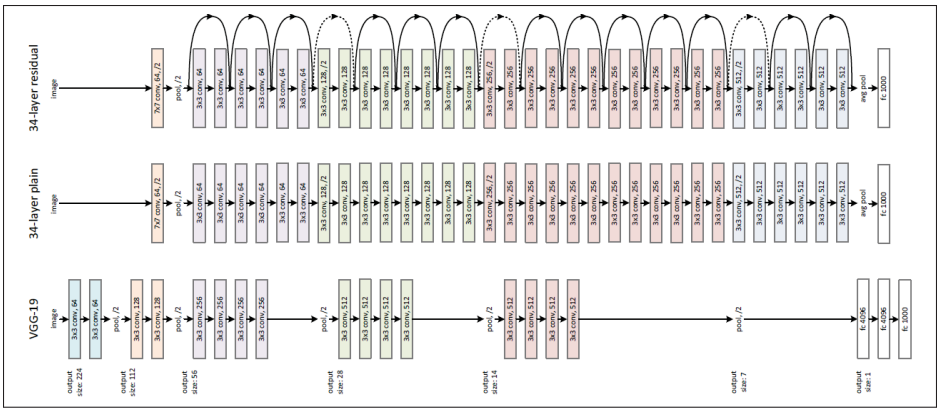


Figure 1.12 Comparaison entre VGG-19, 34 couches d'un CNN classique et ResNet de 34 couches. Tiré de He *et al.* (2016)

connections résiduelles de ResNet. Ils ont montré que l'ajout des couches parallèles entre les connections résiduelles est plus efficace que l'ajout des couches en profondeur ;

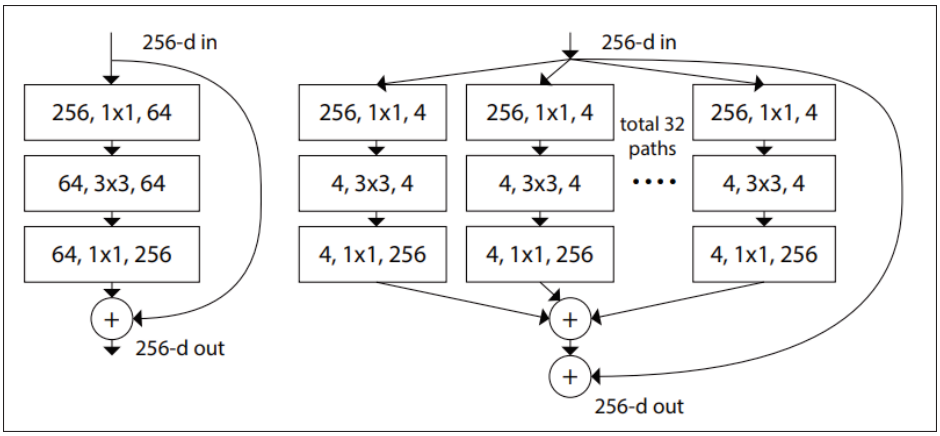


Figure 1.13 Différence entre un bloc résiduel dans ResNet et ResNext. Tiré de Xie *et al.* (2017)

- **SENet** Hu *et al.* (2018)

  SENet propose un bloc qui calibre de manière adaptative les réponses des caractéristiques en modélisant explicitement les interdépendances entre les canaux (figure 1.14). Il a remporté la première place du défi de classement ILSVRC 2017 avec une erreur top-5 à 2,251%,

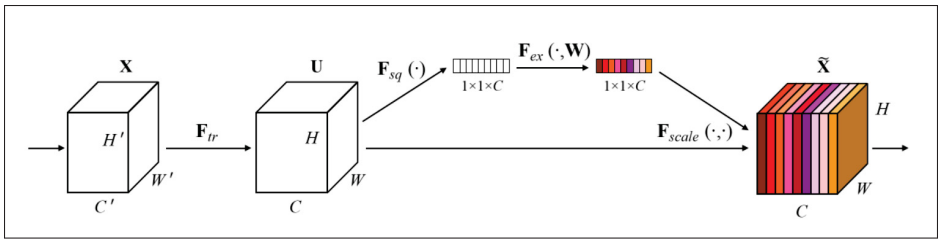


Figure 1.14 SE bloc. Tiré de Hu *et al.* (2018)

ce qui représente une amélioration relative d'environ 25% par rapport à l'architecture gagnante de 2016 ;

Nous avons cité les architectures gagnantes de ILSVRC, mais il y a beaucoup d'autres architectures intéressantes comme DensNet Huang *et al.* (2017), MobileNet Howard *et al.* (2017), SPPNet He *et al.* (2015), etc. La figure 1.15 résume l'évolution des architectures CNNs.

1.2.7 Le transfert d'apprentissage

Le transfert d'apprentissage est devenu plus populaire avec l'apprentissage profond, en particulier sur les réseaux de neurones convolutifs. Il réduit efficacement le temps d'entraînement et améliore également la précision des modèles conçus pour des tâches avec des données d'entraînement minimales ou inadéquates. Le transfert d'apprentissage peut être utilisé des manières suivantes :

- **Modèle pré-entraîné en tant qu'extracteur de vecteurs de caractéristiques de tailles fixes :**

  Dans ce scénario, la dernière couche entièrement connectée (couche de classification) est remplacée par un nouveau classificateur linéaire et cette dernière couche est ensuite entraînée sur un nouvel ensemble de données. De cette façon, les couches d'extraction des caractéristiques (Convolution, ReLU, Pooling,etc) restent fixes et seul le classificateur est affiné. Cette stratégie est mieux adaptée lorsque la nouvelle base de données est insuffisante mais similaire à la base de données d'origine.

- **Ajuster le modèle entier :**

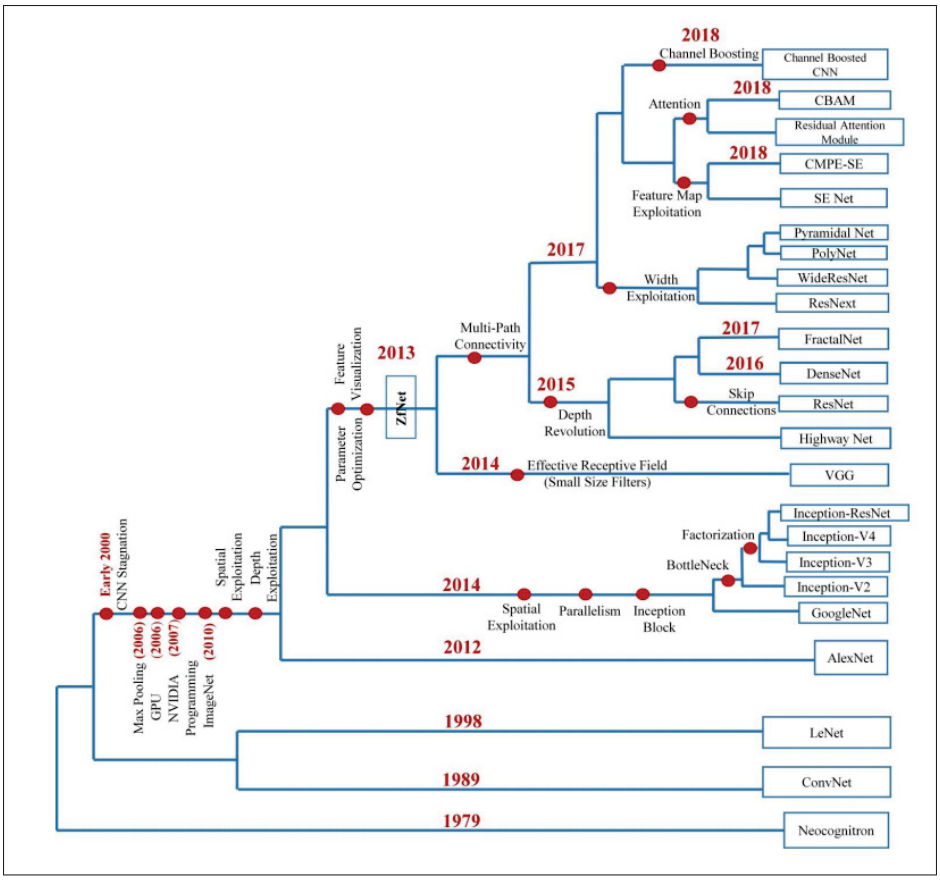


Figure 1.15 Évolution des architectures CNNs. Tirée de Khan *et al.* (2019)

On prend un modèle pré-entraîné et on remplace sa dernière couche entièrement connectée (couche classifiante) par une nouvelle couche entièrement connectée puis on ré-entraîne l'ensemble du réseau sur un nouvel ensemble de données en continuant la rétro-propagation jusqu'au début de réseau. De cette façon, tous les poids sont affinés pour une nouvelle tâche.

1.2.8 Aperçu sur les représentations des images par des vecteurs de caractéristiques locales

Depuis les années 1990, il existe deux types de méthodes pour caractériser l'image, une est basée sur des caractéristiques locales, telles que SIFT Lowe (2004), et l'autre est basée sur des caractéristiques globales, telles que les CNNs.

Les caractéristiques locales telles que SIFT sont robustes à la rotation, aux changements d'échelle, aux changements de vue, à la transformation affine et au bruit. SIFT propose un descripteur de 128 valeurs qui représentent l'histogramme de gradient pour chaque point clé détecté. PCA-SIFT (Ke & Sukthankar (2004)) réduit la dimension de descripteur de 128 à 36 pour accélérer le processus de mise en correspondance. Arandjelović & Zisserman (2012) et Toews & Wells (2009) ont proposé des améliorations sur la représentation des déscripteurs. Une autre alternative de SIFT est SURF (Bay *et al.* (2006)) qui utilise l'image intégrale avec les filtres Haar pour accélérer le processus de détection des points clés. Une bonne étude de comparaison entre SIFT, PCA-SIFT et SURF est présenté dans Juan & Gwun (2009). ORB (rublee2011orb) propose une autre alternative pour accélérer la mise de correspondance entre les points clés par la binarization des descripteurs et l'utilisation de distance Hamming au lieu de la distance euclidienne.

Pour des tâches de classification, la mise en correspondance par des pairs d'images en utilisant des descripteurs des points clés n'est pas efficace. L'utilisation des représentations comme BoW (Sivic & Zisserman (2003)), VLAD (Jégou *et al.* (2010)) et FV (Perronnin *et al.* (2010)) avec des classificateurs comme Kd-Tree (Bentley (1975)) et Random Forest(Ho (1995)) améliorent efficacement la précision.

Pour les CNNs, l'application des filtres par l'opération de convolution sur toute l'image en entrée permet l'extraction des caractéristiques globales. Les réponses des filtres créent des cartes de caractéristiques (ou d'activations) où chaque pixel représente un descripteur de taille 1 x 1 x k (avec k le nombre de filtre pour une couche donnée) qui décrit un motif bien déterminé dans l'image. Azizpour *et al.* (2015) ont montré que les caractéristiques locales sont plus efficaces pour des tâches de reconnaissance d'objets et les caractéristiques profondes (globales) sont meilleures pour des problèmes de classifications avec des grandes bases de données. Parmi les vecteurs de caractéristiques extraites des couches profondes nous avons : Babenko & Lempitsky (2015) mettent la somme des cartes caractéristiques dans un vecteur, Tolias *et al.* (2015) construisent le vecteur de caractéristiques en prélevant le maximum de chaque carte d'activation, cette méthode est nommée par Global-MaxPooling (GMax) ou MAC, au niveau de région,

R-MAC performe le Max Pooling pour chaque région, Radenović *et al.* (2018) généralise le max et average pooling avec un paramètre de pooling.

Enfin, plusieurs recherches ont exploité la fusion des descripteurs locaux avec les descripteurs globaux/profonds, Zhang *et al.* (2013) ont montré que la combinaison du vecteur de caractéristiques FC8 d'AlexNet avec les descripteurs SIFT améliore le taux des bonnes correspondances, Zheng *et al.* (2015) ont proposé de fusionner plusieurs type de caractéristiques (locales et globales) d'une manière adaptative et finalement Chandrasekhar *et al.* (2016) ont combiné FV et CNN avec un paramètre α.

CHAPITRE 2

RECHERCHE DE L'INFORMATION DANS LES COUCHES DE CNN

Dans ce chapitre, nous allons proposer une nouvelle méthode d'extraction de caractéristiques et de mise en correspondance à partir des couches cachées des réseaux de neurones convolutifs pré-entraînés sur la base de données ImageNet. D'abord, nous allons donner notre motivation sur les idées liés à nos méthodes. Puis, nous allons détailler les méthodologies proposées.

2.1 Motivation

Bien que les architectures basées sur le réseau neuronal convolutif (CNN) aient connu un grand succès, il existe encore un certain nombre de domaines qui doivent être étudiés profondément. Tout d'abord, comme les architectures basées sur CNN sont composées d'une série de couches ou de modules, le défi consiste à déterminer les hyperparamètres optimaux requis pour une application donnée. Un autre défi consiste au temps d'entraînement qui peut atteindre plusieurs jours même avec l'utilisation des GPUs. L'utilisation des réseaux de neurones pré-entraînés permet d'exploiter le fruit des jours d'entraînement sur la plus grande base de données existante. Tous les modèles sont disponibles en ligne et sont simples à utiliser.

2.2 Méthodologies

2.2.1 Mise en correspondance des réponses des filtres

La classification cherche à prédire un label de classe inconnu C_i associé à une nouvelle image I_i. Nous développons notre méthode dans le contexte d'une formulation probabiliste bayésienne générale qui cherche à maximiser la probabilité postérieure $p(C_i|I_i)$ du label de classe inconnu

C_i associé à l'image I_i (ou carte de caractéristiques) :

$$p(C_i|I_i) = \frac{p(I_i|C_i)p(C_i)}{p(I_i)} \tag{2.1}$$

$$\propto p(I_i|C_i)p(C_i) \tag{2.2}$$

où l'équation 2.1 est issue de la règle de Bayes avec :

- $p(I_i|C_i)$ est la probabilité que C_i étant donné I_i
- $p(C_i)$ est la probabilité de la classe C_i, ici généralement considérée comme constante
- $p(I_i)$ est constant pour une image fixe I_i et conduit donc à la proportionnalité dans l'équation 2.2

La formulation bayésienne fournit un mécanisme pour incorporer l'incertitude probabiliste. La classification peut être effectuée en estimant la distribution maximale par :

$$C_i^* = \arg\max \{p(C_i|I_i)\} = \arg\max \{p(I_i|C_i)p(C_i)\} \tag{2.3}$$

L'équation 2.3 est équivalent à l'estimation du maximum de similarité dans le cas où $p(C_i)$ est uniforme. L'apprentissage basé sur des données d'entraînement en mémoire définit les données par un couple (C_j, I_j), avec I_j est l'image dans la base d'entraînement et C_j son label.

La probabilité conditionnelle $P(I_i|C_i)$ peut être définie comme suit :

$$p(I_i|C_i) = \sum_j p(I_i, I_j|C_i) \tag{2.4}$$

$$= \sum_j p(I_i|I_j, C_i)p(I_j|C_i) \tag{2.5}$$

$$= \sum_j p(I_i|I_j)p(I_j|C_i) \tag{2.6}$$

ici, l'équation 2.4 provient de la marginalisation probabiliste de I_j de la distribution conjointe $p(I_i, I_j|C_i)$, l'équation 2.5 est le résultat de la théorie de Bayes, et l'équation 2.6 vient de l'hy-

pothèse d'une indépendance conditionnelle de l'image I_i et de classe C_i étant donnée l'image I_j.

Dans l'équation 2.6, $p(I_j|C_i)$ est la probabilité que l'image I_j soit associé à la classe C_i, définie comme :

$$p(I_j|C_i) = \begin{cases} 1, & C_i = C_j \\ 0, & sinon \end{cases} \tag{2.7}$$

Encore, $p(I_i|I_j)$ est défini comme une densité de noyau sur I_j, évaluée au point I_i par :

$$p(I_i|I_j) \propto e^{-h(g(I_i),g(I_j))} \tag{2.8}$$

avec $g(I_i)$ est une transformation de données. Par exemple, g peut être le calcul d'une carte de caractéristiques CNN d'une couche bien déterminée à partir de I_i, et h est une fonction de noyau. Pour une densité gaussienne de moyenne $g(I_i)$ et une variance σ_i, on peut exprimer $h(g(I_i), g(I_j))$ par :

$$h(g(I_i), g(I_j)) = \frac{||g(I_i) - g(I_j)||^2}{(1 + d_i^2)} \tag{2.9}$$

avec $||g(I_i) - g(I_j)||$ est le carré de la distance Euclidienne entre $g(I_i)$ et $g(I_j)$ et $(1 + d_i^2)$ est la variance avec d_i est défini comme la distance au plus proche voisin :

$$d_i = \min_{I_j} ||g(I_i) - g(I_j)||^2 \tag{2.10}$$

Cette théorie va être appliquée sur des vecteurs de caractéristiques comme l'indique la figure 2.1. $g(I)$ représente la sortie du CNNs qui peut être des cartes ou des vecteurs de caractéristiques. La mise en correspondances peut être appliquée entre deux vecteurs de caractéristiques (Global max pooling ou Global average pooling) ou entre les réponses les filtres une par une comme étant des de scripteurs locales (Indépendent pixel features).

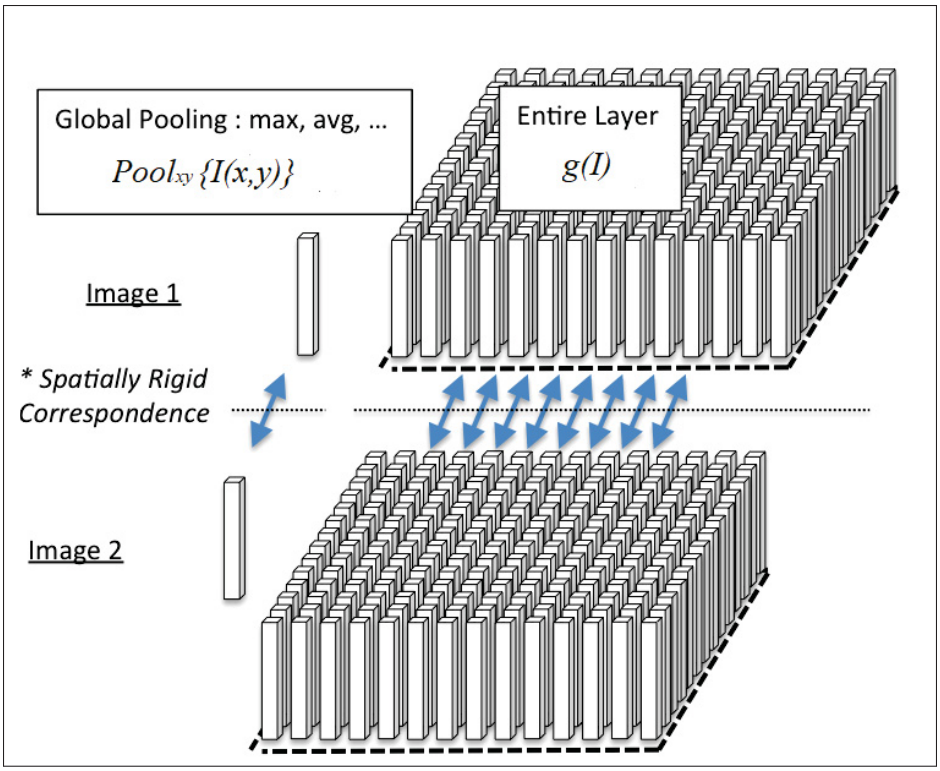


Figure 2.1 Mise en correspondance de deux images avec différentes représentations de données

2.2.2 Normalisation des caractéristiques

La normalisation des vecteurs de caractéristiques est une technique très utilisée dans le domaine de l'apprentissage machine, nous utilisons deux types de normalisation :

- **Standardization (Z score) :** Soit μ la moyenne de $g(I_i)$ et σ son écart-type. La standardization est formulée par :

$$g_{Zscore}(I_i) = \frac{g(I_i) - \mu}{\sigma} \tag{2.11}$$

- **Mise à l'échelle à la longueur unitaire (L2) :** On note $||g(I_i)||$ la longueur euclidienne, la normalisation L2 est formulée par :

$$g_{L2}(I_i) = \frac{g(I_i)}{||g(I_i)||} \tag{2.12}$$

2.2.3 Combinaison des vecteurs de caractéristiques des modèles pré-entraînés différents

La combinaison des caractéristiques peut être effectuée par plusieurs méthodes. La sommation des tenseurs des caractéristiques est la signature de ResNet (He *et al.* (2016)) et la concaténation (figure 2.2) a été d'abord utilisée dans les réseaux Inception (Szegedy *et al.* (2015)) et plus tard dans DenseNet (Huang *et al.* (2017)).

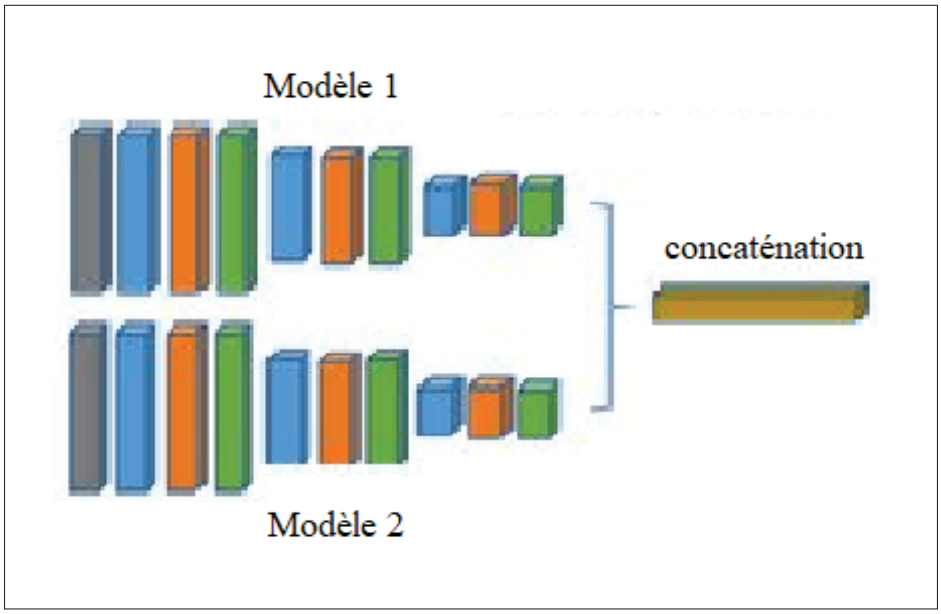


Figure 2.2 Concaténation des vecteurs de caractéristiques

Dans cette approche, la concaténation est la méthode utilisée, nous allons sélectionner les couches qui donnent les caractéristiques les plus distinctives et nous les concaténons pour former un nouveau descripteur. On note F le résultat de la concaténation qui peut être décrit par :

$$F = (Model1_{bestlayer}, Model2_{bestlayer}, ..., ModelN_{bestlayer}) \tag{2.13}$$

2.2.4 Binarisation des vecteurs de caractéristiques

Dans cette approche, nous proposons une méthode de normalisation et de compression des caractéristiques où le concept du gain d'information (ou information mutuelle) a été considéré Quinlan (1986). Le gain d'information (IG) mesure la quantité d'informations qu'une caractéristique nous donne sur la classe. Les caractéristiques qui séparent parfaitement les classes devraient donner un maximum d'informations. En effet, nous adoptons le vecteur de caractéristiques Global max pooling (GMax) qui va être transformé en vecteur binaire basé sur les seuils qui donnent le maximum gain d'information des réponses des filtres sur toute l'ensemble des données d'entraînement. On note H l'entropie Shannon (1948), I_{train} les images de la base de données d'entraînement et $F_t(I_{train})$ les réponses des filtres d'une couche t d'une architecture CNN. Le gain d'information est calculé pour chaque valeur de réponse de filtre. Pour un filtre f_j, le gain d'information pour une valeur de seuil k qui correspond à la classe C_k est donnée par :

$$IG_k = H(F_{tj}(I_{train})[k]) - H(F_{tj}(I_{train})[k]|C_k), \tag{2.14}$$

et la seuil choisie pour binariser la réponse de filtre $F_{tj}(I_{train})$ en se basant sur les labels des données d'entraînement sont formulés par :

$$\tau_j = \arg\max_k \{IG_k\}, \tag{2.15}$$

avec $H(F_{tj}(I_{train})[k])$ et $H(F_{tj}(I_{train})[k]|C_k)$ sont l'entropie binaire et l'entropie conditionnelle binaire, pour k allant de 0 à $N_{I_{train}}$ (nombre d'images d'entraînement). On note que si nous

trouvons juste un exemple par classe, l'entropie conditionnelle $H(F_{tj}(I_{train})[0]|C_0) = 0$ et τ_j maximise l'entropie.

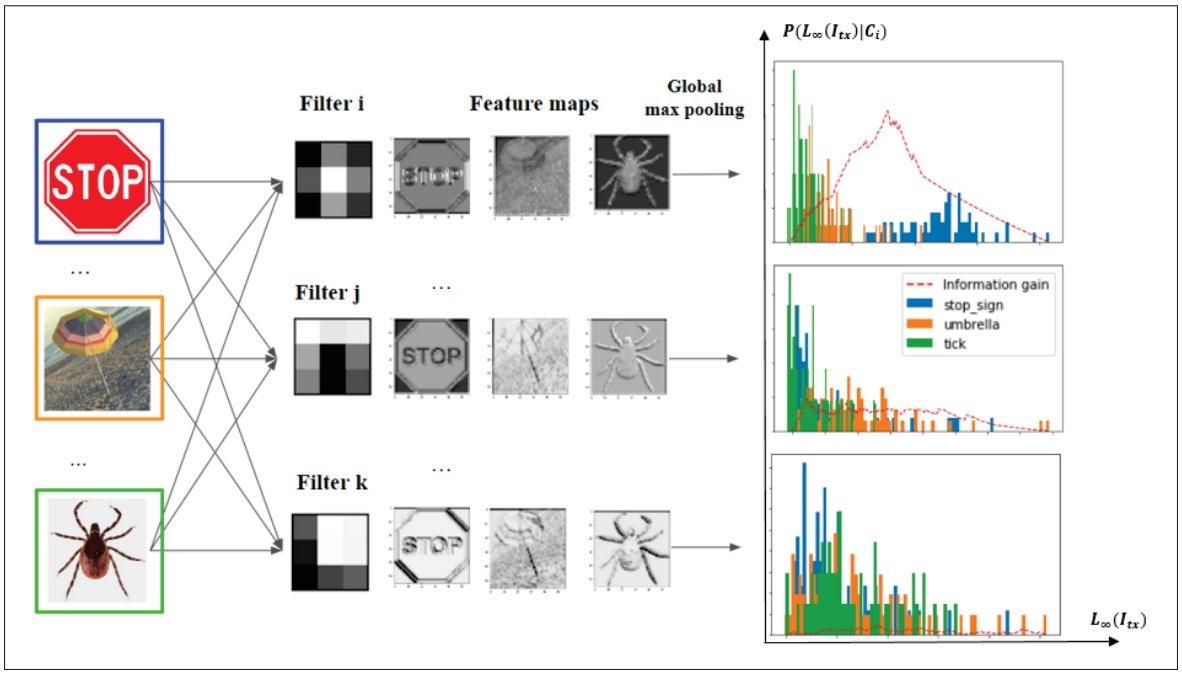


Figure 2.3 L'analyse de gain d'information de trois filtres i, j et k pour trois classes montre que le filtre i est le filtre le plus performant pour séparer ces trois classes.

Dans la figure 2.3 les images sont représentées par des vecteur global max pooling. En fonction de la capacité du classificateur (filtre), les histogrammes peuvent être plus ou moins séparés. Le gain d'information est calculé pour les valeurs de seuil couvrant toute la plage des histogrammes, de la même manière que la dérivation d'une courbe ROC. La performance du filtre pour la classification est alors considérée comme la valeur maximale du gain d'information trouvé, correspondant à la valeur de seuil optimale. Cette opération est répétée pour tous les filtres de chaque couche. Les valeurs maximales des gains d'informations sont ensuite enregistrées résumant les capacités de ses filtres à séparer les données.

CHAPITRE 3

EXPÉRIENCES ET RÉSULTATS

Dans ce chapitre, nous présentons d'abord les résultats de l'étude de recherche de l'information dans les modèles pré-entraînés dont le but est de trouver les couches qui fournissent les caractéristiques les plus distinctives. Puis nous présentons les résultats de combinaison des différents modèles. Enfin, nous montrons l'effet du codage binaire proposé sur les résultats de classification.

3.1 Base de données

Le choix des bases de données a été bien étudié. Toutes les données choisies contiennent des images différentes aux images de la base de données ImageNet [1]. Nous avons choisi Caltech101 (Fei-Fei *et al.* (2004)) car elle représente des catégories d'objets générales, FERET (Phillips *et al.* (1998)) et HCP (Van Essen *et al.* (2013)) contiennent peu de données par catégorie et peuvent être utilisées pour une classification binaire selon le sexe (beaucoup de données pour 2 classes). Ici quelques détails sur ces bases de données :

- **Caltech101**

  C'est une base de données d'objets de 101 catégories (figure 3.1), chacune contient entre 40 et 800 images, mais la majorité contient 50 images ;

- **FERET**

  Facial Recognition Technology est une base de donnée utilisé pour l'évaluation du système de reconnaissance faciale. On va utilisé un sous-ensemble qui contient 994 paires d'images de visages (figure 3.2), avec 2 images par personne. Chaque image est de 256x384 pixels ;

- **HCP**

  C'est une base de données des imageries médicales sur le plan axial de 1010 cerveaux d'adultes en bonne santé où chaque image représente un sujet (figure 3.3). Les sujets sont

1. http ://image-net.org/challenges/LSVRC/2010/

Figure 3.1 Exemples d'images de la base de données Caltech101

Figure 3.2 Exemples d'images de la base de données FERET

âgés de 22 à 36 ans (moyenne : 29 ans), avec 468 hommes et 542 femmes regroupées en 439 familles (dont des jumeaux monozygotes, dizygotes et des frères et sœurs germains) ;

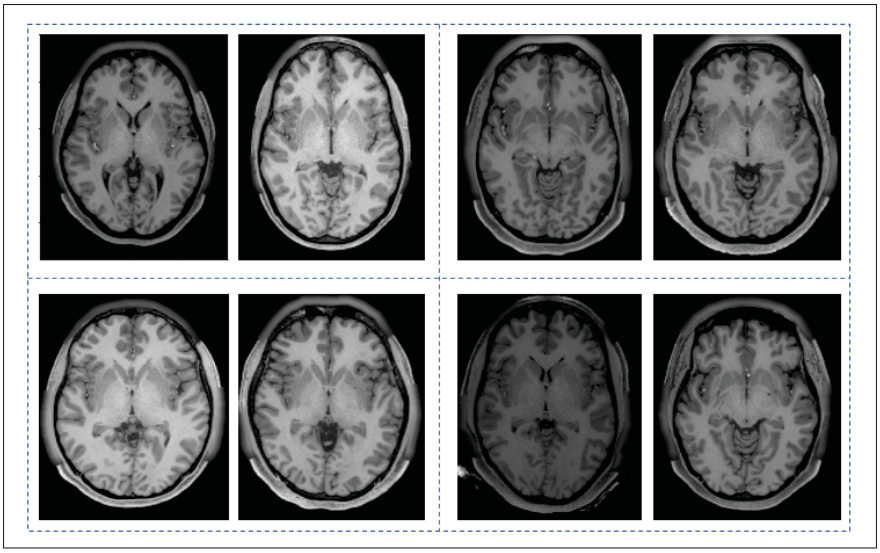


Figure 3.3 Exemples d'images de la base de données HCP.
Chaque paire représente des membres de la même famille

3.2 Protocoles d'expérimentation

3.2.1 Pré-traitement des données

Avant l'extraction des caractéristiques, un pré-traitement est appliqué sur les images. Nous avons redimensionné les tailles des images en 224x224 pour tous les tests effectués. Encore, pour chaque CNN pré-entrainé, nous avons appliqué aussi le même pré-traitement utilisé lors de l'apprentissage. Ce pré-traitement peut varier d'un réseau à un autre.

3.2.2 Détails d'implémentation

Pour nos expériences, nous avons developpé nos algorithmes avec le langage de programmation python. Tous les réseaux pré-entrainés sur ImageNet sont donnés par la bibliothèque Keras [2]. La mise en correspondance est réalisée par l'algorithme de plus proche voisins approximatif «annoy» [3].

2. https ://keras.io/api/applications/
3. https ://pypi.org/project/annoy/1.0.3/

3.3 Résultats

3.3.1 Résultats de mise en correspondances des réponses des filtres

Nous savons que les réseaux de neurones convolutifs pré-entrainés contiennent plus d'informations dans des couches différentes à celle du Softmax (la dernière couche) ou de la couche avant dernière. Nos expériences cherchent à explorer les couches qui donnent plus de caractéristiques génériques dans le cas de la classification générale des objets (Caltech101) et dans le cas de reconnaissance des images bien particulières (reconnaissance de visage avec FERET).

Nous faisons passer les images à travers des CNN pré-entrainés, sans augmentation de données et sans ajustement des modèles, et nous effectuons une indexation simple de k plus proche voisin (k=3 pour une classification sur Caltech101 et k=1 sur Feret). Nous nous intéressons au couches profondes car les caractéristiques sont de plus en plus globales.

- **Résultats sur FERET**

  Dans cette partie seul le réseau VGG-16 (Simonyan & Zisserman (2014)) a été évalué. Nous avons testé les caractéristiques issues de la couche Softmax (couche numéro 22) à la couche 13 (cartes de caractéristiques 28x28x512). Chaque image a été évaluée par la méthode de validation croisée (leave-one-out) en sélectionnant le plus proche voisin dans la base d'entrainement (1-NN). La figure 3.4 montre les courbes de précision de différentes représentations de caractéristiques. Pour chaque couche du modèle, nous avons extrait et normalisé (Z-score) les vecteurs global max pooling (GMAX), global average pooling (GAVG), la moyenne de GMAX et GAVG comme un cas général noté GeM inspiré de Radenović *et al.* (2018). Nous avons également adopté la technique de mise en correspondance de toutes les réponses des filtres comme des descripteurs locaux des images (independent pixel features).

  En analysant les courbes, nous trouvons que les caractéristiques fournies par des couches intermédiaires sont pertinentes comme les couches entièrement connectés et les précisions les plus élevés sont celles des caractéristiques locales qui atteignent 100% de précision pour la couches 13 et 14.

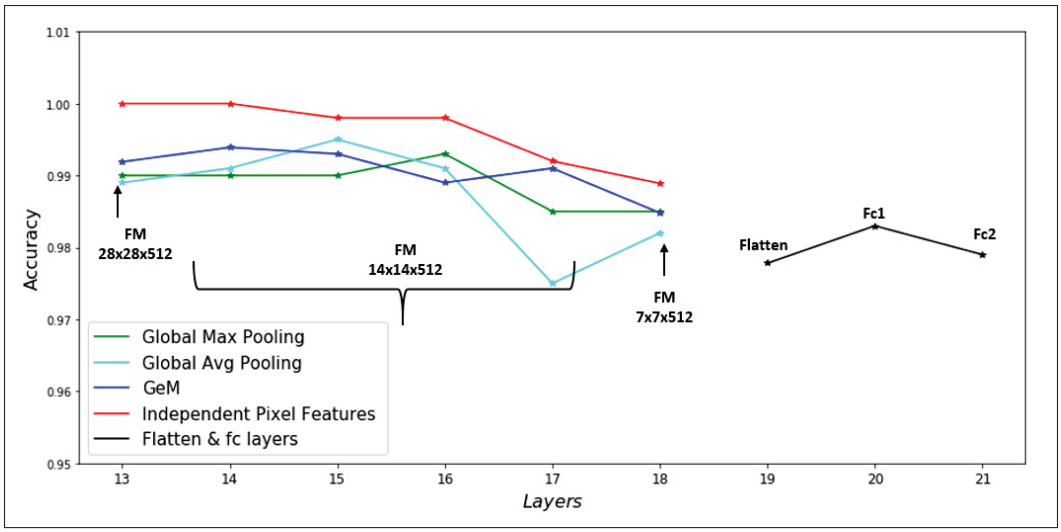


Figure 3.4 Les valeurs de précisions de différentes couches de VGG-16 en utilisant des caractéristiques normalisées avec standardization

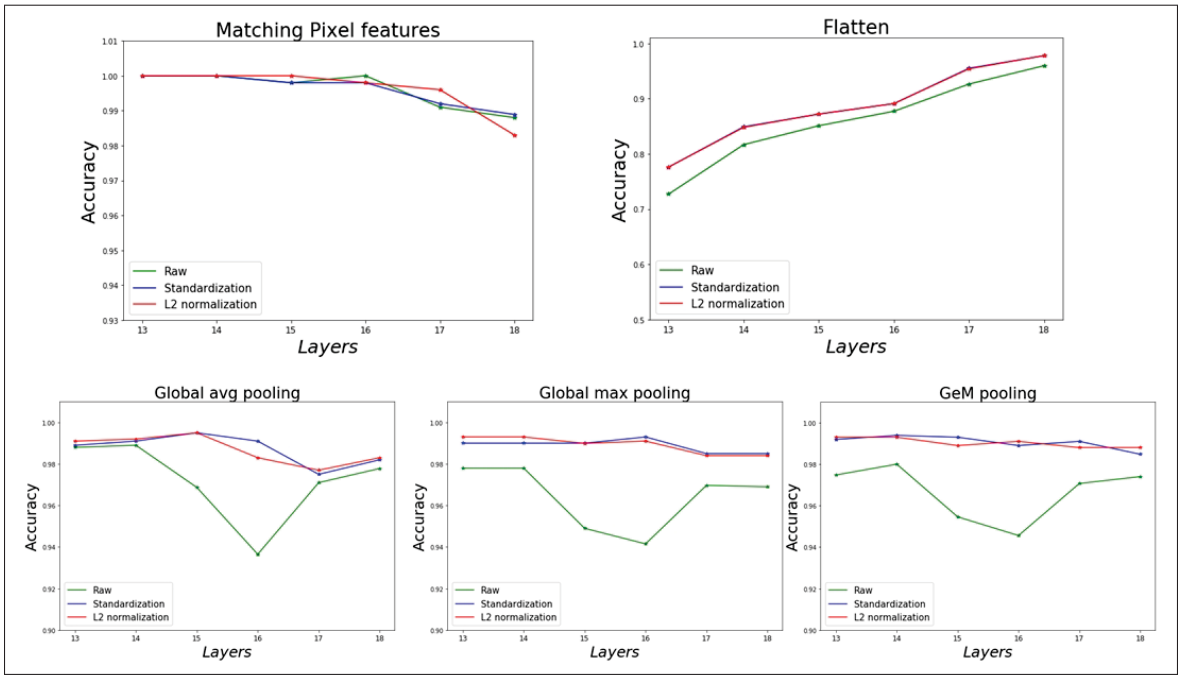


Figure 3.5 Comparaison de l'effet des normalisations sur les précisions des caractéristiques issues de différentes couche de VGG-16

Nos expériences montrent aussi que les normalisations testées (standardization et L2) permettent dans tous les cas d'améliorer les précisions. La figure 3.5 montre la différence entre les précisions données par des caractéristiques Raw (fournis par les filtres) et normalisées.

- **Résultats sur Caltech101**

La figure 3.6 (plus de détails dans l'annexe I) montre les valeurs de précision sur la base de données Caltech101 avec les caractéristiques de 100 dernières couches pour chaque modèle. La normalisation utilisée est la standardization qui performe toujours le taux de précision.

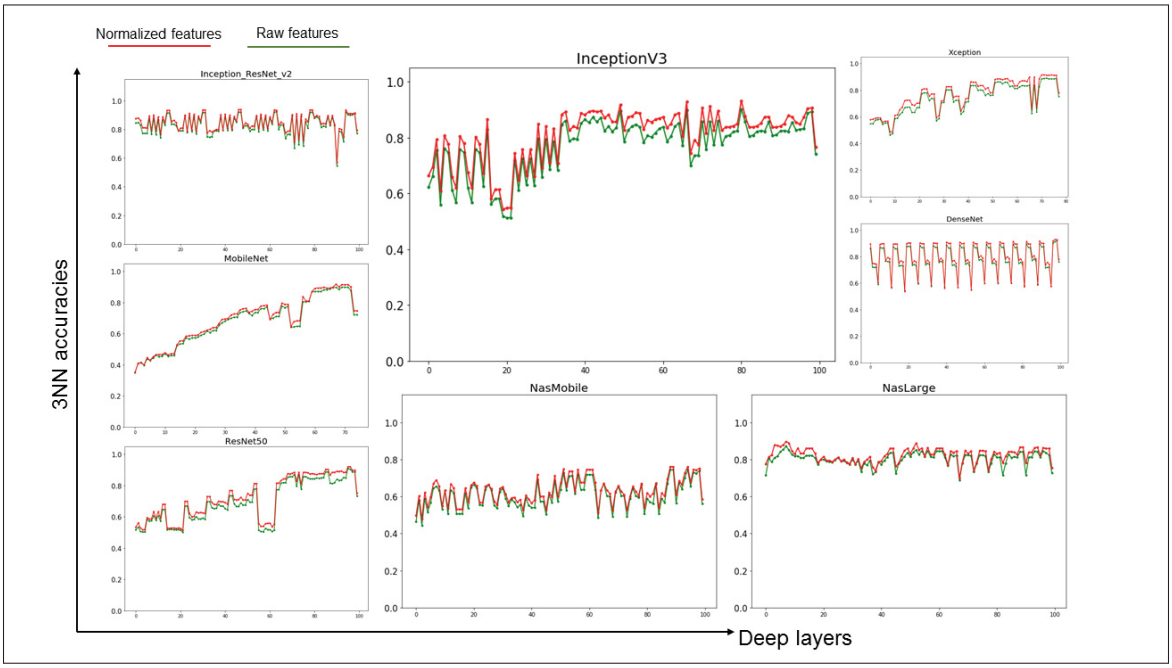


Figure 3.6 Courbes des valeurs de précision des caractéristiques non normalisées (Raw) et normalisées pour les 100 dernières couches des modèles pré-entraînés.

L'analyse de ces courbes montre que la concaténation (utilisée dans NasLarge(Zoph *et al.*), InceptionV3(Szegedy *et al.* (2016)), DenseNet121,169,201(Huang *et al.* (2017)) et InceptionResNetv2) ou l'addition (utilisée dans Xception(Chollet (2017)), ResNet(He *et al.*

(2016)) et NasMobile(Zoph *et al.*)) génèrent les vecteurs de caractéristiques qui mènent aux meilleures précisions. Les chutes des précisions sont les résultats des caractéristiques issues des couches de convolution. En effet, ces couches sont responsables à l'extraction des caractéristiques distinctives de ses entrées (cartes de caractéristiques issues d'autres couches). Tous les réseaux présentés dans la figure 3.6 sont formés par des modules (plusieurs couches en série) qui sont liés entre eux. MobileNet (Howard *et al.* (2017)) fait l'exception, toutes ses couches sont en série et on ne trouve pas ni des liaisons résiduelles avec concaténation ni avec addition. La chute de précision pour ce modèle est dûe aux couches de Zero Padding à la fin du réseau.

La figure 3.7 montre DenseNet comme exemple. Nous trouvons que les valeurs de précision varient d'une manière périodique. Les chutes de précision est toujours due aux couches de convolution, qui sont suivies d'une normalisation et une activation ReLU. La concaténation mène toujours à former des caractéristiques informatives ce qui explique les précisions élevées des sorties de ces couches.

Les couches qui donnent les caractéristiques les plus pertinentes sur Caletch101 pour tout les modèles testés sont données par le tableau 3.1.

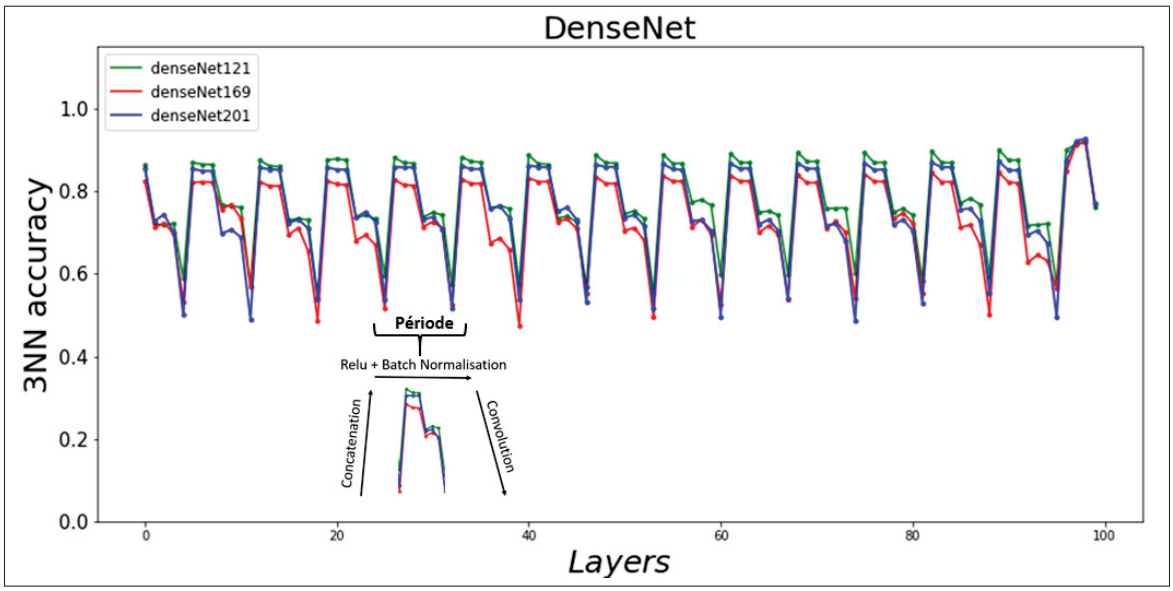


Figure 3.7 Les précisions des différentes couches de DenseNets

Tableau 3.1 Les meilleurs valeurs de précision en % sur Caltech101.

Modèle	Couche	Dim Maps	GMAX	GAVG	GeM	Flatten	Rep filtres
DenseNet121	425	7x7x1024	89.40	90.24	89.98	**92.36**	88.53
DenseNet169	593	7x7x1664	90.50	90.97	90.93	**92.51**	89.19
DenseNet201	704	7x7x1920	90.78	90.90	91.48	**93.02**	89.14
ResNet50	173	7x7x2048	89.85	87.45	90.07	**90.44**	86.26
InceptionV3	293	5x5x2048	90.42	89.94	90.24	**91.34**	89.65
Xception	126	7x7x1536	88.55	**90.09**	89.35	89.58	90.07
InceptionResNetV2	729	5x5x2080	90.82	90.71	90.82	**92.38**	90.60
MobileNet	94	7x7x1024	**89.17**	86.81	88.83	88.88	88.83
NasMobile	739	7x7x176	**73.29**	68.47	72.81	X	X
NasLarge	928	7x7x4032	86.70	85.2	**86.79**	X	X

3.3.2 Résultats de combinaison des vecteurs de caractéristiques des modèles pré-entraînés différents

Comme nous avons mentionné dans la section 2.2.3, la concaténation est la méthode utilisée pour combiner les vecteurs de caractéristiques qui donnent les meilleurs taux de précision suivant le tableau 3.1. La concaténation consiste à conserver toutes les informations là où elles se trouvent ce qui est très bénéfique pour notre approche d'indexation. Nous avons évité l'utilisation des vecteurs Flatten car ils sont très longs et consomment beaucoup de mémoire en traitement. Par conséquent, nous avons essayé de trouver la bonne combinaison avec les vecteurs GMAX, GAVG et GeM. Le tableau 3.2 résume les meilleurs résultats trouvés.

La meilleure précision obtenue est de 94.01% sans aucun entraînement ou affinage de modèles utilisés. Le tableau 3.3 montre une comparaison avec He *et al.* (2015).

3.3.3 Généraliser à partir de peu d'exemples (few shot learning)

Le "few-shot learning" est un domaine de l'apprentissage machine qui cherche à créer des modèles qui ont la capabilité d'obtenir plus de performance possible avec très peu de données. Idéalement, 1 à 5 exemples d'entraînement par classe. Cette idée est née de la capacité que

Tableau 3.2 Résultats de précision des combinaisons des caractéristiques de différents modèles

Modèle	Dim Vectors	Pécision (%)
$Xception \vert InceptionV3_{bestlayers}$	3584	92.23
$Xception \vert ResNet50_{bestlayers}$	3584	92.24
$Xception \vert InceptionResNetV2_{bestlayers}$	3616	92.98
$ResNet50 \vert InceptionResNetV2_{bestlayers}$	4128	93.26
$InceptionV3 \vert ResNet50_{bestlayers}$	4096	92.67
$DenseNet201 \vert ResNet50_{bestlayers}$	3968	93.12
$DenseNet201 \vert InceptionV3_{bestlayers}$	3968	93.09
$DenseNet201 \vert InceptionResNetV2_{bestlayers}$	4000	93.23
$DenseNet201 \vert InceptionResNetV2 \vert ResNet50_{bestlayers}$	6048	93.66
$DenseNet201 \vert InceptionResNetV2 \vert ResNet50 \vert Xception_{bestlayers}$	7584	**94.01**

Tableau 3.3 Comparaison de précision sur Caltech101 avec SPP entraîné sur imageNet (taille d'images 224x224)

Modèle	Pécision (%)
$SPPpool_{5/7}(He\ et\ al.\ (2015))$	89.47
$DenseNet201 \vert InceptionResNetV2 \vert ResNet50 \vert Xception_{bestlayers}$	**94.01**

nous les êtres humains avons le pouvoir de reconnaître un objet dès le premier coup d'oeil sans avoir besoin d'un entraînement intensif.

Dans cette expérience, nous avons essayé de tester la distinctivité des caractéristiques extraites des couches intermédiaires des réseaux pré-entraînés. En se basant sur les résultats générés par Li *et al.* (2020), nous avons fait le même test sur la base de données Caltech101. Nous évaluons les résultats de la classification en calculant la moyenne des précisions sur 600 épisodes. Pour 5-way 1-shot, la précision est calculée en se basant sur la mise en correspondance des images de test avec une seule image prise aléatoirement de chaque 5 classes dans la base d'entraînement. La même chose pour 5-way 5-shot, mais ici la prédiction est basée sur 5 images d'entraînement de 5 classes choisies aléatoirement. Le tableau 3.4 montre les résultats de nos caractéristiques par rapport aux résultats de l'état de l'art.

Tableau 3.4 La moyenne des résultats de précision qui est calculée sur 600 époques de test sur Caltech101.

Dataset		Wang *et al.* (2018)	Li *et al.* (2020)	DenseNet201$_{\textbf{bestlayer}}$
Caltech-101	5-way 1-shot	57.22 ± 0.85	61.00 ± 0.81	$\mathbf{86.25 \pm 0.16}$
	5-way 5-shot	75.34 ± 0.69	75.60 ± 0.66	$\mathbf{90.15 \pm 0.17}$

3.3.4 Résultats de binarisation des vecteurs de caractéristiques

Les résultats de binarisation des vecteurs de caractéristiques ont été combiné avec les travaux de M. Laurent Chauvint (Chauvin *et al.* (2019)) qui démontrent que 3D SIFT est la méthode la plus performante pour la classification des membres de famille et de sexe avec les images du cerveau humain. Dans cette section, nous allons détailler tout d'abord les résultats des vecteurs CNNs sur les données Caltech101, FERET et HCP-2D (images axiales) puis nous allons montrer les résultats de combinaison avec 3D SIFT (Toews & Wells III (2013)).

- **Résultats de Binarisation sur Caltech101**

  Dans cette expérience, nous avons utilisé la technique de validation croisée à k blocs (avec $k = 5$) où chaque bloc contient un nombre d'images identique pour chaque classe. Chaque image est représentée par un vecteur de caractéristiques Global max pooling. Le tableau 3.5 montre une comparaison entre les précisions obtenues en utilisant les caractéristiques non normalisées (Raw), normalisées avec Z-score (Standardization), normalisées en longueur unitaire (normalisation L2) et binaire.

  Ces résultats montrent que pour Caltech101 les précisions données par la méthode de binarisation proposée sont presque dans le même ordre que les précisions données par les caractéristiques Raw. La standardization est la méthode la plus performante pour la majorité des modèles.

- **Résultats de Binarisation sur FERET**

  - **Reconnaissance de visage**

    Dans cette expérience, nous avons deux images pour chaque personne. Nous avons utilisé la méthode de validation croisée pour chaque exemple (leave-one-out), dont le but

Tableau 3.5 Comparaison entre les valeurs moyennes des précisions en % de la validation croisée à 5 blocs sur Caltech101 entre les différents types de normalisation.

Modèle	Dim	Raw	Standardization	Normalisation L2	Binaire
$InceptionV3_{bestlayers}$	2048	85.89	**90.42**	90.40	87.67
$ResNet50_{bestlayers}$	2048	88.37	**89.85**	89.36	88.40
$DenseNet121_{bestlayers}$	1024	88.22	**89.39**	**89.47**	88.12
$DenseNet169_{bestlayers}$	1664	87.38	**90.49**	90.40	86.88
$DenseNet201_{bestlayers}$	1920	88.55	**90.77**	90.40	87.91
$MobileNet_{bestlayers}$	1024	86.79	**89.17**	87.56	87.25
$Xception_{bestlayers}$	1536	84.92	**88.55**	86.48	85.07
$NasMobile_{bestlayers}$	176	71.86	73.30	**73.32**	69.31
$NasLarge_{bestlayers}$	4032	84.85	**86.70**	86.48	83.93
$InceptionResNetV2_{bestlayers}$	2080	88.17	90.82	**90.97**	89.43

de trouver pour chaque image de test sa correspondante en entraînement (en mémoire). Le tableau 3.6 donne les valeurs de précisions pour tous le modèles testés.

Nous remarquons que les caractéristiques non normlisées (Raw) donnent les meilleurs précisions pour la majorité des modèles. En effet, la reconnaissance de visage est basée sur la recherche des similitudes entre les formes de visage, d'où si deux images sont identiques, les réponses des filtres vont être similaires. Dans ce cas, la normalisation peut chevaucher les informations ce qui explique les résultats trouvés.

Tableau 3.6 Comparaison entre les valeurs des précisions en % de reconnaissance de visage sur la base de données FERET.

Modèle	Dim	Raw	Standardization	Normalisation L2	Binaire
$InceptionV3_{bestlayers}$	2048	87.96	86.40	91.09	**91.81**
$ResNet50_{bestlayers}$	2048	**98.48**	90.60	91.71	92.15
$DenseNet121_{bestlayers}$	1024	**97.63**	92.42	92.15	92.46
$DenseNet169_{bestlayers}$	1664	92.09	**92.81**	92.31	92.69
$DenseNet201_{bestlayers}$	1920	**95.21**	93.42	92.57	92.95
$MobileNet_{bestlayers}$	1024	**98.43**	94.15	92.94	93.19
$Xception_{bestlayers}$	1536	**96.57**	94.58	93.18	93.39
$NasMobile_{bestlayers}$	176	81.92	**93.38**	92.71	**93.38**
$NasLarge_{bestlayers}$	4032	73.96	91.77	91.93	**93.06**
$InceptionResNetV2_{bestlayers}$	2080	83.28	91.16	91.68	**93**

- **Classification du sexe à partir des visages**

  La classification selon le sexe est une classification binaire. Les travaux présentés dans cette section sont inspirés de Toews & Arbel (2008). Nous avons utilisé un sous ensemble de 400 images des visages frontales des femmes et d'hommes. En choisissant, les couches qui fournissent les caractéristiques les plus pertinentes selon l'étude de la section 3.3.1, chaque image est représentée par un vecteur de caractéristiques Global max pooling. Les figures 3.8 et 3.9 montrent que la binarisation donne toujours les meilleurs résultats.

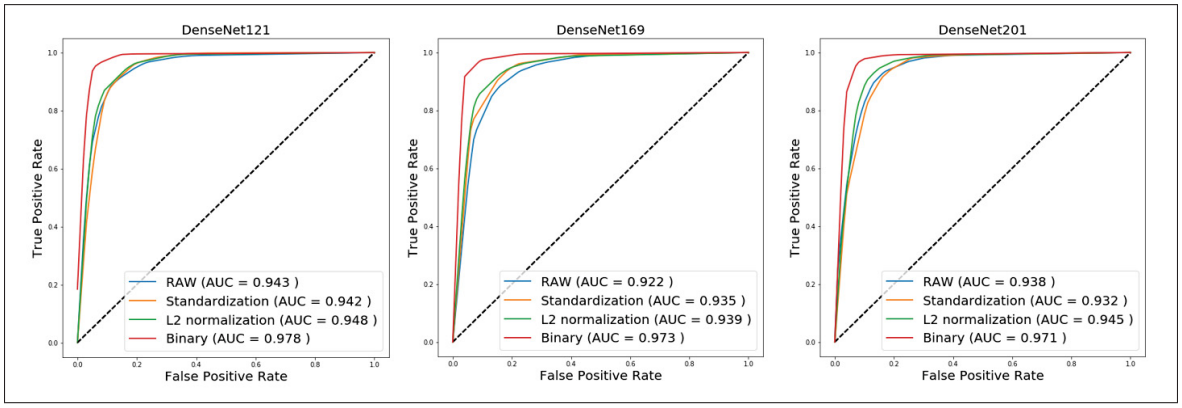


Figure 3.8 Les courbes ROC pour la classification des visages selon le sexe des caractéristiques Raw, normalisées (standardization, normalisation L2) et binaire pour DenseNet 121, 169 et 201 sur FERET.

Nous avons réalisé une comparaison avec l'algorithme SIFT-2D qui est connu par sa robustesse pour la tâche de reconnaissance des formes, et nous avons trouvé que les caractéristiques CNN surpassent la performance de SIFT-2D dans cette application. La figure 3.10 montre une comparaison avec les courbes ROC.

- **Résultats de classification des images IRM du cerveau humain (HCP)**

  - **Classification du sexe à partir des images IRM du cerveau**

    Dans cette partie, nous avons utilisé un sous-ensemble de 424 images de la base HCP en sélectionnant un seul membre de chaque famille avec un nombre identique de mâles et de femelles.

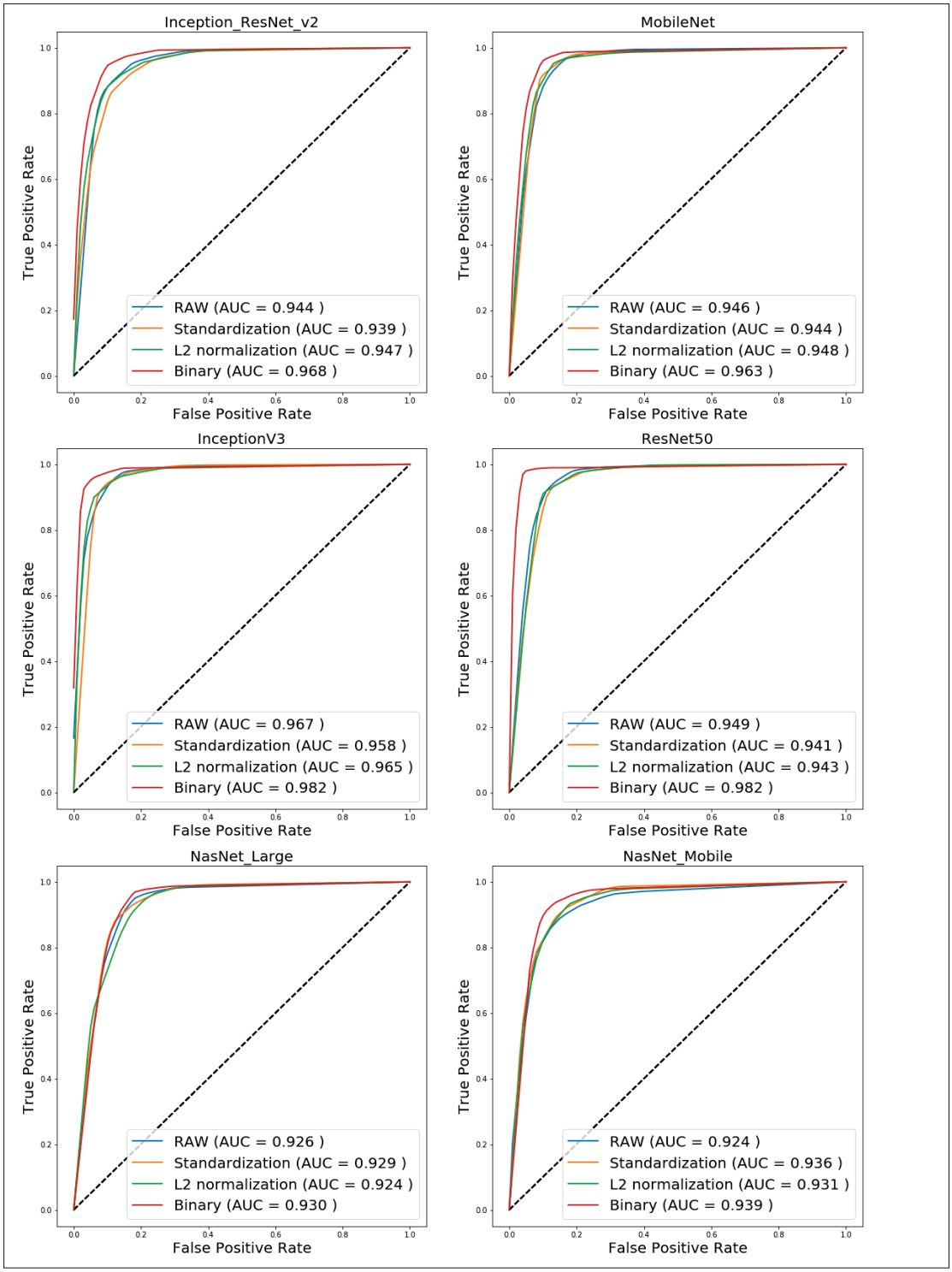


Figure 3.9 Les Courbes ROC de classification des visages selon le sexe par des vecteurs des caractéristiques Raw (GMAX), normalisées (standardization, normalisation L2) et binaire pour les modèles InceptionV3, ResNet50, MobileNet, NasLarge, NasMobile et InceptionResnetV2 sur FERET

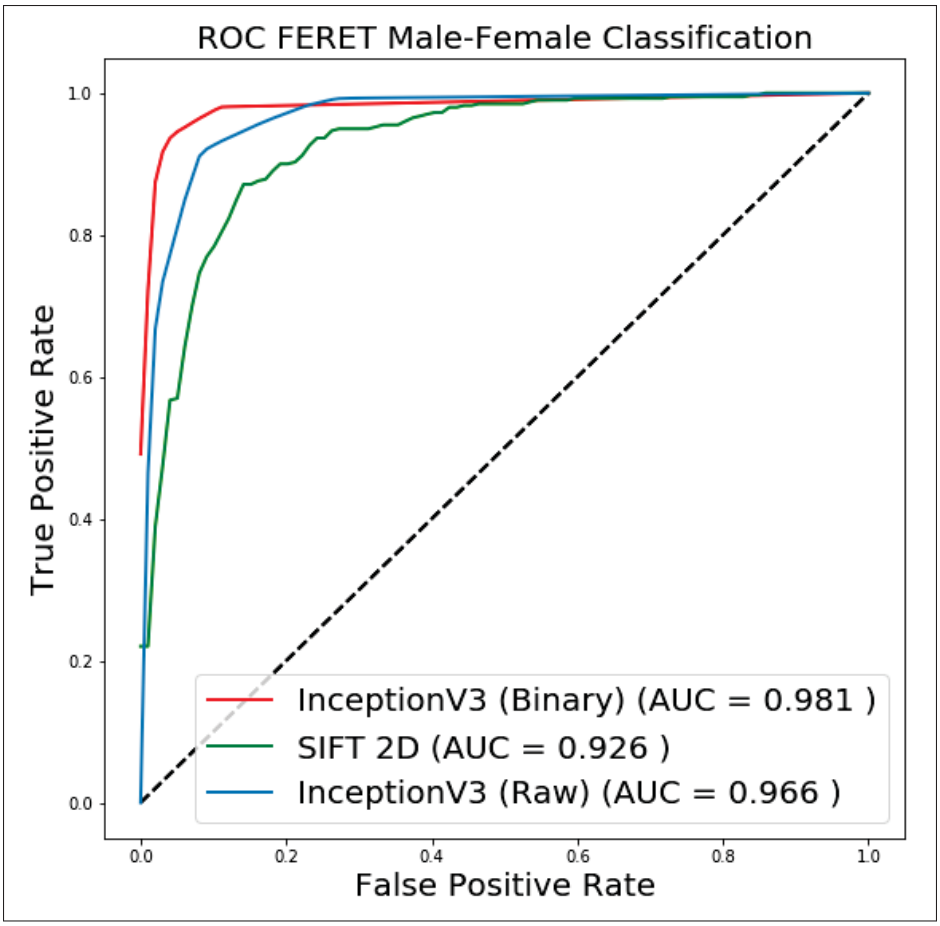


Figure 3.10 Comparaison entre la performance de SIFT-2D et InceptionV3 pour la classification des visages selon le sexe sur FERET

L'évaluation est faite par la méthode de validation croisée pour chaque image (leave one out). La figure 3.11 et 3.12 montrent l'impact de la binarisation sur les valeurs de précisions. En effet, pour tous les modèles utilisés, il y a des améliorations des taux de AUC.

Ces résultats montrent que le modèle le plus performant est DenseNet201 pour la couche 704 qui fournit un vecteur de 1920 caractéristiques par image. Ce modèle est bien utilisé pour les prochaines expériences.

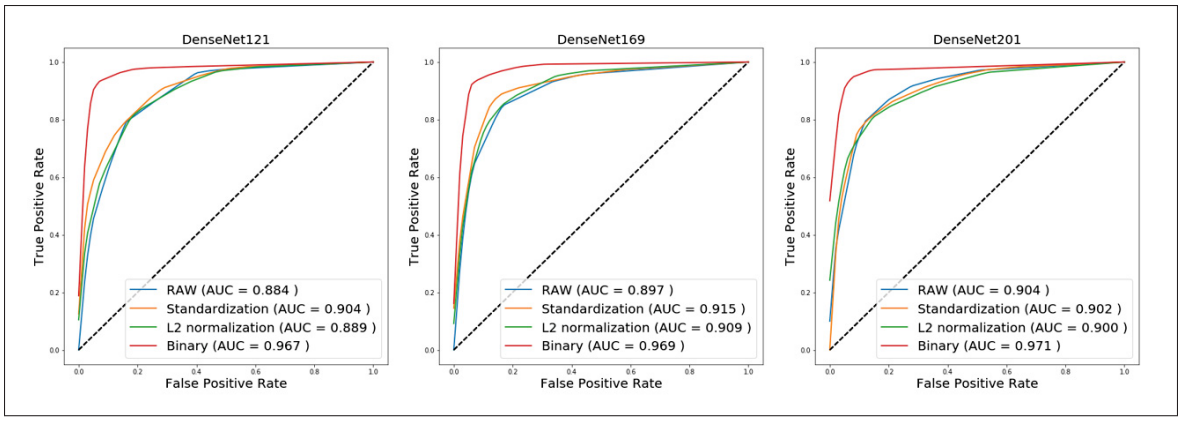


Figure 3.11 Les courbes ROC pour la classification des images IRM du cerveau selon le sexe par des caractéristiques Raw, normalisées (standardization, normalisation L2) et binaire pour DenseNet 121, 169 et 201 sur HCP.

- **Classification des membres de famille**

  Pour cette expérience, nous sommes dans le cas de classification de plusieurs classes avec peu de données (de 1 à 4 membres de famille). Nous avons 1010 images regroupés pour 439 familles selon le ID de la mère, certaines familles ne comprennent qu'une seule image (un seul membre de famille). Le tableau 3.7 montre les meilleurs résultats qui sont donnés par DenseNet201.

Tableau 3.7 Les valeurs de AUC de modèle DenseNet201 pour la classification de membres de famille.

Modèle	**Classification de membres de famille (AUC)**
DenseNet201 (Binaire)	0.8305
DenseNet201 (Raw)	0.6882

- **Résultats Combinaison SIFT-3D avec CNN**

Les résultats des combinaisons entre les différents modèles (SIFT+CNN) sont le résultat de collaboration avec Laurent Chauvint [4] qui travaille aussi sous la direction mon superviseur M.Toews. Le 3D-SIFT-Rank Toews & Wells III (2013) fournit des descripteurs de 64 carac-

4. https ://github.com/lchauvin

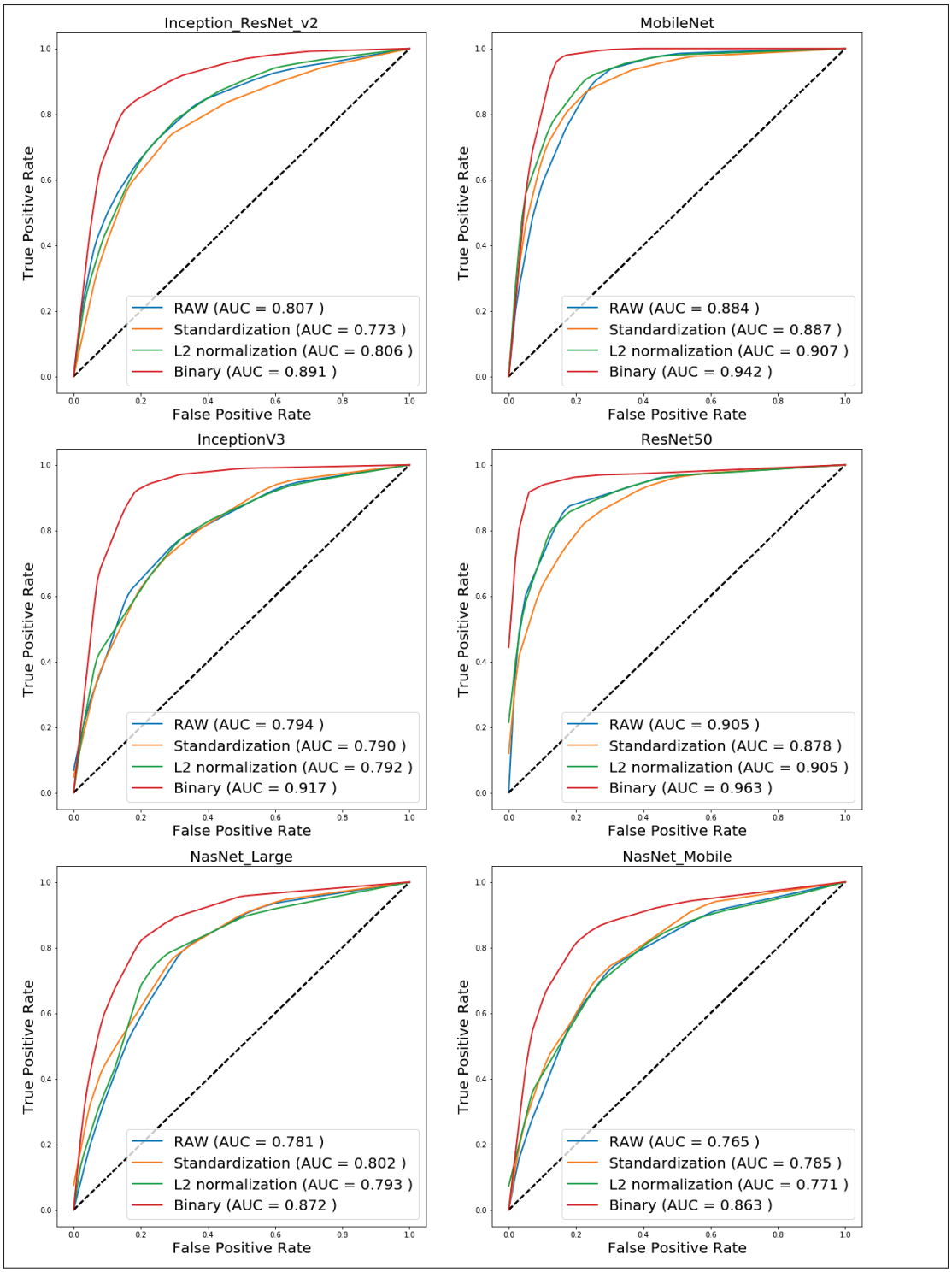
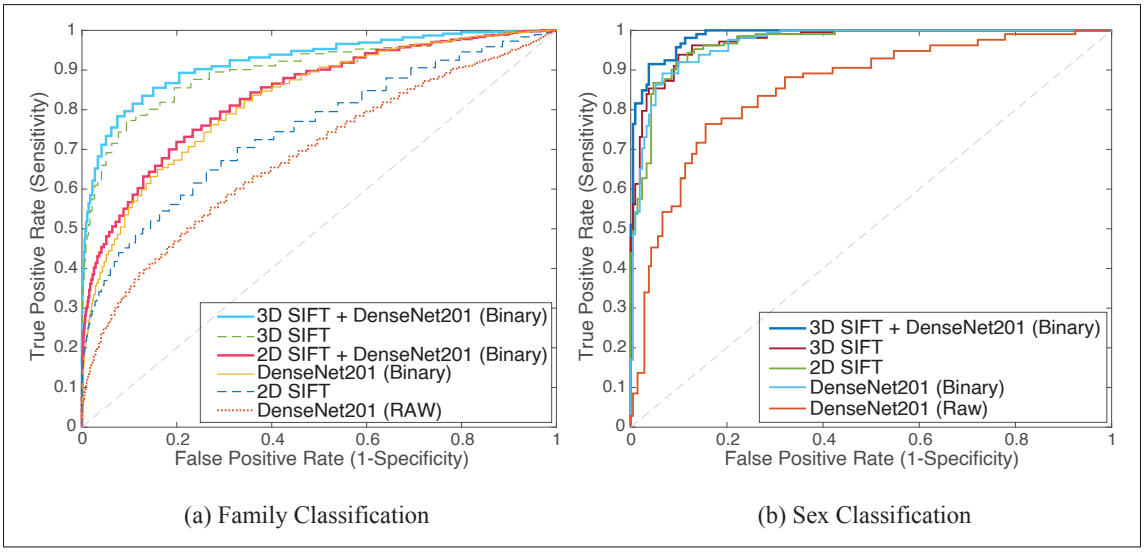


Figure 3.12 Les courbes ROC pour la classification des images IRM du cerveau selon le sexe par des vecteurs caractéristiques (GMAX) Raw, normalisées (standardization, normalisation L2) et binaire pour les modèles InceptionV3, ResNet50, MobileNet, NasLarge, NasMobile et InceptionResnetV2 sur la base de données HCP

téristiques et représente l'état de l'art des meilleurs résultats sur la base données HCP. Nous réalisons une combinaison entre le vecteur binaire CNN et le 3D-SIFT par un paramètre de pondération $\alpha \in [0, 1]$.



(a) Family Classification (b) Sex Classification

Figure 3.13 Les courbes ROC pour des différents modèles et combinaisons sur HCP, la combinaison entre 3D SIFT-Rank et les caractéristiques CNN binaires donnent les meilleurs résultats de AUC pour la classification de sexe (98.75%) et classification des membres de familles (92.58%) .

La figure 3.13 présente les courbes de ROC pour différents modèles (SIFT-2D , SIFT-3D, DenseNet + SIFT-2D, DenseNet + SIFT-3D, etc). Ces résultats montrent que la combinaison de SIFT-3D et DenseNet201 binaire est le modèle le plus performant qui surpasse la performance de SIFT-3D qui est le modèle le plus efficace dans l'état de l'art. Le tableau 3.8 résume tout les valeurs de AUC pour les deux cas de classification avec les modèles les plus pertinents.

3.3.5 Discussion

Les résultats trouvés pour la classification des visages sur la base de données FERET montrent que la mise en correspondance des réponses des filtres comme des descripteurs locaux est la

Tableau 3.8 Résultats de combinaison des différents modèles pour la classification des membres de famille et la classification selon le sexe en se basant sur les ROC de la figure 3.13.

Modèle	Membre de famille (AUC)	Sexe (AUC)
3D SIFT + DenseNet201 (B)	**0.9258**	**0.9875**
3D SIFT	0.9058	0.9712
2D SIFT + DenseNet201 (B)	0.8420	-
DenseNet201 (B)	0.8305	0.971
2D SIFT	0.7590	0.9652
DenseNet201	0.6882	0.904

méthode la plus performante. En plus, les couches intermédiaires du réseau VGG16 semblent pertinentes pour l'extraction des vecteurs de caractéristiques (GMAX,GAVG etc) et aussi à décrire le contenu d'une image d'une façon efficace par des réponses de filtres. La tache de reconnaissance de visage est un cas particulier de reconnaissance d'objets, donc c'est logique que l'utilisation les cartes de caractéristiques comme des descripteurs locaux mène au meilleur résultat.Dans ce genre de problème de classification, les méthodes traditionnelles comme SIFT sont pertinentes car elles sont robustes à la rotation, au changement d'échelle et à la translation. En analysant les résultats trouvés sur Caltech101 pour différents modèles pré-entraînés, on peut dire que ces modèles contiennent des informations génériques dans les couches cachées. La figure 3.6 montre que certaines couches cachées génèrent des caractéristiques plus génériques que les dernières couches. Ces caractéristiques ont montré une efficacité surprenante sur Caltech101. Les résultats aussi montrent que la concaténation de plusieurs vecteurs de caractéristiques de différents modèles est une bonne idée pour combiner les pouvoirs de classification. Le test avec la méthode de few-shot-learning confirme que les caractéristiques, issues des couches bien précises de ces modèles, sont génériques et surpassent les résultats de l'état de l'art.

La binarisation des vecteurs de caractéristiques avec le seuil qui donne le gain d'information le plus élevé, montre une bonne efficacité en le comparant par la standardization et la normalisation L2. Cela est dû aux informations acquises des labels des données d'entraînement. En plus, nos résultats montrent une complémentarité entre les caractéristiques CNN et les carac-

téristiques locaux SIFT. Ce dernier semble plus efficace pour détecter les variabilités fines des images (indexation des membres de la famille), le codage CNN est excellent pour les vastes catégories (par exemple les étiquettes de sexe).

CONCLUSION

4.1 Résumé des contributions

Dans ce travail, nous avons proposé un modèle bayésien basé sur une étude des réseaux de neurones pré-entrainés en cherchant les couches qui donnent les caractéristiques le plus pertinentes. En plus, nous proposons une méthode de normalisation (binarisation) des caractéristiques en utilisant le concept de gain d'information. Dans le chapitre 1, nous avons présenté brièvement le concept de l'apprentissage profond et des réseaux de neurones convolutifs. Ensuite, nous avons présenté les architectures populaires entraînées sur ImageNet et enfin, nous avons présenté un aperçu général sur les représentations des images par des vecteurs caractéristiques. Dans le chapitre 2, nous avons proposé notre méthodologie sur la recherche des couches qui fournissent les caractéristiques les plus robustes en utilisant une approche bayésienne d'indexation et nous avons appliqué la théorie d'information pour former des vecteurs CNN binaires comme une méthode de normalisation et compression de données. Enfin, dans le chapitre 3, nous avons montré l'analyse effectuée sur les CNNs pré-entrainés. Nos expériences ont montré un grand impact sur des problèmes de classification multiples et binaires sur des bases de données d'objets générales, visages et imagerie médicale, et ouvrent de nouvelles perspectives sur la complémentarité des caractéristiques de CNN profond et de SIFT.

4.2 Limites et perspectives

Même si les expériences ont montré de résultats prometteurs, nous font valoir qu'il reste encore certaines limites et recommandations pour un travail futur. Ainsi, nous suggérons les recommandations suivantes pour les prochaines étapes de notre problème de recherche :

- Dans nos expériences, nous avons utilisé 10 réseaux pré-entraînés mais il reste encore beaucoup de modèles qui sont disponibles en ligne à tester ;

- La comparaison avec l'état de l'art est limitée, des tests en futur vont être effectués sur d'autres base de données comme PASCAL VOC Everingham *et al.* (2015) et COCO Lin *et al.* (2014) ;
- La binarisation a montré une bonne performance pour la reconnaissance d'objets et la classification binaire mais ne performe pas assez bien pour des problèmes de catégorisation, ce résultat doit être validé sur d'autres bases de données ;

ANNEXE I

DÉTAILS SUR LES FIGURES DE QUELQUES ARCHITECTURES

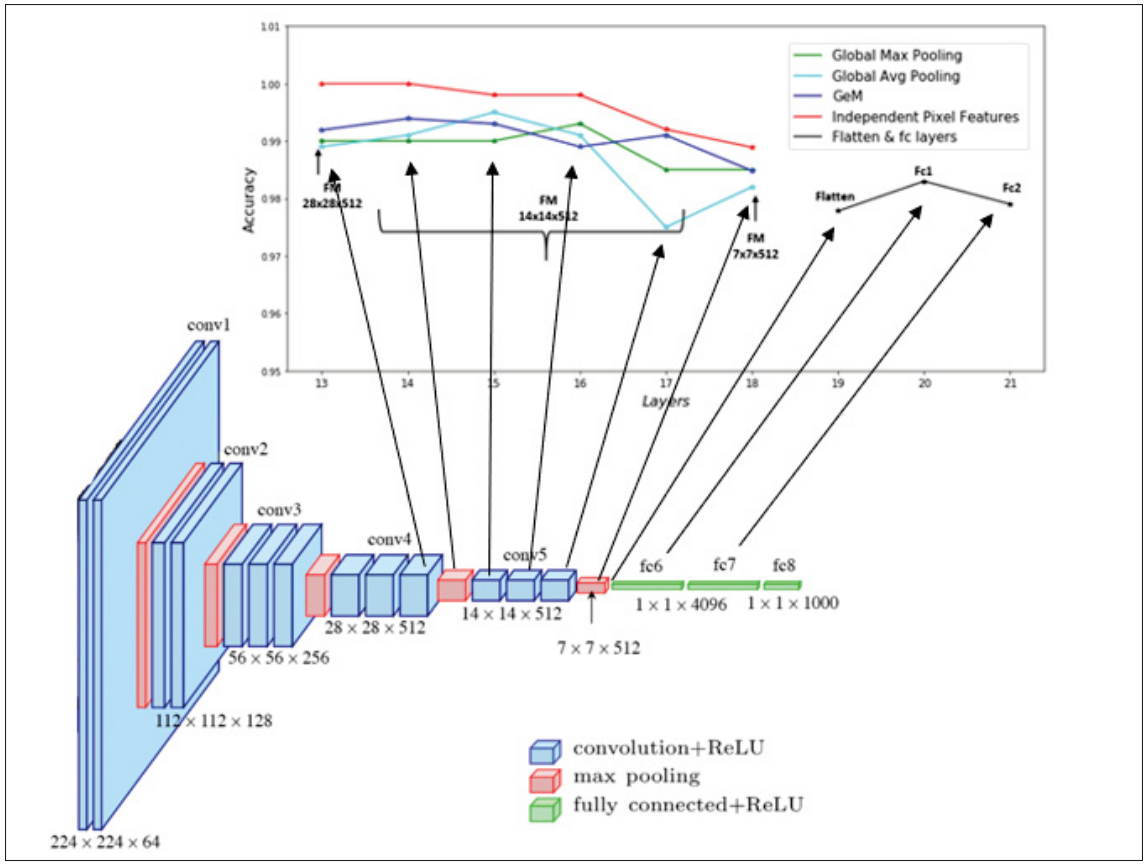


Figure-A I-1 Courbes des valeurs de précision des différents représentations d'images issues du modèle VGG-16 pour la reconnaissance de visage sut FERET

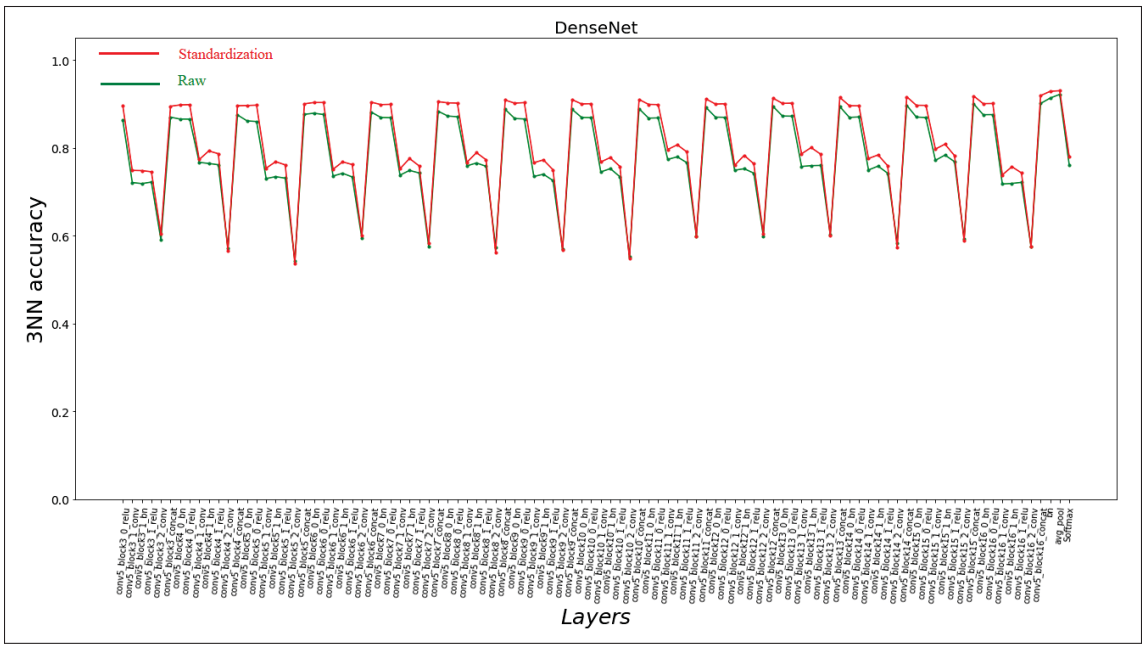


Figure-A I-2 Courbes des valeurs de précision sur Caltech101 des caractéristiques non normalisées (Raw) et normalisées pour les 100 dernières couches du DenseNet121.

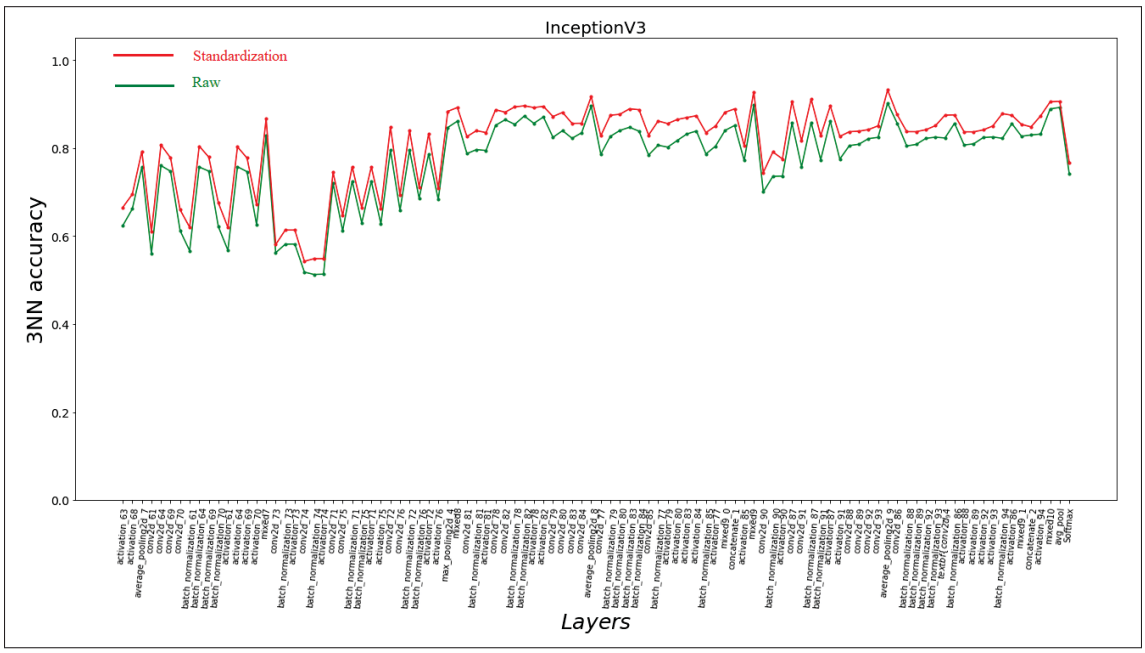


Figure-A I-3 Courbes des valeurs de précision sur Caltech101 des caractéristiques non normalisées (Raw) et normalisées pour les 100 dernières couches du InceptionV3.

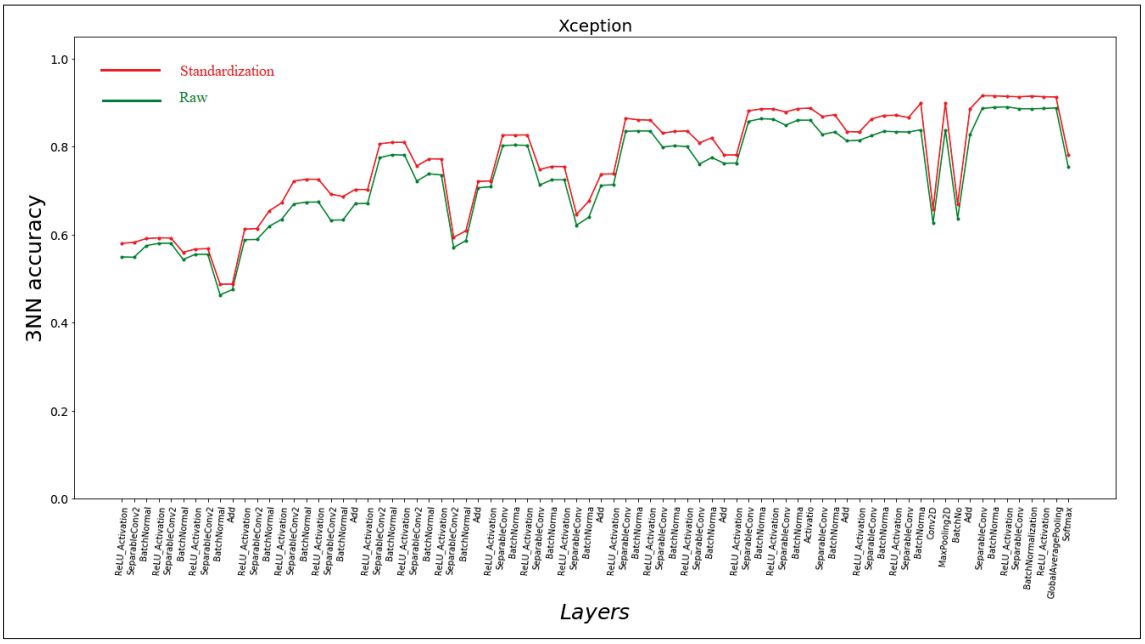


Figure-A I-4 Courbes des valeurs de précision sur Caltech101 des caractéristiques non normalisées (Raw) et normalisées pour les 100 dernières couches du Xception.

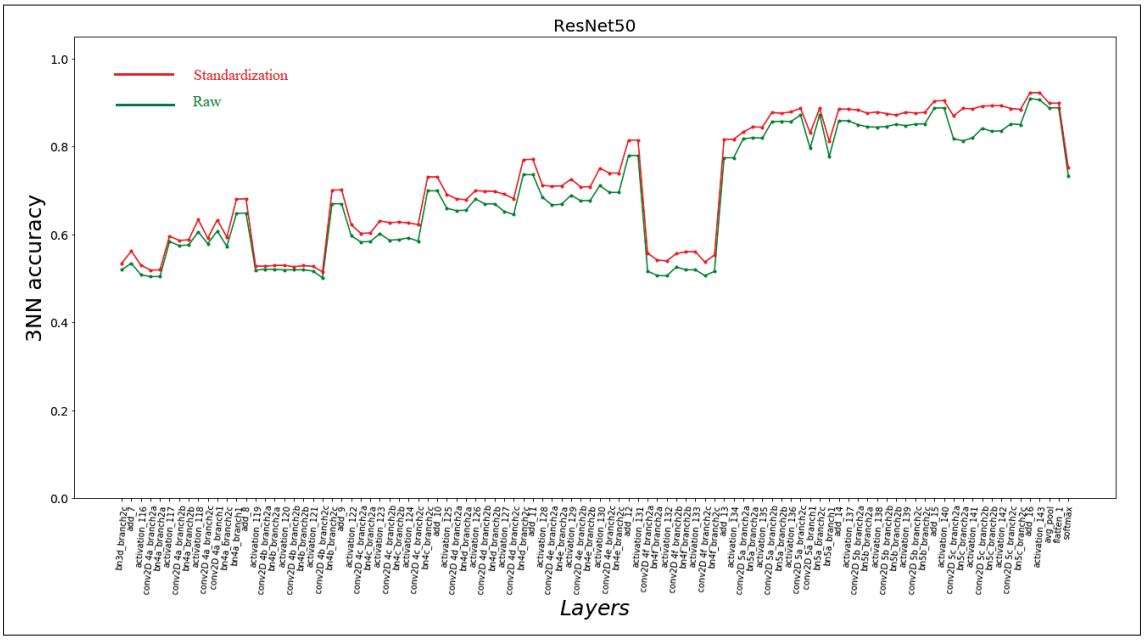


Figure-A I-5 Courbes des valeurs de précision sur Caltech101 des caractéristiques non normalisées (Raw) et normalisées pour les 100 dernières couches du ResNet50.

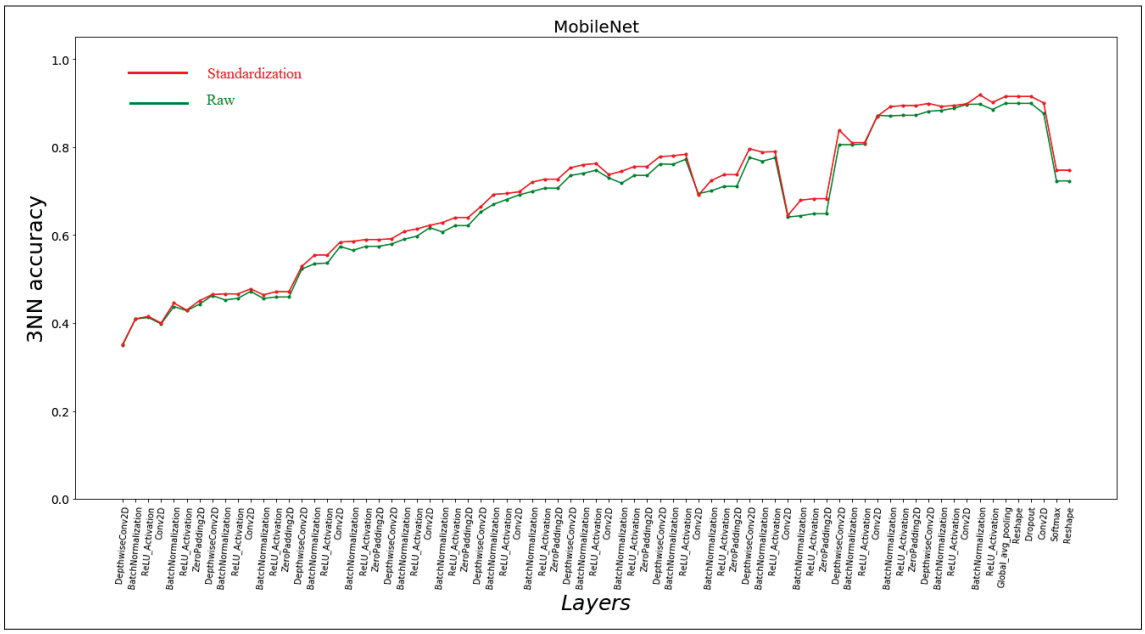


Figure-A I-6 Courbes des valeurs de précision sur Caltech101 des caractéristiques non normalisées (Raw) et normalisées pour les dernières couches du MobileNet.

ANNEXE II

AUTRE TRAVAUX DE RECHERCHE EFFECTUÉS : COMBINAISON ENTRE LES POINTS CLÉS ET RÉSEAUX DE NEURONES CONVOLUTIFS

1. Augmentation de données avec des régions SIFT

L'analyse du contenu d'une image fait référence au processus de compréhension du contenu de cette image afin que nous puissions agir en conséquence. Faisons un pas en arrière et parlons de la façon dont les humains le font. Notre cerveau est une machine extrêmement puissante capable de faire des choses compliquées très rapidement. Lorsque nous regardons quelque chose, notre cerveau crée automatiquement une empreinte basée sur les aspects intéressants de cette image. Ces aspects intéressants en général sont des choses qui sont distincts et uniques.

Les algorithmes de détection des points locaux comme SIFT (Lowe (2004)) et SURF (Bay *et al.* (2006)) permettent de détecter des points bien distinctifs et uniques avec des coordonnées géométriques et un descripteur d'apparence.

L'idée de cette première contribution se concentre sur l'utilisation des régions centrées aux points clés détectés.

Après la détection des points clés, nous sélectionnons les régions centrées dans ces points avec un taille de *Dimx* x *Dimy* (*Dimx* = *Dimy* = 64+ σ) et une orientation θ. Cette façon de représenter les données a pour but de permettre les modèles de réseau de neurones convolutifs d'apprendre les différentes régions d'intérêt dans les visages d'hommes et femmes avec des orientations différentes même si nous donnons en entrée des visages bien alignés en 0 degré. La figure II-1 résume cette approche.

2. Amélioration de la robustesse du réseau de neurones convolutifs contre la rotation

2.1 Base de données : Rot-FERET [1]

1. https ://drive.google.com/file/d/1GnfnjHeb0rDEm6lB6Hjb6Gz4YxyoVf4F/view ?usp=sharing

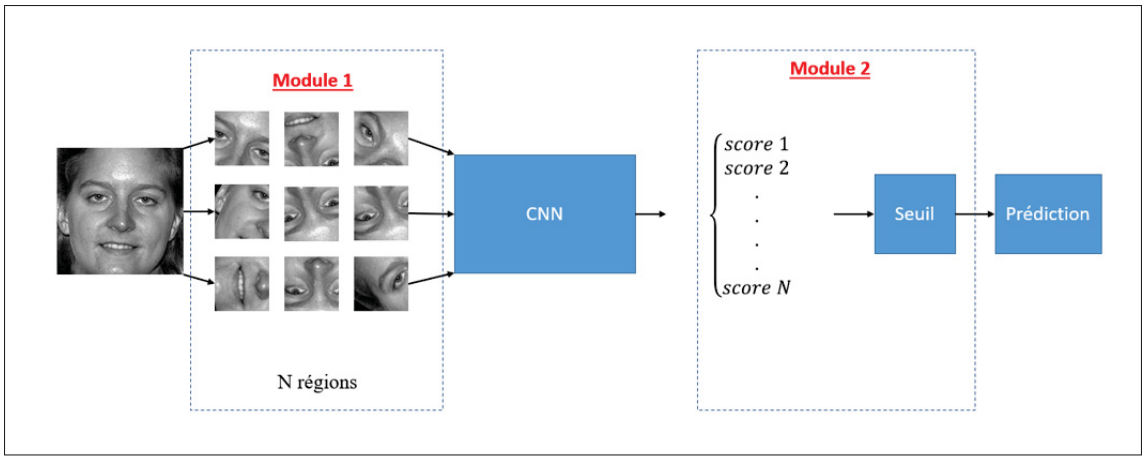


Figure-A II-1 Architecture proposé pour l'augmentation de données avec des régions d'intérêt

nous avons divisé la base de données en une base d'entrainement contenant 700 images (350 hommes et 350 femmes) et une base de test de 100 images (50 hommes et 50 femmes). L'objectif de notre étude est de bien améliorer la robustesse de la classification contre l'invariance aux rotations des données de visage d'hommes et femmes. Afin de bien se concentrer aux information des visages, nous avons segmenté les images avec la méthode Viola & Jones (2004) pour les deux parties de la base de données (entrainement et test).

En plus, nous avons besoin de tester les modèles entraînés sur des visages orientés pour voir l'impact de la rotation sur le taux de classification. Afin de réaliser cette tâche nous avons créé d'autres bases de test avec 100 images (50 hommes / 50 femmes) orientées de 1 à 359 dégrée (avec un pas de 1 dégrée). Pour orienter un visage segmenté à partir d'une image originale, on a développé un algorithme qui permet d'extraire des régions avec une orientation donnée sans avoir des bordures des pixels ajoutés comme indique la figure II-2.

2.2 Classification Multi-CNNs

Dans cette approche, nous avons dévisé nos données (régions de visage) en des clusters avec la algorithme "k-means" MacQueen *et al.* (1967) pour avoir une homogénéité entre les régions du même motif. Nous avons entrainé un CNN pour chaque cluster puis nous avons combiné

Figure-A II-2 Exemple des visages orientés dans la base de test

les prédictions de classification en utilisant un réseau de neurones entièrement connectés ou un foret aléatoire. La figure II-3 montre l'architecture proposée pour k=4.

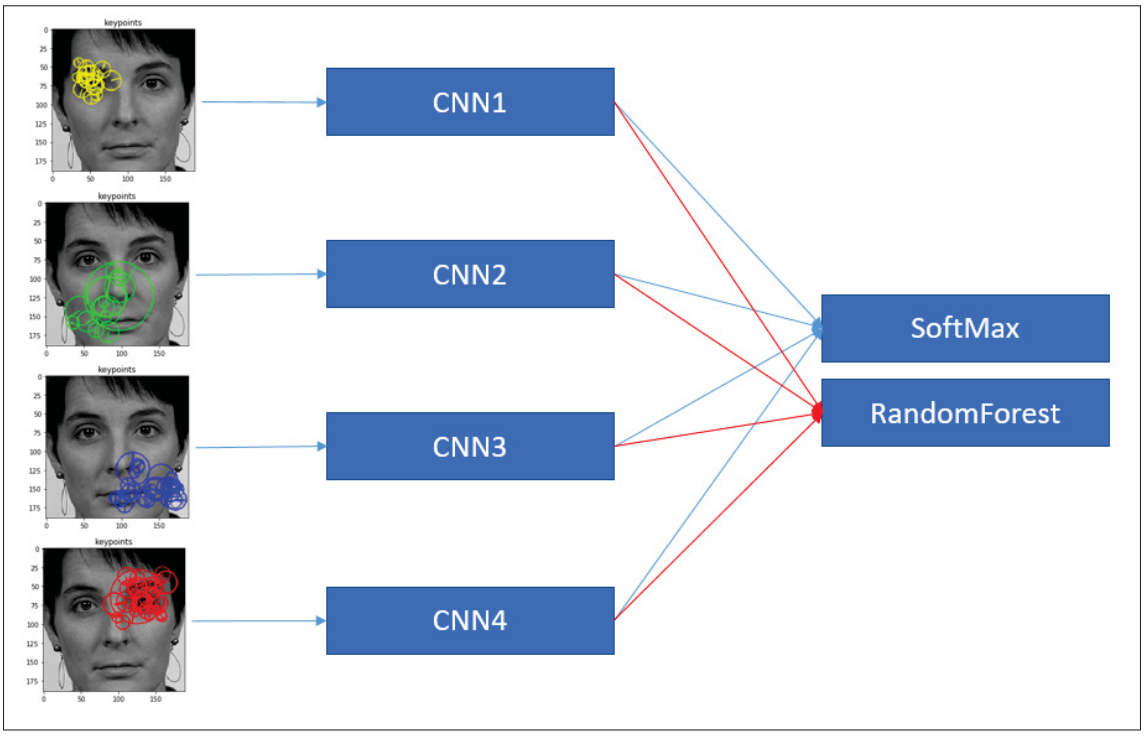


Figure-A II-3 Méthodologie proposée pour la classification des visages orienté avec multi-CNNs entraîné à partir des Régions d'intérêts

2.3 Module de transformation spatial basé sur les points clés combinés avec un réseau de neurones convolutifs

Cette approche est inspirée du papier Rajalingham *et al.* (2010). Elle consiste à extraire les points d'intérêts de toute la base de données d'entraînement puis pour chaque image de test orientée, nous estimons les correspondances des points clés et enfin pour chaque paire de points nous calculons une transformation affine telle que :

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \sigma' \cos\theta' & \sigma' \sin\theta' & dx \\ \sigma' \sin\theta' & \sigma' \cos\theta' & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{A II-1}$$

avec :

$$\sigma' = \frac{\sigma_2}{\sigma_1} \tag{A II-2}$$

$$\theta' = \theta_2 - \theta_1 [2\pi] \tag{A II-3}$$

$$d = \sqrt{dx^2 + dy^2} \tag{A II-4}$$

Nous éliminons les faussent correspondances en se basant sur des seuils appliqués sur θ' ,σ' et d. La prédiction des classes est faite par un modèle entraîné sans augmentation de données.

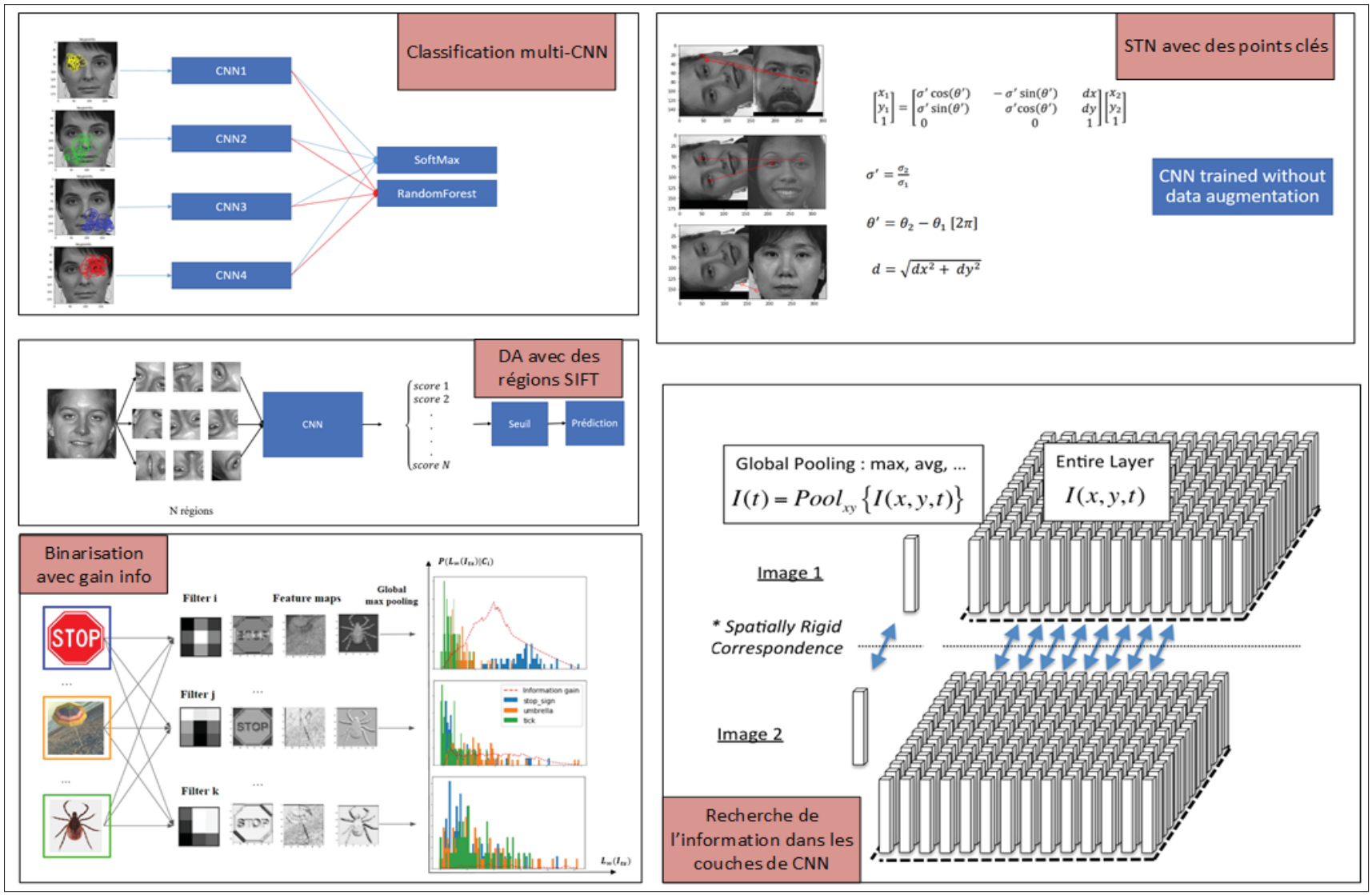


Figure-A II-4 Résumé des travaux réalisés durant ma maîtrise

BIBLIOGRAPHIE

Arandjelović, R. & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2911–2918.

Azizpour, H., Razavian, A. S., Sullivan, J., Maki, A. & Carlsson, S. (2015). Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9), 1790–1802.

Babenko, A. & Lempitsky, V. (2015). Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv :1510.07493*.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf : Speeded up robust features. *European conference on computer vision*, pp. 404–417.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9), 509–517.

Bo, L. & Sminchisescu, C. (2009). Efficient match kernel between sets of features for visual recognition. *Advances in neural information processing systems*, pp. 135–143.

Chandrasekhar, V., Lin, J., Morere, O., Goh, H. & Veillard, A. (2016). A practical guide to CNNs and Fisher Vectors for image instance retrieval. *Signal Processing*, 128, 426–439.

Chauvin, L., Kumar, K., Desrosiers, C., De Guise, J., Wells, W. & Toews, M. (2019). Analyzing brain morphology on the bag-of-features manifold. *International Conference on Information Processing in Medical Imaging*, pp. 45–56.

Chauvin, L., Kumar, K., Wachinger, C., Vangel, M., de Guise, J., Desrosiers, C., Wells, W., Toews, M., Initiative, A. D. N. et al. (2020). Neuroimage signature from salient keypoints is highly specific to individuals and shared by close relatives. *NeuroImage*, 204, 116208.

Chollet, F. (2017, 11). Xception : Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January, 1800–1807. doi : 10.1109/CVPR.2017.195.

Dechter, R. (1986). *Learning while searching in constraint-satisfaction problems*. University of California, Computer Science Department, Cognitive Systems ....

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). ImageNet : A Large-Scale Hierarchical Image Database. *CVPR09*.

Deniz, E., Şengür, A., Kadiroğlu, Z., Guo, Y., Bajaj, V. & Budak, Ü. (2018). Transfer learning based histopathologic image classification for breast cancer detection. *Health information science and systems*, 6(1), 18.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2015). The pascal visual object classes challenge : A retrospective. *International journal of computer vision*, 111(1), 98–136.

Fei-Fei, L., Fergus, R. & Perona, P. (2004). Learning generative visual models from few training examples : An incremental bayesian approach tested on 101 object categories. *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178.

Harris, C. G., Stephens, M. et al. (1988). A combined corner and edge detector. *Alvey vision conference*, 15(50), 10–5244.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904–1916.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition*, 1, 278–282.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017). Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*.

Hu, J., Shen, L. & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Hubel, D. H. & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1), 106–154.

Jégou, H., Douze, M., Schmid, C. & Pérez, P. (2010). Aggregating local descriptors into a compact image representation. *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3304–3311.

Juan, L. & Gwun, O. (2009). A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4), 143–152.

Kalantidis, Y., Mellina, C. & Osindero, S. (2016). Cross-dimensional weighting for aggregated deep convolutional features. *European conference on computer vision*, pp. 685–701.

Kaur, T. & Gandhi, T. K. (2020). Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications*, 31, 1–16.

Ke, Y. & Sukthankar, R. (2004). PCA-SIFT : A more distinctive representation for local image descriptors. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2, II–II.

Khan, A., Sohail, A., Zahoora, U. & Qureshi, A. S. (2019). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 1–62.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541–551.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E. & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, pp. 396–404.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Li, X., Yu, L., Fu, C.-W., Fang, M. & Heng, P.-A. (2020). Revisiting metric learning for few-shot image classification. *Neurocomputing*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014). Microsoft coco : Common objects in context. *European conference on computer vision*, pp. 740–755.

Lindeberg, T. (1994). Scale-space theory : A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2), 225–270.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91–110.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14), 281–297.

Perronnin, F., Sánchez, J. & Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. *European conference on computer vision*, pp. 143–156.

Phillips, P. J., Wechsler, H., Huang, J. & Rauss, P. J. (1998). The FERET database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5), 295–306.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81–106.

Radenović, F., Tolias, G. & Chum, O. (2018). Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7), 1655–1668.

Rajalingham, R., Toews, M., Collins, D. L. & Arbel, T. (2010). Exploring cortical folding pattern variability using local image features. *International MICCAI Workshop on Medical Computer Vision*, pp. 43–53.

Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.

Rosenblatt, F. (1962). *Perceptions and the theory of brain mechanisms*. Spartan books.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011). ORB : An efficient alternative to SIFT or SURF. *2011 International conference on computer vision*, pp. 2564–2571.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211–252.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3), 379–423.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection : CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5), 1285–1298.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.

Sivic, J. & Zisserman, A. (2003). Video Google : A text retrieval approach to object matching in videos. *null*, pp. 1470.

Snell, J., Swersky, K. & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, pp. 4077–4087.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. & Hospedales, T. M. (2018). Learning to compare : Relation network for few-shot learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208.

Szegedy, C., Ioffe, S., Vanhoucke, V., on, A. A. T.-f. A. c. & 2017, u. Inception-v4, inception-resnet and the impact of residual connections on learning. *aaai.org*. Repéré à https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14806.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi : 10.1109/CVPR.2016.308.

Toews, M. & Arbel, T. (2008). Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9), 1567–1581.

Toews, M. & Wells, W. (2009). Sift-rank : Ordinal description for invariant feature correspondence. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 172–177.

Toews, M. & Wells III, W. M. (2013). Efficient and robust model-to-image alignment using 3D scale-invariant features. *Medical image analysis*, 17(3), 271–282.

Tolias, G., Sicre, R. & Jégou, H. (2015). Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv :1511.05879*.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E. J., Yacoub, E., Ugurbil, K., Consortium, W.-M. H. C. P. & others. (2013). The WU-Minn human connectome project : an overview. *Neuroimage*, 80, 62–79.

Viola, P. & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137–154.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z. & Liu, W. (2018). Cosface : Large margin cosine loss for deep face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274.

Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500.

Yue-Hei Ng, J., Yang, F. & Davis, L. S. (2015). Exploiting local features from deep networks for image retrieval. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 53–61.

Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European conference on computer vision*, pp. 818–833.

Zhang, S., Yang, M., Wang, X., Lin, Y. & Tian, Q. (2013). Semantic-aware co-indexing for image retrieval. *Proceedings of the IEEE international conference on computer vision*, pp. 1673–1680.

Zheng, L., Wang, S., Tian, L., He, F., Liu, Z. & Tian, Q. (2015). Query-adaptive late fusion for image search and person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1741–1750.

Zoph, B., Brain, G., Vasudevan, V., Shlens, J. & Le Google Brain, Q. V. *Learning Transferable Architectures for Scalable Image Recognition*.