الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 – Guelma -
Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Système Informatique

Thème :

Analyse perceptuelle pour la reconnaissance d'une scène

Encadré Par :
Guerroui Nadia

Présenté par :
Djabali Lounis

Juillet 2019

Sommaire

L'analyse perceptuelle pour la reconnaissance des scènes est une tâche visuelle basée sur la segmentation sémantique, qui est l'un des problèmes clef de la vision par ordinateur. La difficulté provient principalement du grand nombre de flux de données qui permet un apprentissage approfondi et de l'incapacité de la machine à identifier toutes les scènes et objets existants avec un taux de fiabilité similaire à celui des humains.

L'Homme a l'habitude d'observer et de raisonner en même temps, ce qui le rend plus crédible. Ce travail de fin d'étude est un développement d'une architecture CNN pour la reconnaissance de scène basée sur la localisation d'objets à partir d'un apprentissage approfondi, ce qui se traduira par une intelligence artificielle encore plus authentique et un réseau de neurones assurant une fiabilité presque parfaite du résultat, autrement dit l'élargissement de la précision de la segmentation sémantique.
Ce projet utilise un environnement de développement de quatrième génération spécialisé dans les calculs numériques avec ses boîtes à outils.

Mots-clés: Vision par ordinateur, reconnaissance des scènes, intelligence artificielle, réseau de neurones, apprentissage approfondi, segmentation sémantique, architecture CNN

Remerciements

Merci à mon encadreur Guerroui Nadia qui s’est toujours montré a l’écoute et très disponible tout au long de la réalisation de ce mémoire.

Table des matières

Liste des tableaux

Liste des figures

Introduction générale

La compréhension sémantique des scènes visuelles est un élément de base de la vision par ordinateur. Il est possible de construire des représentations abstraites favorisant les objets et leurs formes en classant tous les pixels d'une image. L'émergence d'ensembles de données d'images à grande échelle tels que setNet, googlenet et alexnet, ainsi que le développement rapide d'approches de réseaux de neurones à convolution profonde ont entraîné à de nombreux progrès dans la perception de la scène visuelle.

Cependant, étant donné la scène visuelle, l'image de cet ensemble de données sera segmentées en détail et comparées sur la base définie avec un algorithme qui calcule le score le plus proche, couvrant un ensemble diversifié de catégories, d'objets et de parties d'objets. L'importance du calcul de ces scores est de donner des résultats fiables. Par contre, l'étiquetage est difficile si la base de données n'est pas définie à l'avance.

La reconnaissance et la segmentation d'objets au niveau des pixels restent l'un des problèmes clés dans la perception de la compréhension de la scène. Au-delà de la reconnaissance au niveau d'image, la reconnaissance de scène au niveau pixel nécessite une annotation avec un grand nombre d'objets. Cependant, les bases de données actuelles ont un nombre limité d'objets et dans de nombreux cas, ils n'ont pas tous les objets les plus courants rencontrés dans le monde, ce qui signifie que les bases de données ne couvrent qu'un nombre limité de scènes.

La motivation de ce travail est d'adopter une architecture plus développée et fiable en coordonnant l'apprentissage de la reconnaissance d'objets et l'apprentissage de la reconnaissance de scènes avec un réseau de neurones adéquats.

Ce mémoire est organisé en 3 chapitres comme suit :

- Une introduction où on situe notre projet de fin étude et son plan.

- Le premier chapitre est consacré à l'état de l'art et introduit la partie descriptive des définitions relatives à l'analyse perceptuelle pour la reconnaissance de scènes avec des concepts importants à utiliser dans ce thème. Ce chapitre sera complété par les différents types d'architecture existants jusqu'à aujourd'hui.

- Dans le deuxième chapitre, nous présentons la conception de l'architecture proposée pour l'analyse perceptuelle d'une reconnaissance de scène fiable, son objectif, les différentes étapes de la conception, ainsi que les différentes méthodes appliquées avec les améliorations apportées.

- Le troisième chapitre est dédié à l'implémentation et la discussion des résultats expérimentaux de notre système.

- Enfin nous terminerons avec une conclusion générale qui englobe ce sujet de fin d'étude ainsi que les perspectives futures.

Chapitre 1

Généralité sur l'analyse perceptuelle d'une scène

1.1 Introduction

L'analyse de données visuelles pour l'être humain est une question simple. Contrairement à une machine supposée intelligente, cette tâche simple est très compliquée pour un ordinateur, car elle appelle trois grands problèmes de l'intelligence artificielle.

1. **Perception :** permet à l'être humain de détecter et de prendre connaissance de la réalité.
2. **Connaissance**: le fait de comprendre, de connaître les propriétés, les caractéristiques, les traits spécifiques de quelque chose, savoir.
3. **Raisonnement:** qui crée avec le regroupement de la perception et de la connaissance.

Ce chapitre présente les principes théoriques de l'analyse perceptuelle des scènes de reconnaissance sur lesquels repose notre projet de fin d'étude.

1.2 Généralités

Notre cerveau a un grand nombre de dimensions, dépassant de loin ce que nous essayons d'imiter avec les machines d'intelligence artificielle aujourd'hui. Le potentiel humain nous permet de résoudre des problèmes très complexes et pluridisciplinaires, nous donne le pouvoir d'imaginer des choses et des situations qui n'ont jamais existé, une créativité générant des émotions très puissantes et une volonté d'atteindre ce qui semble impossible, et qui nous donne un incroyable sens de la conscience et la conscience de soi [1].

1.2.1 Vision par ordinateur

La vision par ordinateur est un ensemble de techniques conçues pour l'interprétation automatique d'images. Au début de l'année 1964, les experts en intelligence artificielle se sont concentrés sur la possibilité de simuler les capacités sensorielles humaines. Comme la vision est le sens humain le plus important, il est naturel que les travaux dans cette direction aient été nombreux. La démocratisation des appareils photo numériques et des appareils photo numériques a encore accéléré les possibilités offertes par le traitement des images, non seulement grâce aux processus de correction intégrés dans les capteurs eux-mêmes, mais également grâce à la nouvelle facilité de post-traitement informatique. Le Graal de la vision par ordinateur est l'émulation de la capacité du cerveau humain à interpréter une scène dynamique à partir de signaux visuels, en particulier pour identifier rapidement des objets et anticiper leurs mouvements. Ces fonctions cognitives sont essentielles pour la navigation en robotique autonome, mais profitent également aux applications d'exploration de données. La numérisation automatique de vieux documents, la recherche d'images similaires en ligne ou la description audio automatique sont des exemples de tâches pouvant s'appuyer sur une brique d'interprétation des images. En particulier, les efforts de la communauté de la vision par ordinateur se sont concentrés sur la reconnaissance des objets dans les images, y compris leur identification et leur localisation. De nombreux descripteurs ad hoc ont été introduits pour des applications aussi diverses que la détection de visage, la classification automatique de photographies d'animaux ou la reconnaissance optique de caractères [2].

Le dénominateur commun de ces œuvres est d'essayer de donner un sens aux images. L'extraction de la sémantique à partir d'informations enregistrer sur une base de données constitue également l'objectif de la reconnaissance d'une scène. Ce projet se situe à l'intersection de la vision par ordinateur et de l'apprentissage automatique. En particulier, nous proposons de mettre en œuvre des méthodes d'apprentissage en profondeur pour l'interprétation automatique des images de reconnaissance d'une scène [2].

Dans la ***figure 1*** nous montrons les domaines les plus connus de la vision par ordinateur ainsi que la différence entre eux et leurs multiples niveaux

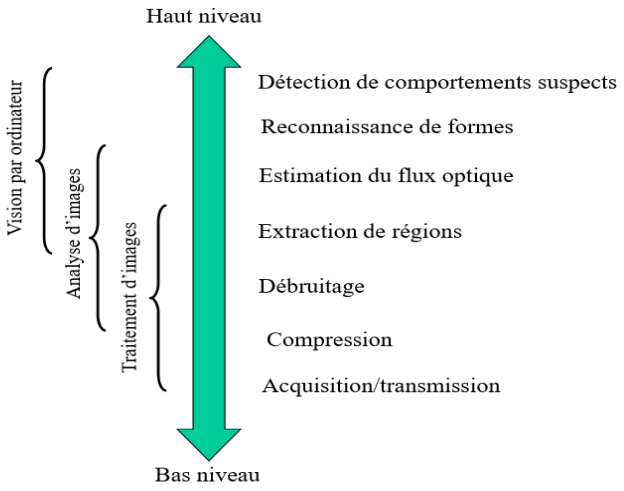


Figure 1 Les domaines de la vision par ordinateur [22].

Dans la ***figure 2*** nous donnons un exemple d'une approche de vision par ordinateur dans le domaine médicale suivie d'un schéma qui montre les différentes étapes suivies pour réaliser cette approche

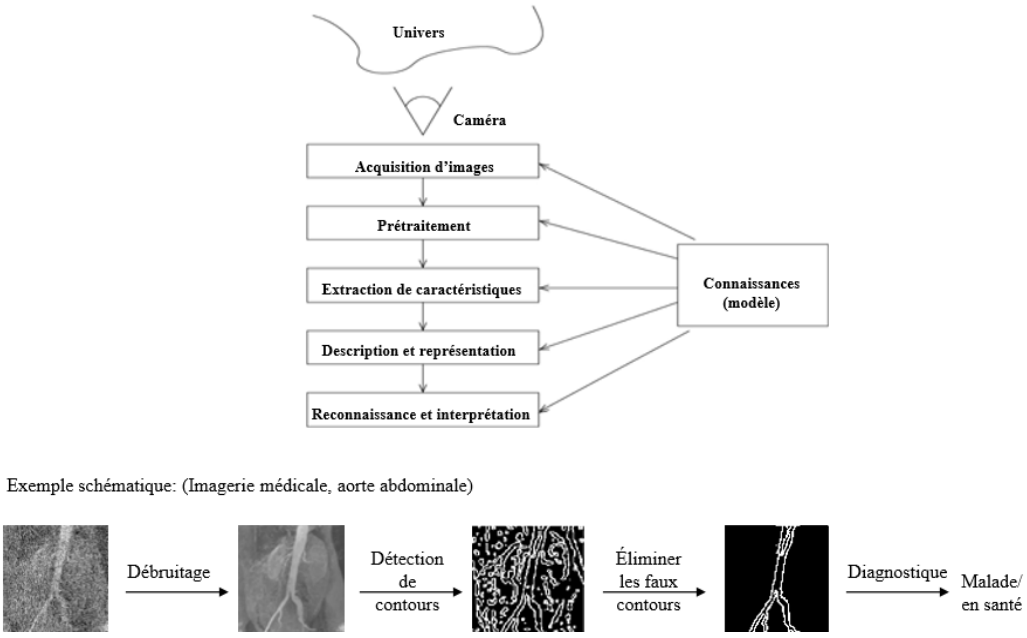


Figure 2 : schéma représentant l'approche de vision par ordinateur [22].

1.2.2 Traitement d’image

Le traitement des images est l'ensemble des méthodes et techniques qui les exploitent, afin de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'extraire les informations jugées pertinentes [3].

D’une manière inverse, le traitement des images et la vision exploitent les connaissances et les techniques de l'intelligence artificielle pour l'adaptation et l’amélioration de la reconnaissance du monde extérieur par les machines intelligentes [4].

1.2.3 Intelligence Artificielle

L'intelligence artificielle (IA) pourrait être considérée comme un ensemble de techniques permettant aux machines d'exécuter des tâches et de résoudre des problèmes normalement réservés à l'homme et à certains animaux [5].

Les tâches d’intelligence artificielle sont parfois très simples pour l’homme, telles que reconnaître et localiser des objets dans une image, planifier les mouvements d’un robot pour attraper un objet ou conduire une voiture. Ils nécessitent parfois une planification complexe, telle que jouer aux échecs. Les tâches les plus complexes nécessitent beaucoup de connaissances et de bon sens, par exemple pour traduire un texte ou mener un dialogue.

Ces dernières années, l'intelligence a presque toujours été associée à des capacités d'apprentissage. C'est grâce à l'apprentissage qu'un système intelligent capable d'effectuer une tâche peut améliorer ses performances avec l'expérience. C'est par l'apprentissage qu'il peut apprendre à effectuer de nouvelles tâches et à acquérir de nouvelles compétences [7].

Le domaine de l'IA n'a pas toujours considéré l'apprentissage comme essentiel à l'intelligence. Construire un système intelligent consistait autrefois à écrire un programme "à la main" pour jouer aux échecs (par la recherche d’arbres), pour reconnaître les caractères imprimés (par rapport aux images prototypes) ou pour poser un diagnostic médical à partir de symptômes (par déduction logique de règles écrites par des experts). Mais cette approche "manuelle" a ses limites [6].

1.3 Segmentation sémantique

La segmentation sémantique, également appelée classification au niveau des pixels, consiste à regrouper chaque parties de l'image qui appartiennent à la même classe d'objets .La classification[1] et la détection au niveau de l'image sont deux autres tâches principales. La segmentation d'image peut être traitée comme une prédiction au niveau du pixel car elle classe chaque pixel dans sa catégorie. En outre, il existe plusieurs types de segmentation. Ce pendant multiples applications, telles que la détection de panneaux de signalisation, la classification de la couverture terrestre la détection[2] des tumeurs dans le cerveau et la reconnaissance des scènes se basent sur cette méthode.

Les architectures de réseaux de neurones entièrement consolatives (CNN) ont permis d'étendre le modèle de réseau convolutif à la segmentation sémantique, en transformant la classification par image en classification dense (par pixel). De nombreux modèles ont par la suite été proposés afin de tirer parti d'un contexte multi échelles, ou d'architectures symétriques encodeur décodeur.

Toutefois, les CNN tendent à produire des segmentations invariantes par translation locale, produisant un effet de flou au niveau des frontières. En outre, les segmentations peuvent être sujettes à des erreurs grossières comme des topologies d'objet non respectées (*Connexité, convexité, a priori polygonal*….). La communauté s'est donc penchée sur des régularisations permettant de diminuer ces erreurs. Ainsi, des modèles graphiques comme les champs aléatoires conditionnels ont été utilisés pour régulariser les frontières des segmentations, d'abord comme prétraitement séparé puis intégré au réseau de façon différentiable. Dans le même esprit, reformule les méthodes variationnelles d'ensemble de niveau de façon à pouvoir les résoudre avec un CNN.

[1] La classification signifie traiter chaque image comme une catégorie identique.
[2] La détection fait référence à la localisation.

1.3.1 Réseau de neurone

Le réseau de neurones artificiels (ANN) est inspiré des neurones biologiques. L'élément de base de ANN est un neurone artificiel. Chaque neurone artificiel possède des entrées qui sont pondérées et résumé. Suivi d'une fonction de transfert ou d'une fonction d'activation, le neurone produit une valeur d'échelle. Un exemple de modèle neural est illustré à la ***Figure 3***.

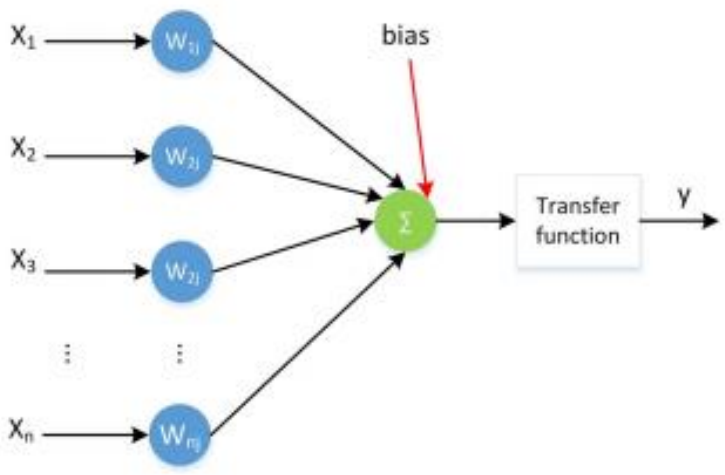


Figure 3: Modèle d'un neurone artificiel

Basé sur un neurone artificiel, différents empilements de neurones forment un auto-encodeur[3] [8]. Il existe plusieurs types de réseaux de neurones ou réseau de neurones récursifs (RNN), réseau de neurones convolutionnels (CNN). L'architecture de base est illustrée à la ***Figure 4***.

Auto-encodeur : est un réseau de neurones artificiels utilisé pour l'apprentissage automatique d'analyse et de reconnaissance dans une image

[3] Auto-encodeur : est un réseau de neurones artificiels utilisé pour l'apprentissage automatique d'analyse et de reconnaissance dans une image

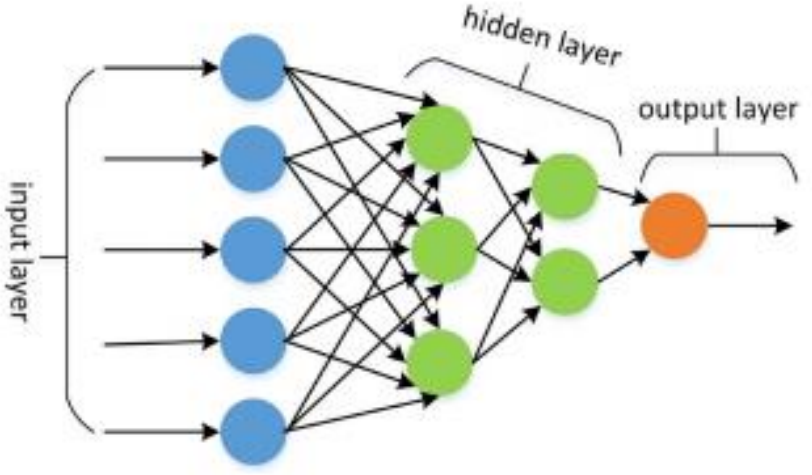


Figure 4 : Modèle d'un réseau neuronal artificiel

Le réseau de neurones convolutifs (CNN) utilisé dans plusieurs architectures de segmentation et traitement d'image, s'inspire des processus biologiques de l'animal .Grâce à l'excellente structure, CNN a obtenu des résultats remarquables en matière de classification, de segmentation et de détection d'images [9]

1.3.2 Travaux Connexes

De nombreux ensembles de données ont été définis dans le but de comprendre de manière imagée les scénarios. Ils ont été examinés selon le niveau de détail de leurs annotations, puis ont brièvement parcouru les travaux antérieurs sur les réseaux de segmentation sémantique *(La classification des objets et des ensembles de données de détection).*

La plupart des fichiers à grande échelle sont généralement définis comme contenant des étiquettes de niveau ou fournissent des cadres de sélection. Les exemples incluent **ImageNet** [10], **Pascal** [11] et **KITTI** [12]. **ImageNet** possède le plus grand ensemble de classes, mais contient des scènes relativement simples. **Pascal** et **KITTI** sont plus difficiles et ont plus d'objets, mais leurs classes sont encore plus limitées.

Ensembles de données de segmentation sémantique Les ensembles de données existants comportant des étiquettes au niveau des pixels fournissent généralement des

annotations uniquement à un sous-ensemble d'objectifs d'avant-plan. Il est beaucoup plus difficile de collecter des annotations denses où tous les pixels sont étiquetés. Pascal fait partie de ces efforts connexes [14] NYUDepthV2 [15], Base de données SUN [16], jeu de données SUN RGB-D [17], jeu de données CityScapes [18] et OpenSurfaces [19] ADE20K Dataset [13].

1.3.3 Deep Learning pour la segmentation sémantique

La segmentation d'images est l'une des premières tâches envisagées dans le cadre de la vision artificielle. La légende raconte qu'en 1964, Minsky avait demandé à son élève, Gerald Sussman, de "passer l'été à connecter une caméra à un ordinateur et à le faire décrire par lui-même". La tâche de Minsky et Papert dans le cadre du projet Summer Vision consistait notamment à "construire un système de programmes divisant une image d'un tube dissecteur en plusieurs régions, telles que: plutôt des objets, plutôt que l'arrière-plan ou le chaos". Avec l'objectif final d'un logiciel "L'identification d'objets, qui nomment chaque objet en les faisant correspondre à un vocabulaire d'objets connus." Dès le début, la reconnaissance de forme concerne donc la division sémantique des images pour comprendre les scènes visuelles qu'elles représentent. Si cette tâche semble triviale pour un humain, cela représente un défi considérable pour la machine. L'équipe de Minsky se heurte rapidement au paradoxe de Moravec "il est relativement facile de placer les ordinateurs au niveau d'un humain adulte dans le cadre d'un test d'intelligence ou d'un jeu de dames, mais difficile sinon impossible de leur donner la perception et la mobilité d'un bébé ".

Depuis, de nombreuses études se sont penchées sur le problème de la reconnaissance d'objet, à savoir l'identification d'un objet présent dans une image. On parlera ici de classification d'images, tâche consistant à associer une image à un type d'objet. Cette tâche a concentré la majorité des efforts de la communauté, de l'utilisation de descripteurs aux caractéristiques apprises à travers les modèles probabilistes. L'arrivée récente de grandes bases de données d'images annotées, telles que **CIFAR-10** [4]et **CIFAR-100**[5], et ImageNet

[4] The CIFAR-10 dataset (Canadian Institute For Advanced Research) est une collection d'images couramment utilisées pour former des algorithmes d'apprentissage automatique et de vision par ordinateur. https://www.cs.toronto.edu/~kriz/cifar.html.

ont permis le succès des réseaux de classification par convolution profonde basés sur les données MNIST, en reconnaissance des signes, en identifiant les caractères chinois et en cours reconnaissant des objets de toutes sortes sur ImageNet **[10]**.

1.3.3.1 Architecture

Une architecture se forme a base de combinaison de CNN le choix d'une architecture est très complexe et c'est plus technique que une science exacte. Il est donc important d'étudier les architectures qui ont s'est avéré efficace et s'inspiré de ces exemples célèbres.

Dans CNN le plus classique, nous enchaînons plusieurs fois une couche de convolution suivie par une couche de regroupement et nous ajoutons à la fin des couches entièrement connectées.

LeNet réseau, proposé par l'inventeur du CNN, Yann LeCun comme la montre la ***Figure 5***. Ce réseau était consacré à la reconnaissance de chiffres et des photos noir et blanc. Il est composé uniquement sur quelques couches et peu de filtres, en raison des limitations de l'ordinateur à ce moment-là.

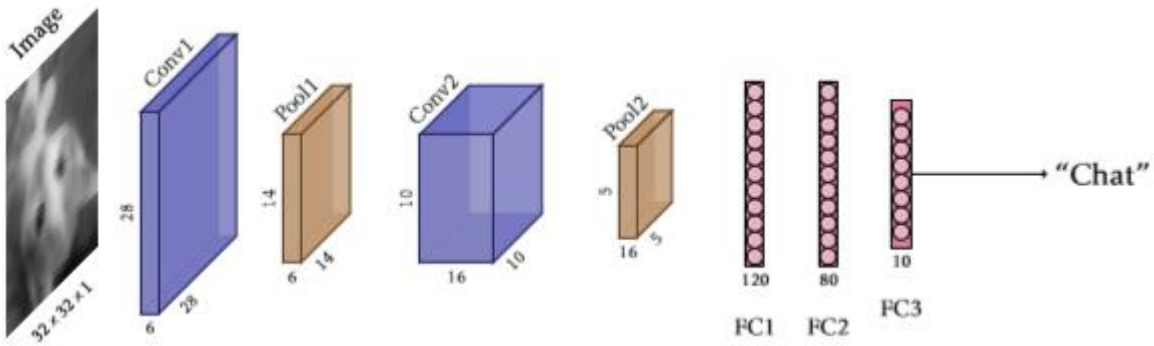


Figure 5 : Architecture LeNet .[2]

Quelques années plus tard, avec l'apparition du GPU (Graphical Processor Unit) , des architectures beaucoup plus complexes pour CNN ont été proposées, comme le réseau

---

[5] THE CIFAR-100 cet ensemble de données est similaire à celui de CIFAR-10, à la différence qu'il contient 100 classes contenant chacune 600 images. Il y a 500 images de formation et 100 images de test par classe. https://www.cs.toronto.edu/~kriz/cifar.html.

AlexNet (conçu par Alex Krizhevsky et publié avec Ilya Sutskever et le directeur de thèse de Krizhevsky, Geoffrey Hinton, qui était à l'origine opposé à l'idée de son élève) ayant remporté le concours ImageNet et pour lequel une version simplifiée est présentée à la **Figure 6**. Ce concours a été consacré à la classification d'un million d'images en couleurs sur 1000 classes. La résolution des images était de 224 × 224.

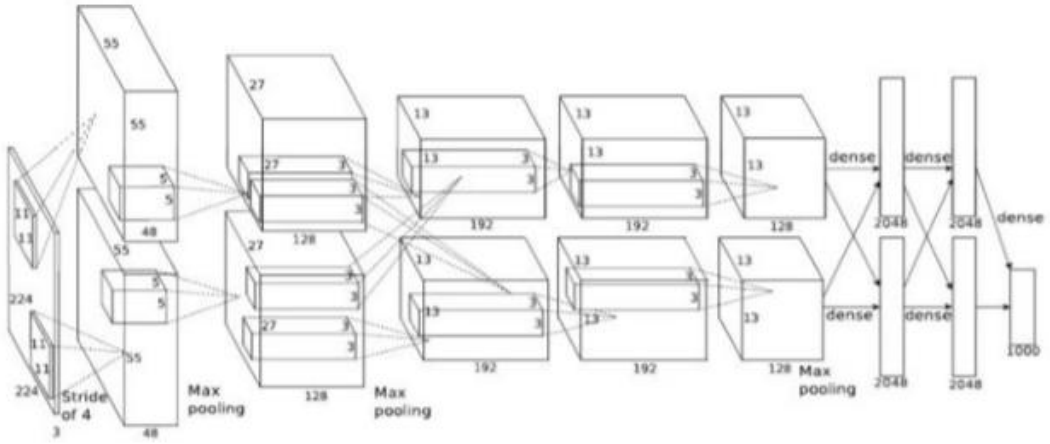


Figure 6 : Architecture AlexNet1 [20]

AlexNet est composé de 5 unités de convolution couches, 3 couches max-pooling qui sont entièrement connectées, la forme du noyau de la première couche de convolution est (11, 11, 3, 96) avec une foulée de s = 4 et la première forme de sortie est (55, 55, 96).

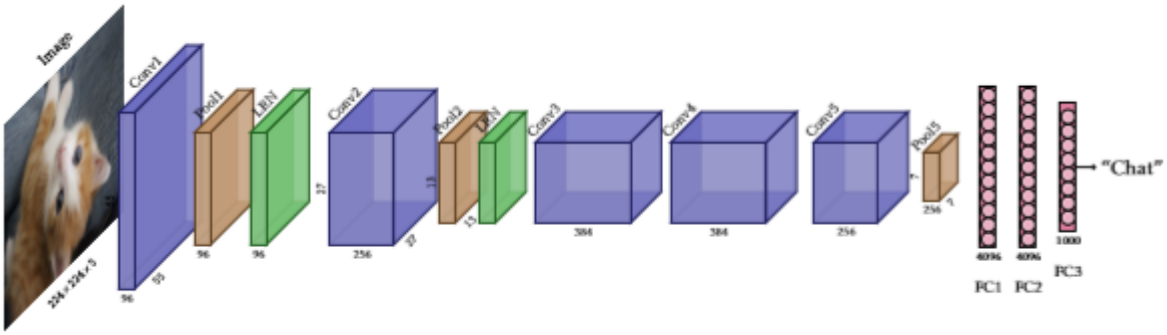


Figure 7 : Architecture AlexNet2 [2]

En 2013, **ZFNet** a été inventé par le Dr. Rob Fergus et son doctorant à l'époque, le Dr. Matthew D. Zeiler à la NYU Sans surprise,c'est le gagnant du ILSVRC 2013(Large Scale Visual Recognition Challenge). Il a atteint un taux d'erreur parmi les 5 premiers de 14,8%, ce qui correspond déjà à la moitié du taux d'erreur non neural mentionné en priorité. Il s'agissait principalement de peaufiner les hyper-paramètres d'AlexNet tout en maintenant la structure avec des éléments d'apprentissage approfondis supplémentaires.

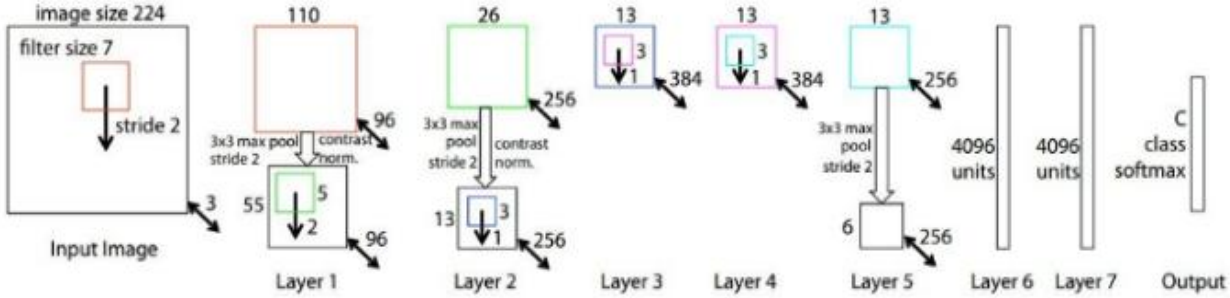


Figure 8 : Architecture ZFNet [23]

Le gagnant du concours ILSVRC 2014 était **GoogleNet** (a.k.a. Inception V1) de Google. Il a atteint un taux d'erreur de 6,67% dans le top 5! C'était très proche de la performance au niveau humain. Les organisateurs du défi étaient maintenant obligés d'évaluer.cela était en fait assez difficile à faire et nécessitait un entraînement humain afin de battre la précision de GoogLeNets. Après quelques jours de formation, l'expert humain (Andrej Karpathy) a réussi à atteindre un taux d'erreur parmi les 5 premiers de 5,1% (modèle unique) et de 3,6% (ensemble).

Le réseau a utilisé un CNN inspiré de LeNet mais a mis en œuvre un nouvel élément appelé module de création. Il a utilisé la normalisation par lots, les distorsions d'image et RMSprop(est un algorithme d'optimisation non publié conçu pour les réseaux de neurones). Ce module est basé sur plusieurs et très petites convolutions afin de réduire considérablement le nombre de paramètres. Leur architecture consistait en un réseau CNN profond de 22 couches mais réduisait le nombre de paramètres de 60 millions (AlexNet) à 4 millions.

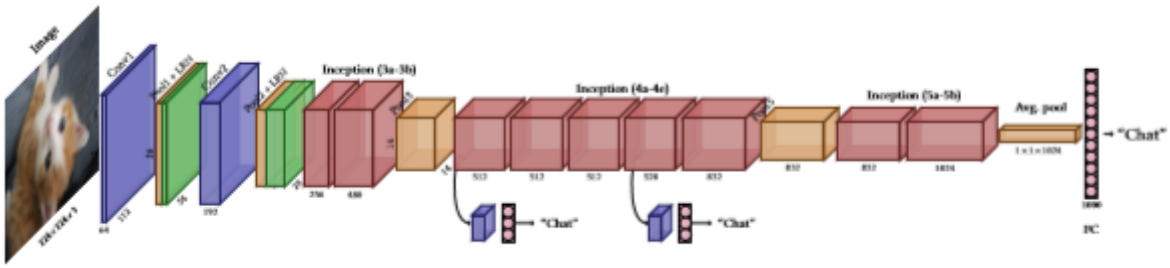


Figure 9 : Architecture googleNet [2]

Le finaliste du concours ILSVRC 2014 est **VGGNet** a été développé par Simonyan et Zisserman. VGGNet se compose de 16 couches convolutives et est très attrayant en raison de son architecture très uniforme. Similaire à AlexNet, seulement 3x3 convolutions, mais beaucoup de filtres. Formé sur 4 GPU pendant 2 à 3 semaines. Il s'agit actuellement du choix le plus populaire dans la communauté pour l'extraction de fonctionnalités à partir d'images. La configuration de poids du VGGNet est disponible publiquement et a été utilisée dans de nombreuses autres applications et défis en tant qu'extracteur de fonctionnalités de base. Cependant, VGGNet comprend 138 millions de paramètres, ce qui peut être un peu difficile à gérer.

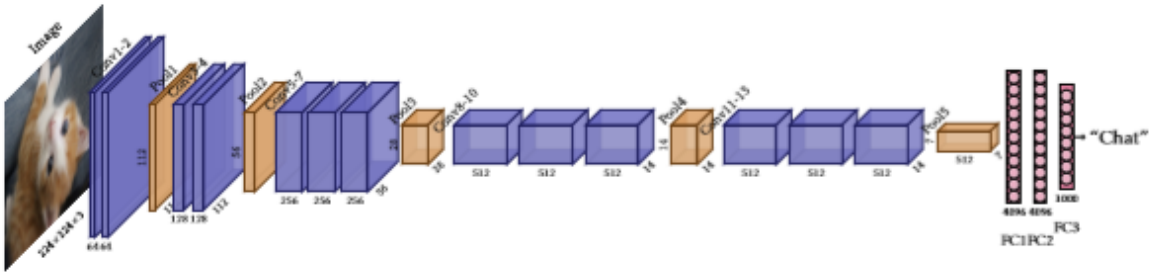


Figure 10 : Architecture VGGNet(16) [2]

Enfin, à l'ILSVRC 2015, le réseau dit de neurones résiduels **ResNet** de Kaiming He et al a présenté la nouvelle architecture avec des «connexions de saut» et présente une normalisation par lots très lourde. Ces connexions de saut sont également appelées gated

units ou gated recurrent units et présentent une forte similitude avec les éléments réussis récents appliqués dans les RNN. Grâce à cette technique, ils ont pu former un NN avec 152 couches tout en ayant une complexité inférieure (recurrent neural network) à celle de VGGNet. Il atteint un taux d'erreur de 3,57% dans le top 5, ce qui est supérieur à la performance humaine de cet ensemble de données.

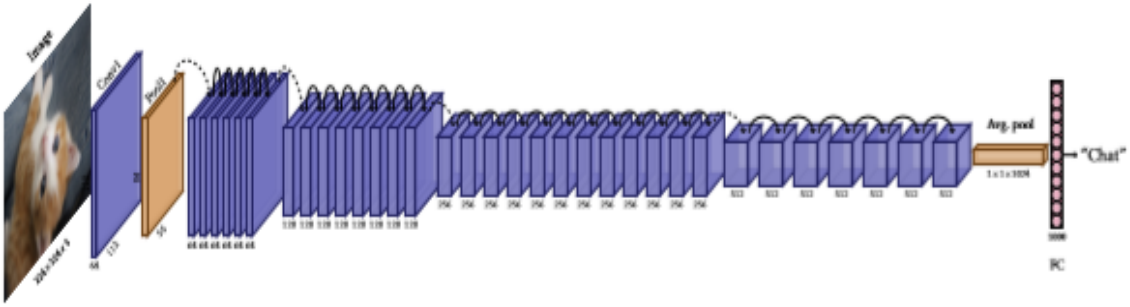


Figure 11 : Architecture Resnet [2]

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

Tableau 1 : Présente une étude comparative entre les différentes architectures [24]

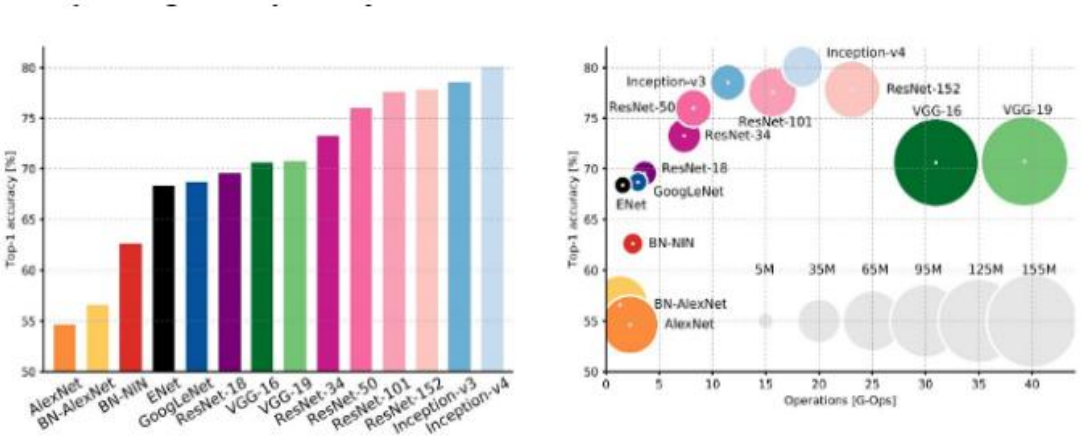


Figure 12 : Taux de précision pour certaines architectures [21]

1.4 Conclusion

La segmentation sémantique est une étape cruciale pour de nombreuses méthodes d'analyse et la compréhension des scènes est un concept très important. Malgré les excellents résultats qui peuvent être obtenus, les architectures nécessitent souvent une modification explicite de leurs paramètres afin de pouvoir obtenir une segmentation visuellement fiable et de réduire le taux d'erreur. Ces étapes seront expliquées dans le chapitre suivant.

Chapitre 2

Conception

2.1 Introduction

Pour la réalisation de ce projet d'analyse de reconnaissance de scène perceptuelle, l'utilisation de couches de convolution dans un modèle d'apprentissage en profondeur est encore plus fiable. Il s'agit d'une tâche extrêmement simple, souvent réalisable dans une seule ligne de code.

De plus, comprendre les convolutions pour la première fois peut parfois sembler gênant, avec des termes tels que noyaux, filtres, canaux, conv, déconv ... etc, superposés. Cependant, les convolutions en tant que concept sont fascinantes, puissantes et très extensibles. Dans cette partie, nous décrivons brièvement les mécanismes de convolution et les différentes méthodes utilisées pour obtenir un résultat satisfaisant.

2.2 Convolution

La convolution est une opération basée sur un noyau, qui est simplement une petite matrice de poids. Ce noyau glisse sur les données d'entrée en effectuant une multiplication élément par élément avec la partie d'entrée sur laquelle elles se trouvent, puis en calculant la somme et en plaçant le résultat dans un seul pixel de sortie. Le noyau répète ce processus pour chaque emplacement indiqué.

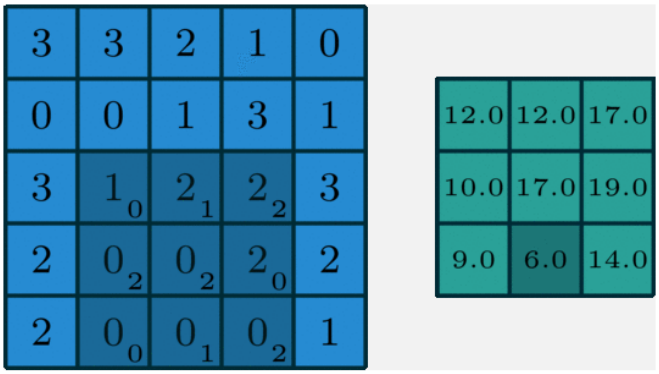
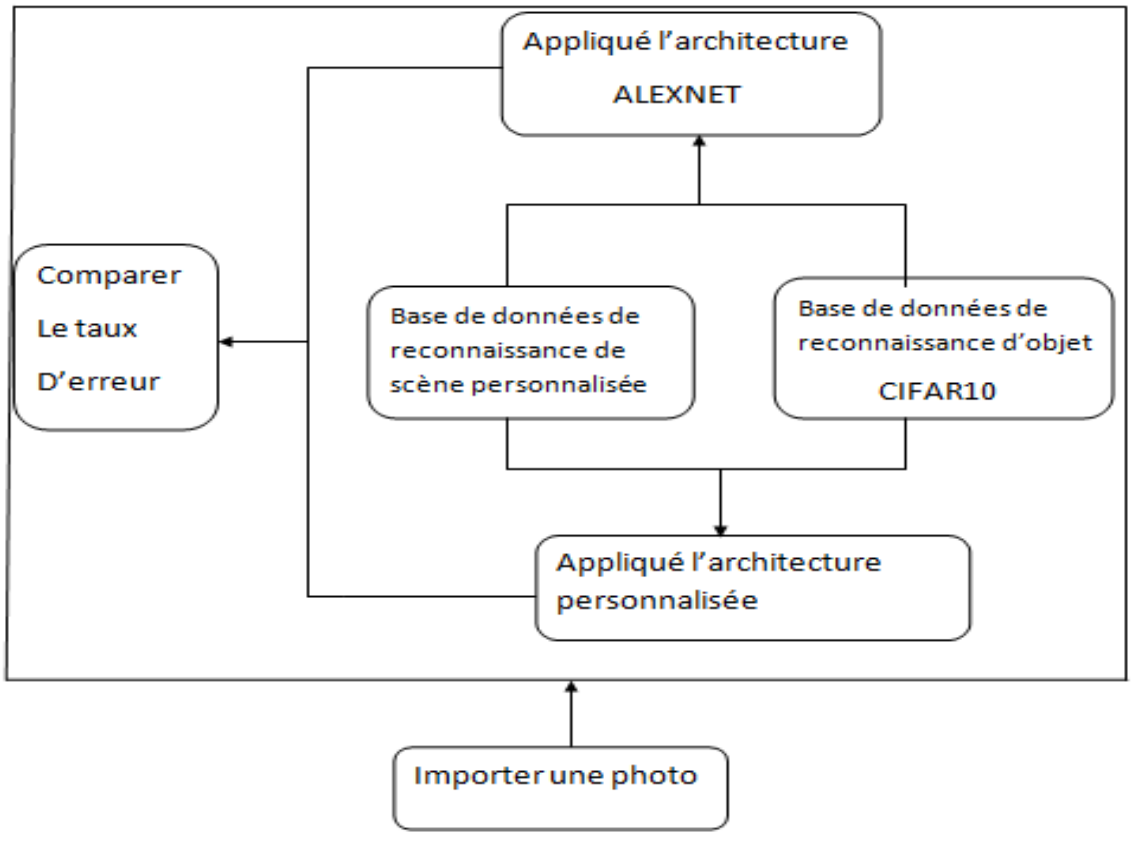


Figure 13 Calcul de convolution

2.3 Conception architectural

L'architecture réalisée a suivi les étapes illustrées dans le schéma ***(Figure 14)*** suivant :



:

Figure 14 Schéma de l'architecture proposée

-Importer une photo

-Base de données **CIFAR-10** pour la reconnaissance des objets télécharger et modifier.

-Base de données pour la reconnaissance scènes télécharger et modifier.

-Application de l'architecture AlexNet.

-Application de notre architecture personnalisée.

-Pour vérifier la fiabilité du résultat nous comparons le taux d'erreur pour chaque architecture.

Pourquoi AlexNet ?

Cette architecture a remporté le concours ImageNet en 2012, avec un taux d'erreur de 15,4%.

En outre, c'est un moyen de visualiser les caractéristiques de convolution, les réseaux de neurones de convolution étaient essentiellement des boîtes noires dont le fonctionnement interne était inconnu.

2.4 Déconvultion

Dans l'apprentissage en profondeur, les déconvultions sont également appelées convolution transposée et convolution de hauteur partielle. La convolution vous permet de passer d'une dimension d'entrée spécifiée à une dimension de sortie. Notez que la forme de la dimension de sortie est définie par les paramètres d'une convolution donnée: taille d'entrée, taille du noyau, étape et remplissage. La déconvultion (convolution transposée) permet d'aller dans le sens opposé: à partir de la dimension de sortie, retournez à la dimension d'entrée. Ceci est appliqué dans le but de deux raisons:

- ✓ La déconvultion est utilisée lors de la propagation des erreurs à travers la couche convultive.
- ✓ La déconvultion est utilisé pour augmenter les contributions à des applications spécifiques d'apprentissage en profondeur telles que les réseaux de ressources humaines.

La rétro-propagation pour la couche convultive est une opération de déconcultation appliquée au gradient entrant de la couche convultive. Si nous considérons l'opération de convultion comme la multiplication de matrice d'entrée par une matrice de convultion C (définie par les poids du noyau de convolution), la propagation de l'erreur revient à multiplier le gradient entrant par la transposition de la matrice de convolution C. C'est pourquoi la déconvultion est aussi appelée convolution transposée. Sachez que la multiplication par la matrice de convolution transposée, CT, vaut une déconvultion avec des filtres pivotés à 180 degrés.

En ce qui concerne l'utilisation de déconvultions dans la conversion ascendante, il serait important de prendre de nouveaux filtres afin d'éviter de réutiliser les filtres de conversion des couches précédentes.

2.4.1 Réseau de déconvolution

Les visualisations produites sont des modèles reconstruits à partir d'une image d'entrée issue de l'ensemble de validation, qui entraînent des activations élevées dans une carte de caractéristiques donnée.

Pour générer des images d'entrée générant un maximum d'activations, un Deconvnet distinct est associé à la sortie de chaque couche dont nous voulons visualiser les caractéristiques. Chaque Deconvnet est lui-même un réseau de neurones dont les entrées sont les activations d'une couche particulière et la sortie est une image décrivant les pixels responsables de l'activation maximale d'un noyau de convolution choisie pour une couche de convolution donnée. Deconvnet est décrit comme une séquence de convolutions transposées, de décomposition et de ReLU modifié spécialement. La séquence d'opérations est l'inverse des étapes qui ont été utilisées dans le réseau neuronal d'origine pour produire la sortie d'une couche particulière. Par exemple, visualiser conv 3 couche d'un réseau de neurones. Le calcul ci-dessous est effectué pour obtenir la sortie de la couche conv3:

Entre I -> convA -> relu -> convB -> relu -> pool -> convC

Deconvnet attaché à la sortie effectuera des opérations en sens inverse:

de-convC -> de-pool -> relu* -> de-convB -> relu* -> de-convA -> Sortie I

La sortie du Deconvnet a les mêmes dimensions que l'image d'entrée dans le convnet visualiser

Notez que cela de-conv signifie déconvultion , cette opération est aussi parfois appelée convultion transposée . Il utilise des noyaux convultifs transposés du réseau original visualiser.

L'opération de décomposition (de-pool) étend la dimensionnalité des données en mémorisant les positions regroupées dans le passage en aval du réseau visualiser.

Modifié ReLU (relu*) ne transmet que l'activation positive, ce qui revient à rétrodécoller uniquement les gradients positifs, ce qui diffère de la rétrogradation par le biais d'une ReLU normale.

Ces étapes sont nécessaires pour la fiabilité du resultat il existe aussi d'autre façon de voir le Deconvnet: il ne s'agit que d'une étape de rétropropagation modifiée du convnet d'origine visualiser l'exemple ci-dessus vue précédemment :

Entrer I -> convA -> relu -> convB -> relu -> pool -> convC

La passe en arrière pour ce calcul ressemble à ceci:

backprop convC -> backprop pool -> backprop convB -> backprop relu -> backprop convA -> gradients de l'entrer de l'image

Maintenant, le backprop à travers une couche convultionnelle est juste une déconvultion avec le noyau pivoté de 180 degrés (convultion transposée), ilest backprop conv = de-conv. Backprop à travers la mise en commun est la même que de-pool celle définie dans le précédemment. Enfin, au lieu d'utiliser la rétro propagation standard à travers ReLU, les Utilisation de ReLU modifié qui ne transmet que les activations positives est nécessaire. Ainsi, l'étape backprop ressemble maintenant à ceci:

de-conv3 -> de-pool -> relu* -> de-conv2 -> relu* -> de-conv1 -> output image

Pour conclure, Deconvnet est une passe légèrement modifiée en arrière dans un convnet visualiser et dont le résultat est une image produisant l'activation maximale d'un noyau de convolution sélectionné pour une couche de convolution sélectionnée. Cette image permet de comprendre le type de fonctionnalité que ce noyau a appris à reconnaître pour assurer un meilleur résultat.

2.5 Fonction de visualisation

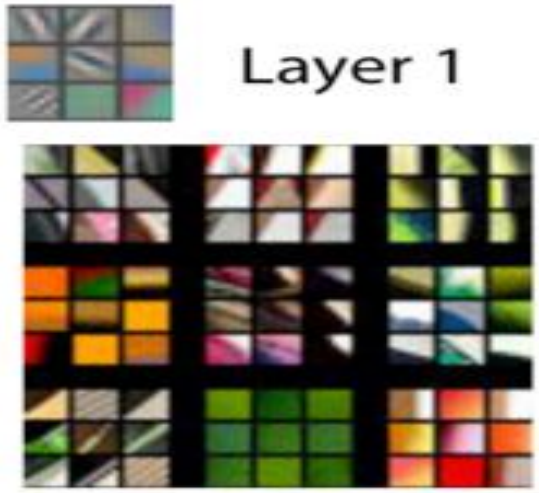


Figure 15 Layer 1.

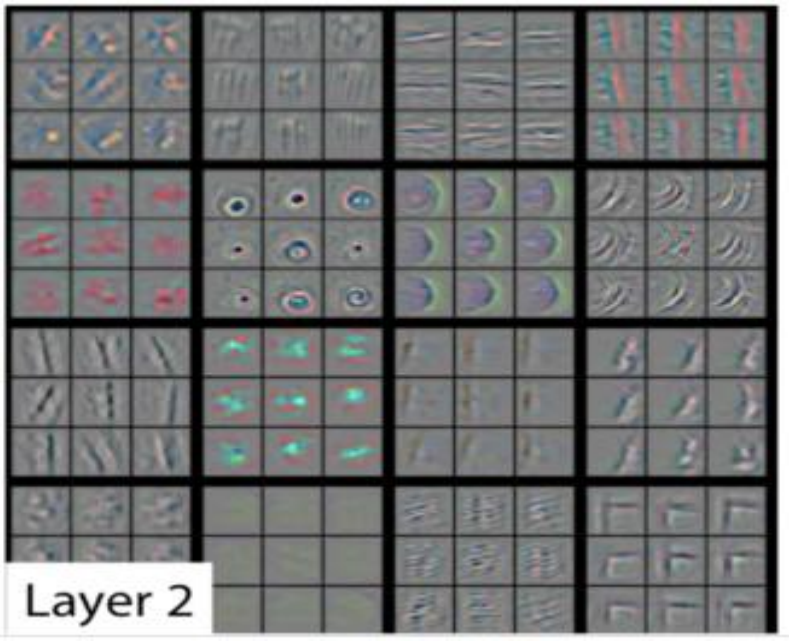


Figure 16 Layer 2

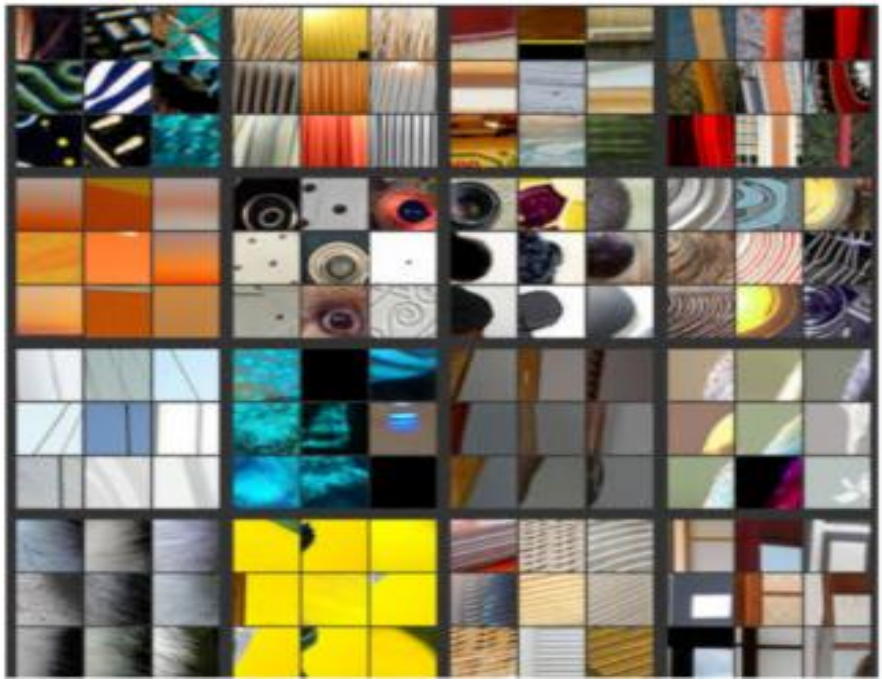


Figure 17 Galerie d’image 1.

Exemples de fonctions visualisées pour les couches 1 et 2. Notez la taille des projections d’activation et des plages d’image sont minuscules pour la couche 1 et plus grandes pour la couche 2. Cela concerne différents champs récepteurs de neurones dans différentes couches.

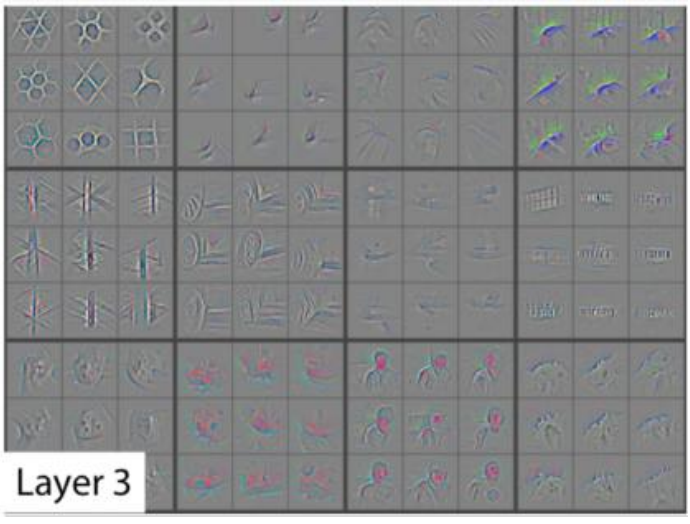


Figure 18 Layer 3.

Figure 19 Galerie d’image 2.

Exemples de fonctionnalités visualisées pour layer3. Notez que la taille des projections d'activation et des zones d'image est quelque peu petite, par contre elle est plus grande que dans les couches 1 et 2 et plus complexe.

Les visualisations présentées offrent un aperçu de la représentation interne des connaissances acquises par le réseau. Il est essentiel de comprendre ce qu'ils disent avec précision. Les images dans la partie supérieure de la feuille sont des projections du neurone activant le plus élevé d’une carte de caractéristiques particulière vers l’espace pixel de l’entrée en utilisant la rétro propagation modifiée. Les images dans la partie inférieure de la feuille font partie des images d'entrée qui produisent une projection donnée en haut. En d'autres termes, la visualisation des activations est réalisée pour chaque image séparément.
Une autre chose essentielle à noter est que la taille de chacune de ces projections est égale au champ de réception de chaque neurone sur l’image d’entrée. Notant que les neurones superposés proches de l'image ont un champ de réception plus petit sur l'image d'entrée et que plus vous vous éloignez de l'image, plus le champ de réception devient grand.

Chaque noyau de convolution glisse sur l’espace d’entrée et produit une activation pour chaque position. En conséquence, une carte de caractéristiques 2D pour ce noyau a été

créée, composée de nombres représentant les activations de neurones. Vérifiez et recherchez l'activation la plus élevée, retracez les champs du récepteur sur le réseau et déterminez quelle partie de l'image d'entrée recherche ce neurone hautement activé. Ces patches sont sur le dessus de chaque galerie.

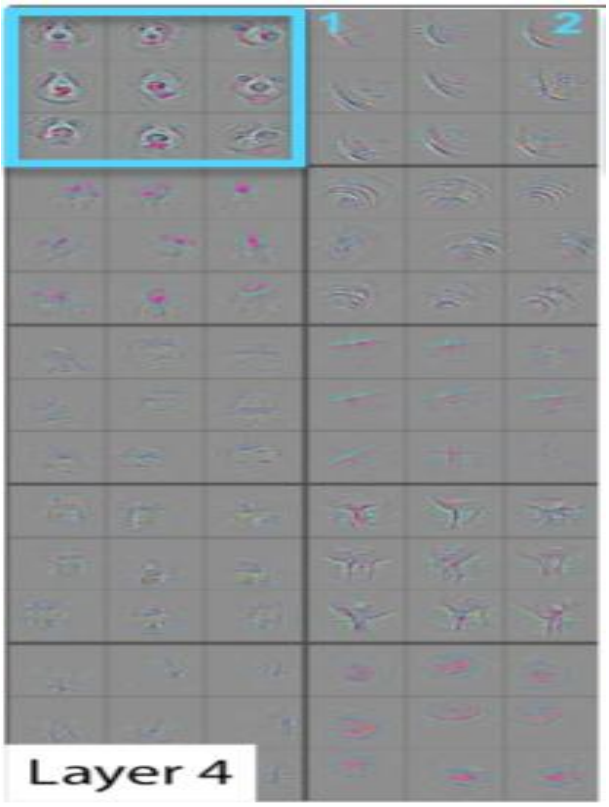


Figure 20 Layer 4.

Figure 21 Galerie d’image 4

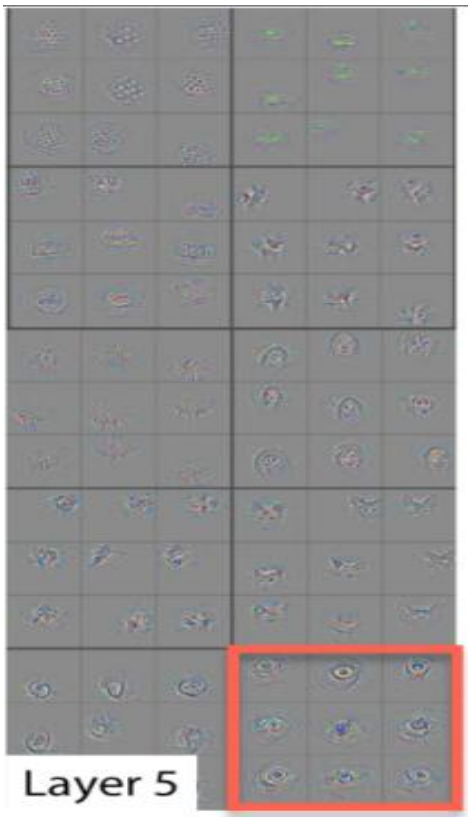


Figure 22 Layer 5.

Figure 23 Galerie d’image 5

Exemples d’entités visualisées pour les couches 4 et 5. la taille des projections d’activation et des corrections d’image est supérieure à celle de la couche 3. La complexité des entités détectées est également supérieure à celle des couches précédentes.

Il convient de mentionner que plus on monte dans le réseau (plus loin de l’entrée), plus les représentations deviennent complexes. Ils augmentent la complexité en combinant des fonctionnalités plus simples à partir des couches inférieures. Par exemple, si la couche 2 détecte des cercles, la couche 5 utilise ces cercles détectés pour détecter les visages de chien, car les yeux du chien ressemblent à des cercles.

2.6 Phase finale

Le principe est de modifier dans les paramètres de l’architecture en supprimant les goulot d’étranglement En particulier ,on réduit la taille du filtre dans la première couche convultive de 11x11 à 5x5, ce qui a eu pour effet de réduire le nombre de caractéristiques

inactives apprises dans la première couche. Une fonctionnalité morte est une situation dans laquelle un noyau convultif n'apprend aucune représentation significative. Visuellement, cela ressemble à une image monotone monochrome, où toutes les valeurs sont proches les unes des autres.

En plus de changer la taille du filtre. Nous triplons le nombre de filtres dans toutes les couches de convolution et le nombre de neurones dans les couches entièrement connectées par rapport à AlexNet. Dans le réseau AlexNet, il y avait **48-128-192-192-128-2048-2048** noyaux / neurones et dans ce réseau, le nombre de ceux-ci a triplé pour atteindre **144-384-576-576-384-6144-6144**. Ce changement a permis au réseau d'accroître la complexité des représentations internes et, par conséquent, de réduire le taux d'erreur.

2.7 Conclusion

Après avoir réalisé la conception, qui aboutit à la présentation de notre architecture fonctionnelle de l'analyse perceptuelle, il serait judicieux de penser à mettre en œuvre notre projet. En effet, cela nous amène au troisième chapitre, ce dernier est consacré à la mise en œuvre de notre architecture.

Chapitre 3

Implémentation

3.1 Introduction

Ce chapitre nous permet de voir d'une manière générale le fonctionnement programmation et l'exécution de notre application les résultats obtenus, ainsi que les différentes technologies utilisées pour y parvenir.

3.2 Matériel utilisé

Processeur : Intel (R) Core (TM) i7
Capacité Mémoire (RAM) : 12 Go.
Vitesse d'horloge : 2.60 Ghz.
Capacité disque dur : 1 To.

3.3 Logiciel utilisé

- System d'exploitation windows 10, 64 bit
  Logiciel de calcul mathématique matlab R2017a de 4 eme génération

3.4 Implémentation de l'architecture proposée :

Le concept consiste à modifier au niveau des paramètres en calculant la précision lors de la comparaison avec AlexNet.

Dans la ***figure 24*** nous montrons un bloque de code standard qui nous permet de calculer le taux d'accuracy de notre architecture

```
EVAL = Evaluate(ACTUAL,PREDICTED)
Input:
ACTUAL = Column matrix with actual class labels of the training examples
PREDICTED = Column matrix with predicted class labels by the classification model
Output:
EVAL = Row matrix with all the performance measures
```

Figure 24 Code source calculant l'accuracy.

Dans la ***figure 25*** nous avons utilise des fonctions modifié pour calculer le taux de performance de notre architecture

```
testSet = imageDatastore(fullfile(rootFolder, categories), 'LabelSource', '
testSet.ReadFcn = @readFunctionTrain;

testFeatures = activations(convnet, testSet, featureLayer);
predictedLabels = predict(classifier, testFeatures);

confMat = confusionmat(testSet.Labels, predictedLabels);
confMat = confMat./sum(confMat,2);
mean(diag(confMat))
```

Figure 25 Code source calculant les performances.

Dans la ***figure 26*** nous appliquent une fonction de reconnaissance de scène sur l'architecture alexNet

```
    label=classify(nnet,picture);

    image(picture);
    title(char(label));
    drawnow;
end

[label,score]=classify(nnet,picture);

clear;
nnet=alexnet;
picture=imread('voiture1.jpg');
picture=imresize(picture,[227,227]);
[label,score]=classify(nnet,picture);
image(picture);
```

Figure 26 Code source testant une photo sur AleXNet.

Dans la ***figure 27*** nous montrons quelque exemple de données existant dans notre base de données

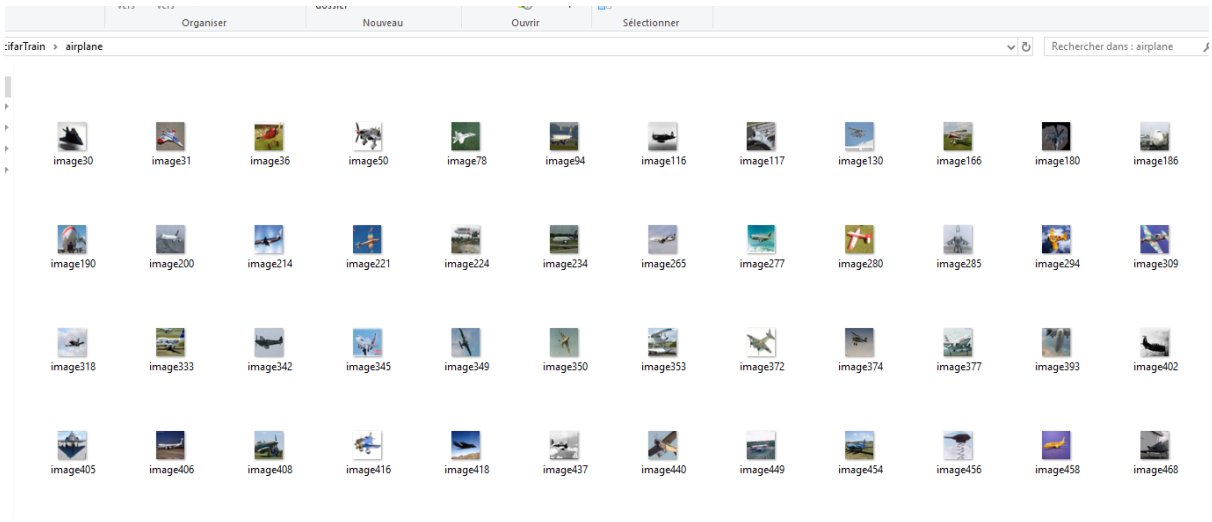


Figure 27 Base de données cifar-10 modifier(airplan).

Dans la ***figure 28*** nous présentons un code source appliqué pour la modification des paramètres de l'architecture alexNet

```
clear

conv1=convolution2dLayer(5,227,'Padding',2,'BiasLearnRateFactor',2);
conv1.Weights=gpuArray(single(randn([5 5 3 227])*0.0001));
fc1=fullyConnectedLayer(64,'BiasLearnRateFactor',2);
fc1.Weights=Array(single(randn([64 576])*0.1));
fc2=fullyConnectedLayer(4,'BiasLearnRateFactor',2);
fc2.Weights=Array(single(randn([4 64])*0.1));

layers=[
    imageInputLayer([227 227 3]);
    conv1;
    maxPooling2dLayer(3,'Stride',2);
    reluLayer();
    convolution2dLayer(5,32,'Padding',2,'BiasLearnRateFactor',2);
    reluLayer();
```

Figure 28 Code source des paramètre modifier.

Dans la ***figure 29*** un teste sur une image importé qui montre la convolution applique sur notre architecture

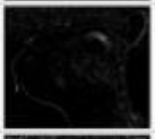
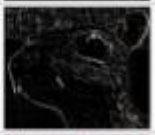
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figure 29 Test de convolution sur une image

Les ***figures 30, 31, 32*** présentent les différents teste appliquer pour vérifier la fiabilité de notre architecture ainsi que les résultats obtenu

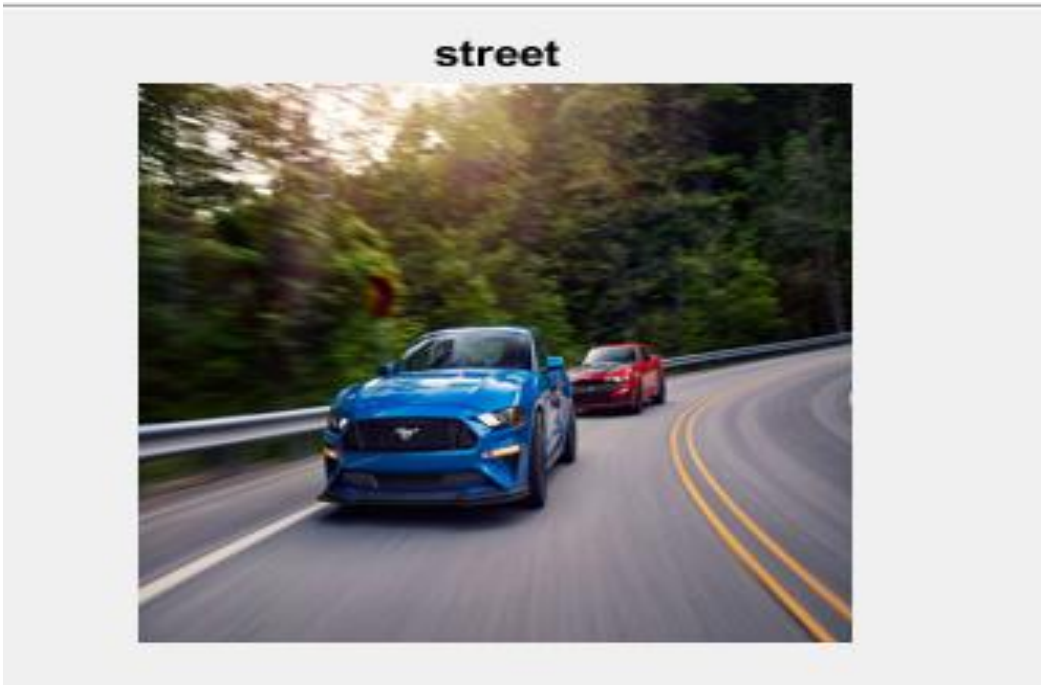


Figure 30 Test 1 de reconnaissance de scène 1.

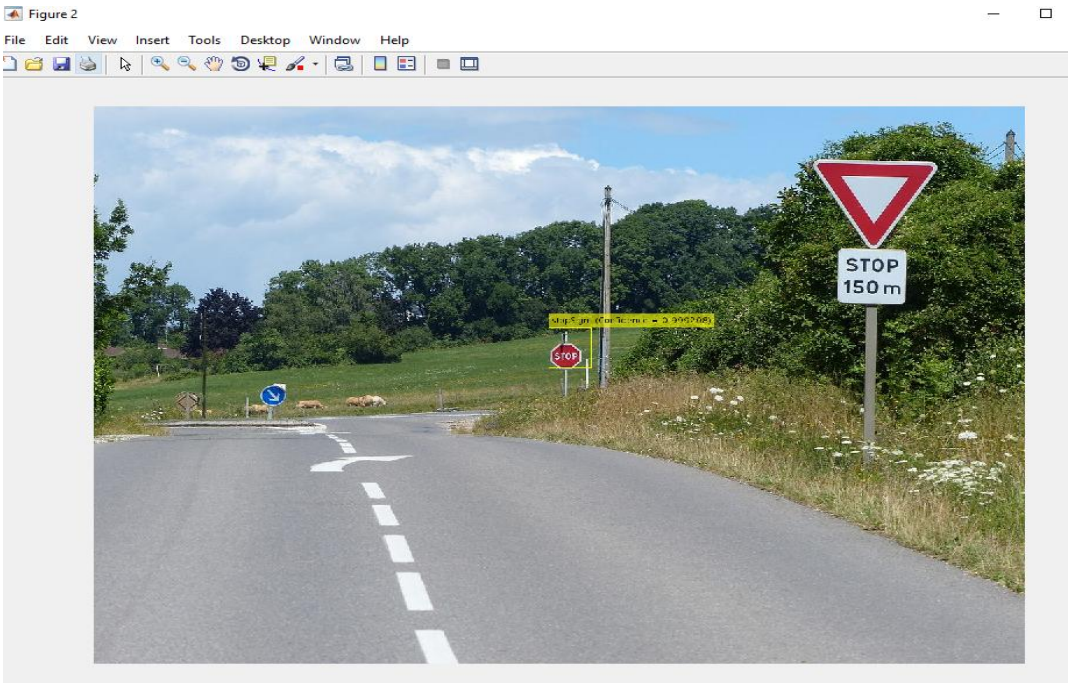


Figure 31 Test 2 de reconnaissance d'une plaque de stop.

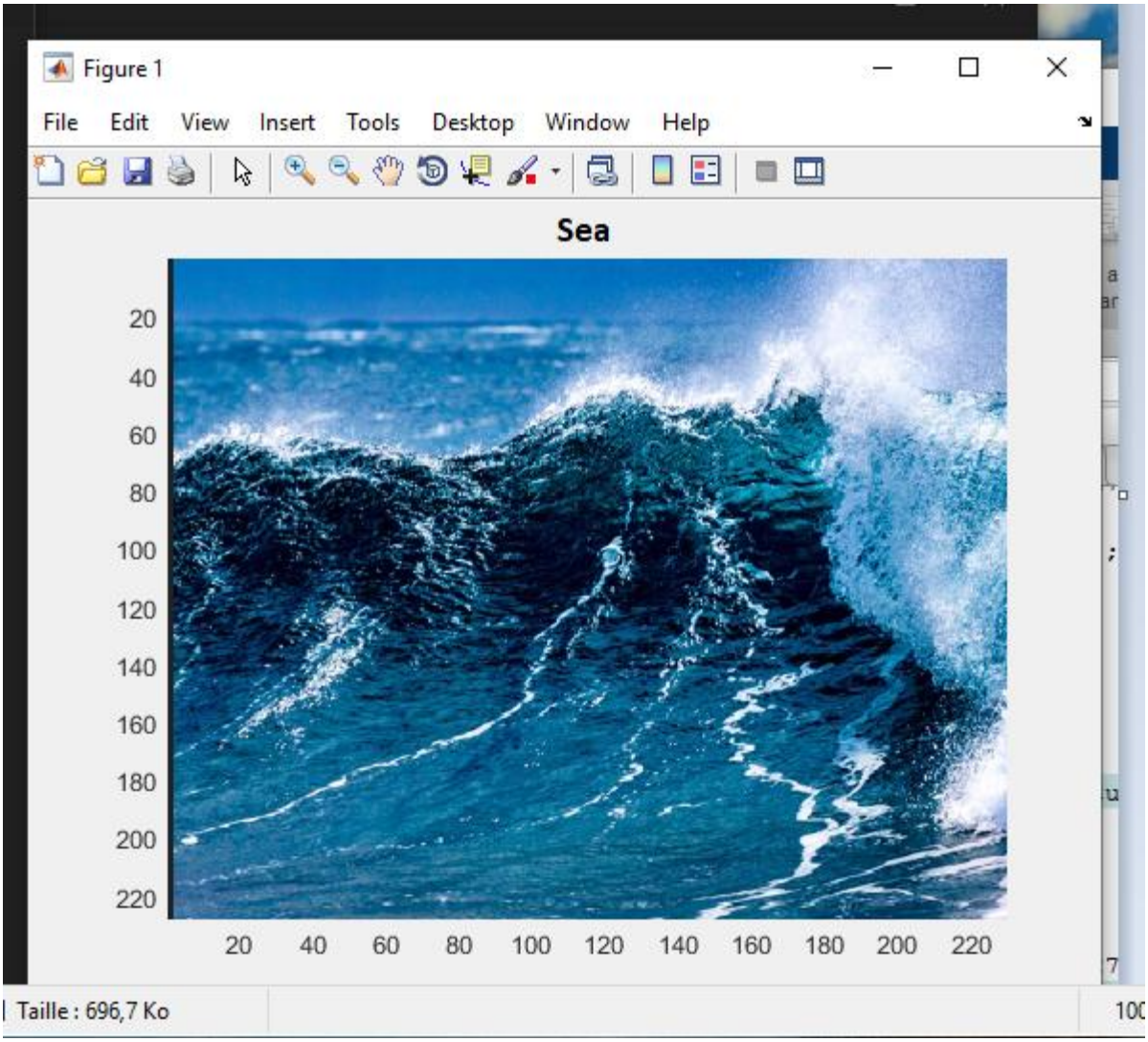


Figure 32 Test 3 de reconnaissance de scène 2.

Ici nous montrons les résultats obtenu durant le calcule des performances entre notre architecture qui représente le S et les performances de l'architecture alexnet qui représente le A

```
s =                    a =

    0.8703                 0.7760
```

Dans les ***figures 33, 34, 35, 36*** nous se basent a expliquer les différents paramètres modifier sur l'architecture alexNet a fin d'obtenir une architecture personnaliser

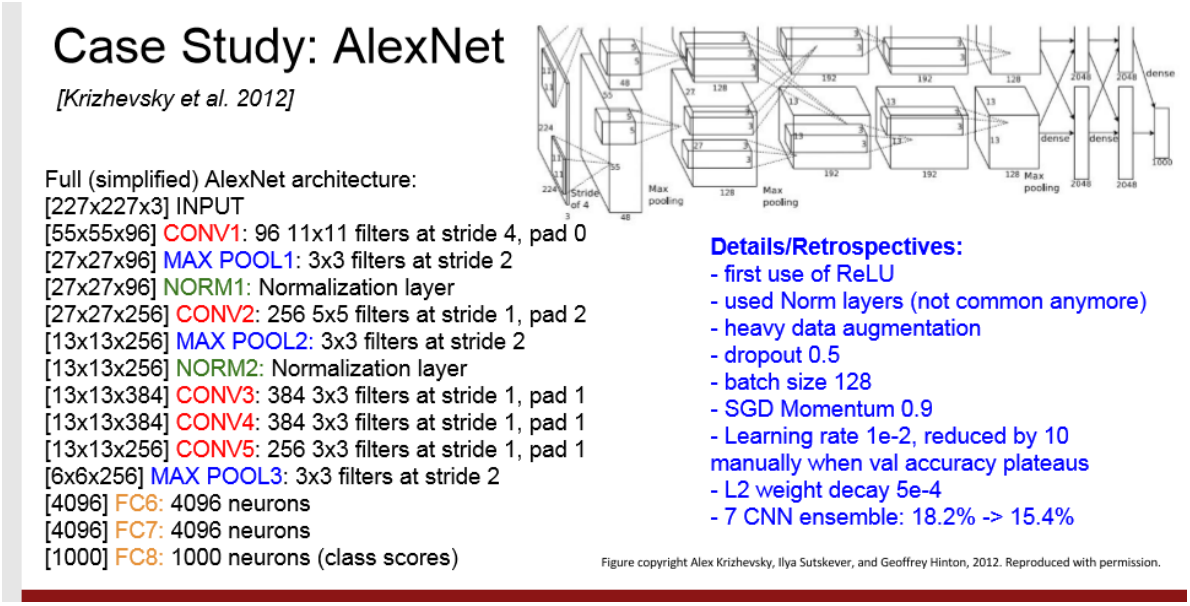


Figure 33 nom de chaque paramètre 1

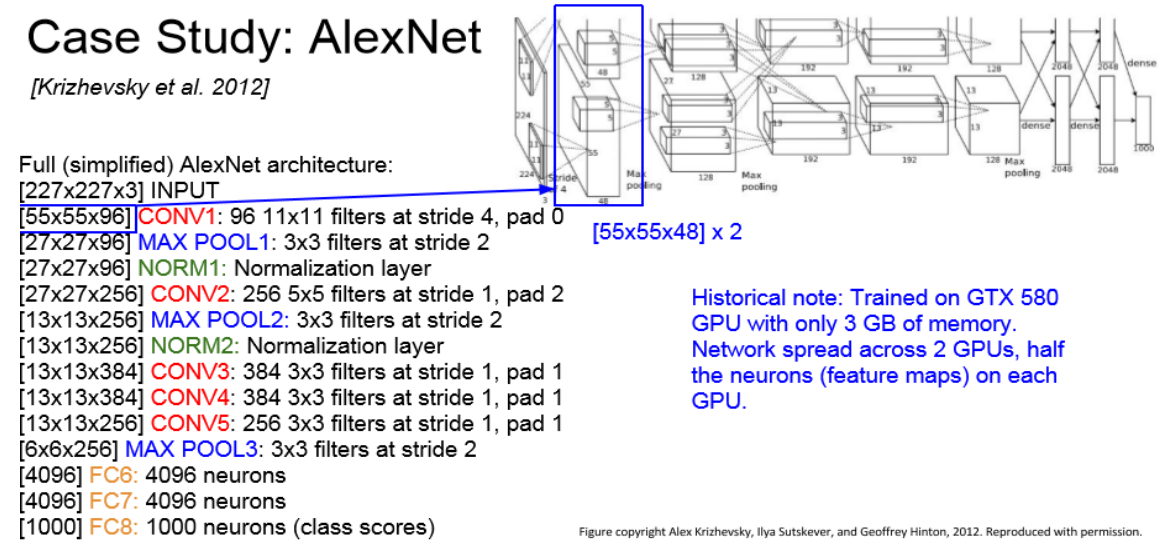


Figure 34 nom de chaque paramètre 2

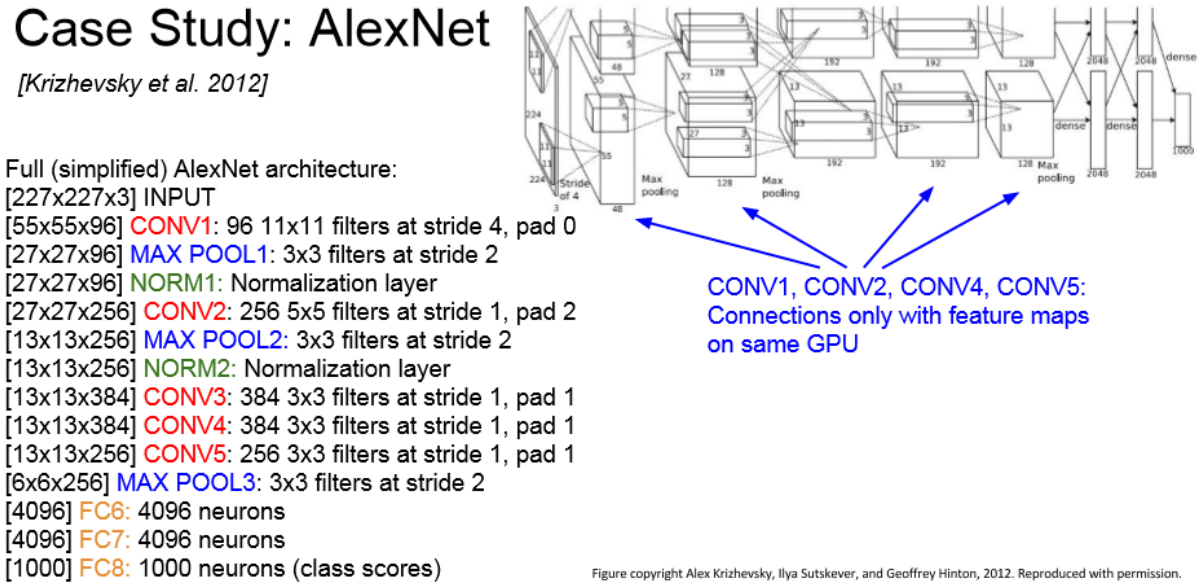


Figure 35 Nom de chaque paramètre 3

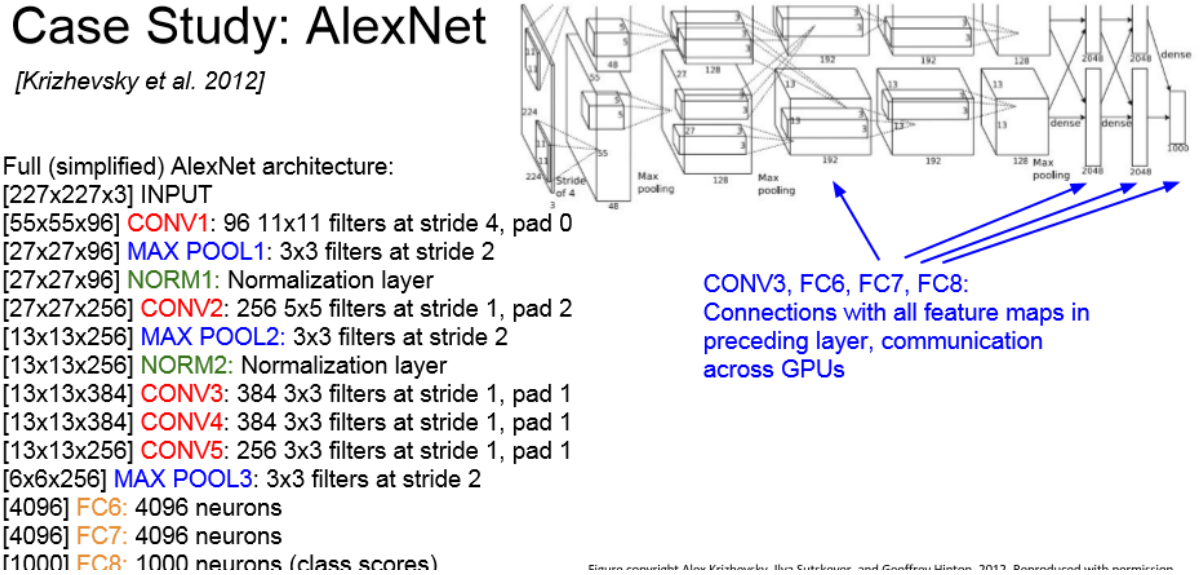


Figure 36 Nom de chaque paramètre 4

Dans la ***figure 37*** nous présentons un histogramme qui compare le taux d'erreur trouvé sur l'architecture alexNet aux différentes architectures existantes.

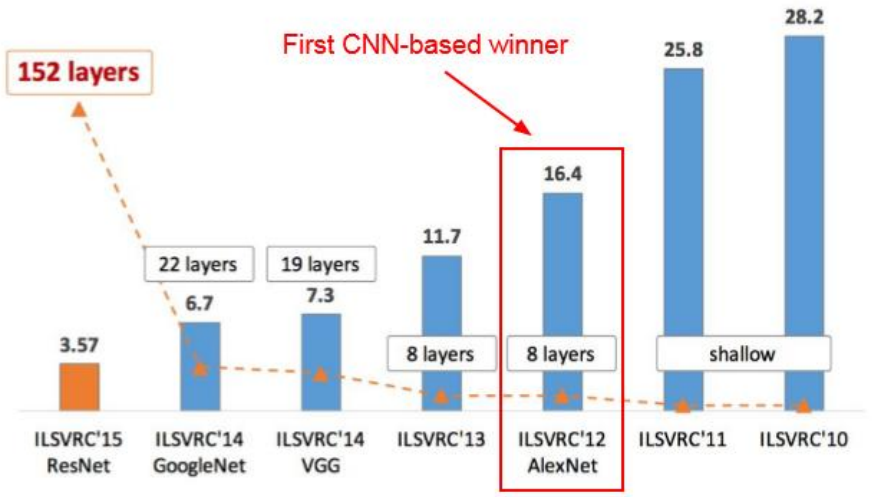


Figure 37 Histogramme calculant le taux d'erreur

Dans la ***figure 38*** nous montrons le schéma de notre architecture proposé qui se base sur l'architecture alexNet

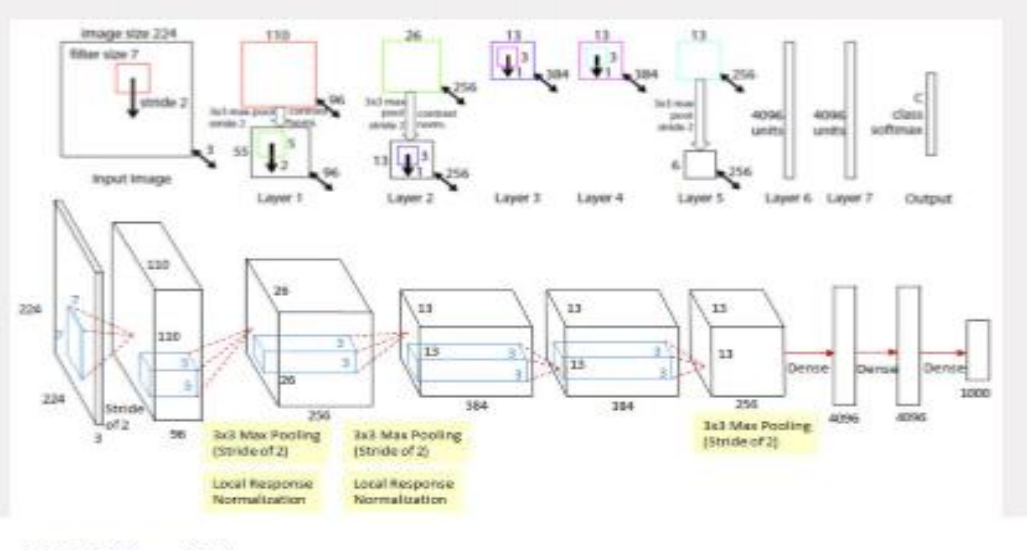


Figure 38 Schéma de l'architecture proposée.

3.5 Conclusion

Dans ce chapitre, nous présentons nos résultats de test avec des captures d'écran tout en décrivant notre architecture d'analyse perceptuelle pour la reconnaissance de scènes. Nous avons également présenté le test de comparaison qui montre la fiabilité de notre application.

Conclusion générale

Le travail effectué dans ce mémoire contribue à l'analyse perceptuelle pour la reconnaissance d'une scène. Notre proposition prend en compte la segmentation (étiquetage de chaque pixel) des images. À cette fin, nous avons proposé une architecture basée sur une autre architecture déjà existante (AlexNet) permettant l'analyse et la reconnaissance de scènes à partir d'une image importée. Les résultats sont préliminaires et sont étroitement liés aux facteurs suivants:

1. Modifiez le téléchargement d'une base de données CIFAR-10 afin de minimiser la grande quantité de données trouvées, réduisant ainsi le temps d'exécution du test.

2. Utilisation du logiciel de calcul mathématique matlab R2017a 4ème génération.

3. L'architecture utilisée pour la reconnaissance perceptuelle d'une scène et les différents paramètres modifiés.

Ces trois éléments ont une influence directe sur les résultats obtenus par nos recherches. Profitant du fait que ce domaine d'étude est encore en développement, nous envisageons d'étendre cette étude en utilisant d'autres architectures de reconnaissance de scène efficaces à partir d'une image importée, notamment l'utilisation de SeNet.

Après avoir beaucoup parlé d'architectures, nous pensons que le cœur de l'apprentissage en profondeur réside dans: Comment construisons-nous ces réseaux? Répondre à cette question nous aidera à développer la fiabilité de l'analyse perceptuelle pour la reconnaissance de scène.

Bibliographie

[1] David Chalmers, The Conscious Mind : In Search of a Fundamental Theory. Oxford University Press. hardcover : ISBN 0-19-511789-1, paperback : ISBN 0-19-510553-2 (1996)

[2] Nicolas Audebert: Classification de données massives de télédétection. (Classification of big remote sensing data). Bretagne Loire University, Rennes, France (2018)

[3] Diane Lingrand ,«Introduction au Traitement d'images» Livre. Vuibert . ISBN-13: 978-2711748662 ,2e édition (2008)

[4] Ian T. Young and Jan J. Gerbrands and Lucas J. van Vliet and Cip-data Koninklijke Bibliotheek and Den Haag and Young Ian Theodore and Gerbrands Jan Jacob and Van Vliet and Lucas Jozef, Fundamentals Of Image Processing, 1995

[5] Russell, S. et Norvig, P., Artificial Intelligence: A Modern Approach (2nd ed.) ISBN 0-13-790395-2; Prentice Hall, 2003

[6] Crevier, Daniel, AI: The Tumultuous Search for Artificial Intelligence, New York, NY: BasicBooks, ISBN 978-0-465-02997-6, 1993

[7] Bengio:Learning Deep Architectures for AI. Foundations and Trends in Machine Learning 2(1): 1-127 (2009)

[8] Liu, Xiaolong & Deng, Zhidong & Yang, Yuhan. Recent progress in semantic image segmentation. Artificial Intelligence Review. 1-18. 10.1007/s10462-018-9641-3. (2018).

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, Fei-Fei Li: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115(3): 211-252 (2015)

[10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, Andrew Zisserman: The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88(2): 303-338 (2010).

[12] Andreas Geiger, Philip Lenz, Raquel Urtasun: Are we ready for autonomous driving? The KITTI vision benchmark suite. CVPR 2012: 3354-3361

[13] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, Antonio Torralba: Semantic Understanding of Scenes Through the ADE20K Dataset. International Journal of Computer Vision 127(3): 302-321 (2019)

[14] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, Alan L. Yuille: The Role of Context for Object Detection and Semantic Segmentation in the Wild. CVPR 2014: 891-898

[15] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, Rob Fergus: Indoor Segmentation and Support Inference from RGBD Images. ECCV (5) 2012: 746-760

[16] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, Antonio Torralba: SUN database: Large-scale scene recognition from abbey to zoo. CVPR 2010: 3485-3492

[17] Shuran Song, Samuel P. Lichtenberg, Jianxiong Xiao: SUN RGB-D: A RGB-D scene understanding benchmark suite. CVPR 2015: 567-576

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele: The Cityscapes Dataset for Semantic Urban Scene Understanding. CVPR 2016: 3213-3223

[19] Sean Bell, Paul Upchurch, Noah Snavely, Kavita Bala: Material recognition in the wild with the Materials in Context Database. CVPR 2015: 3479-3487

[20] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet classification with deep convolutional neural networks. Commun. ACM 60(6): 84-90 (2017)

[21] Site Internet : http://image-net.org/challenges/LSVRC/ .

[22] Diaporama Automne 2011 Vision par ordinateur IMN 559 Presenter Par Pierre-Marc Jodoi page 4 et 5.

[23] Main architectures of deep learning - ZFNet 28 January 2019 ZFNet [2013, article by Zeiler et al.]

[24] CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more .... posted by Siddharth Das on November 16, 2017