République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Larbi Tébessi – Tébessa

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématiques et Informatique

Mémoire Fin D'étude

En Vu de l'obtention du Diplôme Master En Informatique

Option : Système Multimédia

Apprentissage Profond pour la Classification d'Images

Effectué par :

TAILEB AMMAR

Membre du Jury

Dr. Bennour Akram	MCA	Université Larbi Tébessi	Président
Dr. Gahmous Abdelatif	MAA	Université Larbi Tébessi	Examiner
Dr. Gattal Abdeljalil	MCA	Université Larbi Tébessi	Encadreur

Juin 2022

Université Larbi Tébessi- Tébessa

Faculté des sciences exactes et des sciences de la nature et de la vie

Département des Mathématiques et Informatique.

Filière : *Informatique.*

Année universitaire 2021/2022

Déclaration sur l'honneur de non-plagiat (A joindre obligatoirement avec le mémoire)

Je, soussigné(e)

Nom et prénom : Taileb Ammar

Régulièrement inscrit (e) : Master

N de carte d'étudiant : 1997502330

Année universitaire :2021/2022

Domaine : MATHS ET INFORMATIQUE

Filière : Informatique

Spécialité : SYM

Intitulé : Apprentissage profond pour la classification d'images

Atteste que mon mémoire est un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité, je certifie également que je n'ai ni copié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Sanctions en cas de plagiat prouvé :

L'étudiant sera convoqué devant le conseil de discipline, les sanctions prévues selon la gravité de plagiat sont :

- *L'annulation du mémoire avec possibilité de refaire sur un sujet différent.*
- *L'exclusion d'une année de Master.*
- *L'exclusion définitive.*

2022 جوان 27

Fait à Tébessa, le :10/06/2022

Signature de l'étudiant (e)

Université Larbi Tebessi- Tébessa

Faculté des sciences exactes et des sciences de la nature et de la vie

Département des Mathématiques et d'Informatique

Filière : (Mathématiques/Informatique)

Année universitaire 2021/2022

Autorisation de Soutenance
D'un Mémoire de (Licence/Master)

je, soussigné, Abdeljalil GAFFAR *encadreur de l'étudiant:* TAILEB Ammar

..... *ayant traité un sujet de (Licence/Master) qui a pour titre :*

.....

Apprentissage profond pour la classification d'images

.....

Atteste que la concernée a mené à son terme le travail qui lui a été exigé. Le mémoire qu'il a rédigé a été lu et corrigé par mon soin.

Par conséquent, Je lui autorise à déposer son mémoire en vue de soutenir son (master/licence) devant un jury que désignera le département.

Date : 07.06.2022

Signature de l'encadreur

Remerciement

Je remercie tout d'abord le bon dieu pour m'avoir donnée le courage et la santé pour accomplir ce travail.

Ce travail n'aurait pas pu aboutir à des résultats sans l'aide et les encouragements de plusieurs personnes que je remercie.

Mes vifs remerciements accompagnés de toute ma gratitude vont ensuite à mon encadreur Dr Gattal Adeljalil, pour ses conseils judicieux, sa grande disponibilité et pour m'avoir suivie et orientée et aussi Dr Menasel Rafik pour tout c'est conseils déterminé.

J'exprime ma profonde gratitude à Monsieur Dr BENOUR Akram , , qui m'a fait l'honneur de présider le jury de thèse. J'adresse mes sincères remerciements à mes prof durant Mon parcours étudiant, pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant d'en être examinateur.

Enfin, que tous ceux qui nous ont aidés et encouragés de près ou de loin dans la concrétisation de ce projet, trouvent ici ma gratitude et mes sincères remerciements.

Je tiens aussi à exprimer ma reconnaissance à toutes mes amies pour leur soutien inconditionnel.

T.Amar

Liste des Figures

Liste des Tableaux

Sommaire

Introduction Générale

Introduction générale

En général les méthodes de classification d'image s'exécutent en plusieurs étapes. L'étape la plus importante consiste à élaborer des règles de classification à partir de connaissances disponibles à priori; il s'agit de la phase d'apprentissage. Cette dernière utilise un apprentissage soit déductif soit inductif. Les algorithmes d'apprentissage inductif dégagent un ensemble de règles (ou de normes) de classification à partir d'un ensemble d'exemples déjà classés. Le but de ces algorithmes est de produire des règles de classification afin de prédire la classe d'affectation d'un nouveau cas. Parmi les méthodes de classification utilisant ce type d'apprentissage, on cite les méthodes des *k* plus proches voisins, la méthode bayésienne, la méthode d'analyse discriminante, l'approche des réseaux de neurones et la méthode d'arbre de décision ainsi Les Support Vector Machine (SVM) sont des algorithmes de classification binaire non linaire très puissant.. Dans les algorithmes d'apprentissage déductif, les règles d'affectation sont déterminées à priori par l'interaction avec le décideur, ou l'expert. À partir de ces règles on détermine les classes d'affectation des objets.

La complexité des algorithmes utilisés reste un facteur qui pose d'énormes problèmes. Il n'est pas évident de déterminer dans des temps raisonnables par exemple l'unité d'information qui va nous permettre de parcourir le contenu d'une image. Et plus on dispose de temps pour un traitement plus celui-ci a des chances de déboucher sur des résultats pertinents.

Dans le cas contraire les résultats n'auraient aucun intérêt.

Le Deep Learning est basé sur l'idée des réseaux de neurones artificielles et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de deep learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime.

Sur les cinq dernières années, le deep learning est passé d'un marché de niche ou seulement une poignée de chercheurs s'y intéressait au domaine le plus prisé par les chercheurs.. Le deep learning a classification des images , appris à conduire une voiture , diagnostiquer le cancer et l'autisme et même devenu un artiste

Problématique étudié

Nous allons nous intéresser à la problématique de la classification d'image plus spécifiquement s'intéressent à l'augmenter de taux de précision en lorsque en fait l'extraction des caractéristiques de texture Obif qui est la tâche d'attribuer à une image d'entrée *x* un label *y* à partir d'un ensemble fixe de catégories. C'est l'un des problèmes fondamentaux de la vision par ordinateur qui, malgré sa simplicité, a une grande variété d'applications pratiques.

Plusieurs bases ont été présentées dans la littérature, nous avons choisis les bases CIFAR-10, CIFAR-100,STL10, Nous allons commencer par entraîner avec deux type comme VGG16 et Alexnet de réseau de neurones convolutif .

Ce type de réseau possède des paramètres entraînables et il consiste en des couches convolutives, une couche de Max pooling est placé après chaque série de couches convolutives et enfin couches cachées entièrement connectées et une couche de sortie entièrement connectée.

De nombreuses stratégies utilisées dans le Deep Learning sont explicitement conçues pour réduire l'erreur de test, éventuellement au détriment d'une augmentation de l'erreur d'apprentissage. Ces stratégies sont connues sous le nom de régularisation.

Les techniques d'augmentation de données génèrent artificiellement différentes versions d'un jeu de données réel pour augmenter sa taille. Les modèles de vision par ordinateur utilise une stratégie d'augmentation des données pour gérer la rareté et la diversité insuffisante des données.

Les algorithmes d'augmentation de données peuvent augmenter la précision des modèles d'apprentissage automatique. Selon une expérience, un modèle d'apprentissage en profondeur après augmentation d'image est plus performant en termes de perte d'apprentissage (c'est-à-dire de pénalité pour une mauvaise prédiction) et de perte de précision et de validation qu'un modèle d'apprentissage en profondeur sans augmentation pour la tâche de classification d'images.

Notre objectif de travail est utilisée les images de texture Obifs avec différent paramètre pour augmenter le nombre d'échantillon dans la base d'apprentissage entiers.

Les images Obifs sont des informations de texture et d'orientation extraite à partir des images originale.

Notre mémoire est organisé en trois chapitres. Dans le premier, nous présenterons la problématique de la classification des images en présentant un résumé sur les techniques de classification les plus utilisées, son application à l'indexation d'images, les limites de ces méthodes et les contraintes.

Dans le second chapitre, nous nous intéressions à la méthode de classification de deep learning utilisé leur description et architecture.

Et pour satisfaire notre modèle de deep learning, **il faut appliquer des techniques de Data Augmentation aux images d'entraînement**, dans deuxième point.

La troisième point c'est L'analyse (ou caractérisation) de texture joue un rôle important dans la segmentation d'image ou dans sa classification. Les éléments apportant le plus de précisions dans la segmentation sont les fréquences spatiales et la moyenne du niveau RGB, les techniques utilisées et Obif.

Le chapitre trois en a proposé des modèles de classification d'image en trois expérimentation but.

Nous présentons dans la première expérimentation le réseau Vgg16 sur les bases de données Cifar10 ainsi que l'évaluation de nos résultats.

Dans le deuxième expérimentation en augment notre modèle avec Obif et en évalué nos résultats.

En troisième expérimentation, Nous terminerons notre mémoire par la conclusion et les perspectives de notre approche. .

Chapitre 01

Introduction

Les **réseaux de neurones à convolution profonde** sont devenus les méthodes de pointe pour les tâches de **classification d'images**. Cependant, l'une de leurs plus grandes limites est qu'ils nécessitent beaucoup de données annotées (images dont la classe à prédire est connue). Par exemple, un réseau ayant pour unique tâche de reconnaître des chats, devra être entraîné avec des milliers de photos de chats avant qu'il ne puisse discerner cet animal d'une autre entité avec une bonne précision. Autrement dit, plus le jeu de données d'apprentissage est important, meilleure sera la **précision de l'algorithme**.

Cette contrainte n'est pas négligeable car il est difficile voire parfois impossible de collecter des quantités aussi importantes de données. Dans de nombreux cas on aimerait que les réseaux de neurones apprennent de nouveaux concepts avec peu de données, c'est à dire, qu'ils aient un comportement proche de l'homme.

1. Type de Machine Learning :

1.1 Machine Learning classique L'apprentissage automatique (Machine learning) est un domaine de recherche en informatique qui traite des méthodes d'identification et de mise en œuvre de systèmes et algorithmes par lesquels un ordinateur peut apprendre, ce domaine a souvent été associé à l'intelligence artificielle et plus spécifiquement l'intelligence computationnelle[1].

1.2 Deep Learning ou apprentissage profond Le Deep Learning est basé sur l'idée des réseaux de neurones artificielles et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de deep learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime[2].

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

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

2. L’architecture de la machine Learning

La théorie de l'apprentissage utilise des outils mathématiques dérivés de la théorie des probabilités et de la théorie de l'information. Cela vous permet d'évaluer l'optimalité de certaines méthodes par rapport aux autres. On peut citer trois types d'algorithme d'apprentissage automatique : • Apprentissage supervisé. • Apprentissage non supervisé. • Apprentissage par renforcement. [2]

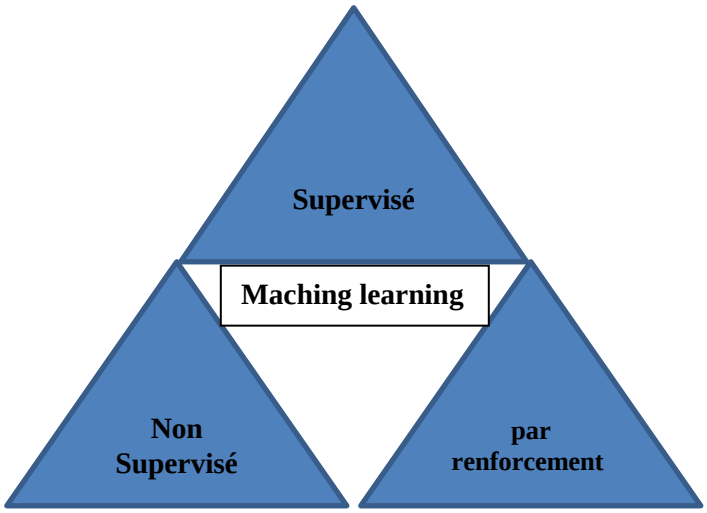

Fig 01 : les types de machine learning

on distingue principalement trois types d’apprentissages

- **L'apprentissage supervisé** Il est basé sur un certain nombre d'exemples pré classifiés, dans lesquels est connu à priori la catégorie à laquelle appartient chacune des entrées utilisées comme exemples. Dans ce cas, la question cruciale est le problème de généralisation, après l'analyse d'un échantillon d'exemples, le système devrait produire un modèle qui devrait fonctionner pour toutes les entrées possibles. L'ensemble de données pour l'entrainement, est

constitué de données étiquetées, c'est-à-dire d'objets et de leurs classes associées. Cet ensemble d'exemples étiquetés constitue donc l'ensemble d'apprentissage.

- Apprentissage non supervisé La deuxième classe d'algorithmes d'apprentissage automatique est appelée apprentissage non supervisé, dans ce cas, nous n'étiquetons pas les données au préalable, nous laissons plutôt l'algorithme arriver à sa conclusion. Ce type d'apprentissage est important car il est beaucoup plus commun dans le cerveau humain que l'apprentissage supervisé. Les algorithmes d'apprentissage non supervisé sont particulièrement utilisés dans les problèmes de clustering, dans lesquels, étant donné une collection d'objets, nous voulons être en mesure de comprendre et de montrer leurs relations.

- Apprentissage par renforcement L'apprentissage par renforcement est une approche de l'intelligence artificielle qui met l'accent sur l'apprentissage du système à travers ses interactions avec l'environnement. Avec l'apprentissage par renforcement, le système adapte ses paramètres en fonction des réactions reçues de l'environnement, qui fournit ensuite un retour d'information sur les décisions prises. [3]

	APPRENTISSAGE SUPERVISÉ	APPRENTISSAGE NON-SUPERVISÉ	APPRENTISSAGE PAR RENFORCEMENT
DÉFINITION	L'algorithme apprend à partir de données labellisées	L'algorithme est entraîné à partir de données non labellisées sans indications particulières	L'algorithme interagit avec son environnement en réalisant des actions et en apprenant de ses erreurs et succès
TYPE DE PROBLÈMES	Régression et classification	Association et Clustering	Basés sur un système de récompense
TYPE DE DONNÉES	Données labellisées	Données non labellisées	Pas de données fournies au préalable
APPROCHE	Étudie les relations sous-jacentes qui lient les données en entrée aux labels	Découvre les motifs communs au sein des données d'entrée	Apprend une stratégie de comportement en fonction d'expériences passées et des récompenses perçues

Table 01 : méthode d'apprentissage

3. Architecture De Deep Learning

Il est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d'abstraction. Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral. [4].

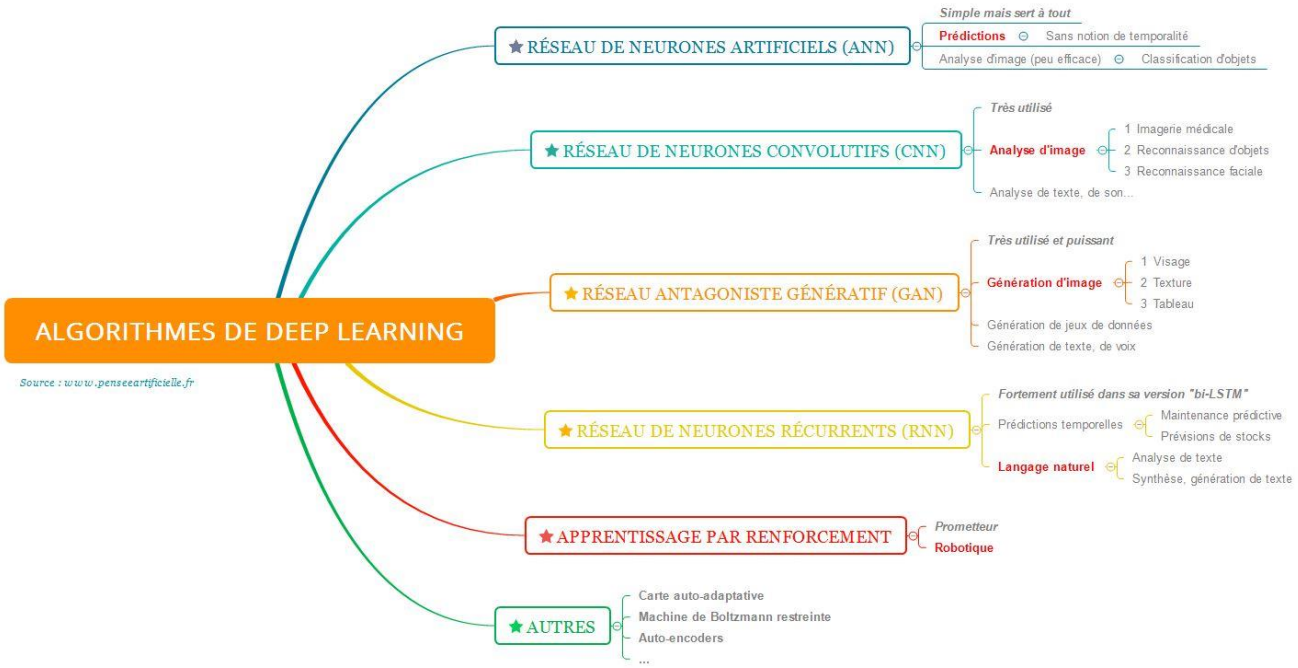

Fig 02 : les Algorithme de deep learning

4- Classification d'images via le deep learning

Dans le cas d'une classification standard, l'image d'entrée est introduite dans une série de couches de convolution, qui génère une **distribution de probabilités sur toutes les classes** (généralement à l'aide de la fonction softmax). Par exemple, si on essaye de classer une image comme étant un « chat », un « chien », un « cheval » ou un « éléphant », pour chaque image d'entrée appartenant à l'une de ces classes, quatre probabilités seront générées, indiquant le niveau de confiance avec lequel le réseau a étiqueté l'image.

Deux points importants doivent être notés ici.

Tout d'abord, pendant le processus d'apprentissage, le réseau a besoin d'un grand nombre d'images pour chaque classe (chat, chien, cheval et éléphant).

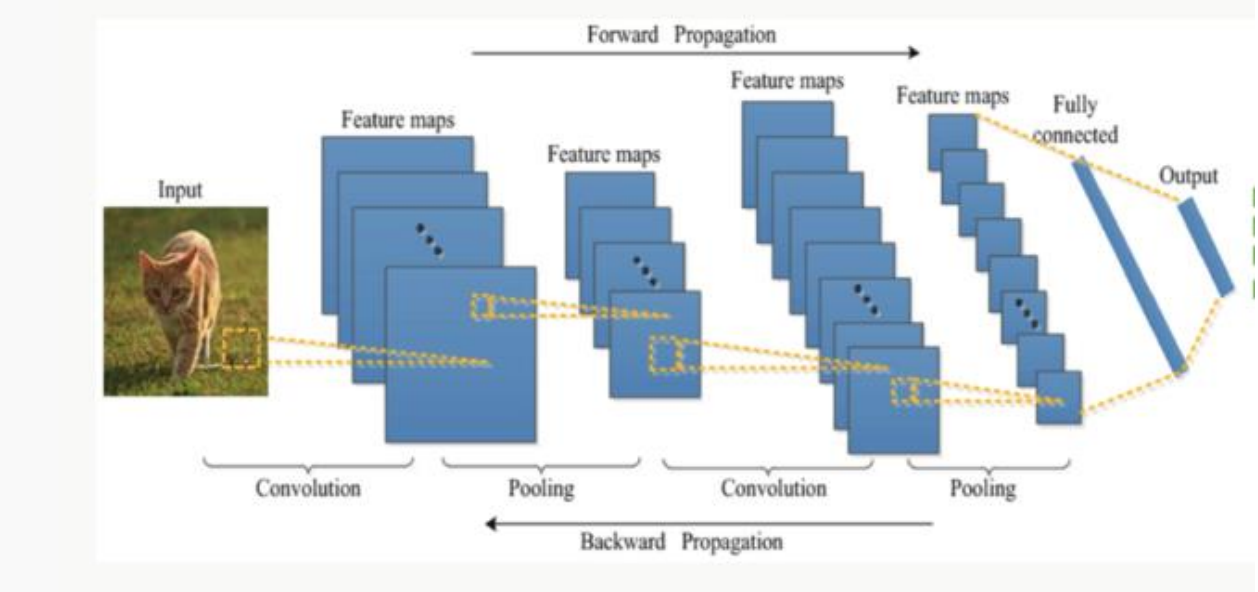

Fig 03 : structure réseaux de neurones convolutifs [5]

Ensuite, si le réseau est entraîné uniquement sur ces 4 classes, « chat », « chien », « cheval » et « éléphant », comme dans l'exemple ci-dessus, il ne sera pas en mesure d'étiqueter correctement l'image d'un zèbre. En effet lorsque l'image du zèbre sera donnée en entrée au réseau, ce dernier sortira quatre probabilités, chacune exprimant le niveau de confiance avec lequel l'image appartient aux classes « chat », « chien », « cheval » ou « éléphant ». Pour que le réseau reconnaisse également les images de zèbre, il faudra d'abord obtenir un grand nombre d'images de cet animal, puis ré-entraîner le modèle.

Il existe des applications pour lesquelles très peu de données sont disponibles pour un très grand nombre de classes. De plus, il peut être nécessaire de modifier la base de données, en supprimant ou en ajoutant une classe. Le coût de la collecte de données et du réentraînement devient donc trop élevé, ce qui est problématique. Un exemple typique est la reconnaissance faciale. Une organisation qui souhaite mettre en place un tel système pour ses employés doit construire une base de données contenant beaucoup d'images de visages différents. Si l'organisation en question se basait sur un modèle de **deep learning classique** (un réseau ConvNet, par exemple), elle serait aussi obligée de ré-entraîner son modèle de classification à chaque fois qu'un nouvel employé la rejoint. Cela est en pratique impossible, particulièrement pour les grandes organisations où le recrutement est un processus continu quasi quotidien.

5- Les bases de données de classification d'images

5.1 MNIST :

L'acronyme MNIST (*Modified* ou *Mixed National Institute of Standards and Technology)*, est une base de données de chiffres écrits à la main. C'est une base de données très utilisé en apprentissage automatique.

La reconnaissance de l'écriture manuscrite est un problème difficile, et un bon test pour les algorithmes d'apprentissage. La base MNIST est devenue un test standard. Elle regroupe 60,000 images d'apprentissage et 10,000 images de test, issues d'une base de données antérieure, appelée simplement NIST. Ce sont des images en noir et blanc, normalisées centrées de 28 pixels de côté. [6]

5.2 CIFAR 10

La base de données CIFAR-10 comprend 60000 images 32×32 ombrées dans 10 classes, avec 6000 images pour chaque classe. L'ensemble de données est divisé en cinq groupes de préparation et un groupe de test, chacun contenant 10 000 images. La grappe de test contient précisément 1000 images choisies au hasard dans chaque classe. Les groupes de préparation contiennent le reste des images des requêtes arbitraires, mais certains groupes de préparation peuvent contenir un plus grand nombre d'images d'une classe que d'une autre. Entre eux, les groupes de préparation contiennent précisément 5000 images de chaque classe.

Voici les classes de l'ensemble de données, ainsi que 10 images irrégulières de chacune d'entre elles : Il y a 50000 images de préparation et 10000 images d'essai. [7]

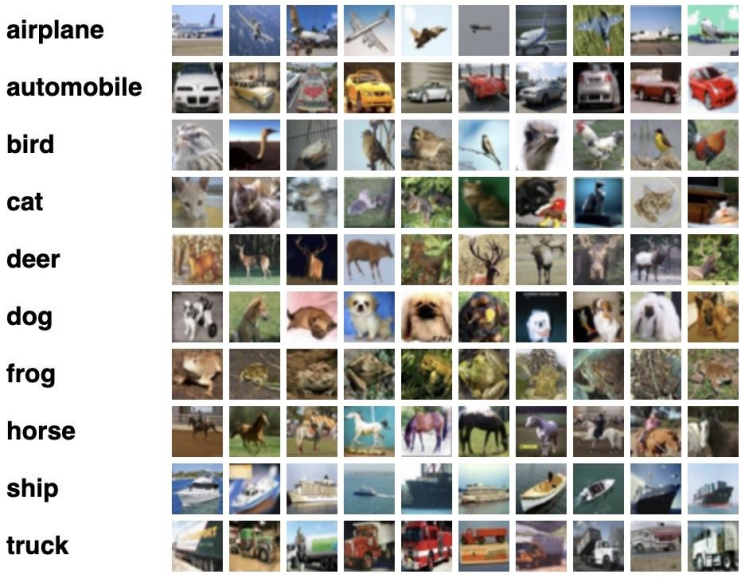

Fig 04 : les classes de CIFAR 10

5.3 CIFAR 100

La base de données CIFAR-100 (Institut canadien de recherches avancées, 100 classes) est un sous-ensemble de la base de données Tiny Images et se compose de 60 000 images couleur 32x32. Les 100 classes du CIFAR-100 sont regroupées en 20 superclasses. Il y a 600 images par classe. Chaque image est accompagnée d'une étiquette "fine" (la classe à laquelle elle appartient) et d'une étiquette "grossière" (la superclasse à laquelle elle appartient). Il y a 500 images d'apprentissage et 100 images de test par classe.

Les critères pour décider si une image appartient à une classe étaient les suivants :

Le nom de la classe doit figurer en haut de la liste des réponses probables à la question « Qu'y a-t-il dans cette image ? »

L'image doit être photo-réaliste. Les étiqueteurs ont reçu pour instruction de rejeter les dessins au trait.

L'image ne doit contenir qu'une seule instance bien visible de l'objet auquel la classe fait référence.

L'objet peut être partiellement occulté ou vu d'un point de vue inhabituel tant que son identité est encore claire pour l'étiqueteur.

5.4 STL-10

La base de données STL-10 est un ensemble de données de reconnaissance d'images pour le développement d'algorithmes d'apprentissage de caractéristiques non supervisé, d'apprentissage en profondeur et d'apprentissage autodidacte. Il est inspiré de la base de données CIFAR-10 mais avec quelques modifications. En particulier, chaque classe a moins d'exemples de formation étiquetés que dans CIFAR-10, mais un très grand nombre d'exemples non étiquetés est fourni pour apprendre des modèles d'image avant la formation supervisée. Le principal défi consiste à utiliser les données non étiquetées (qui proviennent d'une distribution similaire mais différente des données étiquetées) pour construire un a priori utile. Nous prévoyons également que la résolution plus élevée de cet base de données (96x96) en fera une référence difficile pour développer des méthodes d'apprentissage non supervisées plus évolutives.

- 10 classes : avion, oiseau, voiture, chat, cerf, chien, cheval, singe, bateau, camion.
- Les images sont 96x96 pixels, couleur.
- 500 images d'entraînement (10 plis prédéfinis), 800 images de test par classe.
- 100 000 images non étiquetées pour un apprentissage non supervisé. Ces exemples sont extraits d'une distribution similaire mais plus large d'images. Par exemple, il contient d'autres types d'animaux (ours, lapins, etc.) et de véhicules (trains, bus, etc.) en plus de ceux de l'ensemble étiqueté.
- Les images ont été acquises à partir d'exemples étiquetés sur ImageNet. [8]

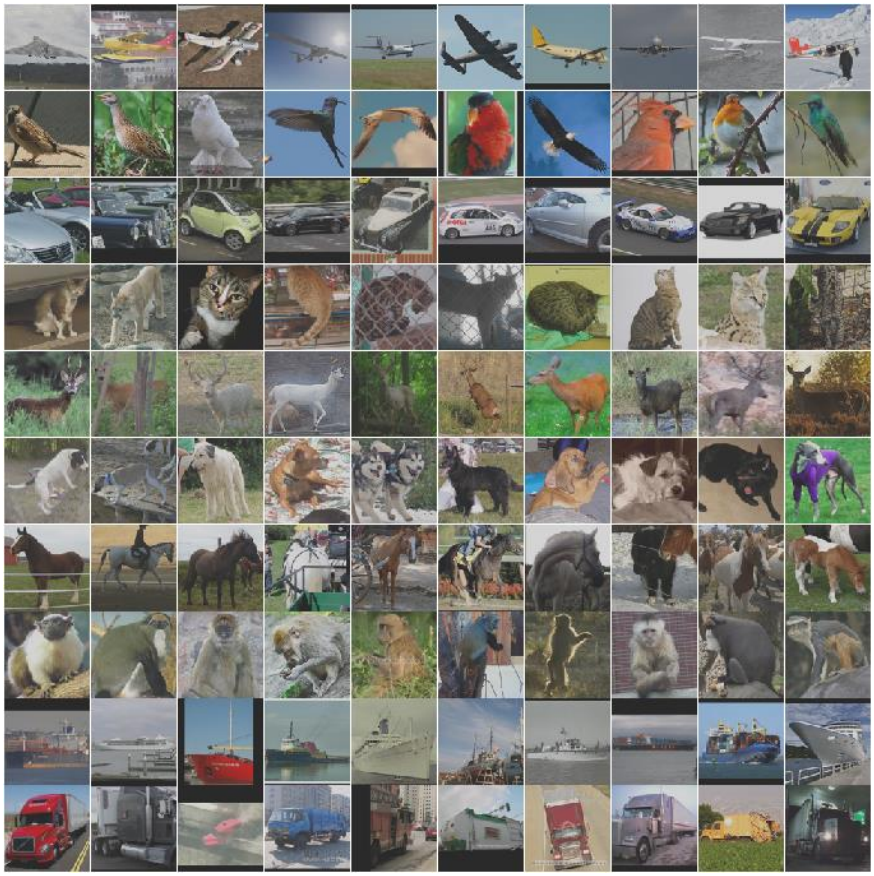

Fig 05 : les classe de STL 10

5.5 Fashion-MNIST

Cette base de données est un ensemble de données d'images d'articles de Zalando, composé d'un ensemble d'apprentissage de 60 000 exemples et d'un ensemble de test de 10 000 exemples. Chaque exemple est une image en niveaux de gris 28x28, associée à une étiquette de 10 classes. Zalando souhaite que Fashion-MNIST remplace directement l'ensemble de données MNIST d'origine pour l'analyse comparative des algorithmes d'apprentissage automatique. Il partage la même taille d'image et la même structure de fractionnements d'entraînement et de test.

La base de données MNIST d'origine contient de nombreux chiffres manuscrits. Les membres de la communauté AI/ML/Data Science adorent cet ensemble de données et l'utilisent comme référence pour valider leurs algorithmes. En fait, MNIST est souvent le premier ensemble de données que les chercheurs essaient. "Si cela ne fonctionne pas sur MNIST, cela ne

fonctionnera pas du tout", ont-ils déclaré. "Eh bien, si cela fonctionne sur MNIST, cela peut encore échouer sur d'autres."

Zalando cherche à remplacer l'ensemble de données original du MNIST[9]

5.6 Pascal VOC

La base de données PASCAL Visual Object Classes (VOC) 2012 contient 20 catégories d'objets, y compris les véhicules, les ménages, les animaux et autres : avion, vélo, bateau, bus, voiture, moto, train, bouteille, chaise, table à manger, plante en pot, canapé, TV/moniteur, oiseau, chat, vache, chien, cheval, mouton et personne. Chaque image de cet ensemble de données comporte des annotations de segmentation au niveau du pixel, des annotations de cadre de délimitation et des annotations de classe d'objets. Cet ensemble de données a été largement utilisé comme référence pour les tâches de détection d'objets, de segmentation sémantique et de classification. L'ensemble de données PASCAL VOC est divisé en trois sous-ensembles : 1 464 images pour la formation, 1 449 images pour la validation et un ensemble de test privé. . [10]

5.7 MS COCO

La base de données MS COCO (Microsoft Common Objects in Context) est un ensemble de données de détection d'objets à grande échelle, de segmentation, de détection de points clés et de sous-titrage. Le jeu de données se compose de 328K images.

Divisions : la première version de l'ensemble de données MS COCO a été publiée en 2014. Elle contient 164 000 images réparties en ensembles d'entraînement (83 Ko), de validation (41 Ko) et de test (41 Ko). En 2015, un ensemble de test supplémentaire de 81 000 images a été publié, comprenant toutes les images de test précédentes et 40 000 nouvelles images.

Sur la base des commentaires de la communauté, en 2017, la répartition formation/validation est passée de 83K/41K à 118K/5K. La nouvelle division utilise les mêmes images et annotations. La base de test 2017 est un sous-ensemble de 41K images de l'ensemble de test 2015. De plus, la version 2017 contient un nouvel ensemble de données non annoté de 123 000 images. [11]

Conclusion

Les Réseaux de neurones sont un domaine de recherche très actif, qui ne cesse de progresser afin d‘améliorer les performances des résultats. L‘apprentissage profond (DL) a pu s‘imposer et révolutionner plusieurs domaines technologiques. Traduction automatique moderne, moteurs de recherche, assistants informatiques et plusieurs applications de notre vie quotidienne sont tous alimentés par un apprentissage profond. Les modèles d'apprentissage profond tentent d‘imiter au maximum les modes de traitement de l'information et de communication observés dans le système nerveux biologique. Ce chapitre a présenté les architectures des réseaux de neurones les plus connues et les plus utilisées ainsi que les nouvelles architectures qui semblent avoir un avenir prometteur dans différents domaines d‘application de la technologie contemporaine.

Nous avons montré le rôle des réseaux neurones convolutifs. Ces réseaux sont capables d’extraire des caractéristiques d’images présentées en entrée et de classifier ces caractéristiques. Ils sont fondés sur la notion de « champs récepteurs » (receptive fields),

Chapitre 02

Introduction

Les Réseaux de neurones sont un domaine de recherche très actif, qui ne cesse de progresser afin d'améliorer les performances des résultats.

L'apprentissage profond (*Deep Learning*) a pu s'imposer et révolutionner plusieurs domaines technologiques. Traduction automatique moderne, moteurs de recherche, assistants informatiques, classification d'image et plusieurs applications de notre vie quotidienne sont tous alimentés par un apprentissage profond. Les modèles d'apprentissage profond tentent d'imiter au maximum les modes de traitement de l'information et de communication observés dans le système nerveux biologique.

Ce chapitre a présenté les différents architectures des réseaux de neurones les plus connues et les plus utilisées ainsi que les techniques de data augmentation puis les caractéristiques de texture.

1- Les Méthodes de l'apprentissage profond

1.1 Réseaux neurones convolutifs (CNN) (LEcun)

Les réseaux de neurones convolutifs sont des réseaux couramment utilisés pour des problèmes de classification d'images; ces derniers se sont montrés efficaces dans divers travaux de recherche grâce à leur capacité à extraire des caractéristiques dans les images.

La structure d'un réseau de neurones convolutif est essentiellement composée de couche(s) de convolution, de couche(s) de Pooling et de couche(s) d'un réseau de neurones artificiels entièrement connectée. La couche principale et la plus importante de ce type de réseau est la couche de convolution d'où son appellation : réseau de neurones convolutif.

1.1.1 Architecture globale de CNN

1.1.2. Couche convolutive

Les couches convolutives constituent le noyau du réseau convolutif. Ces couches se composent d'une grille rectangulaire de neurones qui ont un petit champ réceptif étendu à travers toute la profondeur du volume d'entrée. Ainsi, la couche convolutive est juste une convolution d'image de la couche précédente, où les poids spécifient le filtre de convolution. La couche convolutive déterminera la sortie des neurones qui sont connectés aux régions locales de l'entrée par le calcul du produit scalaire entre leurs poids et la région connectée au volume d'entrée. ReLu vise à appliquer une fonction d'activation «élémentaire» telle qu'une fonction sigmoïde à la sortie de l'activation produite par la couche précédente. [12]

1.1.3. Couche de pooling

Après chaque couche convolutive, il peut y avoir une couche de pooling. Cette couche sous échantillonne le long de la dimensionnalité spatiale de l'entrée donnée, ce qui réduira davantage le nombre de paramètres au sein de cette activation. Il y a plusieurs façons de faire cette mise en commun, comme prendre la moyenne ou le maximum, ou une combinaison linéaire prise par des neurones dans le bloc.

1.1.4. Couche totalement connectée

Après les couches de convolution et pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches totalement connectées.

Dans les réseaux de neurones convolutifs, chaque couche fait comme un filtre de détection pour la présence de caractéristiques spécifiques ou de motifs présents dans les données d'origine. Les premières couches d'un réseau convolutif détectent des caractéristiques qui peuvent être reconnues et interprétées facilement. Les couches ultérieures détectent de plus en plus des caractéristiques plus abstraites. La dernière couche du réseau convolutif est capable de faire une classification ultra-spécifique en combinant toutes les caractéristiques spécifiques détectées par les couches précédentes dans les données d'entrée.

Les couches totalement connectées font les mêmes tâches que celles des ANN standard et tenteront de produire des notes de classe à partir des activations, pour les utiliser pour la classification. Il est également suggéré d'utiliser ReLu entre ces couches pour améliorer les performances. [13]

1.1.5. Couche de correction (ReLu)

C'est une couche pour améliorer l'efficacité du traitement en additionnant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLu : **F(x)=max (0, x)** Cette fonction force les neurones à retourner des valeurs positives.

1.2 LeNet (1998)

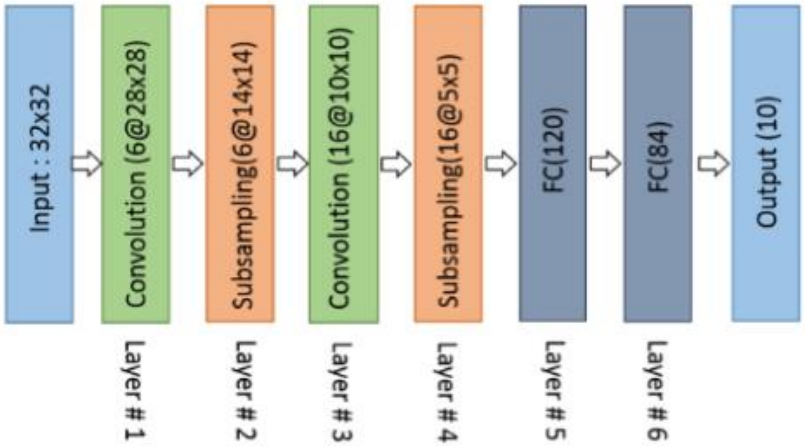

Fig 06 : Architecture de LeNet

LeNet ait été proposé dans les années 1990, la capacité de calcul et la capacité de mémoire limitées ont rendu l'algorithme difficile à mettre en œuvre jusqu'en 2010 environ.
LeCun, cependant, a proposé des CNN avec l'algorithme de rétro propagation et a expérimenté sur un ensemble de données de chiffres manuscrits pour atteindre des précisions de pointe.

Son architecture est bien connue sous le nom de LeNet-5. La configuration de base de LeNet-5 est : 2 couches de convolution (conv), 2 couches de sous-échantillonnage, 2 couches entièrement connectées et une couche de sortie avec connexion gaussienne.
Le nombre total de pondérations et de multiplication et d'accumulation (MAC) est respectivement de 431k et 2,3M. [14]

1.3 Alexnet (2012)

On ne peut pas parler de Deep Learning sans mentionner Alexnet. En effet, c'est l'un des pionniers du Deep Neural Net qui vise à classifier les images. Il a été développé par Alex Krizhevsky, Ilya Sutskever et Geoffrey Hinton et a remporté de loin le Défi de la classification des images (ILSVRC) en 2012.

À l'époque, les autres algorithmes concurrents n'étaient pas basés sur le deep learning. Aujourd'hui, et depuis lors, ils le sont presque tous. Ce réseau a eu un impact énorme sur le domaine et la plupart des réseaux suivants étaient plus ou moins basés sur son architecture. AlexNet est composé de 5 couches convolutionnelles (C1 à C5 sur le schéma) suivies de deux couches entièrement connectées (FC6 et FC7), et d'une couche finale de sortie softmax (FC8). Il a été initialement formé pour reconnaître 1000 objets différents.

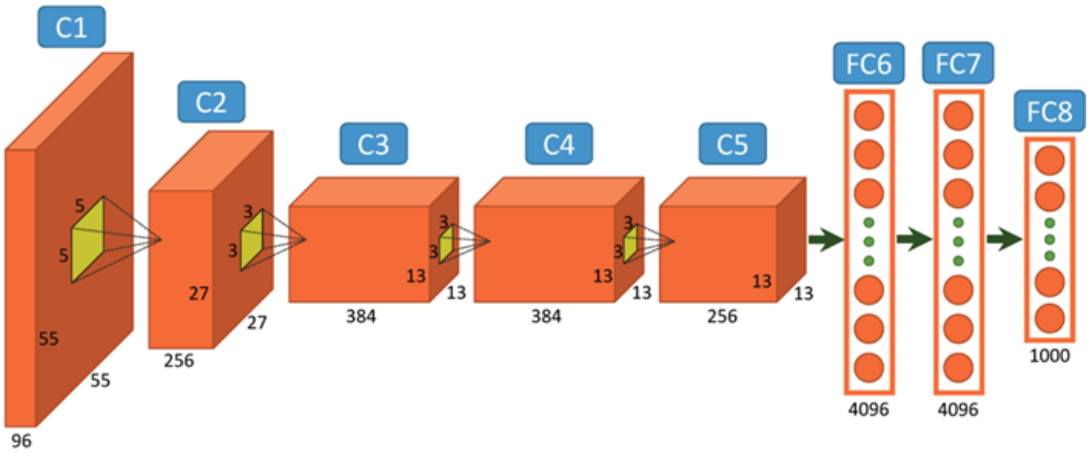

Fig 07 :Architecture Alexnet [15]

La perception derrière de ce réseau est que chaque couche convolutionnelle apprend une représentation plus détaillée des images (feature map) que la précédente. Par exemple, la première couche est capable de reconnaître des formes ou des couleurs très simples, et la dernière des formes plus complexes comme des visages complets par exemple.

Les couches convolutionnelles représentent l'image de manière bien plus efficace pour la classification. Après les convolutions, chaque image est représentée comme un vecteur de 4096 caractéristiques (alors qu'elles étaient initialement des vecteurs de 227*227*3 = 154 587 caractéristiques).

Les deux couches entièrement connectées et Softmax sont similaires à une perception multicouche et pourraient en fait être remplacées par d'autres types de classificateurs tels que les Random Forests ou les SVM. Cependant, elles sont vraiment importantes pour la phase d'apprentissage du réseau neuronal.

1.4 ZFNet (2013)

ZFNet est un réseau neuronal convolutif classique. La conception a été motivée par la visualisation des couches d'entités intermédiaires et le fonctionnement du classificateur. Par rapport à AlexNet, les tailles de filtre sont réduites et la foulée des convolutions est réduite., il se compose de 5 couches convolutionnelles, deux couches entièrement connectées et une couche softmax de sortie. Les différences sont par exemple des noyaux convolutionnels mieux dimensionnés.

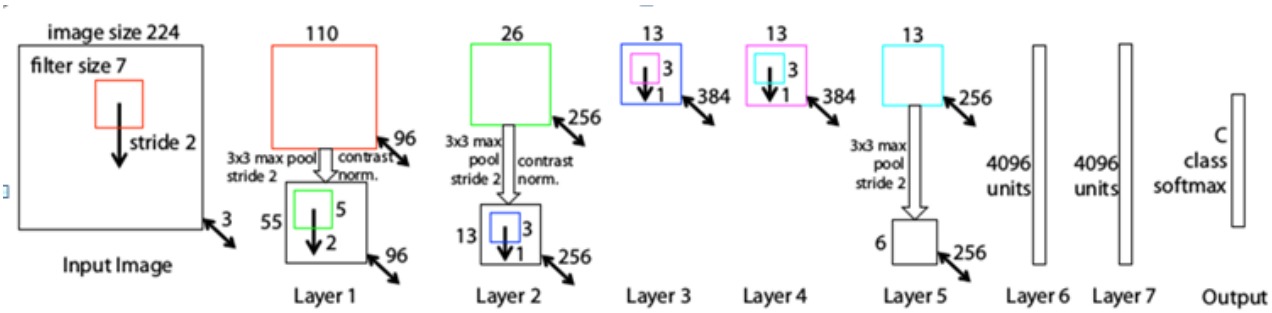

Fig 08 : Architecture ZFNet [16]

1.5 VGG (2014)

VGG est un réseau très profond et simple. Dans la version la plus courante, il comporte 16 couches. Cependant, l'architecture globale est très similaire à celle d'Alexnet. En fait, les couches convolutionnelles d'Alexnet sont ici représentées par deux ou trois couches convolutionnelles successives. Une autre différence est que chaque couche convolutionnelle a un noyau 3×3 contrairement aux autres réseaux qui ont des noyaux de taille différente pour chaque couche.

Les couches convolutionnelles représentent l'image d'une manière beaucoup plus efficace pour la classification. Après les convolutions, chaque image est représentée comme un vecteur de 4096 caractéristiques (alors qu'elles étaient initialement des vecteurs de 227*227*3 = 154 587 caractéristiques).

Les deux couches entièrement connectées et softmax sont similaires à une perception multicouche et pourraient en fait être remplacées par d'autres types de classificateurs tels que Random Forests ou les SVM. Cependant, elles sont vraiment importantes pour la phase de formation du réseau neuronal.

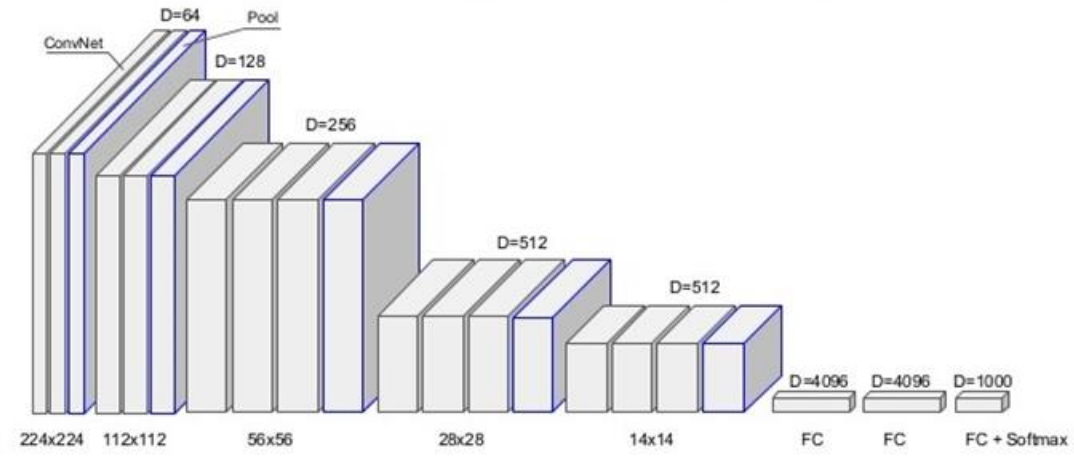

Fig 09 : Architecture VGGNet [17]

Ces réseaux importants sont construits pour classifier les images, c'est-à-dire pour produire une classe lorsqu'une image est montrée. Ce problème est assez bien résolu puisque les résultats d'aujourd'hui dépassent les performances humaines.

Mais concentrons-nous maintenant sur le sujet principal : la détection d'objets dans les images. Ce problème est bien plus difficile car l'algorithme doit non seulement trouver tous les objets dans une image mais aussi leur emplacement exact. En d'autres termes, l'algorithme doit être capable de détecter qu'une zone spécifique de l'image (à savoir une « boîte ») contient un certain type d'objet.

1.6 ResNet :

ResNet est un modèle d'apprentissage en profondeur bien connu qui a été présenté pour la première fois dans un article de Shaoqing Ren, Kaiming He, Jian Sun et Xiangyu Zhang. En 2015, une étude intitulée « Deep Residual Learning for Image Recognition » a été publiée.

Les ResNets sont constitués de ce qu'on appelle un bloc résiduel. Ceci est construit sur le concept de "connexions sautées" et utilise beaucoup de normalisation par lots pour lui permettre de former des centaines de couches avec succès sans sacrifier la vitesse au fil du temps.

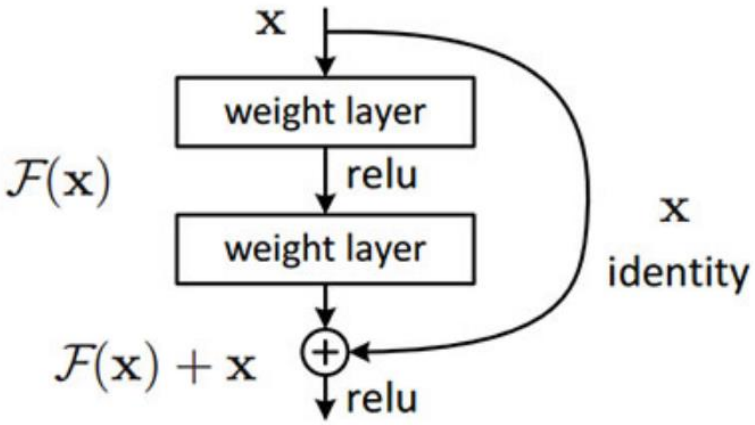

Fig 10 : diagramme de ResNet[18]

La première chose que nous remarquons dans le diagramme ci-dessus est qu'il existe un lien direct qui saute plusieurs niveaux du modèle. La « connexion de saut », comme on l'appelle, se trouve au cœur des blocs résiduels. En raison de la connexion sautée, la sortie n'est pas la même. Sans la connexion de saut, l'entrée 'X est multipliée par les poids de la couche, puis en ajoutant un terme de biais.

1.7 GoogleNet :

Le concours ILSVRC 2014 a été remporté par GoogleNet ou Inception Network, qui avait un taux d'erreur parmi les 5 premiers de 6,67 %, ce qui correspondait à des performances pratiquement humaines. Google a créé le modèle, qui intègre une mise en œuvre améliorée de la conception originale de LeNet. Ceci est basé sur le concept de module de démarrage. GoogLeNet est une variante du réseau Inception, qui est un réseau de neurones à convolution profonde à 22 couches.

GoogLeNet est maintenant utilisé pour une variété d'applications de vision par ordinateur, y compris la détection et l'identification des visages, la formation contradictoire, etc. [19]

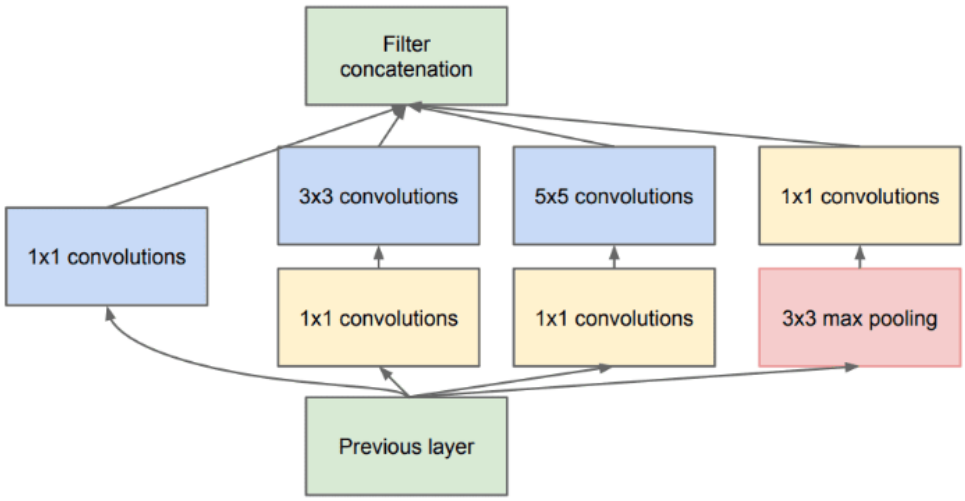

Fig 11 : Le module de démarrage de GoogleNet

1.8 MobileNetV1 :

Le modèle MobileNet est construit sur des convolutions séparables en profondeur, qui sont un type de convolution factorisée qui divise une convolution régulière en une convolution en profondeur et une convolution ponctuelle. La convolution en profondeur utilisée par MobileNets applique un seul filtre à chaque canal d'entrée. Les sorties de la convolution en profondeur sont ensuite combinées à l'aide d'une convolution 11 par la convolution ponctuelle. En une étape, une convolution conventionnelle filtre et mélange les entrées pour créer un nouvel ensemble de sorties. La convolution séparable en profondeur divise cela en deux couches : une pour le filtrage et l'autre pour la combinaison. [20]

1.9 Réseau densément connecté (DenseNet en 2017)

DenseNet développé par Gao Huang et d'autres en 2017, qui se compose de couches CNN densément connectées, les sorties de chaque couche sont connectées à toutes les couches successeurs dans un bloc dense. Par conséquent, il est formé d'une connectivité dense entre les couches qui lui vaut le nom de "DenseNet"[21].

Ce concept est efficace pour la réutilisation des fonctionnalités, ce qui réduit considérablement les paramètres du réseau. DenseNet se compose de plusieurs blocs denses et blocs de transition, qui sont placés entre deux blocs denses adjacents. Le schéma conceptuel d'un bloc dense est illustré à la Fig

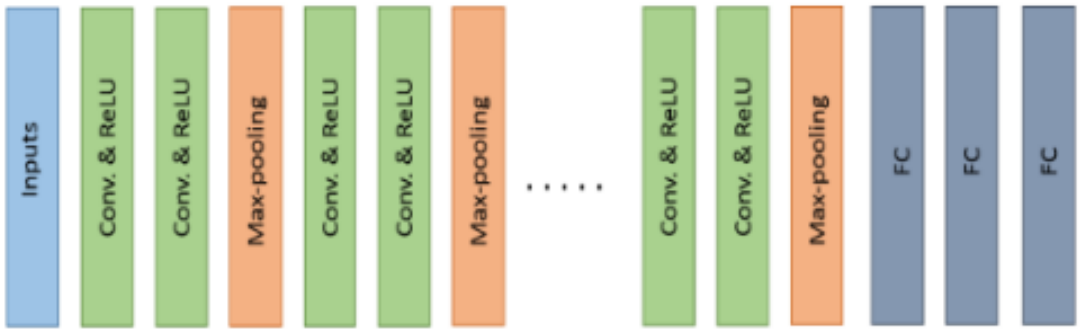

Fig 12 : schéma conceptuel d'un bloc dense

2- Data Augmentation :

Les techniques d'augmentation de données créées artificiellement différentes versions d'un jeu de données réelles pour augmenter sa taille. Les modèles de vision par ordinateur et de traitement du langage naturel (NLP) utilisent une stratégie d'augmentation des données pour gérer la rareté et la diversité insuffisante des données.

Les algorithmes d'augmentation de données peuvent augmenter la précision des modèles d'apprentissage automatique. Selon une expérience, un modèle d'apprentissage en profondeur après augmentation d'image est plus performant en termes de perte d'apprentissage (c'est-à-dire de pénalité pour une mauvaise prédiction) et de perte de précision et de validation qu 'un modèle d'apprentissage en profondeur sans augmentation pour la tâche de classification d'images.

Il existe des méthodes d'augmentation d'images afin de créer une diversité d'images dans le modèle. Il est facile de trouver de nombreux exemples de codage pour ces transformations d'augmentation dans des bibliothèques open source et dans des articles sur le sujet. [22]

2.1 Transformations géométriques

2.1.1. Rotation

Cette transformation, comme son nom l'indique, consiste à effectuer une rotation de l'image originale selon un angle souhaité.

2.1.2. Réflexions (en anglais flips)

Les réflexions d'images s'effectuent selon un axe de symétrie. Pour l'augmentation des images, elles peuvent être effectuées aussi bien verticalement qu'horizontalement,

2.1.3. Translations (en anglais shift)

Cette transformation peut être effectuée selon l'axe x et/ou y aléatoirement . L'image transformée conserve la même orientation que l'image originale et est déplacée selon la direction appliquée.

2.1.4. Recadrage (en anglais crop) et zoom

Ces deux méthodes possèdent visuellement le même résultat, certaines parties de l'image sont rognées afin de n'en conserver qu'une partie.

Lors d'un recadrage, les dimensions de l'image seront diminuées (e.g. recadrage d'une image en 512x512 vers une image en 250x250). Le ratio hauteur/largeur n'est pas nécessairement conservé.

Lors d'un zoom, le recadrage conserve le même rapport de dimensions que l'image d'origine. L'image est redimensionnée identiquement à l'image d'origine. Pour cela, les pixels créés pour récupérer les dimensions originales ont une valeur interpolée à partir d'une fonction définie en amont de l'application du zoom à l'image originale (e.g. la luminosité moyenne des pixels voisins). Par conséquent, une perte de qualité peut être constatée. Ce type de transformation est généralement appliqué aléatoirement au jeu de données.

2.1.5. Recadrage avec remplissage (en anglais padding)

Sur le même principe que la section précédente, un recadrage est effectué mais afin de restaurer/conserver les dimensions de l'image d'origine, un remplissage est effectué (avec une valeur de pixel égale à 0) autour de l'image recadrée.

2.1.6. Cisaillement (en anglais shear)

Le cisaillement horizontal (selon l'axe des x) et/ou vertical (selon l'axe des y) est une transformation affine qui consiste à décaler dans des directions opposées le haut et le bas de l'image (cisaillement horizontal) ou la droite et la gauche de l'image (cisaillement vertical). Contrairement aux méthodes précédentes, l'image est déformée.

2.2. Transformations diverses

2.2.1. Ajout de bruit

Le bruit est une variation aléatoire de la luminosité à l'intérieur de l'image. Il dégrade la qualité de l'image originale et peut être de plusieurs types : bruit blanc gaussien, bruit de Poisson, bruit impulsionnel (poivre et sel).

2.2.2. Filtre gaussien

Ce filtre de convolution possède un noyau (en anglais kernel) de forme gaussienne. Celui-ci balaye la totalité de l'image originale pour la débruiter voire la flouter, grâce à l'application d'une fonction gaussienne aux pixels de l'image. Celle-ci réalise une moyenne pondérée des pixels convolutionnés en accordant un poids plus important aux pixels à

proximité du centre du noyau (contrairement à un filtre moyen où chaque pixel présent dans le noyau possède le même poids).

2.2.3. Contraste

Le contraste d'une image est augmenté lorsque les pixels foncés de cette image sont assombris et les pixels clairs sont éclaircis : une image contrastée contiendra donc une plus grand quantité de noir et de blanc . Il est possible de visualiser sur l'histogramme de l'image l'augmentation du contraste, car l'écart entre les pixels les plus clairs et les pixels les plus foncés de l'image est plus important.

2.2.4. Luminosité

Ces images sont dépendantes de la luminosité (en anglais value) ; c'est-à-dire que plus la valeur de cette dernière est faible, plus l'image est assombrie. Afin d'augmenter la taille de la base de données .

2.2.5. Accentuation (en anglais sharpening)

Cette transformation augmente le contraste au niveau des contours dans l'image. Des voxels clairs et foncés situés dans une même zone seront éclaircis et assombris suite à cette transformation. L'application est réalisée avec un certain rayon autour du pixel de bord, il correspond au nombre de pixels entourant ce pixel cible et plus le rayon est grand, plus l'accentuation des contours est importante. Il en est de même pour le bruit présent dans l'image si celui-ci n'est pas atténué au préalable . En se focalisant ainsi sur les frontières entre les régions à l'intérieur de l'image et non sur le contraste global, l'image paraît alors plus nette.[23]

3- Les méthodes de classification d'image basé sur Deep Learning

Plusieurs méthodes de classification d'image basé sur le deep learning ont été développées en utilisant les différentes méthodes de reconnaissance techniques. Dans ce qui suit, nous présenterons quelques travaux de différents auteurs sur l'image classification.

3.1. Foret et al. (2021)

Les auteurs présentent la minimisation sensible à la netteté (SAM), une nouvelle méthode qui améliore la généralisation du modèle en minimisant en même temps la valeur de la perte et la netteté de la perte en reformule le cycle d'optimisation qui se produit lors de la formation des organisations neuronales.

Plutôt que de découvrir un vecteur de poids, qui minimise effectivement la fonction de perte pour l'ensemble de formation donné, cette stratégie propose de résoudre un problème d'optimisation différent, en trouvant les minima de la fonction de perte environnante.

Plutôt que d'utiliser la méthode Gradient-Descent pour déterminer les pour décider du minimum global (absolu) de la fonction de perte et pour rafraîchir les poids du réseau dans cette direction, leur algorithme ajuste le réseau, de sorte qu'autour de ce point, la fonction de perte produise des valeurs minimales.[24]

Les bases de données : CIFAR-10, CIFAR-100, Fashion-MNIST…

Voici quelques résultats lorsque SAM est utilisé dans certains ensembles de données:

Tache	Base de données	Taux de classification
Classification d'image	CIFAR 10	99.70
	CIFAR 100	96.08
	Fashion-MNIST	96.41

Tableau 02 : Minimisation sensible sur différentes base de données.

3.2. Tanveer et al. (2020)

Le réglage fin est une méthode éprouvée pour améliorer les performances d'un réseau neuronal. Il s'agit d'un d'un processus qui utilise un réseau neuronal déjà formé à une tâche donnée et lui fait exécuter une autre tâche similaire en aval.

Une autre tâche similaire en aval. S'inspirant du réglage fin, les auteurs proposent d'incorporer des opérations fixes pour affiner le réglage de DARTS.[25]

Les bases de données : CIFAR-10, CIFAR-100, Fashion-MNIST

Tache	Base de donnée	Précision taux de classification
Classification d'image	CIFAR 10	97.87
	CIFAR 100	84.33
	Fashion-MNIST	3.09 (Taux d'erreur)

Tableau 03 : Réglage fin des DARTS sur différentes bases de données.

3.3. Hugo et al. (2021)

Les transformateurs ont récemment été utilisés pour la classification d'images à grande échelle et ont atteint des niveaux élevés scores, ébranlant la domination à long terme des réseaux de neurones convolutifs. Cependant, jusqu'à présent, il y a eu peu d'études d'optimisation sur les processeurs d'images. Dans ce travail, les auteurs construit et optimise un réseau de transformateurs plus profond pour la classification des images. Ils ont spécifiquement étudié l'interaction entre l'architecture et l'optimisation de ces processeurs spécialisés. Ils ont apporté des modifications aux deux architectures Transformer, ce qui a considérablement amélioré la précision de Deep Transformer. [26]

Base de données : CIFAR-10, CIFAR-100,

Voici quelques résultats lorsque cette méthode est utilisée dans certains bases de données :

Tache	Base de donnée	Précision taux de classification
Classification d'image	CIFAR 10	99.40
	CIFAR 100	93.10

Tableau 4: les résultats des transformateurs d'image sur différentes base de données.

3.4. Dosovitskiy et al. (2020)

Inspiré par le succès du Transformer Scaling en NLP(**Natural Language Processing**), les auteurs a essayé d'appliquer le Transformer standard directement aux images avec le moins de modifications possible.

Ils ont l'image en patchs et ont fourni des séquences d'incorporation linéaires de ces patchs comme entrée au transformateur. Dans les applications NLP, les patchs d'images sont traités de la même manière que les tokens (mots).

Elles entraînent des modèles de classification d'images de manière supervisée. Sur des ensembles de données de taille moyenne tels que ImageNet, ces modèles produisent une précision modérée.

Ces résultats ont constaté que l'apprentissage à grande échelle l'emportait sur le biais inductif. La Vision Transformer a obtenu d'excellents résultats lorsqu'il a été pré-entraîné à une échelle suffisante et transféré à des tâches comportant moins de points de données. .[27]

Base de données : CIFAR-10, CIFAR-100,

Tache	**Base de donnée**	**Précision taux de classification**
Classification d'image	CIFAR 10	99.50
	CIFAR 100	94.55

Tableau 5 : les résultats de cette méthode sur différentes base de données

3.5. Harris et al., (2020)

L'augmentation des données d'échantillons mixtes (Mixed Sample Data Augmentation - MSDA) a reçu de plus en plus d'attention ces dernières années. Les auteurs ont proposé FMix, (**Enhancing Mixed Sampled Data Augmentation**') qui est un type de MSDA qui utilise un masque binaire aléatoire obtenu en appliquant un seuil à des images basse fréquence échantillonnées dans l'espace de Fourier.

Ces masques aléatoires peuvent prendre diverses formes et peuvent être générés pour un, deux et trois dimensions. FMix améliore les performances de MixUp et CutMix sans augmenter le temps de formation. Il convient à plusieurs modèles sur une série d'ensembles de données et paramètres de problème. Un nouveau modèle unique peut être obtenu sur CIFAR-10 sans données externes.

Enfin, ils ont montré que le résultat de l'interpolation de la différence entre MSDA (comme MixUp) et le masquage MSDA (comme FMix) est que les deux peuvent être combinés pour améliorer les performances. [28]

Bases de données : CIFAR-10, CIFAR-100, Fashion-MNIST.

Voici quelques résultats lorsque cette méthode est utilisée dans certains jeux de données :

Tache	Base de donnée	Précision taux de classification
Classification d'image	CIFAR 10	98.64
	CIFAR 100	83.95
	Fashion-MNIST	3.64 (taux d'erreur)

Tableau 6: FMix avec différentes bases de données

3.6. Dipu et al.(2020)

L'auteur propose SpinalNet, qui est créé avec certains points communs avec le travail de la moelle épinière humaine, ce qui permet d'obtenir une plus grande précision avec moins de ressources de calcul. Dans le SpinalNet proposé, la structure de la couche cachée est allouée à trois secteurs :

- Rangée d'entrée.
- Rangée intermédiaire.
- Rangée de sortie.

La rangée intermédiaire de SpinalNet contient quelques neurones. La fonction de la segmentation de l'entrée est permettre à chaque couche cachée de recevoir une partie de l'entrée et de la sortie de la couche précédente.

Par conséquent, le nombre de poids entrant dans la couche cachée est significativement plus faible que celui des DNNs (Deep network neurone) traditionnels. Parce que toutes les couches du SpinalNet contribuent directement à la ligne de sortie. Ils ont également étudié la couche entièrement connectée du SpinalNet par rapport à plusieurs modèles DNN bien connus.

modèles DNN bien connus, et ont effectué un apprentissage traditionnel et un apprentissage par transfert.[29]

Bases de données : CIFAR-10, CIFAR-100, STL-10, Fashion-MNIST.

Voici quelques résultats lorsque cette méthode est utilisée dans certains jeux de données :

Tache	Base de donnée	Précision taux de classification
Classification d'image	CIFAR 10	91.98
Classification d'image	CIFAR 100	65.51
Classification d'image	STL 10	98.66
Classification d'image	Fashion-MNIST	94.68

Tableau 7 : SpinalNet sur différentes base de données

3.7. Mingxing and Quoc (2021)

Ce travail présent EfficientNetV2, qui est une nouvelle famille de réseaux convolutifs, qui a une vitesse d'apprentissage plus rapide et une meilleure efficacité des paramètres que les modèles précédents. Afin de développer cette série de modèles, les auteurs ont utilisé une combinaison de recherche et de mise à l'échelle d'architecture neuronale sensible à l'apprentissage et la mise à l'échelle pour optimiser conjointement la vitesse de l'apprentissage et l'efficacité des paramètres.
Ces modèles sont recherchés dans un espace de recherche riche en nouvelles opérations.

Les expériences montrent que la vitesse de formation du modèle EfficientNetV2 est beaucoup plus rapide que celle de l'état de l'art, et peut être réduite jusqu'à 6,8 fois.

La vitesse de formation peut être encore accélérée en augmentant graduellement la taille de l'image pendant l'apprentissage, mais cela entraîne généralement une diminution de la précision. Pour compenser cette diminution, les auteurs suggèrent d'ajuster de manière adaptative la régularisation (abandon et augmentation des données) afin d'obtenir des résultats rapide et une bonne précision.[30]

Tache	Base de donnée	Précision taux de classification
Classification d'image	CIFAR 10	99.10
	CIFAR 100	92.3

Tableau 8 : EfficientNetV2 sur différentes base de données

4. Comparaison entre les travaux ultérieurs

La classification des images apparaît comme un sujet de recherche toujours vivant, qui fait l'objet d'un grand nombre d'œuvres, grâce à ses diverses et potentielles méthodes de classement qui faciliter le processus de classement.

Dans ce qui suit nous citerons les principaux systèmes de classification utilisés, les différentes bases de données et le taux obtenu par chaque méthode de classement tel qu'illustré dans le tableau suivant :

Base de donné	Auteur	Approche utilisé	Année	Taux de classification
CIFAR 10	(Pham et al)	(Pham et al) Sharpness-Aware Minimization for Efficiently Improving Generalization	2021	99.70
	(Dosovitskiy et al)	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale	2020	99.50
	(Dosovitskiy et al)	Going deeper with Image Transformers	2021	99.40
	(Mingxing et Quoc V)	EfficientNetV2: Smaller Models and Faster Training	2020	99.10
	(Harris et al.)	FMix: Enhancing Mixed Sample Data Augmentation	2020	98.64
	(Tanveer et al)	fine-Tuning DARTS for Image Classification	2O2O	97.52
	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	91.98
CIFAR 100	(Pham et al)	(Pham et al) Sharpness-Aware Minimization for Efficiently Improving Generalization	2021	96.08
	(Dosovitskiy et al)	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale	2020	94.55
	(Dosovitskiy et al)	Going deeper with Image Transformers	2021	92.3

	(Mingxing et Quoc V)	EfficientNetV2: Smaller Models and Faster Training	2020	93.10
	(Harris et al.)	FMix: Enhancing Mixed Sample Data Augmentation	2020	83.95
	(Tanveer et al)	fine-Tuning DARTS for Image Classification	2O2O	84.10
	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	65.51
MNIST FASHION	(Tanveer et al)	fine-Tuning DARTS for Image Classification	2O2O	96.91
	(Harris et al.)	FMix: Enhancing Mixed Sample Data Augmentation	2020	96.41
	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	96.36
	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	94.68
MNIST	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	99.72
STL 10	(Dipu & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	98.66

Tableau 9 : Comparaison entre les résultats des travaux ultérieurs sur différentes bases de données.

Conclusion

Nous avons exploré différentes approches pour développer un modèle de classification des images. Nous pouvons encore optimiser davantage notre modèle en modifiant différents hyperparamètres tels que les taux d'apprentissage, le nombre d'époques, etc. CIFAR-10 est un problème résolu, nous pouvons trouver de nombreuses implémentations différentes pour cet ensemble de données en ligne. Ceci n'est qu'une introduction sur le réseau de neurones convolutifs et quelques techniques d'optimisation que nous pouvons mettre en œuvre pour améliorer la précision du modèle.

Chapitre 03

Introduction

De nos jours, nous avons beaucoup de modèles plus connus avec d'excellents résultats qui font des classifications rapides et avec une grande précision. Par conséquent, nous devons utiliser ces outils pour les appliquer dans nos prévisions quotidiennes en nous concentrant sur les objectifs de nos modèles et pas seulement sur leur empreinte. Pour cette raison, nous devons comprendre notre base de données et essayer d'appliquer le bon modèle, en effectuant le prétraitement nécessaire de la base de données et les corrections dans ces fameux modèles si nécessaire.

Dans ce chapitre, on va parler sur la performance de notre méthode avec une précision supérieure à 90% avec Cifar-10 en utilisant l'apprentissage par transfert, on a utilisé VGG16 avec la taille modifiée de l’image entrée.

1- Matériel et méthodes

Dans cette pratique, on a utilisé :

"Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU. [31]

2 -Modèle VGG16 utilisée :

VGG16 est un modèle de réseau neuronal convolutif conçu par K. Simonyan et A. Zisserman [41]. Ce modèle atteint une précision de 92,7 % dans ImageNet, qui regroupe plus de 14 millions d'images appartient 1000 classes.

Durant l'apprentissage du modèle, l'input de la première couche de convolution est une image RGB de taille 224 x 224. Pour toutes les couches de convolution, le noyau de convolution est de taille 3×3: la plus petite dimension pour capturer les notions de haut, bas, gauche/droite et centre. C'était une spécificité du modèle au moment de sa publication. Jusqu'à VGG16 beaucoup de modèles s'orientaient vers des noyaux de convolution de plus grande dimension (de taille 11 ou bien de taille 5 par exemple).

La base de données Cifar 10 contenant 60000 images en couleur (RGB) de 32x32 partagé entre 10 classes, avec 6000 images par classe. Cette base est divisée en deux parties, 50 000 échantillons d'apprentissage et 10 000 échantillons de test.

3. La méthode de texture “oriented Basic Image Features (BIFs)”

Les caractéristiques basées sur **Basic Image Features** (BIF) sont une contribution significative pour extraire les informations de texture dans l’image. Le calcul des BIF consiste à classer chaque pixel d'une image dans l'une des sept catégories en fonction des structures et des symétries locales, créant une esquisse primale de l'image d'entrée. Les BIF sont calculés efficacement sur la base des réponses à une banque de filtres Derivative-of-Gaussian (DtG).

Le calcul des BIF est contrôlé par un paramètre d'échelle (σ, l'écart type des filtres DtG) et un paramètre de seuil (ε) dictant la fraction d'une image qui doit être considérée comme "plate" (c'est-à-dire sans structure particulière).

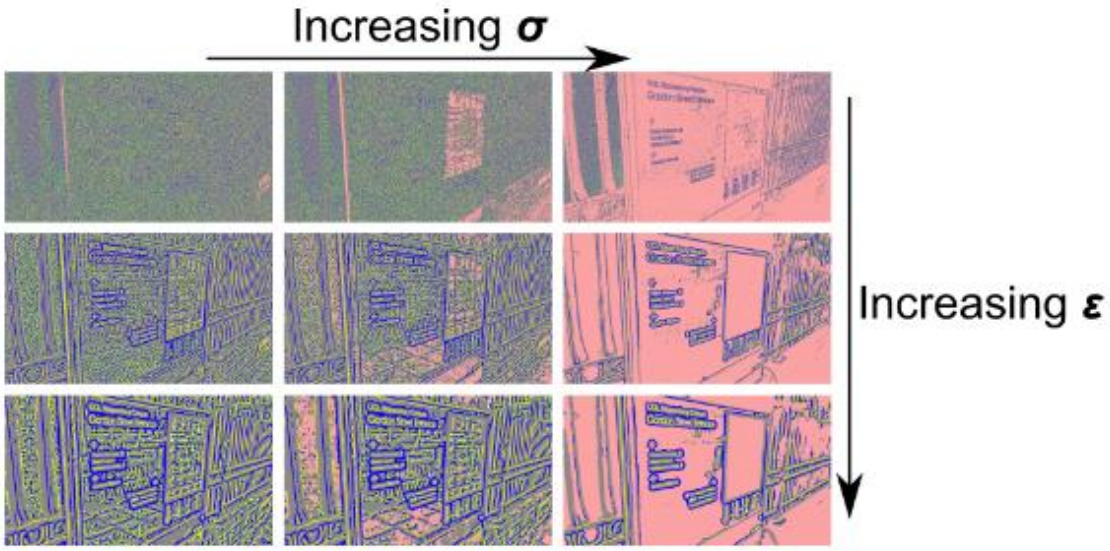

Fig 13 : paramètre d'échelle de BIF

Les BIF peuvent être utilisés pour coder des représentations compactes d'images (par exemple, des histogrammes à 7 cases). Les sept catégories peuvent être étendues à 23 par quantification des caractéristiques non symétriques en rotation. Ces 23 caractéristiques sont appelées oriented Basic Image Features ou oBIF.[32]

4. Système proposé

Pour notre système on a proposé d'utiliser VGG16 en modifiant les paramètres d'entrée « 32x32x3 » au lieu de « 224x224x3 » et taille de kernel est fixé par (3 ,3) et stride (1,1).

La chose unique à propos de VGG16 est qu'au lieu d'avoir un grand nombre d'hyper-paramètres, ils se sont concentrés sur des couches de convolution de filtre 3x3 avec une pas de 1 et ont toujours utilisé le même rembourrage et la même couche maxpool de filtre 1x1 de pas 1. Il suit cet arrangement de couches de convolution et de pool maximum de manière cohérente dans toute l'architecture. Au final, il a deux FC (couches entièrement connectées) suivies d'un Softmax pour la sortie. Le 16 dans VGG16 fait référence à 16 couches qui ont des poids. Ce réseau est un réseau assez vaste et compte environ 138 millions de paramètres (environ).

La structure suivante est présentée le modèle de VGG16 utilisé dans la méthode proposée.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_26 (Conv2D)          (None, 32, 32, 64)        1792

 activation_30 (Activation)  (None, 32, 32, 64)        0

 batch_normalization_28 (Bat  (None, 32, 32, 64)       256
 chNormalization)

 dropout_20 (Dropout)        (None, 32, 32, 64)        0

 conv2d_27 (Conv2D)          (None, 32, 32, 64)        36928

 activation_31 (Activation)  (None, 32, 32, 64)        0

 batch_normalization_29 (Bat  (None, 32, 32, 64)       256
 chNormalization)

 max_pooling2d_10 (MaxPoolin  (None, 16, 16, 64)       0
 g2D)

 conv2d_28 (Conv2D)          (None, 16, 16, 128)       73856

 activation_32 (Activation)  (None, 16, 16, 128)       0

 batch_normalization_30 (Bat  (None, 16, 16, 128)      512
 chNormalization)

 dropout_21 (Dropout)        (None, 16, 16, 128)       0

 conv2d_29 (Conv2D)          (None, 16, 16, 128)       147584
```

```
activation_33 (Activation)  (None, 16, 16, 128)       0

batch_normalization_31 (Bat  (None, 16, 16, 128)      512
chNormalization)

max_pooling2d_11 (MaxPoolin  (None, 8, 8, 128)        0
g2D)

conv2d_30 (Conv2D)          (None, 8, 8, 256)         295168

activation_34 (Activation)  (None, 8, 8, 256)         0

batch_normalization_32 (Bat  (None, 8, 8, 256)        1024
chNormalization)

dropout_22 (Dropout)        (None, 8, 8, 256)         0

conv2d_31 (Conv2D)          (None, 8, 8, 256)         590080

activation_35 (Activation)  (None, 8, 8, 256)         0

batch_normalization_33 (Bat  (None, 8, 8, 256)        1024
chNormalization)

dropout_23 (Dropout)        (None, 8, 8, 256)         0

conv2d_32 (Conv2D)          (None, 8, 8, 256)         590080

activation_36 (Activation)  (None, 8, 8, 256)         0

batch_normalization_34 (Bat  (None, 8, 8, 256)        1024
chNormalization)

max_pooling2d_12 (MaxPoolin  (None, 4, 4, 256)        0
g2D)

conv2d_33 (Conv2D)          (None, 4, 4, 512)         1180160

activation_37 (Activation)  (None, 4, 4, 512)         0

batch_normalization_35 (Bat  (None, 4, 4, 512)        2048
chNormalization)

dropout_24 (Dropout)        (None, 4, 4, 512)         0

conv2d_34 (Conv2D)          (None, 4, 4, 512)         2359808

activation_38 (Activation)  (None, 4, 4, 512)         0

batch_normalization_36 (Bat  (None, 4, 4, 512)        2048
chNormalization)

dropout_25 (Dropout)        (None, 4, 4, 512)         0

conv2d_35 (Conv2D)          (None, 4, 4, 512)         2359808

activation_39 (Activation)  (None, 4, 4, 512)         0
```

```
batch_normalization_37 (Bat  (None, 4, 4, 512)        2048
chNormalization)

max_pooling2d_13 (MaxPoolin  (None, 2, 2, 512)        0
g2D)

conv2d_36 (Conv2D)           (None, 2, 2, 512)        2359808

activation_40 (Activation)   (None, 2, 2, 512)        0

batch_normalization_38 (Bat  (None, 2, 2, 512)        2048
chNormalization)

dropout_26 (Dropout)         (None, 2, 2, 512)        0

conv2d_37 (Conv2D)           (None, 2, 2, 512)        2359808

activation_41 (Activation)   (None, 2, 2, 512)        0

batch_normalization_39 (Bat  (None, 2, 2, 512)        2048
chNormalization)

dropout_27 (Dropout)         (None, 2, 2, 512)        0

conv2d_38 (Conv2D)           (None, 2, 2, 512)        2359808

activation_42 (Activation)   (None, 2, 2, 512)        0

batch_normalization_40 (Bat  (None, 2, 2, 512)        2048
chNormalization)

max_pooling2d_14 (MaxPoolin  (None, 1, 1, 512)        0
g2D)

dropout_28 (Dropout)         (None, 1, 1, 512)        0

flatten_2 (Flatten)          (None, 512)              0

dense_4 (Dense)              (None, 512)              262656

activation_43 (Activation)   (None, 512)              0

batch_normalization_41 (Bat  (None, 512)              2048
chNormalization)

dropout_29 (Dropout)         (None, 512)              0

dense_5 (Dense)              (None, 10)               5130

activation_44 (Activation)   (None, 10)               0

=================================================================
Total params: 15,001,418
Trainable params: 14,991,946
Non-trainable params: 9,472
_________________________________________________________________
```

5- Etude Exprimental

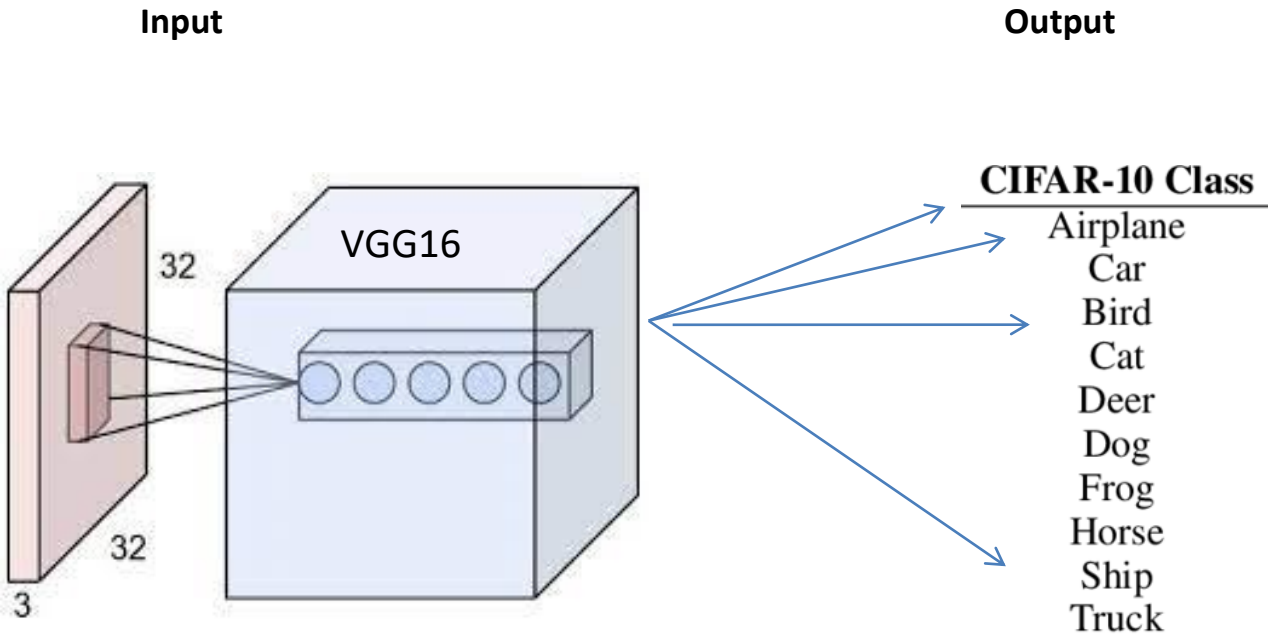

Fig 14 : Model proposé VGG16 en modifiant l'image d'entrée de taille 32x32x3

Dans cette étude expérimentale, nous proposons deux protocoles d'évaluation :

- **Premier protocole :** On a employé les images RGB de CIFAR 10 dans le Deep VGG 16. On a utilisé les images de CIFAR 10 en couleur comme des entrées de taille 32x32x3 .

- **Deuxième protocole :** On a proposé d'utiliser oBIF en trois dimensions avec VGG16. Pour chaque d'image CIFAR10 en couleur RGB, on a créé une nouvelle image de texture basé sur oBIF avec $\sigma = 1$ et $\varepsilon = 0.1$ correspondante pour chaque type de couleur R(Rouge), G(Vert) et B(Blue). On a superposé les trois types d'image de texture afin de réaliser une image de trois dimensions basées sur l'oBIF. Ensuite, on augmente la base de l'apprentissage avec les images en oBIF pour obtenir la taille doublée de la base.

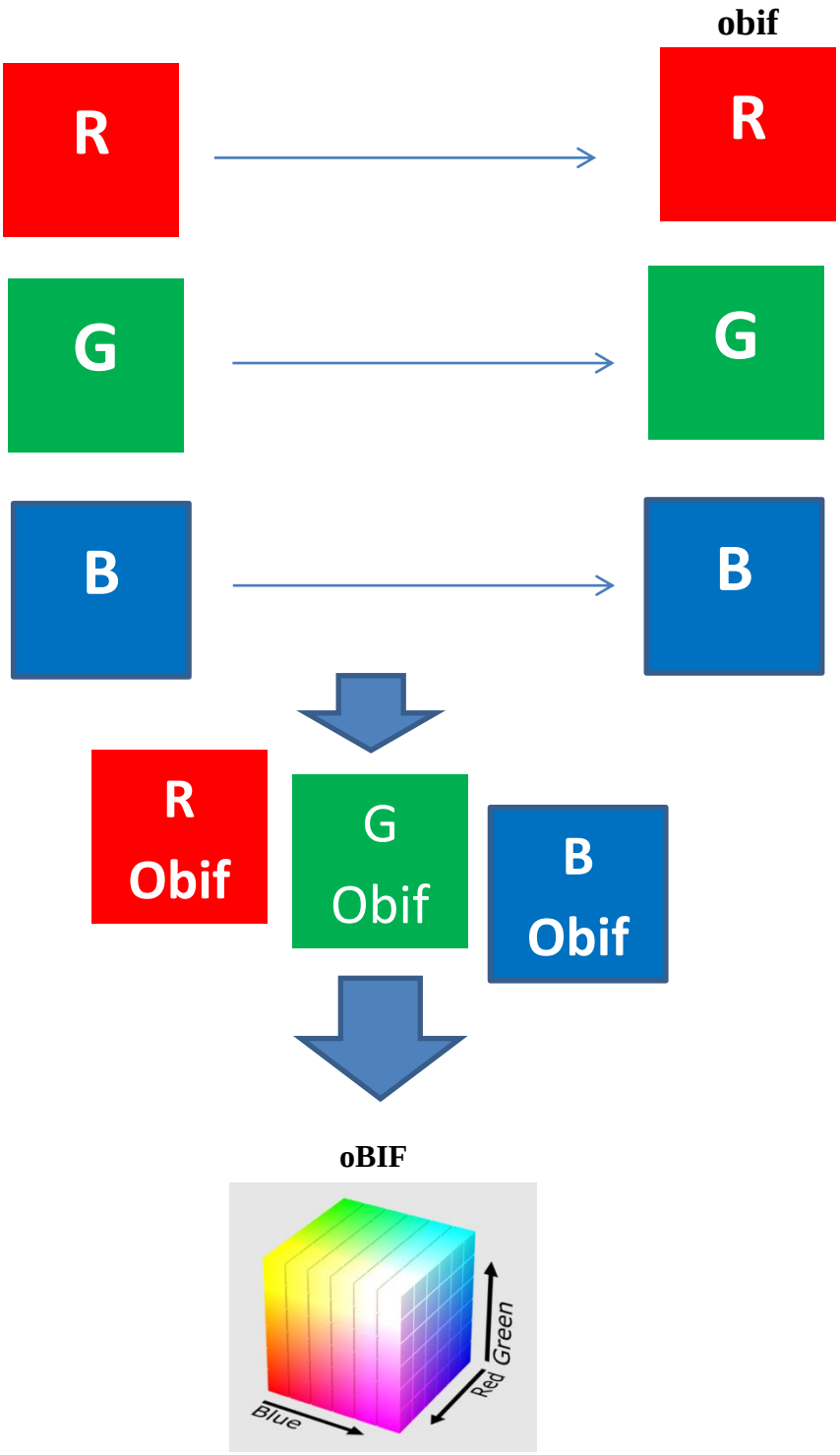

Fig 15 :Schémas générale du modéle

Protocole	Base de donné	Taux de classification %
1er protocole	CIFAR 10	91.15
2eme protocole	CIFAR 10	91.75

Tableau 10 : les résultats obtenu du modèle proposé

Discutions et résultat

Dans tous les cas, les deux protocoles ont pu apprendre la base de données de l'apprentissage. C'est un bon signe, car cela montre que le problème est apprenable et que les deux protocoles ont une capacité suffisante pour apprendre le problème. Les résultats du deuxième protocole sur la base de test ont montré une amélioration du taux de la classification. Il est possible que la tendance vers data-augmentation en utilisant les images en oBIF s'améliore la performance de notre système. D'un autre côté, les résultats des deuxièmes protocoles ont été excellents, en particulier pour CIFAR-10. Par conséquences, les nouvelles images en oBIF sont considérées comme des informations complémentaires des images originales.

Conclusion

Dans ce chapitre, nous avons présenté les outils, l'environnement de développement et la méthode proposée dans ce travail et les résultats. Tout ça a été consacrée principalement à l'évaluation des méthodes proposées pour l'image classification pour la base de données CIFAR 10. L'idée de ce travail est augmentée la base de l'apprentissage avec les images en oBIF pour obtenir la taille doublée de la base, ces nouvelles images sont considérées comme des informations complémentaire des image originales.

Le but principal de ce mémoire est de nous aider d'améliorer le système en utilisant la phase data-augmentation avec des images de texture en oBIF ainsi que le meilleur modèle de VGG16 qui donne un taux élevé.

Conclusion Générale

Nous rappelons que l‘objectif de notre travail est l‘étude et la classification d’mage basée sur l‘apprentissage profond, de ce fait nous avons proposé dans ce mémoire une solution basée, comme demandé, sur l‘apprentissage profond comprenant deux modules principaux. Dans la première partie de notre travail, nous avons proposé le modél VGG16 avec la base de donnée CIFAR10 en Modifiant les paramétre d’entréé « 32X32X3 ». Le deuxième objectif visé est de créés des images texture basé sur l’obif a partir dela base cifar10 entrainé avec le modèle vgg16. Elles ont donné des résultats prometteurs et de très bonne qualité, donc nous pouvons qualifier la fiabilité et l‘efficacité de notre approche comme très satisfaisante.

Le travail sur notre projet nous a permis d‘approfondir notre connaissance dans le domaine de l‘apprentissage profond, il nous a permis, aussi, de connaître et expérimenter de nouvelles bibliothèques python dédiées à ce domaine d‘avant-garde et de savoir le mode de fonctionnement de plusieurs algorithmes de l‘apprentissage automatique tels que les différentes architectures du deep learning.

Bien que les objectifs visés, au préalable, ont été atteints, mais il reste toujours des perspectives et des améliorations possibles qui peuvent encore être réalisés dans le future, telles que :

• L‘amélioration de ce travail en utilisant d‘autres méthodes du monde du deep learning et faire une étude comparative entre ces dernières au niveau des résultats, performance et rapidité.

• Penser à la création d'un modèle de réseau en temps réel sans avoir besoin d‘utiliser l‘ensemble de données CIFAR ou autre.

Liste de Référence

[1] ZACCONE Giancarlo, MD REZAUL Karim, MENSHAWY Ahmed. Deep learning with tensorflow. 2017

[2] Y. Bengio et al., "Learning deep architectures for ai," Foundations and trends R in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.

[3] PARIZEAU Marc. Réseaux de neurones. 2004

[4] ZACCONE Giancarlo, MD REZAUL Karim, MENSHAWY Ahmed. Deep learning with tensorflow. 2017.

[5]http://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/[Consulté le 30 Avril 2022]..

[6] http://yann.lecun.com/exdb/mnist/ [Consulté le 30 Avril 2022].

[7] http://www.cs.toronto.edu/~kriz/cifar.html [Consulté le 30 Avril 2022].

[8] https://cs.stanford.edu/~acoates/stl10/ [Consulté le 30 Avril 2022].

[9] https://www.kaggle.com/datasets/zalando-research/fashionmnist [Consulté le 15 Mai 2022].

[10] http://host.robots.ox.ac.uk/pascal/VOC/index.html [Consulté le 30 Mai 2022].

[11] https://cocodataset.org/#home [Consulté le 30 Mai 2022].

[12] O'SHEA Keiron, NASH Ryan. An introduction to convolutional neural networks. 2015

[13] WANG Peng, XU Jiaming, XU Bo, LIU Chenglin, ZHANG Heng, WANG Fangyuan HAO Hongwei. Semantic clustering and convolutional neural networks for short text categorization. 2015

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, pp. 1097–1105, 2012.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[17] NAKAMOTO Pat. Neural networks and deep learning : deep learning explained to your granny a visual introduction for beginners who want to make their own deep learning neural network. 2017

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

[20] CHOLLET Francois. Deep learning with python. 2017

[21] A. Kölsch, M. Z. Afzal, M. Ebbecke et M. Liwicki, «Real-time document image classification using deep CNN and extreme learning machines,» arXiv preprint arXiv:1711.05862, 2017.

[22] NAKAMOTO Pat. Neural networks and deep learning : deep learning explained to your granny a visual introduction for beginners who want to make their own deep learning neural network. 2017

[23] CIRESAN Dan C, MEIER Ueli, MASCI Jonathan, MARIA GAMBARDELLA Luca, SCHIDHUBER Jurden. Flexible, high performance convolutional neural networks for image classification. 2011

[24] Sharpness-aware minimization for efficiently improving generalization P Foret, A Kleiner, H Mobahi, B Neyshabur – 2020arXiv201001412F Published as a conference paper at ICLR 2021

[25] Fine-Tuning DARTS for Image Classification (Tanveer & al., arXiv:2006.09042v1 [cs.CV] 16 Jun 20203

[26] Going deeper with Image Transformers Hugo et al.2021

[27] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale 2020 (Dosovitskiy et al., 2020).

[28] FMix: Enhancing Mixed Sample Data Augmentation (Harris & al., 2020).

[29] SpinalNet: Deep Neural Network with Gradual Input (Dipu K H M & al., 2020).

[30] EfficientNetV2: Smaller Models and Faster Training (Mingxing & Quoc, 2021).

[31] https://research.google.com/colaboratory [Consulté le 30 Avril 2022].

[32] Crosier M., and Griffin L. D. (2010). *Using Basic Image Features for Texture Classification*. International Journal of Computer Vision, 88(3), 447-460

Résumé

Une image numérique est composée d'unîtes élémentaires appelées· pixels, contenant chacune différentes informations (intensité, lumineuse, couleur, etc.).

L’arrivée de ces images constitue un véritable atout dans beaucoup de domaines scientifiques. En effet, les images numériques apportent un support de représentation permettant de modéliser et de véhiculer beaucoup d'informations, les rendant ainsi accessibles a l'interprétation et ce, pour une utilisation optimale dans le domaine socio-économique. L'analyse d’une 'image consiste souvent à extraire un certain nombre de propriétés caractéristiques et à les exprimer sous forme paramétrique.

La texture joue un rôle très important dans l'étape de l’identification et de l’extraction des informations thématiques contenues dans une image et il semble que l’avenir de la maîtrise de l’image réside dans la compréhension plus approfondie de sa texture, donc dans sa définition plus complète et cohérente. Dans l'analyse des textures, on trouve une méthode théoriquement simple mais très puissante avec un opérateur appelé **oriented Basic Image Features (OBIFs)**. L’opérateur de texture **(OBIFs)** est devenu une approche populaire dans diverses applications. Grâce à ses récentes extensions, il a été transformé en une très puissante mesure de la texture de l'image. Ce travail concerne l’étude de différentes méthodes de traitements d’images et d’analyse de texture et essentiellement l’application de l’opérateur **oriented Basic Image Features (OBIFs)**.

Abstract

A digital image is made up of elementary units called • pixels, each containing different information (intensity, light, color, etc.).

The arrival of these images is a real asset in many scientific fields. Indeed, digital images provide a support of representation making it possible to model and convey a lot of information, thus making them accessible to interpretation and this, for an optimal use in the socio-economic field. The analysis of an image often consists of extracting a number of characteristic properties and expressing them in parametric form.

Texture plays a very important role in the stage of identifying and extracting the thematic information contained in an image and it seems that the future of mastering the image lies in the deeper understanding of its texture. , therefore in its most complete and coherent definition. In texture analysis, there is a theoretically simple but very powerful method with an operator called oriented Basic Image Features (OBIFs). The texture operator (OBIFs) has become a popular approach in various applications. Thanks to its recent extensions, it has been transformed into a very powerful measure of image texture. This work concerns the study of different methods of image processing and texture analysis and essentially the application of the operator oriented Basic Image Features (OBIFs).

ملخص

تتكون الصورة الرقمية من وحدات أولية تسمى وحدات البكسل ، كل منها يحتوي على معلومات مختلفة (شدة ، ضوء ، لون ، إلخ).

تنوع هذه الصور هو رصيد حقيقي في العديد من المجالات العلمية ، توفر الصور الرقمية دعمًا للتمثيل مما يجعل من الممكن نمذجة ونقل الكثير من المعلومات ، مما يجعلها في متناول التفسير وهذا ، من أجل الاستخدام الأمثل في المجال الاجتماعي والاقتصادي. غالبًا ما يتكون تحليل الصورة من استخراج عدد من الخصائص المميزة والتعبير عنها في شكل حدودي.

يلعب النسيج دورًا مهمًا للغاية في مرحلة تحديد واستخراج المعلومات الموضوعية الموجودة في الصورة ، ويبدو أن مستقبل إتقان الصورة يكمن في الفهم الأعمق لنسيجها. ، لذلك في تعريفه الأكثر اكتمالا وتماسكًا. في تحليل النسيج ، هناك طريقة بسيطة من الناحية النظرية لكنها قوية جدًا مع عامل يسمى ميزات الصورة الأساسية الموجهة (OBIFs). أصبح عامل النسيج (OBIFs) أسلوبًا شائعًا في العديد من التطبيقات. بفضل امتداداته الأخيرة ، تم تحويله إلى مقياس قوي جدًا لملمس الصورة. يتعلق هذا العمل بدراسة الطرق المختلفة لمعالجة الصور وتحليل النسيج وأساسًا تطبيق ميزات الصورة الأساسية الموجهة نحو المشغل (OBIFs).