

Projet de module Réseau informatique

Report de Projet | 2022

Deep Learning et optimisation des transferts des images ou stream

Réalisé par :

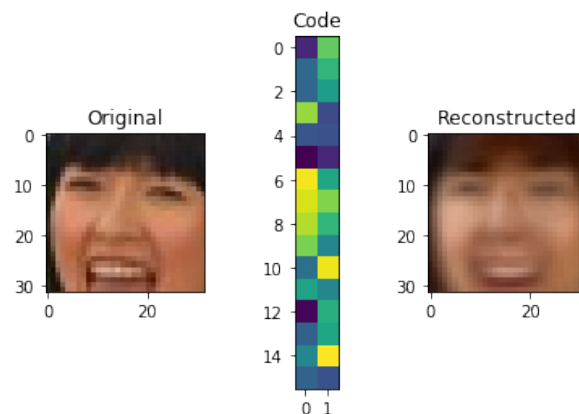
Othmane ELAZRI
Meryem MOHIDDINE
Anass EL HALLANI
Younes EL BELGHITI

Encadrée par :

Pr. Abdelladim HADIOUI

Résumé

Dans ce projet nous avons ces deux modelés avec le deep learning en utilisons des couches de convolution pour implémenter un code qui capable de compresser les images avant de l'envoyer au serveur pour gagner le temps de transmission des images et gagner aussi dans espace de stockage. Pour cela nous avons cherché une dataset avec 500 images pour entrainer notre modèle. Après nous avons implémenté notre modèle pour l'entrainer. Et avant de lancer notre entrainement nous avons préparé notre data pour extraire de l'information depuis les images. En fin nous avons utiliser un serveur local pour visualiser les résultats de cette compression et leur influence sur le temps de transmission et la taille ou cette image et stocker. Donc après qu'on a test, on a trouvé que les images prennent 5 fois moins de temps pour le transmettre et 75% mois d'espace.



keyword : machine learning,deep learning,transfert d'image,cloud computing, Python ,intelligence artificielle,reséau informatique

Sommaire

1	Introduction	6
2	Les technologies modernes en l'intelligence artificielle (AI) et machine learning (ML), Deep Learning (DL)	8
2.1	Intelligence artificielle (IA) :	8
2.2	machine learning (ML)	9
2.3	deep learning (DL)	9
2.3.1	Fonctionnement du deep Learning	9
2.4	convolutional neural networks	10
2.4.1	Définition :	10
2.4.2	Element principal :	10
2.4.3	Pourquoi CNN ?	11
2.5	Conclusion	12
3	Transfer des images	13
3.1	serveur	13
3.1.1	Définition	13
3.1.2	Fonctionnement	13
3.1.3	Types de serveurs	14
3.2	cloud	15
3.2.1	Définition	15
3.2.2	Principe	15
3.3	les protocole de communication	16
3.3.1	Définition	16
3.3.2	Les protocoles de communication	16
3.3.3	Types de protocoles de communication	17
3.4	Conclusion	18
4	Outils et techniques	19
4.1	Réseaux de neurones convolutifs	19
4.1.1	Réseaux de neurones convolutionnels ponctuels	20
4.1.2	Réseaux de neurones convolutifs en profondeur	20

4.2	Mesures de qualité d'image	21
4.3	python	21
4.3.1	Utilisation	22
4.3.2	Mots-clés du langage	22
4.4	Tensorflow	22
4.5	OpenCV	23
4.5.1	Fonctionnalités	23
4.6	Google Colab	23
5	Experimentation et discussions des résultats	25
5.1	Étapes d'Algorithme	25
5.1.1	Preparation des données	25
5.1.2	Création de modele	26
5.1.3	Entrainement de model	26
5.2	Resultat	27
5.3	Test	27
5.4	Conclusion	28

Table des figures

2.1	IA, ML, deep Learnnig et Data Sience	9
2.2	Fonctionnement du deep Learning	10
2.3	Comparison between image classification, object detection and instance segmentation.	10
2.4	Classification ImageNet avec des CNN profonde	11
3.1	Le modèle client-serveur	14
3.2	les protocoles de communication	17
4.1	Architecture Alexnet [2]	20
4.2	CNN	21
4.3	Python	22
4.4	Tensorflow	23
4.5	OpenCV	23
4.6	Python	24
5.1	Train images	25
5.2	Model de compression	26
5.3	Model de decompression	26
5.4	Train images	26
5.5	Loss compression	27
5.6	Loss decompression	27
5.7	Resultat de test	27
5.8	Resultat de test (qualite)	28

Chapitre 1

Introduction

Aujourd'hui, la plupart de la population des pays développés possède un téléphone mobile si bien que le nombre d'utilisateurs de téléphones mobiles a dépassé les 5 milliards en 2019. De plus, avec le développement des réseaux 3G/4G/5G il est devenu possible accès d'internet donc une augmentation de données images/vidéos. En revanche, les opérateurs mobiles limitent la bande passante, ce qui entraîne une limitation de la transmission des données.

Etant donné que les images représentent une des sources majeures des activités de stockage et de transmission mobile, ce type de comportement induit une forte consommation de ressources. De plus, les téléphones mobiles dépassent massivement, en termes d'utilisation quotidienne, tout type d'appareils photos. Une des raisons de ce phénomène est liée notamment à possibilité de partager instantanément les images acquises sur des réseaux sociaux et autres plates-formes, ainsi que de les stocker, parfois même simultanément, sur des clouds personnels.

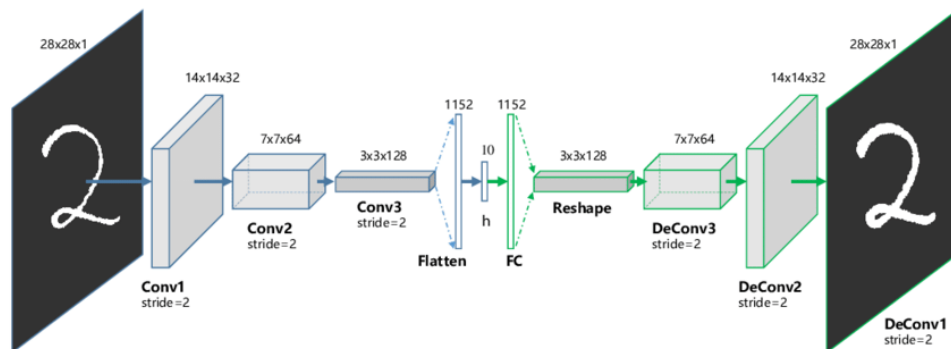
Nous sommes donc ici face à un problème à deux faces. D'un côté, nous nous retrouvons dans le cadre d'un besoin de compression d'images numériques, pour réduire le temps de transmission.

De l'autre côté, pour objectif de réduire la consommation de stockage (et par conséquent, énergétique) due aux utilisations de ces images, tout en garantissant une qualité adéquate au regard des différentes applications.

La solution sont totalement applicables à ces différentes applications, notamment grâce à une utilisation adaptée d'algorithmes d'apprentissage par réseaux de neurones profonds. Le principe consiste à effectuer les processus de compression et de décompression à l'aide d'un même réseau.

Durant ces dernières années, l'apprentissage profond, que nous nommons dans ce manuscrit par sa dénomination anglophone « deep learning » (DL), a connu un développement fulgurant grâce à la disponibilité croissante de puissantes cartes graphiques (GPU) grand public, permettant de faire fonctionner efficacement et dans un temps raisonnable, de réseaux complexes de neurones profonds. En associant l'efficacité de ces réseaux de neurones et l'abondance de très grandes bases de données publiques d'images

d'une grande variété, des progrès significatifs ont pu être réalisés dans les domaines du traitement et de l'analyse d'images en général, et plus particulièrement dans le champ de la compression d'image.



Notre project est organisée de manière suivante. Tout d'abord, nous commençons par une introduction pour montrer la vue générale de notre projet, puis nous allons exposer quelques rappels et définitions et informations sur le domaine de l'intelligence artificielle et ses différents aspects et techniques.

dans le chapitre III nous présentons une étude générale sur le serveur et le cloud et les différents protocoles de communication

Ensuite, le chapitre IV explique tous divers concepts et détails de mise en œuvre sur le projet.

Et en fin, nous allons nous intéresser à la réalisation de Notre projet avec langage python.

Chapitre 2

Les technologies modernes en l'intelligence artificielle (AI) et machine learning (ML), Deep Learning (DL)

Dans ce chapitre nous présentons une étude générale sur le domaine de l'intelligence artificielle et ses différents aspects et techniques.

2.1 Intelligence artificielle (IA) :

L'intelligence artificielle (IA) simplement imite l'intelligence humaine et effectue des défis à améliorer les systèmes ou les machines selon les informations collectées basant sur l'iteration. L'intelligence artificielle se compose sous plusieurs formes. Voici quelques exemples. Chatbot peut utiliser des problèmes de contact plus rapides et plus efficaces. Smart Assists Utilisez des informations clés d'analyse AI à partir de gros enregistrements de texte ouverts afin d'améliorer les plans. La machine recommandée peut recommander automatiquement un programme de télévision basé sur les habitudes du spectateur. L'IA se concentre sur les processus et les capacités de réflexion. Il y a autant d'analyses de données détaillées que de formats et de fonctionnalités spécifiques.

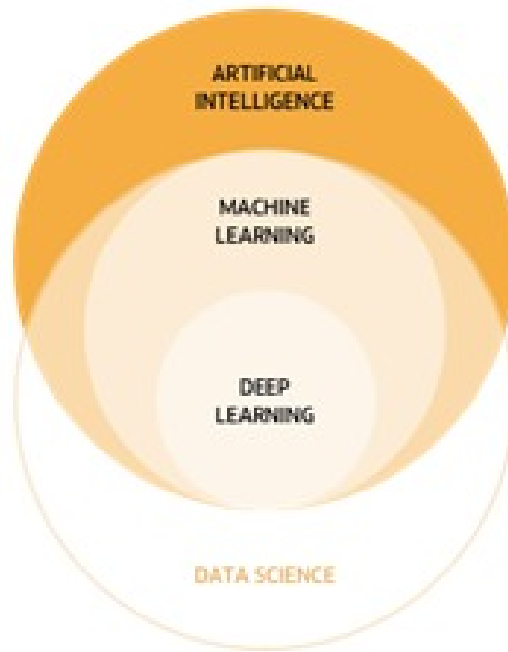


FIGURE 2.1 – IA, ML, deep Learning et Data Science

2.2 machine learning (ML)

2.3 deep learning (DL)

Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

2.3.1 Fonctionnement du deep Learning

Les réseaux de deep learning sont entraînés sur la base de structures complexes de données auxquelles ils sont confrontés. Ils élaborent des modèles de calcul composés de plusieurs couches de traitement pour créer plusieurs niveaux d'abstraction afin de représenter les données. [?].

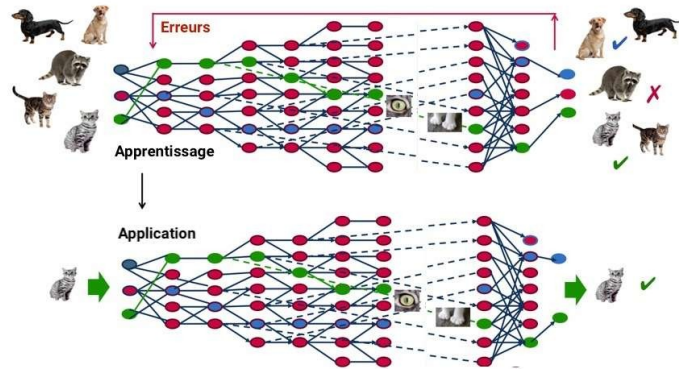


FIGURE 2.2 – Fonctionnement du deep Learning

2.4 convolutional neural networks

2.4.1 Definition :

Les réseaux de neurones convolutifs (CNN) sont de l'intelligence artificielle algorithmes basés sur des réseaux de neurones multicouches qui apprennent caractéristiques des images, étant capable d'effectuer plusieurs tâches comme classification, détection et segmentation d'objets.[?]

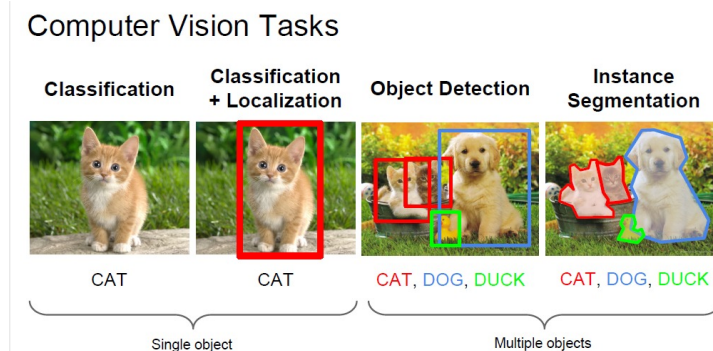


FIGURE 2.3 – Comparison between image classification, object detection and instance segmentation.

2.4.2 Element principal :

Pour bien comprendre les réseaux de neurones convolutifs, il est important de connaître les bases des réseaux de neurones tout court Les éléments principaux à retenir sont les suivants : Un CNN est topiquement composé de quatre types de couches : Convolutional, Pooling ,Relu et Fully Connected

1- Une couche convolutive : Une couche convolutive est composée d'un ensemble de filtres, également appelés noyaux, qui glisse sur les données d'entrée. Chaque noyau a

une largeur, une hauteur et des poids largeur \times hauteur utilisé pour extraire les caractéristiques des données d'entrée. Dans l'étape d'apprentissage, les poids dans le noyau commencent par aléatoires valeurs, et apprendra sur la base de l'ensemble de formation.

2-La couche RELU :

Les couches ReLU (Rectified Linear Units) sont une couche d'activation lié après une couche convolutive pour générer une non-linéarité dans le réseau. Le ReLU aide le réseau à apprendre des fonctions de décision plus difficiles et réduire le surajustement.

3-Pooling : The pooling layer, or down-sampling layer, est appliquée pour réduire la dimensionnalité des cartes d'entités de manière à enregistrer les informations les plus pertinentes des cartes d'entités. Dans la couche de regroupement, un filtre glisse sur les données d'entrée et applique l'opération de regroupement (max, min, moy). Le max pooling est le plus utilisé dans la littérature.

4- Fully Connected Layer : Un CNN est topiquement divisé en deux parties : la convolutionnelle et la étapes denses. Le premier apprend les meilleures caractéristiques à extraire du images et ce dernier apprend à classer les caractéristiques dans différents categories.

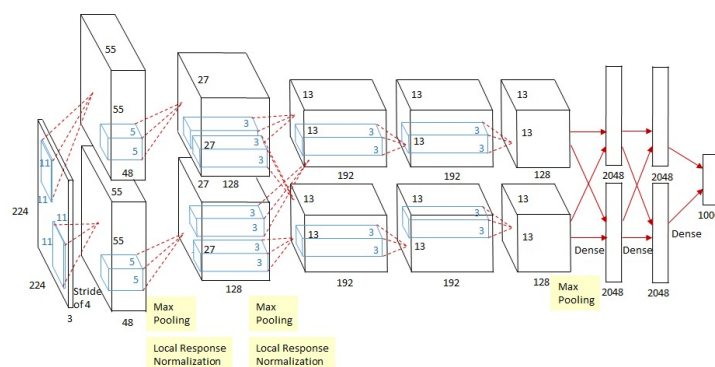


FIGURE 2.4 – Classification ImageNet avec des CNN profonde

La couche entièrement connectée est un Perceptron multicouche (MLP), composé de trois types de couches : couches d'entrée, masquées et de sortie.

2.4.3 Pourquoi CNN ?

L'avantage des CNN par rapport aux autres algorithmes de classification (SVM, K-NN, Random-Forest et autres) est que les CNN apprennent le mieux caractéristiques pour représenter les objets dans les images et a un haut capacité de généralisation, être capable de classer précisément des nouveaux objets.

2.5 Conclusion

Dans ce chapitre nous avons présenté une vue générale sur le domaine de deep learning qui fait partie d'intelligence artificielle et aussi l'algorithme du CNN qui connu par la représentation des objets dans les images et a un haut capacité de généralisation et de precision,qui va nous aidons de bien genere notre modele.

Chapitre 3

Transfer des images

Dans ce chapitre nous présentons une étude générale sur le serveur et le cloud et les différents protocoles de communication.

3.1 serveur

3.1.1 Définition

Le terme de « serveur » possède deux significations en informatique. On qualifie par serveur non seulement l'ordinateur qui fournit les ressources d'un réseau informatique, mais aussi le programme fonctionnant sur cet ordinateur. Nous vous donnons les deux définitions d'un « serveur » ci-dessous :

- **définition Serveur (Hardware)** : un serveur matériel (hardware) est un réseau d'ordinateurs reliés par une machine physique et sur lequel fonctionnent un ou plusieurs serveurs logiciels (software). Une alternative au terme de serveur (Hardware) est « hébergeur » (Host en anglais). En principe chaque ordinateur est utilisé avec un serveur logiciel.
- **définition Serveur (Software)** : un logiciel serveur est un programme effectuant des interactions en réseau avec d'autres programmes appelés logiciels clients. Le service apporté dépend du type de logiciel serveur. La base de la communication en réseau est cette relation Client-serveur. Lors de l'échange de données, différents protocoles de transmission entrent en jeu.

3.1.2 Fonctionnement

La mise à disposition de services de serveurs sur un réseau s'effectue sur le modèle client-serveur. Ce concept permet de répartir les tâches sur différents ordinateurs et les rendre accessibles à plusieurs utilisateurs finaux, de manière indépendante. Chaque service disponible sur un réseau est administré par un logiciel serveur. Il s'agit de la seule manière

d'assurer un accès permanent au serveur et selon les besoins pour des clients tels que des navigateurs Web ou des programmes de messagerie électronique.

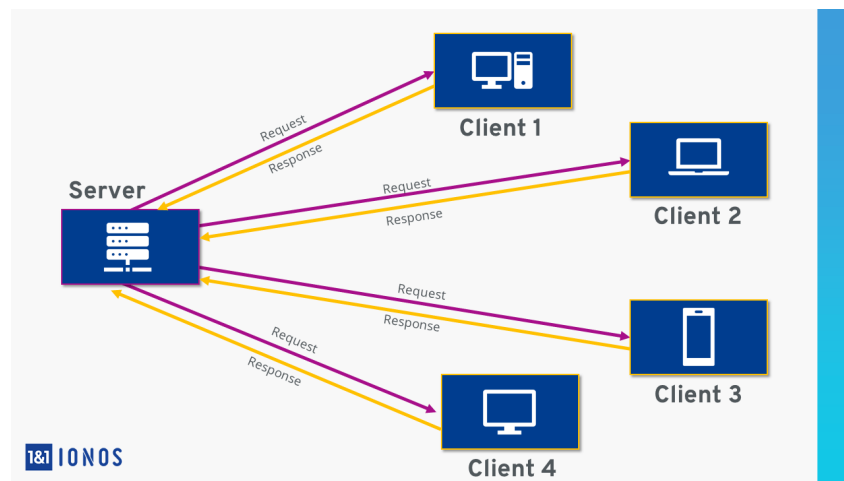


FIGURE 3.1 – Le modèle client-serveur

3.1.3 Types de serveurs

Le mode de communication utilisé entre client et serveur est déterminé par le protocole de communication. Les deux types de serveurs courants définis ci-dessous permettront d'éclaircir ce principe :

- **Serveur Web** : La tâche primaire d'un serveur Web est de stocker les pages Web, de les traiter et les livrer à des clients comme un navigateur ou à un robot d'indexation (Crawler) d'un moteur de recherche. La communication entre le serveur (software) et le client s'établit grâce au protocole HTTP (Hypertext Transfer Protocol) ou sa variante sécurisée HTTPS. En général les images, feuilles de style ou script sont transmis via des documents HTML. Les serveurs Web courants sont des serveurs http Apache, Microsoft Internet information Services (IIS) ou Nginx.
- **Serveur de fichiers (File-Server)** : un serveur de fichiers sert à l'enregistrement central de fichiers accessibles depuis différents clients sur un réseau. Les entreprises misent sur de tels systèmes de gestion de fichiers dans le but que plus de groupes de travail puissent accéder aux mêmes fichiers.
- **Serveur email** : un serveur mail est composé de plusieurs modules logiciels dont l'assemblage permet la réception et l'expédition de courriers électroniques. En général, le protocole utilisé est le Simple Mail Transfer Protocol (SMTP). Les utilisateurs qui voudraient accéder à un serveur email ont besoin d'un client email dont le but est de récupérer les messages du serveur et de les expédier dans la boîte de réception électronique. Ce chargement passe par le protocole IMAP (Internet Message Access Protocol) ou POP (Post Office Protocol).
- **Serveur de bases de données** : on définit un serveur de bases de données comme un programme informatique permettant à d'autres programmes l'accès en

réseau à un ou plusieurs systèmes de bases de données. Les logiciels de ce type ayant les plus grandes parts de marché sont Oracle, MySQL, Microsoft SQL Server, PostgreSQL et DB2. Les serveurs de bases de données sont en général compatibles avec les serveurs Web pour l'enregistrement et l'expédition de données.

3.2 cloud

3.2.1 Définition

Le cloud computing, en français l'informatique en nuageau , correspond à l'accès à des services informatiques (serveurs, stockage, mise en réseau, logiciels) via Internet (le « cloud » ou « nuage ») à partir d'un fournisseur. On distingue généralement trois types de cloud : le cloud public est accessible par Internet , le cloud d'entreprise ou privé est accessible uniquement sur un réseau privé , le cloud intermédiaire ou hybride qui est une combinaison entre le cloud public et le cloud privé. Le cloud computing permet aux entreprises de minimiser les coûts d'infrastructure informatique et de bénéficier d'une adaptation des ressources en fonction des fluctuations de l'usage, avec cependant une augmentation notable des coûts d'exploitation du réseau.

Les grandes entreprises du secteur des technologies de l'information et des télécoms développent massivement le cloud computing, allant des services nuagiques cités ci-dessus à la location de logiciel (application de téléphonie mobile, logiciels de bureautique, de dessin, publication, etc.), mais aussi de contenu, comme la vidéo à la demande (VAD). Les technologies du cloud computing ont permis la mise au point de solutions de mégadonnées et de l'informatique ubiquitaire. L'internet des objets (IdO) et la 5G s'appuient sur des infrastructures nuagiques.

3.2.2 Principe

Un cloud (« nuage ») est un ensemble de matériels, de raccordements réseau et de logiciels⁹ fournissant des services qu'individus et collectivités peuvent exploiter depuis n'importe où dans le monde¹⁰. Un basculement de tendance fait qu'au lieu d'obtenir de la puissance de calcul par acquisition de matériel et de logiciel, le consommateur se sert de puissance mise à sa disposition par un fournisseur via l'Internet.

Un nuage est caractérisé par plusieurs principes récurrents :

- ressources en libre-service avec adaptation automatique à la demande de la capacité de stockage et puissance de calcul, selon le besoin du consommateur, tant qu'il peut payer. Ceci contraste avec la technique classique des hébergeurs où une demande écrite doit être faite au fournisseur pour obtenir une augmentation de capacité demandant en outre un certain temps. En cloud computing la demande est automatique et la réponse immédiate ;
- ouverture : les services de cloud computing sont accessibles via l'Internet, via

des techniques standardisées, tant pour un ordinateur qu'un téléphone ou une tablette ;

- mutualisation : elle permet de combiner des ressources hétérogènes (matériel, logiciel, trafic réseau) pour servir plusieurs consommateurs à qui les ressources sont automatiquement attribuées¹¹. La mutualisation améliore l'évolutivité et l'élasticité ; elle facilite l'adaptation automatique des ressources aux variations de la demande ;
- paiement à l'usage : la quantité de service consommée dans le cloud est mesurée, à des fins de contrôle, d'adaptation des moyens techniques et de facturation .

Le nuage (public, privé ou communautaire) bénéficie de technologies telles que la virtualisation du matériel informatique, les grilles, l'architecture orientée services et les services web. Un nuage public est mis à disposition du grand public. Les services sont généralement mis à disposition par une entreprise utilisant une infrastructure lui appartenant⁵. Un nuage privé est destiné exclusivement à une organisation qui peut le manipuler elle-même ou faire appel à des services fournis par des tiers. Un nuage communautaire utilise une infrastructure provenant d'un ensemble de membres partageant un intérêt commun, comme dans le cas des milieux universitaires pour des études de grande envergure .

3.3 les protocole de communication

3.3.1 Définition

Un protocole de communication est compris comme le système de règles d'un système de communication qui permet la transmission d'informations entre deux entités ou plus.

3.3.2 Les protocoles de communication

Les protocoles de communication, dans télécommunications et informatique, sont définis comme un système de règles par lequel deux entités ou plus faisant partie d'un système de communication sont autorisées, ne peut que communiquer mutuellement, pour transmettre des informations à travers toute sorte de variation d'une grandeur physique.

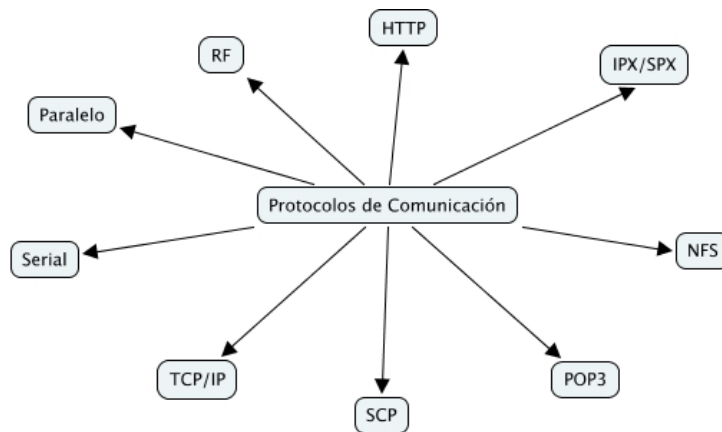


FIGURE 3.2 – les protocoles de communication

Ainsi, le protocole de communication comprend les règles ou un standard avec lesquels la syntaxe, la sémantique et la synchronisation de la communication sont définies, ainsi que les méthodes possibles de reprise sur incident. Enfin, un protocole peut être mis en œuvre par matériel informatique ou logiciel et si vous voulez, dans une combinaison des deux.

3.3.3 Types de protocoles de communication

Le protocole de communication peut être compris comme un ensemble de règles formelles, qui sont respectées pour permettre la communication entre des périphériques. Les types sont :

- TCP / IP Protocole de base défini pour la communication de les réseaux. Avec votre aide, il existe une transmission d'informations entre des ordinateurs appartenant à un réseau. Comme plusieurs ordinateurs d'un réseau, vous pouvez communiquer avec d'autres ordinateurs et ce réseau virtuel est appelé Internet.
- TCP ou protocole de contrôle de transmission Il est axé sur les communications et la transmission de données est fiable. Il est responsable de l'assemblage des données provenant de charges supérieures aux packages standard.
- HTTP (protocole de transfert hypertexte) Il permet de récupérer des informations et d'effectuer des recherches indexées permettant des sauts intertextuels de manière efficace. Vous pouvez transférer des textes dans plusieurs formats.
- FTP (protocole de transfert de fichier) Il est utilisé pour les transferts de fichiers à distance.
- SSH (Secure Shell) Il a été développé pour améliorer la sécurité de la communication Internet. Élimine l'envoi de mots de passe non cryptés et les informations sont toujours cryptées.
- UDP (User Datagram Protocol) Il est destiné aux communications hors ligne et ne dispose pas de mécanismes pour transmettre des datagrammes.

- SNMP (Simple Network Management Protocol) Il utilise UDP pour le transport des données et utilise différents termes de TCP / IP en tant qu'administrateurs et agents, au lieu de clients et serveurs.
- TFTP (Trivial File Transfer Protocol) Pour les transferts, c'est très simple et simple. Il n'y a pas de sécurité.
- SMTP (protocole de transfert de courrier simple) Il se compose de plusieurs règles qui régiront le format et le transfert des données lors de l'envoi de courriels.
- ARP (protocole de résolution d'adresse) Les tâches sont réalisées qui cherchent l'association d'un périphérique IP, qui est identifié avec une adresse IP avec un périphérique réseau, qui a une adresse réseau physique. Il est utilisé pour les périphériques réseau Ethernet locaux.

3.4 Conclusion

Dans ce chapitre nous avons présenté une vue générale sur le serveur et le cloud et les protocoles de communication.

Chapitre 4

Outils et techniques

Cette section explique tous divers concepts et détails de mise en œuvre sur le projet.

4.1 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs [11] sont principalement utilisés pour des données de type grille. Cela implique souvent des images ainsi que d'autres données disposées en forme de grille. Dans les CNN, chaque noyau aboutit à une nouvelle couche alambiquée. Ces couches sont également appelées cartes d'activation.

Comme son nom l'indique, dans ce réseau, l'opération de convolution est utilisée à la place de la multiplication matricielle. Comme le montre la figure 5.1, dans cette opération, un noyau ou un masque se déplace sur une image et une représentation convolutée est calculée en tant que sortie. Les CNN peuvent capturer des dépendances spatiales et temporelles dans une image. Cette propriété Les réseaux de neurones convolutifs [11] sont principalement utilisés pour des données de type grille. Cela implique souvent des images ainsi que d'autres données disposées en forme de grille. Dans les CNN, chaque noyau aboutit à une nouvelle couche alambiquée. Ces couches sont également appelées cartes d'activation.

Comme son nom l'indique, dans ce réseau, l'opération de convolution est utilisée à la place de la multiplication matricielle. Comme le montre la figure 3, dans cette opération, un noyau ou un masque se déplace sur une image et une représentation convolutée est calculée en tant que sortie. Les CNN peuvent capturer des dépendances spatiales et temporelles dans une image.

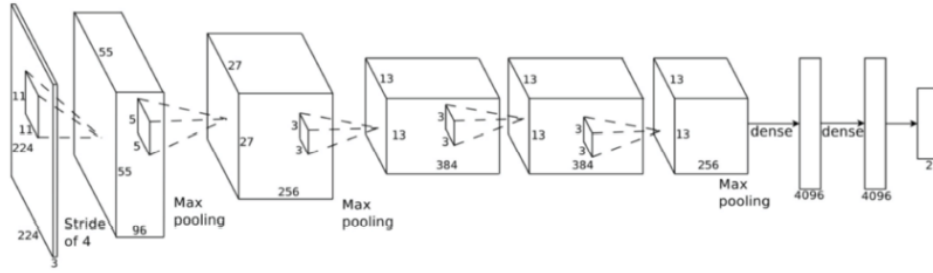


FIGURE 4.1 – Architecture Alexnet [2]

Cette propriété est extrêmement utile dans les tâches de vision par ordinateur telles que la classification d'images et la classification d'images. Dans un réseau de classification d'images [2, 12, 13, 14], des séries de couches de convolution sont suivies de couches entièrement connectées qui aident à la classification. Profondeur de Les réseaux de neurones convolutifs décident de la nuance des observations d'un réseau. Les couches inférieures des CNN sont responsables des caractéristiques de niveau inférieur telles que les lignes, les coins, etc., tandis que les couches de niveau supérieur sont responsables des caractéristiques de niveau supérieur. La taille des résultats intermédiaires peut être réduite en utilisant la foulée. La foulée est un décalage par lequel les noyaux sont déplacés pendant l'opération de convolution.

4.1.1 Réseaux de neurones convolutifs ponctuels

L'opération de convolution ponctuelle est un type d'opération spéciale où la taille du noyau est toujours de 1 1. Cette opération renvoie une couche avec les mêmes dimensions que celle des couches d'entrée.

4.1.2 Réseaux de neurones convolutifs en profondeur

Comme leur nom l'indique, les CNN en profondeur fonctionnent en profondeur. Chaque noyau peut avoir n'importe quelle hauteur et largeur, cependant, sa profondeur est toujours égale à un. Des noyaux séparés agissent sur chaque niveau de profondeur. L'empilement de toutes ces couches donne une image. Les convolutions ponctuelles et en profondeur impliquent beaucoup moins de multiplications. Cette peut être démontré par un calcul simple [15] - supposons qu'il y a 16 noyaux au total. Supposons que la taille du noyau est 5x5x3 et nous le déplaçons 8 fois pour la longueur et largeur. D'où les multiplications totales requises sont : $16 \cdot 3 \cdot 5 \cdot 5 \cdot 8 \cdot 8 = 76800$ Si nous décidons d'utiliser à la place une convolution en profondeur et ponctuelle, alors Pour la convolution en profondeur, nous aurons trois noyaux au lieu d'un. D'où le le calcul sera - $3 \cdot 5 \cdot 5 \cdot 8 \cdot 8 = 4800$ Pour la convolution ponctuelle, nous avons $16 \cdot 1 \cdot 1 \cdot 3 \cdot 8 \cdot 8 = 3072$ Par conséquent, le nombre de multiplications requises est $4800 + 3072 = 7872$ C'est bien moins que les multiplications requises pour les couches de convolution normales. Étant

donné que les convolutions séparables dans l'espace entraînent moins de paramètres que le CNN normaux, les réseaux de neurones peu profonds avec des convolutions séparables dans l'espace peuvent ne pas parvenir à apprendre pendant la formation.

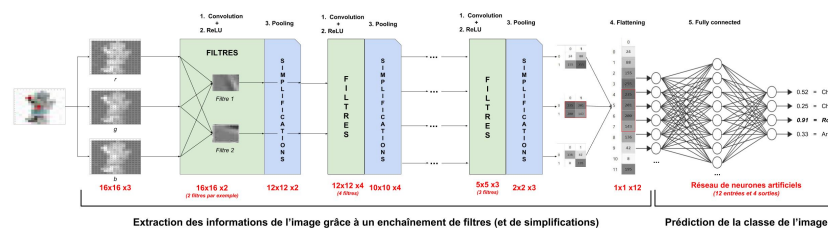


FIGURE 4.2 – CNN

4.2 Mesures de qualité d'image

Les métriques de qualité d'image nous sont utiles pour mesurer les performances d'une architecture. Ceux-ci peuvent également être utilisés pour définir la perte dans les réseaux de neurones. Les métriques de qualité d'image sont de deux types : les métriques de qualité d'image de référence et les métriques d'image non-référentes métriques de qualité.

Les métriques de qualité d'image non-référentes telles que le contraste moyen soustrait normalisé (MSCN) ne nécessitent pas d'image de référence pour comparer une image.

Puisque nous allons comparer l'image non compressée avec l'image originale dans ce projet, nous n'avons utilisé que des mesures de qualité d'image de référence. Ici, nous passerons en revue certaines mesures de qualité d'image de référence.

4.3 python

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux5, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation6.



FIGURE 4.3 – Python

4.3.1 Utilisation

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives . On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.

4.3.2 Mots-clés du langage

Les mots-clés réservés du langage Python sont fournis dans la liste `keyword.kwlist` du module `keyword`²⁷.

Les mots-clés de Python 2.7.5 sont les suivants : `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `not`, `or`, `pass`, `print`, `raise`, `return`, `try`, `while`, `with`, `yield`.

4.4 Tensorflow

TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.



FIGURE 4.4 – Tensorflow

4.5 OpenCV

OpenCV (pour Open Computer Vision) est une bibliothèque libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage et la société ItSeez se sont succédé au support de cette bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel.

Cette bibliothèque est distribuée sous licence BSD.

NVidia a annoncé en septembre 2010 qu'il développerait des fonctions utilisant CUDA pour OpenCV.



FIGURE 4.5 – OpenCV

4.5.1 Fonctionnalités

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

4.6 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre

ordinateur à l'exception d'un navigateur. Cool, n'est ce pas ? nous rappellerons ce qu'est un Jupyter Notebook.

Jupyter Notebook est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données, du Machine Learning, etc. Elle est disponible par défaut dans la distribution Anaconda



FIGURE 4.6 – Python

Chapitre 5

Experimentation et discussions des résultats

Dans cette partie nous avons implémenté un modèle avec deep learning pour compresser et décompresser les images qu'on va stocker au serveur pour avoir une durée d'envois qui petite et moins d'espace pour le stocker dans le serveur. Ce modèle est exécuté à chaque fois qu'il est envoyé au serveur et aussi lors du téléchargement depuis le serveur.

5.1 Étapes d'Algorithme

5.1.1 Preparation des données

Dans cette partie nous avons cherché défirent images pour entrainer notre modèle. Finalement, nous avons été en mesure d'obtenir 500 images pour former notre réseau de convolution pour compresser et décompresser les images de ces données à 45 Mo. Après on prépare notre image pour l'entraînement par le modifier à une matrice pour faciliter entraînement pour compresser les images pour avoir moins de temps pour envois au serveur.



FIGURE 5.1 – Train images

5.1.2 Création de modele

Nous avons implémenté un modèle qui utilise les couches de convolution pour extraire la maximent d'information depuis les images pour les reconstruire les images en décompression des images. Donc ce modèle est construit de deux couches de convolution de 64 filtres avec une kernel de (5*5), et une couche de maxpooling pour réduire la taille de matrice de convolution. En fin nous avons trouver une model avec 3,265 paramètres qui un bon nombre de paramètre.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 512, 512, 64)	1664
max_pooling2d (MaxPooling2D)	(None, 256, 256, 64)	0
conv2d_1 (Conv2D)	(None, 256, 256, 1)	1601
Total params: 3,265		
Trainable params: 3,265		
Non-trainable params: 0		

FIGURE 5.2 – Model de compression

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 256, 256, 64)	1664
up_sampling2d (UpSampling2D)	(None, 512, 512, 64)	0
conv2d_3 (Conv2D)	(None, 512, 512, 1)	1601
Total params: 3,265		
Trainable params: 3,265		
Non-trainable params: 0		

FIGURE 5.3 – Model de décompression

5.1.3 Entrainement de model

Après qu'on a implémenter notre model on va l'entrainer sur 200 epochs et avec une batch size de 64 pour avoir deux model pour compression et pour la décompression.

train the model
history = model.fit(img, imgT, epochs=200, batch_size=32, validation_data=(img, imgT))
Epoch 72/100
17/17 [=====] - 6s 388ms/step - loss: 0.0232 - accuracy: 0.0777 - val_loss: 0.0214
Epoch 73/100
17/17 [=====] - 6s 388ms/step - loss: 0.0235 - accuracy: 0.0777 - val_loss: 0.0210
Epoch 74/100
17/17 [=====] - 6s 388ms/step - loss: 0.0206 - accuracy: 0.0777 - val_loss: 0.0204
Epoch 75/100
17/17 [=====] - 6s 387ms/step - loss: 0.0199 - accuracy: 0.0777 - val_loss: 0.0194
Epoch 76/100
17/17 [=====] - 6s 387ms/step - loss: 0.0197 - accuracy: 0.0777 - val_loss: 0.0193
Epoch 77/100
17/17 [=====] - 6s 387ms/step - loss: 0.0224 - accuracy: 0.0777 - val_loss: 0.0198
Epoch 78/100
17/17 [=====] - 6s 387ms/step - loss: 0.0198 - accuracy: 0.0777 - val_loss: 0.0191
Epoch 79/100
17/17 [=====] - 6s 388ms/step - loss: 0.0195 - accuracy: 0.0777 - val_loss: 0.0195
Epoch 80/100
17/17 [=====] - 7s 390ms/step - loss: 0.0200 - accuracy: 0.0777 - val_loss: 0.0218
Epoch 81/100
17/17 [=====] - 6s 389ms/step - loss: 0.0205 - accuracy: 0.0777 - val_loss: 0.0193
Epoch 82/100
17/17 [=====] - 6s 388ms/step - loss: 0.0193 - accuracy: 0.0777 - val_loss: 0.0203
Epoch 83/100
17/17 [=====] - 6s 389ms/step - loss: 0.0192 - accuracy: 0.0777 - val_loss: 0.0195

FIGURE 5.4 – Train images

5.2 Resultat

Après l'entraînement on trouve les résultats suivants :

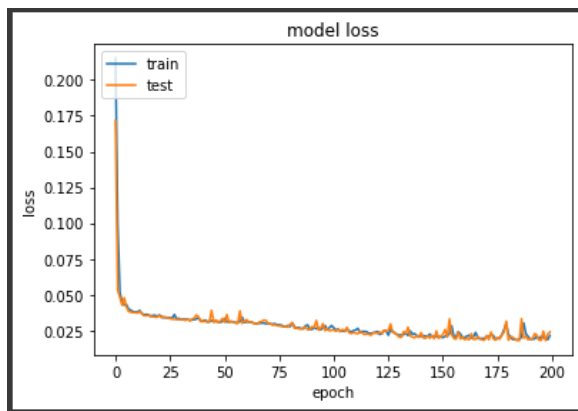


FIGURE 5.5 – Loss compression

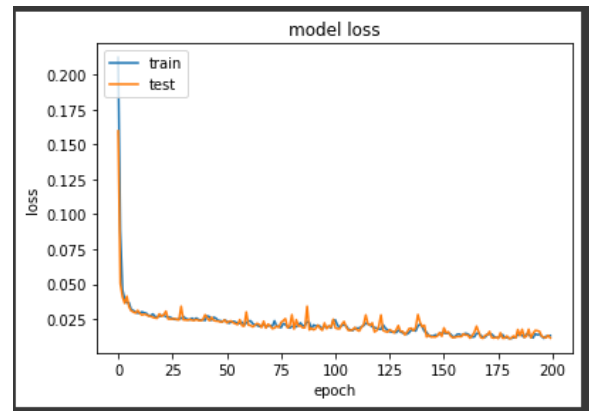


FIGURE 5.6 – Loss decompression

Comme il présente les graphes on peut dire que notre model à bien entrainer sur notre data pour qu'on puit compresser et décompresser les images.

5.3 Test

Pour ce test on a utilisé un serveur local pour l'envoi des images pour tester le temps d'envoi de chaque type des images original et compresser. Donc si on stocke les images compresser on gagne dans le temps de téléchargement et de transmission et aussi sur espace de stockage.

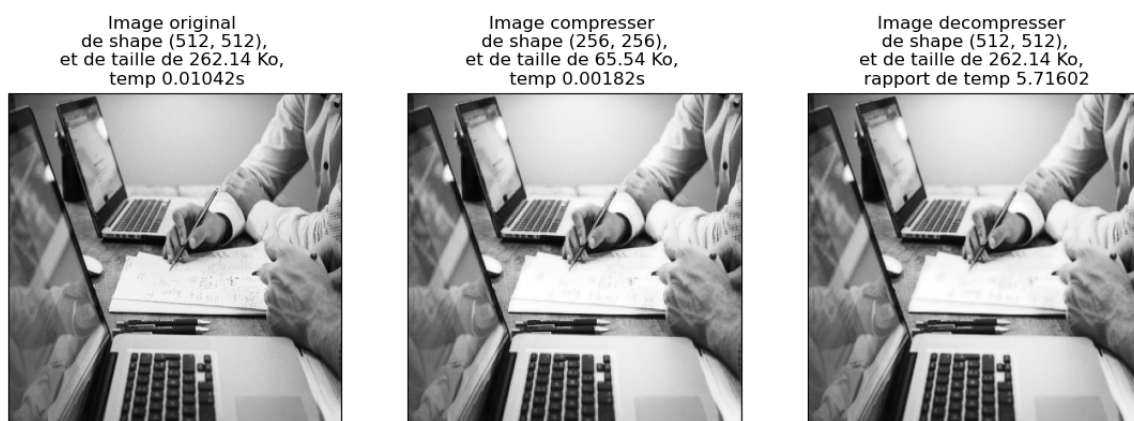


FIGURE 5.7 – Resultat de test

Donc comme la figure présente image original et compresser et décompresser, comme on voit que chaque image a une taille et une en une temps pour le transmettre à un serveur et dans la dernière image on a le rapport de temps entre le temps pour transmis image

originale au serveur et d'image compresser comme on vois que la durée pour transmis image compresser prend 5 fois moins de temps et 75% moins de taille donc on gagne dans le temps et dans espace avec une qualité de récupération très élever.



FIGURE 5.8 – Resultat de test (qualite)

5.4 Conclusion

Donc on a pu implémenter ce modèle avec des très bon résultat dans l'entraînement et dans le test.

bibliography

- [2] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” In *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [12] A. Krizhevsky, I. Sutskever, and G. Hinton, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [13] C. Szegedy, V. V. S. Ioffe, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [15] C. Wang, “A basic introduction to separable convolutions,” <https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728>, Aug 2019, (Accessed on 04/27/2019).