



REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE



UNIVERSITE LARBI TEBESSI - TEBESSA
FACULTE DES SCIENCES ET TECHNOLOGIES
DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE

**DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE MASTER EN
INSTRUMENTATION**

THEME

**Detection des catastrophes naturelles par les algorithmes
de l'apprentissage profond**

Présenté par le binôme :

- Merzoug Sabah
- Khamam Soulef

Devant le jury :

- | | |
|---------------------|-----------|
| - AOUICHE Abdelaziz | Président |
| - KHEMAISSIA Seddik | Encadreur |
| - CHERIET Laila | Examineur |

Année Universitaire 2019 / 2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Nous aimerions profiter de ces quelques lignes pour remercier tous ceux qui ont contribué de près ou de loin à ce mémoire de maîtrise.

Nous tenons d'abord à exprimer notre gratitude à l'encadreur de notre mémoire **Dr : Khemaissia seddik**, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion. Un grand remerciement aux membres des jurys pour avoir bien voulu examiner et juger

Nous souhaitons adresser nos remerciements les plus sincères au corps professoral et administratif de Département Génie électrique, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Un grand merci à mes parents, mes familles, pour leurs amours, leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui nous a permis de réaliser les études que nous voulions et par conséquent ce mémoire.

Enfin, Nous aimerions exprimer notre reconnaissance envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long notre démarche.

Dédicaces

A mes très chers parents, source de vie,

D'amour et d'affection

A mes chers frères, et leurs enfants source de

Joie et bonheur

A tous mes amis, surtout frère et collègue

YAHIA REDJIMI

Khamam soulef

Dédicaces

Je dédie ce travail en signe de reconnaissance :

*A l'âme de mon très cher et aimé père qui restera à jamais
une source de lumière qui illuminera mon chemin*

*A ma chère mère qui m'a toujours comblé
d'amour et tendresse durant les moments difficiles*

A mes très chers frères

A toutes mes amies et surtout notre collègue

YAHIA REDJIMI.

Merzoug sabah

Table des matières

Liste des figures.....	1
Introduction générale.....	2

Chapitre I : Classification des images

I.1 Introduction	3
I.2 Les motivations de la classification des images	3
I.3 Notions de bases	4
I.3.1 Définition d'une image.....	4
I.3.2 Les différents types du format d'image.....	5
I.3.3 Caractéristiques de l'image	5
I.3.3.1 Pixel	5
I.3.3.2 Dimension Résolution.....	6
I.3.3.3 Voisinage.....	6
I.3.3.4 Niveau de gris	7
I.3.3.5 Contraste	7
I.3.3.6 Luminance.....	7
I.3.3.7 Bruit	8
I.3.3.8 Contour	8
I.4 Méthodes de classification	8
I.4.1 Méthodes supervisées.....	8
I.4.2 Méthodes non supervisées.....	9
I.5 Indicateurs de performance en classification.....	9
I.5.1 Matrice de confusion.....	9
I.5.2 Courbe ROC.....	10
I.5.3 Classification des images et l'apprentissage machine.....	11
I.5.4 Classification des images et les réseaux de neurones.....	12
I.8 Conclusion	13

Chapitre II : l'apprentissage profond (deep learning).....	14
II.1 Introduction	15
II.1.1 L'intelligence artificiel, l'apprentissage automatique et l'apprentissage profond .	15
II.1.1.1 Intelligence artificielle	15
II.1.1.2 Apprentissage automatique	15
II.1.1.3 L'apprentissage en profondeur.....	16
II.1.2 Différence entre machine learning et le deep learning	17
II.2. Deep learning (l'apprentissage profond)	18
II.3 Présentation des réseaux de neurones artificiels	21
II.3.1 Couche d'entrée	21
II.3.2 Couche cachée.....	21
II.3.3 Couche de sortie.....	22
II.3.4. Connexions entre les couches.....	22
II.4. Les fonctions d'activation.....	22
II.4.1. La fonction Sigmoidale.....	22
II.4.2. La fonction ReLu.....	23
II.4.3 La fonction Soft max.....	23
II.5. Les architecture des réseaux de neurones.....	24
II.6. Les réseaux entièrement connectés.....	24
II.6.1 . La couche convolutive	24
II.6.2. La couche de pooling.....	24
II.6.3. La couche entièrement connectée	24
II.7. Les réseaux neuronaux récurrents et LSTM	25

II.8. Les réseaux de neurones convolutifs (CNN)	26
Chapitre III : L'apprentissage par transfert.....	27
III .1 Introduction	28
III .2 L'architecture d'un CNN comporte 2 parties	29
III .2 .1 L'entraînement d'un nouveau CNN est difficile	30
III .3 réseaux de neurones convolutifs (CNN).....	30
III .3 .1 L'étape de convolution	33
III .3 .2 Profondeur	37
III .3 .3 Stride	38
III .3 .4 Zéro-remplissage	39
III .3 .5 Présentation de la non linéarité (ReLU).....	39
III .3 .6 L'étape de la mise en commun	40
III .4 L'histoire jusqu'à présent	42
III .4.1 Couche entièrement connectée	43
III .4.2 Formation à l'aide de la rétropropagation.....	44
III.4 :3 Visualisation des réseaux de neurones convolutifs	47
III.5 Autres architectures ConvNet	48
III.6 Conclusion	49

Chapitre IV: Détection des catastrophes naturelles à l'aide du Deep Learning	
IV. 1 Introduction	50
IV.2 Logiciels et bibliothèques Utilisés dans l'implémentation	51
IV.2.1 Python	52
IV.2.2 TensorFlow	53
IV.2.3 Keras	54
IV.2.4 Scikit learn	55
IV.2.5 Configuration Utilisé dans l'implémentation	56
IV.3 Collection de bases de données	57
IV.4 Formation du modèle	58
IV.5 valuation	59
IV.6 configuration programme	60
IV.7 Résultats obtenus et discussion	61
IV.7.1 Résultats obtenus pour 3 différents Epochs	62
IV.7.2 Résultats obtenus pour 3 différentes résolutions	63
IV.7.3 Résultats obtenus pour les 3 différents nombre d'image	64
IV.7.4 Résultat expérimental	65
IV.8 Détection de calamité en vidéo	66
IV.9 Message d'alerte	67
IV.10 Conclusion	68
Conclusion général	69

Liste des figures

Figure I. 1: Voisinage à 4

Figure I. 2 : Voisinage à 8

Figure I.3 : Courbe ROC

Figure II.1:différence entre AI, ML et Deep Learning

Figure II.2:différence entre L'apprentissage profond et L'apprentissage automatique

Figure II.3: neurone artificiel et biologique

Figure II.4: Un neurone artificiel

Figure II.5:Réseau de neurones artificiels

Figure II.6 : Représentation graphique de la fonction Sigmoidé

Figure II.7:Représentation graphique de la fonction Relu

Figure II.8: fonction Softmax

Figure II.9: Les types de séquences d'entrée pour un réseau récurrent

Figure II.10:Réseau de neurones a convolution (CNN)

Figure III.1 : Les réseaux de neurones convolutifs

Figure III.2 : réseaux de neurones convolutifs (CNN)

Figure III.3 : exemple de ConvNets utilisé pour reconnaître des objets

Figure III.4 : Un simple ConvNet

Figure III.5 : chaque image est une matrice de valeurs de pixels

Figure III.6 :L'opération de convolution. La matrice de sortie est appelée entité convolutive ou Carte d'entités

Figure III.7 :L'opération de convolution

Figure III.8 : nombre de filtres

Figure III.9 :l'opération ReLU

Figure III.10 : opération ReLU

Figure III.11 : mise en commun maximale

Figure III.12: regroupement appliqué aux cartes d'entités rectifiées

Figure III.13: Mise en commun

Figure III.14 : modèle CNN

Figure III.15 : couche entièrement connectée - chaque nœud est connecté à tous les autres nœuds de la couche adjacente

Figure III.16 : Formation du ConvNet

Figure III.17 : opérations des ConvNets

Figure III.18: Fonctionnalités apprises d'un réseau convolutif de croyances profondes

Figure IV.1: ensemble de données d'images de catastrophes naturelles

Figure IV.2: réglage fin de ResNet 50

Figure IV.3: Précision et Erreur pour le Modèle & Confusion Matrix. (20-400)

Figure IV.4: Precision-Recall.& courbe ROC (20-400)

Figure IV.5: Précision et Erreur pour le Modèle & Confusion Matrix. (40-400)

Figure IV.6: Precision-Recall.& courbe ROC (40-400)

Figure IV.7: Précision et Erreur pour le Modèle & Confusion Matrix. (80-400)

Figure IV.8: Precision-Recall.& courbe ROC (80-400)

Figure IV.9: Précision et Erreur pour le Modèle & Confusion Matrix. (Resolution 80x80)

Figure IV.10: Precision-Recall.& courbe ROC (Resolution 80x80)

Figure IV.11: Précision et Erreur & Confusion Matrix. (Résolution 224 x 224)

Figure IV.12: Precision-Recall.& courbe ROC (Résolution 224 x 224)

Figure IV.13: Précision et Erreur & Confusion Matrix. (Résolution 224 x 224 filtre 7x7)

Figure IV.14: Precision-Recall.& courbe ROC (Résolution 224 x 224 filtre 7x7)

Figure IV.15: Précision & Confusion Matrix. (80-400) (Résolution 224 x 224 filtre 7x7)

Figure IV.16: Precision-Recall.& courbe ROC (80-400) (Résolution 224 x 224 filtre 7x7)

Figure IV.17: Précision et Erreur pour le Modèle & Confusion Matrix. (80-800)

Figure IV.18: Precision-Recall.& courbe ROC (80-800)

Figure IV.19: Exemples d'images de sortie : inondation

Figure IV.20: Exemples d'images de sortie : inondation ou pluies normale

Figure IV.21: Exemples d'images de sortie erronées (80 Epochs, 400 Images)

Figure IV.22: Exemples d'images de sortie (80 epochs, 800 images)

Figure IV.23: Alerte générée par le système

Liste des tableau

Tableau IV.1: a précision et Erreur VS Epochs (400 Images)

Tableau IV.2: a précision et Erreur VS Epochs (800 Images)

Introduction générale

Introduction générale

Les catastrophes naturelles se produisent de façon souvent imprévisible. Elles échappent à la volonté humaine et sont amplifiées par l'implication des personnes. Les catastrophes naturelles sont aussi d'une gravité croissante, compte tenu de leur ampleur, de leur complexité, de leur fréquence et de leur impact sur l'économie. Chaque pays doit assurer la sécurité et la protection de ses habitants, de ses infrastructures et des autres éléments de son patrimoine contre les effets des catastrophes naturelles et environnementales.

Comme pour la défense militaire, la défense civile utilise une série de mesures pour réduire les risques causés par les « agressions » que sont les séismes, les inondations, les glissements de terrain, la sécheresse et les catastrophes environnementales. Elle a pour fonctions la surveillance, la défense et la gestion du risque. La surveillance a pour ultime objectif l'évaluation la plus précise possible des risques quelle que soit leur origine. Toute surveillance est le produit d'observations, aussi permanentes et précises que possible, et de

« Modèles » permettant de les interpréter. La focalisation sur les « zones sensibles » a pour objet de détecter l'apparition d'un phénomène dangereux et de déterminer comment il progresse.[1]

L'automatisation présente alors une alternative intéressante. Conférer aux machines la capacité d'interpréter les images de la Terre permettrait de multiplier les observations, pour en tirer à la fois informations et modèles. À l'heure actuelle, l'état de l'art en vision par ordinateur repose majoritairement sur les réseaux de neurones profonds, dont les performances en classification d'images, détection d'objets et reconnaissance de formes ont permis des avancées significatives en intelligence artificielle. Un processus idéal de cartographie itératif pour l'observation de la Terre est détaillé dans la figure 1.2. Cette thèse s'articule ainsi de la façon suivante. Nous cherchons à concevoir, implémenter et valider des modèles de réseaux de neurones artificiels profonds pour l'interprétation automatisée d'images aériennes et satellites.[2]

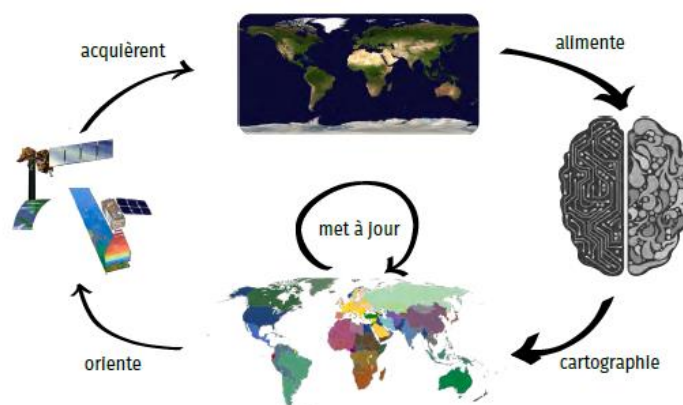


Figure 1 – Processus d'interprétation automatique des images d'observation de la Terre[2]

Introduction générale

L'analyse des images satellites nécessite encore des efforts et du temps, ce qui entraîne un retard dans la prédiction. Ce défi entraîne une perte substantielle d'infrastructures et de vies. Par conséquent, un système entièrement automatisé est un besoin, qui peut identifier ces catastrophes naturelles avec un minimum de retard. Ce système peut empêcher la perte de vies humaines précieuses et d'autres ressources.

Notre recherche se place à l'intersection de trois domaines scientifiques : la télédétection, la vision artificielle et l'apprentissage automatique. La littérature en vision par ordinateur est abondante concernant l'interprétation de données visuelles, y compris pour la télédétection. Récemment, les méthodes dites d'apprentissage profond ont permis de réaliser des avancées considérables en interprétation automatique d'images.

Cette thèse se trouve ainsi au croisement entre la télédétection, la vision par ordinateur et l'apprentissage automatique. En particulier, nous nous proposons de mettre en œuvre des méthodes d'apprentissage profond permettant la détection de catastrophes naturelles automatisées utilisant le deep learning, qui peut prévoir les catastrophes en temps réel et envoyer un message d'alerte[3]. À cet effet, nous avons entraîné le modèle ResNet50 CNN, et les performances sont mesurées en calculant la matrice de confusion. Le modèle est également testé avec des vidéos préenregistrées acquises à partir de satellites et de drones. Les résultats expérimentaux donnent une précision de 91% et fonctionnent bien lorsqu'ils sont testés avec des vidéos collectées sur YouTube.

Si les réseaux de neurones artificiels sont populaires dans l'état de l'art pour la vision par ordinateur, leurs succès sont récents et leur introduction pour la télédétection est encore nouvelle. Il sera donc nécessaire dans un premier temps dans notre premier chapitre de à présenter les notions de base de la classification des images, les différents types des images et les caractéristiques, ainsi que l'utilisation des réseaux de neurones dans la classification des images.

Le Chapitre II est l'occasion de rappeler les fondamentaux théoriques de l'apprentissage profond, et plus particulièrement des réseaux de neurones convolutifs ainsi que leurs applications en perception artificielle.

Le Chapitre III démontre l'utilité de L'apprentissage par transfert. Approche dans laquelle les connaissances acquises dans une ou plusieurs tâches source sont transférées et utilisées pour améliorer l'apprentissage d'une tâche cible connexe. On peut utiliser les connaissances acquises par un réseau qui a été formé pour une tâche particulière et le déployer pour une autre tâche. On n'a pas besoin de construire un modèle à partir de zéro.

Dans le quatrième chapitre, on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus et à la fin on termine par une conclusion générale et discuter des pistes de recherches futures envisageables à la suite de ces travaux.



CHAPTER I

CLASSIFICATION DES IMAGES

I.1 Introduction :

La classification automatique des images consiste à attribuer automatiquement une classe à une image à l'aide d'un système de classification. On retrouve ainsi la classification d'objets, de scènes, de textures, la reconnaissance de visages, d'empreintes digitale et de caractères. Il existe deux principaux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non-supervisé. Dans l'approche supervisée, chaque image est associée à une étiquette qui décrit sa classe d'appartenance. Dans l'approche non-supervisée les données disponibles ne possèdent pas d'étiquettes. Dans notre travail on s'intéresse de l'approche supervisée.

I.2 Les motivations de la Classification des images :

La classification des images consiste à répartir systématiquement des images selon des classes établies au préalable, classer une image lui fait correspondre une classe, marquant ainsi sa parenté avec d'autres images.

En général reconnaître une image est une tâche aisée pour un humain au fil de son existence, il a acquis des connaissances qui lui permettent de s'adapter aux variations qui résultent de conditions différentes d'acquisition .Il lui est par exemple relativement simple de reconnaître un objet dans plusieurs orientations partiellement caché par un autre de près ou de loin et selon diverses illuminations.

Toute fois les progrès technologiques en terme d'acquisition d'images (microscopes, caméras, capteurs) et de stockage engendrent des bases de données riche en information et multiplient les domaines d'applications, il devient alors difficile pour l'humain d'analyser le nombre important d'images, le temps requis le caractère répétitif de la tâche et la concentration nécessaire sont problématiques. Toutefois celle-ci n'est pas forcément aisée pour un programme informatique pour lequel une image est un ensemble de valeur numérique

L'objectif de la classification d'images est d'élaborer un système capable d'affecter une classe automatiquement à une image. Ainsi, ce système permet d'effectuer une tâche d'expertise qui peut s'avérer coûteuse à acquérir pour un être humain en raison notamment de contraintes physiques comme la concentration, la fatigue ou le temps nécessité par un

volume important de données images.

Les applications de la classification automatique d'images sont nombreuses et vont de l'analyse de documents à la médecine en passant par le domaine militaire. Ainsi on retrouve des applications dans le domaine médical comme la reconnaissance de cellules et de tumeurs, la reconnaissance d'écriture manuscrite pour les chèques les codes postaux. Dans le domaine urbain comme la reconnaissance de panneaux de signalisation la reconnaissance de piétons la détection de véhicules la reconnaissance de bâtiments pour aider à la localisation. Dans le domaine de la biométrie comme la reconnaissance de visage, d'empreintes, d'iris.

Le point commun à toutes ces applications est qu'elles nécessitent la mise en place d'une chaîne de traitement à partir des images disponibles composée de plusieurs étapes afin de fournir en sortie une décision. Chaque étape de la mise en place d'un tel système de classification nécessite la recherche de méthodes appropriées pour une performance globale optimale à savoir la phase d'extraction de caractéristiques et la phase d'apprentissage. Typiquement, nous disposons de données images desquelles il nous faut extraire des informations pertinentes traduites sous formes de vecteurs numériques. Cette phase d'extraction nous permet de travailler dans un espace numérique. Il s'agit ensuite d'élaborer dans la phase d'apprentissage, à partir de ces données initiales, une fonction de décision pour décider de l'appartenance d'une donnée nouvelle à l'une des classes en présence.[04] .

I.3 Notions de base :

I.3.1 Définition d'une image :

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel, elle est issue du contact des rayons lumineux provenant des objets formants la scène avec un capteur (caméra, scanner, rayons X, ...). Il ne s'agit en réalité que d'une représentation spatiale de la lumière.

L'image est considérée comme un ensemble de points auquel est affectée une grandeur physique (luminance, couleur). Ces grandeurs peuvent être continues (image analogique) ou bien discrètes (images digitales). Mathématiquement, l'image représente une fonction continue IF , appelée fonction image, de deux variables spatiales représentée par $IF(x, y)$ mesurant la nuance du niveau de gris de l'image aux coordonnées (x, y) . [02]

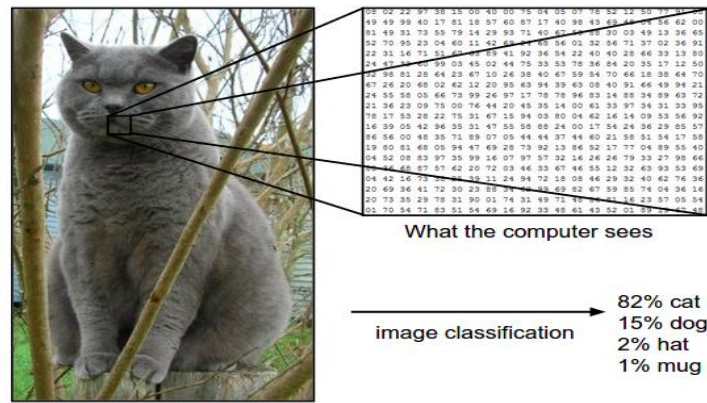


Figure I.1 : présentation d'une image

I.3.2 Les différents types de format d'image :

- **Image couleur RGB** : L'œil humain analyse la couleur à l'aide de trois types de cellules photo 'les cônes'. Ces cellules sont sensibles aux basses, moyennes, ou hautes fréquences (rouge, vert, bleu). Pour représenter la couleur d'un pixel, il faut donc donner trois nombres, qui correspondent au dosage de trois couleurs de base : Rouge, Vert, Bleu. On peut ainsi représenter une image couleur par trois matrices chacune correspondant à une couleur de base.

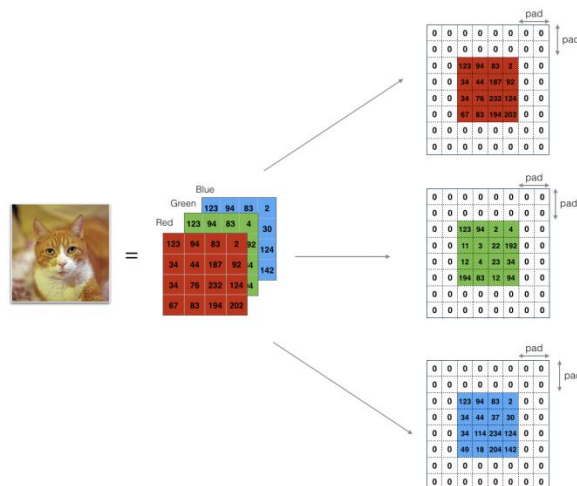


Figure I.2 : présentation d'une image RGB

- **Image d'intensités** : C'est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris, car les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris.

- **Image binaire** : Une image binaire est une matrice rectangulaire dans l'élément valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc.[05]

I.3.3 Caractéristiques de l'image :

L'image est un ensemble structuré d'information caractérisé par les paramètres suivants :

I.3.3.1 Pixel :

Le pixel est l'abréviation du mot « Picture élément », est une unité de surface permettant de définir la base d'une image numérique. Il matérialise un point donné (x, y) du plan de l'image. L'information présentée par le pixel est le niveau de gris (ou la couleur) prélevée à l'emplacement correspondant dans l'image réelle. La différence entre image monochrome et image couleur réside dans la quantité d'informations contenue dans chaque pixel, par exemple dans une image couleur (RGB : Rouge, Vert, Bleu) la valeur d'un pixel est représentée sur trois octets pour chaque couleur.

I.3.3.2 Dimension & Résolution :

La dimension est la taille de l'image. Elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

Par contre, la résolution est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure.

I.3.3.3 Voisinage :

Le plan de l'image est divisé en termes de formes rectangulaires ou hexagonales permettant ainsi l'exploitation de la notion de voisinage (voir figure 1). Le voisinage d'un pixel est formé par l'ensemble des pixels qui se situent autour de ce même pixel. On définit aussi

l'assiette comme étant l'ensemble de pixels définissant le voisinage pris en compte autour d'un pixel.

On distingue deux types de voisinage :

Voisinage à 4 : On ne prend en considération que les pixels qui ont un coté commun avec le pixel considéré.

Voisinage à 8 : On prend en compte tous les pixels qui ont au moins un point en liaison avec le pixel considéré.

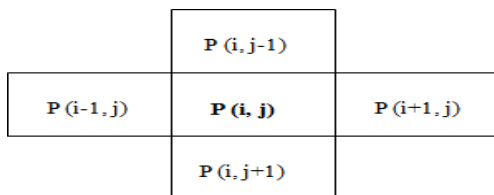


Figure I. 3: Voisinage à 4

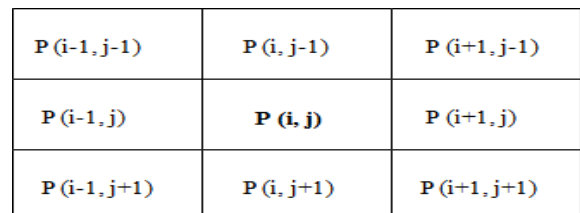


Figure I. 4 : Voisinage à 8

I.3.3.4 Niveau de gris :

C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir (0) jusqu'au blanc (255) en passant par les nuances qui sont contenues dans l'intervalle [0, 255]. Elle correspond en fait à la quantité de la lumière réfléchie.

Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue. Plus le nombre de bit est grand plus les niveaux sont nombreux et plus la représentation est fidèle.

[06]

I.3.3.5 Contraste :

C'est l'opposition marquée entre deux régions d'une image. Une image contrastée présente une bonne dynamique de la distribution des valeurs de gris sur tout l'intervalle des valeurs possibles, avec des blancs bien clairs et des noirs profonds. Au contraire une image peu contrastée a une faible dynamique, la plupart des pixels ayant des valeurs de gris très proches.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste est défini par le rapport : $C=L_1-L_2/L_1+L_2$

I.3.3.6 Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

I.3.3.7 Bruit :

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. C'est un parasite qui représente certains défauts (poussière, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...etc.). Il se traduit par des taches de faible dimension et dont la distribution sur l'image est aléatoire. [07]

I.3.3.8 Contour :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentant une différence significative.

Dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum; il s'agit alors de contour « en toit »

I.4 Méthodes de classification :

De nombreuses méthodes classiques ont été consacrées, elles peuvent être séparées en deux grandes catégories: les méthodes de classification supervisée et les méthodes de classification non supervisée.

I.4.1 Méthodes supervisées :

L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets. On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement.

Il est nécessaire d'étudier la fiabilité de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage ou de sur apprentissage (complexité du modèle). On utilise souvent un deuxième échantillon indépendant, dit de validation ou de test.

I.4.2 Méthodes non supervisées :

Procède de la façon contraire. C'est à dire ne nécessitent aucun apprentissage et aucune tâche préalable d'étiquetage manuel. Elle consiste à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster. Il est généralement lié au domaine de l'analyse des données comme l'ACP. Un «Cluster» est une collection d'objets qui sont «similaires» entre eux et qui sont «dissemblables

» Par rapport aux objets appartenant à d'autres groupes. [08]

I.5 Indicateurs de performance en classification :

I.5.1 Matrice de confusion :

Prenons l'exemple d'un classifieur binaire, c'est-à-dire, qui prédit 2 classes notées classe 0 et classe 1.

Pour mesurer les performances de ce classifieur, il est d'usage de distinguer 4 types d'éléments classés pour la classe voulue :

1. **Vrai positif VP** : Elément de la classe 1 correctement prédit.
2. **Vrai négatif VN** : Elément de la classe 0 correctement prédit.
3. **Faux positif FP** : Elément de la classe 1 mal prédit.
4. **Faux négatif FN** : Elément de la classe 0 mal prédit.

Ces informations peuvent être rassemblés et visualisés sous forme de tableau dans une matrice de confusion. Dans le cas d'un classifieur binaire, on obtient :

		Classe prédite	
		Classe 0	Classe 1
Classe réelle	Classe 0	VN	FN
	Classe 1	FP	VP

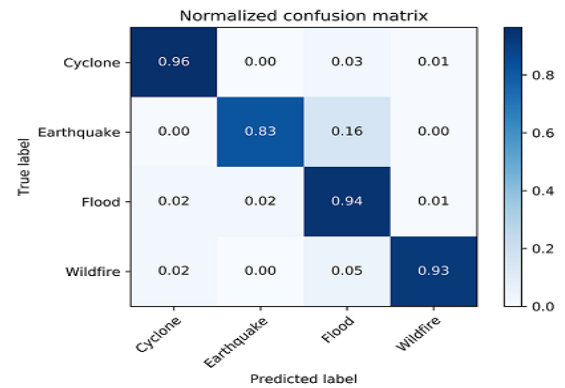


Figure I .5 : matrice de confusion

En particulier, si la matrice de confusion est diagonale, le classifieur est parfait.

Il est possible de calculer plusieurs indicateurs résumant la matrice de confusion. Par exemple si nous souhaitons rendre compte de la qualité de la prédiction sur la classe 1, on définit :

- **Précision.** Proportion d'éléments bien classés pour une classe donnée:

$$Precision_{de\ la\ classe\ 1} = \frac{VP}{VP + FP} \tag{1}$$

- **Rappel.** Proportion d'éléments bien classés par rapport au nombre d'éléments de la

$$Rappel_{de\ la\ classe\ 1} = \frac{VP}{VP + FN} \tag{2}$$

- **F-mesure.** Mesure de compromis entre précision et rappel:

$$F\text{-mesure}_{de\ la\ classe\ 1} = \frac{2 \cdot (Précision \cdot Rappel)}{Précision + Rappel} \tag{3}$$

Il est possible de calculer tous ces indicateurs pour chaque classe. La moyenne sur chaque classe de ces indicateurs donne des indicateurs globaux sur la qualité du classifieur.

$$Precision = \frac{1}{k} \sum_{i=1}^k \frac{VP_i}{VP_i + FP_i} \quad (4)$$

$$Rappel = \frac{1}{k} \sum_{i=1}^k \frac{VP_i}{VP_i + FN_i} \quad (5)$$

$$F - mesure = \frac{2 * (Précision * Rappel)}{Précision + Rappel} \quad (6)$$

I.5.2 Courbe ROC (Received Operating Characteristic):

Dans le cas d'un classifieur binaire, il est possible de visualiser les performances du classifieur sur ce que l'on appelle une courbe ROC. La courbe ROC est une représentation du taux de vrais positifs en fonction du taux de faux positifs. Son intérêt est de s'affranchir de la taille des données de test dans le cas où les données sont déséquilibrées.

Cette représentation met en avant un nouvel indicateur qui est l'aire sous la courbe. Plus elle se rapproche de 1, plus le classifieur est performant. [12]

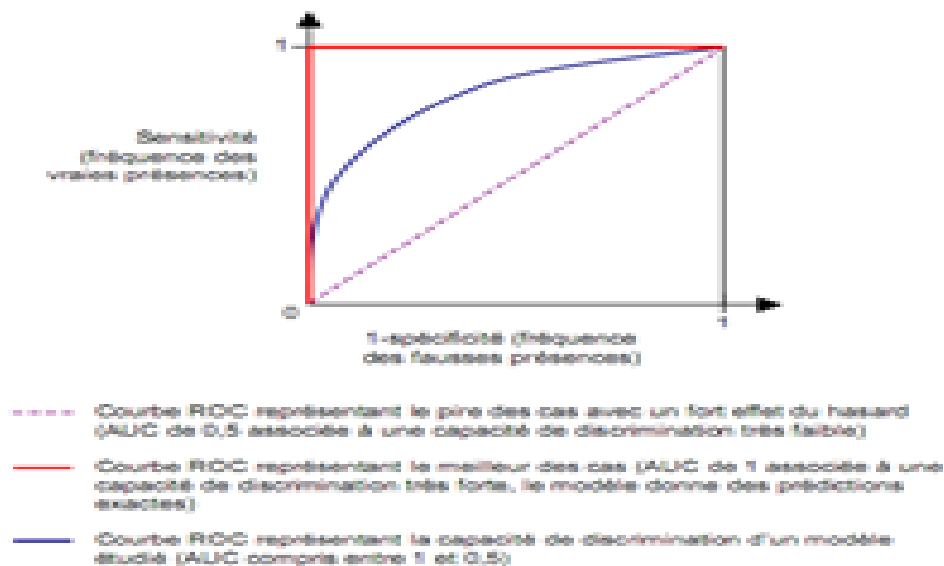


Figure I.6: Courbe ROC

I.6 Classification des images et l'apprentissage machine :

Les méthodes manuelles se sont avérées très difficiles à appliquer pour des tâches en apparence très simples comme la classification des images, la reconnaissance d'objets dans les images ou la reconnaissance vocale. Les données venant du monde réel les échantillons d'un son ou les pixels d'une image sont complexes, variables et entachées de bruit.

Pour une machine, une image est un tableau de nombres indiquant la luminosité (ou la couleur) de chaque pixel, et un signal sonore une suite de nombres indiquant la pression de l'air à chaque instant.

Comment une machine peut-elle transcrire la suite de nombres d'un signal sonore en série de mots tout en ignorant le bruit ambiant, l'accent du locuteur et les particularités de sa voix? Comment une machine peut-elle identifier un chien ou une chaise dans le tableau de nombres d'une image quand l'apparence d'un chien ou d'une chaise et des objets qui les entourent peut varier infiniment?

Il est virtuellement impossible d'écrire un programme qui fonctionnera de manière robuste dans toutes les situations. C'est là qu'intervient l'apprentissage machine (que l'on appelle aussi apprentissage automatique). C'est l'apprentissage qui anime les systèmes de toutes les grandes entreprises d'Internet. Elles l'utilisent depuis longtemps pour filtrer les contenus indésirables, ordonner des réponses à une recherche, faire des recommandations, ou sélectionner les informations intéressantes pour chaque utilisateur.

Un système entraînable peut être vu comme une boîte noire avec une entrée, par exemple une image, un son, ou un texte, et une sortie qui peut représenter la catégorie de l'objet dans l'image, le mot prononcé, ou le sujet dont parle le texte. On parle alors de systèmes de classification ou de reconnaissance des formes.

Dans sa forme la plus utilisée, l'apprentissage machine est supervisé: on montre en entrée de la machine une photo d'un objet, par exemple une voiture, et on lui donne la sortie désirée pour une voiture. Puis on lui montre la photo d'un chien avec la sortie désirée pour un chien. Après chaque exemple, la machine ajuste ses paramètres internes de manière à rapprocher sa sortie de la sortie désirée. Après avoir montré à la machine des milliers ou des millions d'exemples étiquetés avec leur catégorie, la machine devient capable de classer correctement la plupart d'entre eux. Mais ce qui est plus intéressant, c'est qu'elle peut aussi classer correctement des images de voiture ou de chien qu'elle n'a jamais vues durant la

phase l'apprentissage. C'est ce qu'on appelle la capacité de généralisation.

Jusqu'à récemment, les systèmes de reconnaissance des images classiques étaient composés de deux blocs: un extracteur de caractéristiques (feature extractor en anglais), suivi d'un classifieur entraînable simple. L'extracteur de caractéristiques est programmé «à la main», et transforme le tableau de nombres représentant l'image en une série de nombres, un vecteur de caractéristiques, dont chacun indique la présence ou l'absence d'un motif simple dans l'image. Ce vecteur est envoyé au classifieur, dont un type commun est le classifieur linéaire. Ce dernier calcule une somme pondérée des caractéristiques: chaque nombre est multiplié par un poids (positif ou négatif) avant d'être sommé. Si la somme est supérieure à un seuil, la classe est reconnue. Les poids forment une sorte de «Prototype» pour la classe à laquelle le vecteur de caractéristiques est comparé. Les poids sont différents pour les classifieurs de chaque catégorie, et ce sont eux qui sont modifiés lors de l'apprentissage. Les premières méthodes de classification linéaire entraînable datent de la fin des années cinquante et sont toujours largement utilisées aujourd'hui. Elles prennent les doux noms de perceptron ou régression logistique. [13]

I.7 Classification des images et les réseaux de neurones :

Le problème de l'approche classique de la reconnaissance des images est qu'un bon extracteur de caractéristiques est très difficile à construire, et qu'il doit être repensé pour chaque nouvelle application.

C'est là qu'intervient l'apprentissage profond ou deep learning en anglais. C'est une classe de méthodes dont les principes sont connus depuis la fin des années 1980, mais dont l'utilisation ne s'est vraiment généralisée que depuis 2012, environ.

L'idée est très simple: le système entraînable est constitué d'une série de modules, chacun représentant une étape de traitement. Chaque module est entraînable, comportant des paramètres ajustables similaires aux poids des classifieurs linéaires. Le système est entraîné de bout en bout: à chaque exemple, tous les paramètres de tous les modules sont ajustés de manière à rapprocher la sortie produite par le système de la sortie désirée. Le qualificatif profond vient de l'arrangement de ces modules en couches successives.

Pour pouvoir entraîner le système de cette manière, il faut savoir dans quelle direction et de combien ajuster chaque paramètre de chaque module. Pour cela il faut calculer un gradient, c'est-à-dire pour chaque paramètre ajustable, la quantité par laquelle l'erreur en sortie augmentera ou diminuera lorsqu'on modifiera le paramètre d'une quantité donnée. Le calcul

de ce gradient se fait par la méthode de rétro propagation, pratiquée depuis le milieu des années 1980.

Dans sa réalisation la plus commune, une architecture profonde peut être vue comme un réseau multicouche d'éléments simples, similaires aux classifieurs linéaires, interconnectés par des poids entraînaibles. C'est ce qu'on appelle un réseau neuronal multicouche.

Pourquoi neuronal? Un modèle extrêmement simplifié des neurones du cerveau les voit comme calculant une somme pondérée et activant leur sortie lorsque celle-ci dépasse un seuil. L'apprentissage modifie les efficacités des synapses, les poids des connexions entre neurones. Un réseau neuronal n'est pas un modèle précis des circuits du cerveau, mais est plutôt vu comme un modèle conceptuel ou fonctionnel. Le réseau neuronal est inspiré du cerveau un peu comme l'avion est inspiré de l'oiseau.

Ce qui fait l'avantage des architectures profondes, c'est leur capacité d'apprendre à représenter le monde de manière hiérarchique. Comme toutes les couches sont entraînaibles, nul besoin de construire un extracteur de caractéristiques à la main. L'entraînement s'en chargera. De plus, les premières couches extrairont des caractéristiques simples (présence de contours) que les couches suivantes combineront pour former des concepts de plus en plus complexes et abstraits: assemblages de contours en motifs, de motifs en parties d'objets, de parties d'objets en objets, etc. [14]

I.8 Conclusion :

Nous avons consacré ce chapitre à la présentation des notions de la classification ainsi que leurs intérêts dans le domaine d'imagerie et on a parlé aussi sur l'utilisation des réseaux de neurones dans ce domaine. Dans le deuxième chapitre on va bien détailler l'apprentissage profond (deep learning).



CHAPITER II

L'APPRENTISSAGE PROFOND

II.1 Introduction :

Le cerveau est constitué de milliers de circuits neuronaux qui contrôlent chacun de nos pensées, raisonnements et comportements. L'exemple de la couture d'un masque facial (COVID19): qui est un ensemble de neurones liées de manière spécifique pour s'assurer que le couturier ne piquer pas son doigt avec une aiguille pointue. Au premier essai, il échouera puis peu à peu après plusieurs essais il saura comment éviter de se piquer tout en cousant des points uniformes avec efficacité.

La question est, qui, parmi ces neurones (ou connexions) ont été responsables de ses blessures pour les blâmer et permettre finalement de coudre aisément par la suite.

Dans un nouvel article , certains des esprits les plus brillants en intelligence artificielle - y compris le Dr Geoffrey Hinton, le parrain de l'apprentissage en profondeur , et les gens de DeepMind de Google soutiennent que les idées derrière un algorithme de base qui pilote l'apprentissage en profondeur fonctionnent également dans le cerveau biologique. L'algorithme, appelé rétro-propagation, a été l'étincelle qui a déclenché la révolution actuelle de l'apprentissage profond en tant que monstre de l'apprentissage automatique de facto. À sa base, le «back propagation» est un moyen extrêmement efficace d'attribuer le blâme aux connexions dans les réseaux de neurones artificiels et de conduire à de meilleurs résultats d'apprentissage. Bien qu'il n'y ait pas encore de preuve solide que le principe de cet algorithme fonctionne également dans le cerveau , les auteurs ont avancé plusieurs idées que les neuroscientifiques pourraient potentiellement tester dans des tissus cérébraux vivants.

Imaginons qu'un réseau de neurones est un peu comme une démocratie avec des individus qui ne sont en contact qu'avec leurs voisins. Tout neurone reçoit uniquement des données de son partenaire en amont et transmet des informations à ses partenaires en aval. Dans le langage des neurosciences, la force de ces connexions dépend des «poids synaptiques» - pensez-y comme d'une poignée de main plus ferme ou plus lâche, ou d'un transfert d'informations. Le principal objectif de l'apprentissage est de «régler» en quelque sorte le poids de l'ensemble de la population afin que le résultat final soit celui que nous voulons, c'est-à-dire la couture de tissus plutôt que la piqûre du doigt.

La réalité est que dans les réseaux de neurones profonds, l'apprentissage en suivant le gradient d'une mesure de performance fonctionne efficacement, ont déclaré les auteurs. Notre seul

souci est de montrer qu'il y a une chance que les idées derrière le back propagation fonctionnent également pour le cerveau?

II.1.1 L'intelligence artificiel, l'apprentissage automatique et l'apprentissage profond :

Le développement dans le domaine de la technologie c'est amélioré au fil des années. Avec le temps, nous obtenons des termes comme l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond en technologie. Nous confondons souvent ces termes et les définissons de manière similaire. Mais ce n'est pas une définition précise car ces termes sont différents les uns des autres. Dans ce chapitre, nous allons essayer de voir la différence entre ces trois termes AI, ML et Deep Learning.

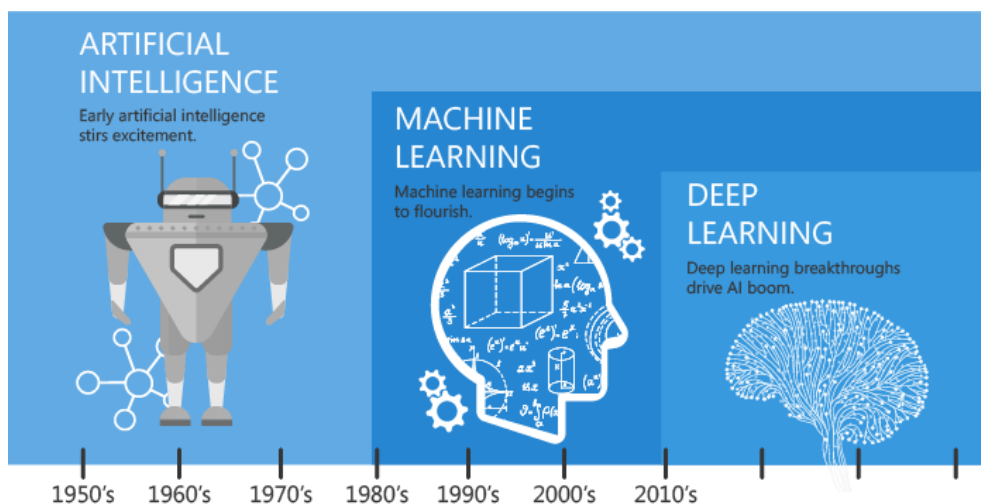


Figure II.1:différence entre AI, ML et Deep Learning [8]

II.1.1.1 Intelligence artificielle :

L'intelligence artificielle, comme son nom l'indique c'est l'intelligence créée par les humains. En effet, ce sont des machines complexes utilisant des propriétés informatiques et effectuant diverses actions comme nous, les humains. En un mot, il intègre l'intelligence humaine dans des machines.

II.1.1.2 Apprentissage automatique :

L'apprentissage automatique fait partie de l'intelligence artificielle. La plupart des gens le considèrent comme une intelligence artificielle, mais ce n'est pas le cas. Les machines peuvent apprendre. Les robots apprennent d'eux-mêmes à partir des données qui leur sont

fournies. Cela ressemble plus à une technique qui nous fait prendre conscience de la présence de l'Intelligence Artificielle. Cette technique utilise des algorithmes pour obtenir des données, apprendre, puis analyser les données. Les résultats sont obtenus sous forme de prédictions. Vous l'avez peut-être remarqué lorsque vous avez reçu des recommandations sur des sites commerciaux, Google ou Facebook. Vous obtenez des suggestions en fonction de vos intérêts. Cela se fait avec des algorithmes d'apprentissage automatique qui sont développés de manière à analyser les recherches récentes, l'historique et d'autres informations. Cette technique influence également les secteurs marketing et bancaire.

«L'apprentissage automatique est la tendance des machines à tirer des enseignements de l'analyse des données et à atteindre l'intelligence artificielle.»

Les nouveaux algorithmes d'apprentissage automatique étaient limités à l'IA de base, mais maintenant ils sont devenus une partie essentielle de ce système. De nombreux algorithmes complexes sont préparés pour offrir une meilleure expérience. Cela a révolutionné la façon de regarder des émissions et des films. L'industrie du divertissement utilise cet algorithme pour fournir des suggestions appropriées à ses téléspectateurs sur des chaînes Web comme Netflix et Amazon Prime. L'apprentissage automatique est le concept d'analyse des données et offre d'excellentes recommandations basées sur l'apprentissage à partir de ces points [8].

II.1.1.3L'apprentissage en profondeur

C'est le sous-ensemble de l'apprentissage automatique, ou on peut dire l'intelligence artificielle, qui est la raison derrière les capacités de travail des machines. Cette technique est similaire à l'apprentissage automatique dans certains contextes. La différence entre ces deux éléments est que l'apprentissage automatique a besoin de conseils pour effectuer une tâche, tandis que l'apprentissage en profondeur du modèle le fera lui-même sans l'interférence du programmeur. Le Deep Learning a amélioré l'expertise des utilisateurs. Le meilleur exemple d'apprentissage en profondeur est une voiture automatique.

«La technique utilisée pour implémenter le machine learning est connue sous le nom de deep learning.»

L'apprentissage en profondeur a permis aux machines de fonctionner et de penser comme des humains. Dans l'apprentissage automatique, les programmeurs doivent corriger l'algorithme si les résultats sont inappropriés. Mais les modèles d'apprentissage profond font ceux-là tout comme le cerveau humain.

II .1.2 Différence entre machine learning et le deep learning

Le deep learning et le machine learning sont deux concepts liés à l'intelligence artificielle. Les deux se sont combinés pour améliorer l'avenir de l'IA, mais ce n'est pas de l'intelligence artificielle. Ils sont différents à bien des égards qui doivent tenir compte pour créer l'AI qui est encore mieux que le cerveau humain. Lorsque les machines font preuve d'intelligence humaine, cela s'appelle l'intelligence artificielle. L'apprentissage automatique des perspectives pour obtenir l'intelligence artificielle et sa mise en œuvre est appelé apprentissage en profondeur. C'est la différence que chacun doit connaître au moment d'utiliser ces termes. Maintenant, c'est la différence fondamentale que ces trois termes ont entre eux.

Une des grandes différences entre le Deep Learning et les algorithmes de ML traditionnelles c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleurs. Contrairement à plusieurs algorithmes de ML classiques qui est limitée par la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", les modèles de Deep Learning n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme l'image processing.

Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme [9].

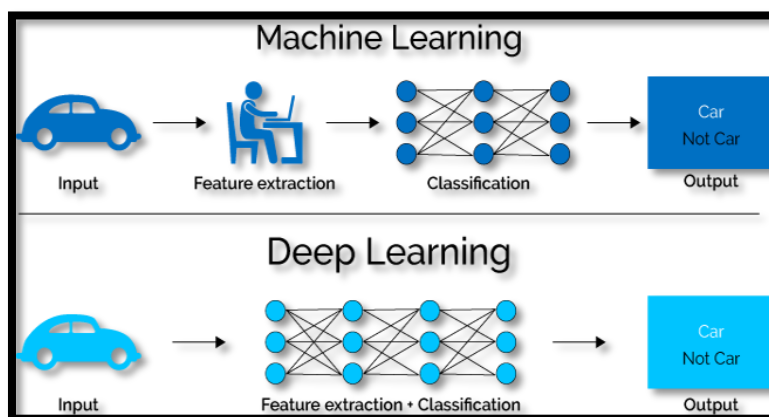


Figure II.2:différence entre L'apprentissage profond et L'apprentissage automatique [9]

II .2. Deep learning (l'apprentissage profond) :

Au milieu des années 80 et au début des années 90, de nombreux progrès architecturaux importants ont été réalisés dans les réseaux de neurones. Toutefois, le temps et les données nécessaires pour obtenir de bons résultats ont ralenti leur adoption, ce qui a réduit leur intérêt. Au début des années 2000, la puissance de calcul s'est accrue de façon exponentielle et l'industrie a assisté à une explosion de techniques de calcul qui. De ce fait, L'apprentissage en profondeur a émergé de la croissance informatique explosive de cette décennie en tant que concurrent sérieux sur le terrain, remportant de nombreux concours d'apprentissage automatique importants. L'intérêt n'a pas ralenti à partir de 2017.

L'apprentissage profond est l'un des domaines les plus en vogue dans le domaine de la science des données. De nombreuses études de cas ont donné des résultats étonnants dans les domaines de la robotique, de la reconnaissance des formes et de l'intelligence artificielle (IA).

Plusieurs publications ont énormément influencé le développement du domaine:

- Apprentissage basé sur le gradient appliqué par la NYU à la reconnaissance de documents (1998), qui introduit le réseau de neurones convolutionnels dans le monde de l'apprentissage automatique [10].
- Deep Boltzmann Machines de Toronto (2009), qui présente un nouvel algorithme d'apprentissage pour les machines Boltzmann, qui contient de nombreuses couches cachées [11].
- Stanford & Google a créé des fonctionnalités de haut niveau à l'aide de l'apprentissage non supervisé à grande échelle (2012), qui résout le problème de la création de détecteurs de fonctionnalités de haut niveau et spécifiques à une classe à partir de données non étiquetées uniquement [15].
- DeCAF - Une fonctionnalité d'activation par convolution profonde de Berkeley pour la reconnaissance visuelle générique (2013), qui publie DeCAF, une implémentation à code source ouvert des fonctionnalités d'activation par convolution profonde, ainsi que tous les paramètres de réseau associés permettant aux chercheurs en vision de mener des expériences avec représentations à travers une gamme de paradigmes d'apprentissage de concepts visuels [16].

- Deep Ateliers de DeepMind avec Deep Reinforcement Learning (2016), qui présente le premier modèle d'apprentissage en profondeur permettant de maîtriser les politiques de contrôle directement à partir de données sensorielles de haute dimension utilisant l'apprentissage par renforcement [17].

Dans ce chapitre nous allons présenter les bases de l'apprentissage en profondeur.

➤ Le neurone

Le neurone biologique (voir figure II.3) est une cellule nerveuse qui constitue l'unité fonctionnelle fondamentale du système nerveux de tous les animaux. Les neurones existent pour communiquer les uns avec les autres et transmettre des impulsions électrochimiques à travers les synapses, d'une cellule à l'autre, à condition que l'impulsion soit suffisamment puissante pour activer la libération de produits chimiques à travers une fente synaptique. La force de l'impulsion doit dépasser un seuil minimal, sinon les produits chimiques ne seront pas libérés.

La figure présente les principales parties de la cellule nerveuse : le soma, les dendrites, les axones et les synapses.

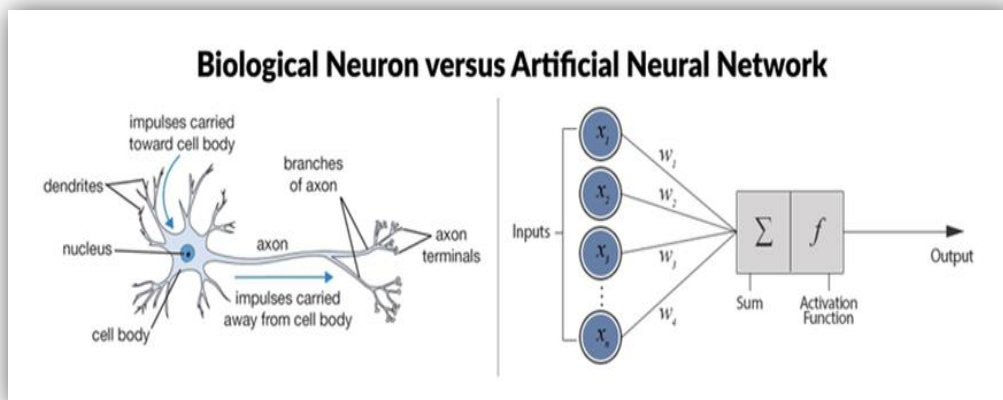


Figure II.3:neurone artificiel et biologique [14]

Le neurone est constitué d'une cellule nerveuse constituée d'un soma (corps cellulaire) et de plusieurs dendrites mais un seul axone. L'axone unique peut cependant se ramifier des centaines de fois. Les dendrites sont des structures minces issues du corps cellulaire principal. Les axones sont des fibres nerveuses ayant une extension cellulaire particulière qui provient du corps de la cellule.

➤ **Flux d'informations à travers le neurone biologique**

Les synapses qui augmentent le potentiel sont considérées comme excitatrices, et celles qui le diminuent sont considérées comme inhibitrices. La plasticité fait référence aux changements à long terme de la force des connexions en réponse au stimulus d'entrée. Les neurones se sont également avérés former de nouvelles connexions au fil du temps et même migrer. Ces mécanismes combinés au changement de connexion dirigent le processus d'apprentissage dans le cerveau biologique.

➤ **De biologique à artificiel**

Il a été démontré que le cerveau de l'animal est responsable des composants fondamentaux de l'esprit. Nous pouvons étudier les composants de base du cerveau et les comprendre. La recherche a montré des moyens de cartographier les fonctionnalités du cerveau et de suivre les signaux lorsqu'ils se déplacent dans les neurones.

➤ **Neurone artificielle**

Les neurones artificiels peuvent être définis par le type d'entrée qu'ils sont capables de recevoir (binaire ou continue) et par le type de transformation (fonction d'activation) qu'ils utilisent pour produire une sortie.

Différents éléments caractéristiques importantes de ces neurones sont

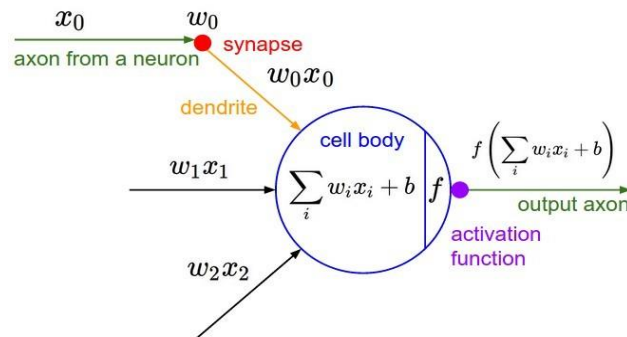


Figure II.4: Un neurone artificiel [14]

➤ **Poids de connexion**

Les poids sur les connexions dans un réseau de neurones sont des coefficients qui modifient (amplifient ou minimisent) le signal entrant dans un neurone donné du réseau.

➤ **Les biais**

Les biais sont des valeurs scalaires ajoutées à l'entrée pour garantir qu'au moins quelques nœuds par couche sont activés, quelle que soit la puissance du signal. Les biais permettent l'apprentissage en donnant au réseau une action en cas de faible signal. Ils permettent au réseau d'essayer de nouvelles interprétations ou comportements [18].

II.3 Présentation des réseaux de neurones artificiels :

Un réseau de neurones artificiels est une tentative de simulation du réseau de neurones constituant le cerveau humain, de sorte que l'ordinateur puisse apprendre des choses et prendre des décisions de manière humaine. Les réseaux de neurones artificiels sont créés en programmant des ordinateurs ordinaires pour qu'ils se comportent comme des cellules cérébrales interconnectées.

Dans un réseau de neurones artificiels, nous avons des neurones artificiels disposés en groupes appelés couches comme suit :

- une couche d'entrée ;
- une ou plusieurs couches cachées, entièrement connectées ;
- une seule couche de sortie

Comme le montre la figure 5, les neurones de chaque couche (représentés par les cercles) sont entièrement connectés à tous les neurones de toutes les couches adjacentes. Pour la couche d'entrée, l'entrée est l'entrée vectorielle brute. L'entrée des neurones des autres couches est la sortie (activation) des neurones de la couche précédente.

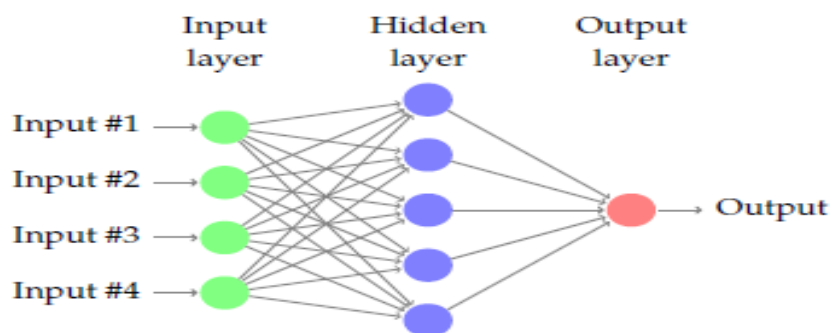


Figure II.5:Réseau de neurones artificiels [15]

II.3 .1 Couche d'entrée :

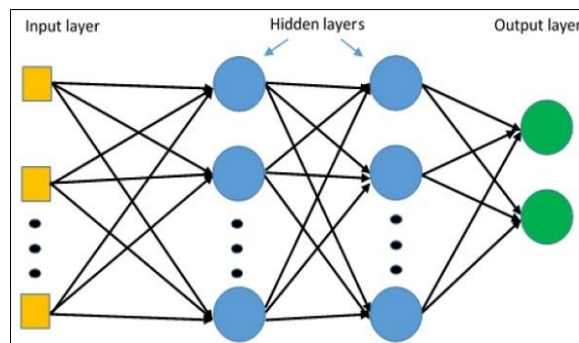
La couche d'entrée rassemble les données d'entrée dans le réseau neuronal. Les données d'entrée doivent être numériques. Le processus de manipulation des données avant leur entrée dans le réseau neuronal est appelé **traitement des données**.

II .3.2 Couche cachée:

Les couches cachées sont composées de la plupart des neurones du réseau neuronal et constituent le cœur de la manipulation des données pour obtenir la sortie souhaitée. Les données traverseront les couches cachées et seront manipulées par de nombreux poids et biais.

II.3.3 Couche de sortie:

La couche de sortie est le produit final de la manipulation des données dans le réseau neuronal. Souvent, la couche de sortie est constituée de neurones qui représentent chacun un objet et la valeur numérique attachée est la probabilité de cet objet spécifique. L'idée principale est que les couches de sortie sont le résultat des données lors de leur passage à travers le réseau neuronal, et le but à atteindre.



II.3.4. Connexions entre les couches

Dans un réseau entièrement connecté, les connexions entre les couches sont les connexions sortantes de tous les neurones de la couche précédente à tous les neurones de la couche suivante.

Les poids seront progressivement modifiés au fur et à mesure que notre algorithme trouve la meilleure solution possible durant la phase d'apprentissage de la rétropropagation.

Les poids peuvent être considérés comme une optimisation du vecteur de paramètres afin de minimiser les erreurs.

II.4. Les fonctions d'activation:

La fonction d'activation est une composante essentielle d'un réseau neuronal. Cette fonction décide si le neurone sera activé ou non. C'est une transformation non linéaire de la valeur d'entrée.

La non-linéarité est si importante dans les réseaux de neurones. Il existe de nombreux types de ces fonctions, parmi lesquelles nous trouvons [19]:

II. 4.1. La fonction Sigmoid:

Cette fonction est l'une des plus couramment utilisées. Elle est bornée entre 0 et 1, et elle peut

être interprétée stochastiquement comme la probabilité que le neurone s'active, et elle est généralement appelée la fonction logistique ou le sigmoïde logistique.

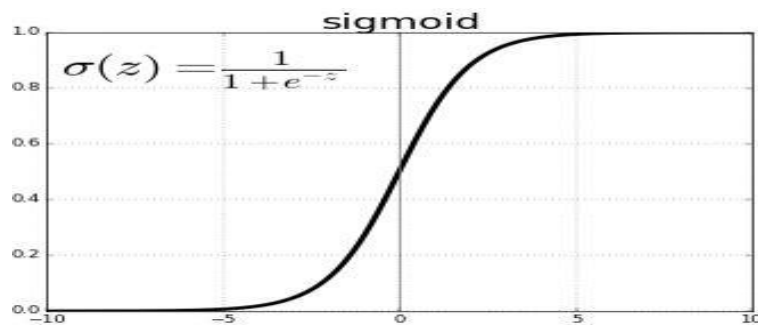


Figure II.6 : Représentation graphique de la fonction Sigmoïde [15].

II.4.2. La fonction ReLU:

La fonction RELU est probable mental a plus proche de sa correspondante biologique.

Cette fonction est récemment de venue le choix de nombreuses tâches (notamment en computer vision).

Comme dans la formule Ci-dessus, cette fonction renvoie 0 si l'entrée z est inférieure à 0 être tournez lui- mêmes 'il est plus grand que 0.

Dans [15], nous avons un exemple de reconnaissance

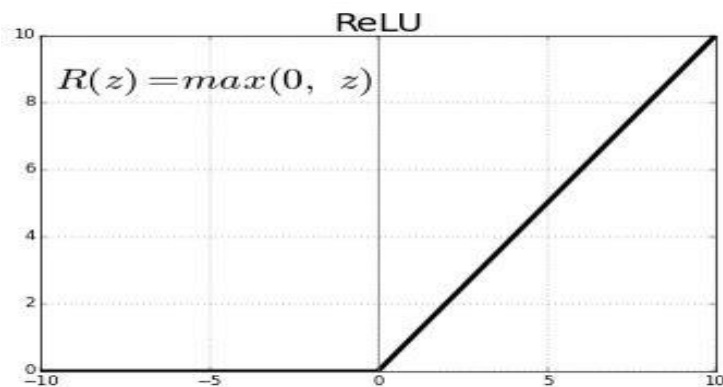


Figure II.7: Représentation graphique de la fonction Relu [15]

II.4.3 La fonction Softmax:

Régression Softmax (synonymes : Logistique multinomiale, Classificateur d'entropie maximale, ou simplement Régression logistique multi-classe) est une généralisation de la régression logistique que nous pouvons utiliser pour la classification multi-classes [20]. Contrairement à d'autres types de fonctions, la sortie d'un neurone d'une couche utilisant la fonction soft max dépend des sorties de tous les autres neurones des a couche.

Cela s'explique par le fait qu'il nécessite que la somme de toutes les sorties soit égale à 1.

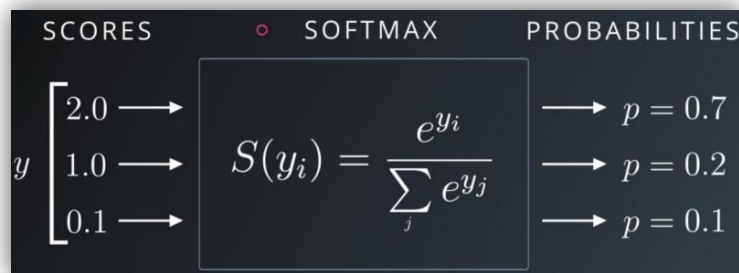


Figure II.8: fonction softmax [17]

II.5. Les architectures des réseaux de neurones : La plupart des architectures profondes sont réalisées en combinant et recombinaison un ensemble limité de primitives architecturales (des couches de réseaux neuronaux) [21]. Selon Hinton, dans [22], nous disons que notre réseau de neurones est profond lorsque le nombre des couches cachées est supérieur à 1. Dans cette section, nous présenterons un bref aperçu des structures communes que l'on retrouve dans de nombreux réseaux profonds.

II.6. Les réseaux entièrement connectés:

Un réseau entièrement connecté (Fully Connected en Anglais) permet de transformer une liste d'entrées en une liste de sorties. La transformation est appelée totalement connectée car toute valeur d'entrée peut affecter toute valeur de sortie. Dans ce type de réseaux, les activités des neurones de chaque couche sont une fonction non-linéaire des activités de la couche inférieure.

II.7 Les réseaux neuronaux récurrents et LSTM:

Les couches de réseaux neuronaux récurrentes (RNN : Récurrent Neural Networks) sont des entités primitives qui permettent aux réseaux neuronaux d'apprendre à partir de séquences d'entrées. La figure représente les types de séquences d'entrée que le réseau traite:

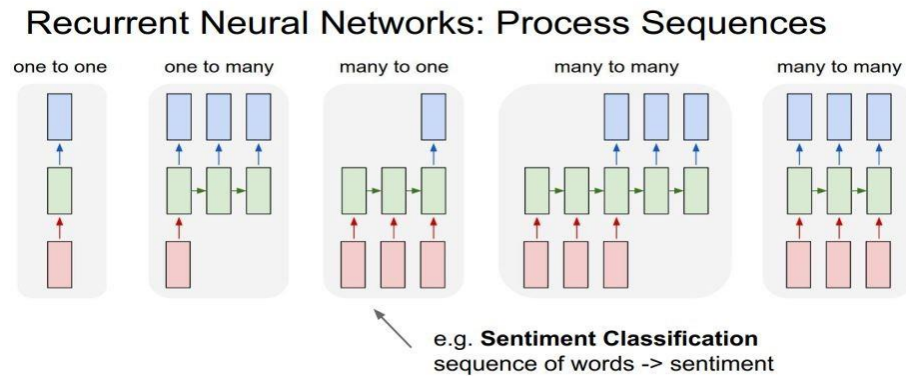


Figure II.9: Les types de séquences d'entrée pour un réseau récurrent [19]

II.8 Les réseaux de neurones convolutifs (CNN) :

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes.

En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris.

La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive à proprement parler.

Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

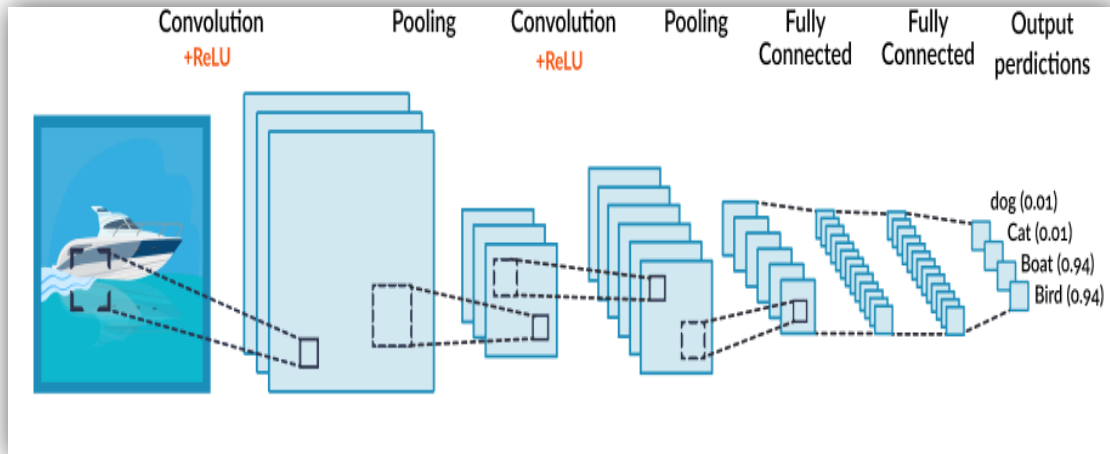


Figure II.10 : Réseau de neurones a convolution (CNN) [19]



CHAPITER III

L'APPRENTISSAGE PAR TRANSFERT

III.1 Introduction :

La motivation pour l'apprentissage par transfert utilisé pour le Machine Learning et le Deep Learning est basée sur le fait que les développeurs peuvent appliquer intelligemment les connaissances acquises précédemment pour une tâche ou un domaine spécifique qui peuvent être utilisés pour résoudre de nouveaux problèmes plus rapidement ou avec de meilleures solutions.

L'apprentissage par transfert est l'approche dans laquelle les connaissances acquises dans une ou plusieurs tâches source sont transférées et utilisées pour améliorer l'apprentissage d'une tâche cible connexe.

Habituellement, les réseaux de neurones convolutionnels sont des réseaux très profonds avec beaucoup de couches convolutionnelles pour extraire des caractéristiques des images. De plus, ces réseaux sont formés sur d'énormes ensembles de données qui se composent de millions d'images. ImageNet qui contient 1,2 million d'images avec 1000 catégories. Il n'est pas toujours pratique d'obtenir de tels énormes ensembles de données. De plus, les réseaux profonds peuvent prendre des jours ou des semaines d'apprentissage sur de telles données si vous les exécutez sur du matériel spécialisé comme un GPU.

C'est là que nous pouvons utiliser les connaissances acquises par un réseau qui a été formé pour une tâche particulière et le déployer pour une autre tâche. Vous n'avez pas besoin de construire un modèle à partir de zéro.

III.2 apprentissage par transfert :

L'apprentissage par transfert fait généralement référence à un processus ou un modèle formé sur un problème est utilisé d'une manière ou d'une autre pour un deuxième problème connexe.

Dans l'apprentissage en profondeur, l'apprentissage de transfert est une technique par laquelle un modèle de réseau de neurones est d'abord formé sur un problème similaire au problème qui est résolu. Une ou plusieurs couches du modèle formé sont ensuite utilisées dans un nouveau modèle formé sur le problème d'intérêt.

L'apprentissage par transfert est une méthode populaire en vision par ordinateur car elle nous permet de construire des modèles précis en économisant du temps. Avec l'apprentissage par transfert, au lieu de démarrer le processus d'apprentissage à partir de zéro, vous partez de

modèles qui ont été appris lors de la résolution d'un problème différent. De cette façon, vous tirez parti des apprentissages précédents et évitez de recommencer à zéro.

L'apprentissage par transfert a l'avantage de réduire le temps de formation pour un modèle de réseau neuronal et peut entraîner une erreur de généralisation plus faible.

Les poids des couches réutilisées peuvent être utilisés comme point de départ du processus de formation et adaptés en réponse au nouveau problème. Cette utilisation traite l'apprentissage par transfert comme un type de schéma d'initialisation de poids. Cela peut être utile lorsque le premier problème connexe contient beaucoup plus de données étiquetées que le problème d'intérêt et que la similitude dans la structure du problème peut être utile dans les deux contextes.

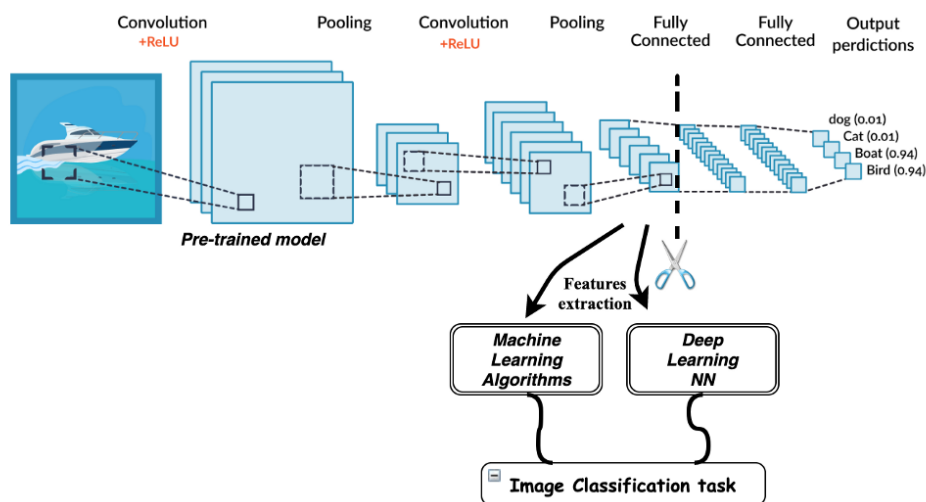


Figure III .1 : architecture d'un réseau d'apprentissage de transfert [26]

III .3 L'architecture d'un CNN comporte 2 parties :

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive proprement dit. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN [20].

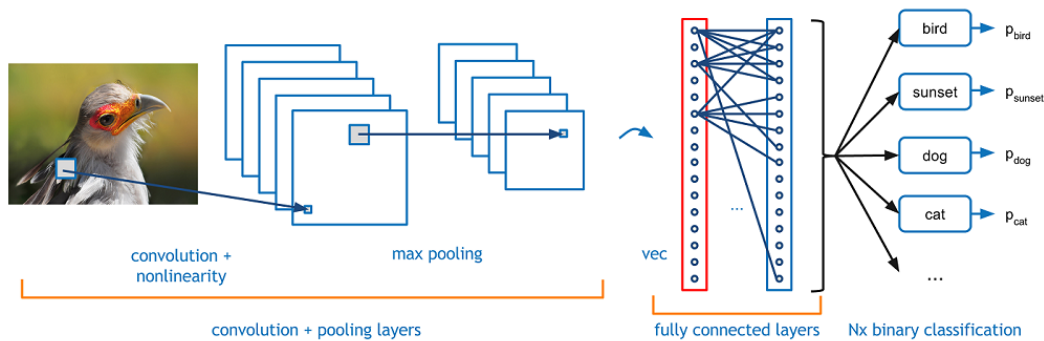


Figure III .2: Les réseaux de neurones convolutifs. [20]

La sortie du CNN de la partie convolutive est ensuite branchée en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

III .3.1 Difficulté d'entraînement du CNN

Créer un nouveau CNN est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires.

Il s'agit d'abord de fixer l'architecture du réseau, c'est-à-dire le nombre de couches, leurs tailles et les opérations matricielles qui les connectent. L'entraînement consiste alors à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées.

Des équipes de recherche se spécialisent dans l'amélioration des CNN. Elles publient leurs innovations techniques, ainsi que le détail des réseaux entraînés sur des bases de données de

références. Le challenge ImageNet (ILSVRC) fournit par exemple 1.2 millions d'images classées en 1000 catégories. Ces CNN entraînés sont disponibles nativement dans la librairie Keras, ou sont regroupés dans des dépôts git hub pour d'autres librairies (par exemple Model Zoo pour Café) [20].

III.4 réseaux de neurones convolutifs(CNN)

Les (ConvNets ou CNN) sont une catégorie de réseaux de neurones qui se sont révélés très efficaces dans des domaines tels que la reconnaissance et la classification d'images. Les ConvNets ont réussi à identifier les visages, les objets et les panneaux de signalisation en plus d'alimenter la vision des robots et des voitures autonomes.

La **Figure III.3** montre un exemple de ConvNets utilisé pour reconnaître des objets du quotidien, les humains et les animaux. Dernièrement, ConvNets a également été efficace dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases).

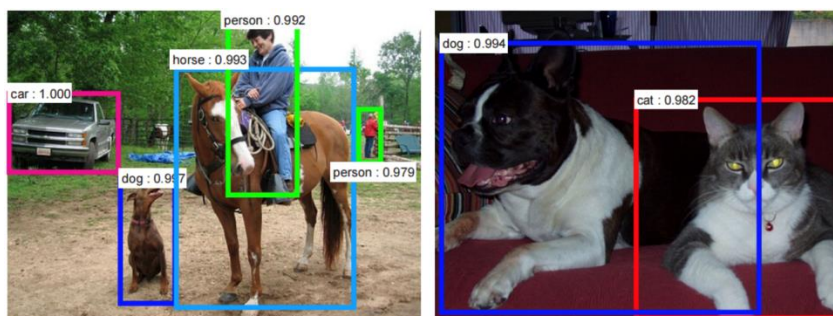


Figure III.3 : exemple de ConvNets utilisé pour reconnaître des objets [21]

Les ConvNets sont donc aujourd'hui un outil important pour la plupart des praticiens de l'apprentissage automatique. Cependant, comprendre les ConvNets et apprendre à les utiliser pour la première fois peut parfois être une expérience intimidante. Le principal objectif de ce billet de blog est de développer une compréhension du fonctionnement des réseaux de neurones convolutionnels sur les images [30].

LeNet a été l'un des tout premiers réseaux de neurones convolutifs qui a contribué à propulser le domaine du Deep Learning. Ce travail pionnier de Yann LeCun a été nommé LeNet5 après de nombreuses itérations réussies antérieures depuis 1988 [31]. À cette époque, l'architecture LeNet était principalement utilisée pour des tâches de reconnaissance de caractères telles que la lecture de codes postaux, de chiffres, etc.

Ci-dessous, nous développerons une intuition de la façon dont l'architecture LeNet apprend à reconnaître les images. Plusieurs nouvelles architectures ont été proposées au cours des dernières années qui sont des améliorations par rapport à LeNet, mais elles utilisent toutes les principaux concepts de LeNet et relativement plus faciles à comprendre si vous avez une compréhension claire de la première.

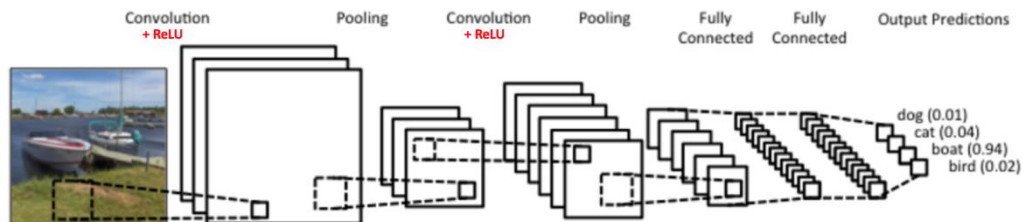


Figure III.4 : Un simple ConvNet[21]

Le réseau neuronal convolutif de la **figure III.4** est similaire en architecture au LeNet d'origine et classe une image d'entrée en quatre catégories: chien, chat, bateau ou oiseau (le LeNet original a été utilisé principalement pour des tâches de reconnaissance de caractères). Comme la montre la figure ci-dessus, lors de la réception d'une image de bateau en entrée, le réseau attribue correctement la probabilité la plus élevée de bateau (0,94) parmi les quatre catégories. La somme de toutes les probabilités dans la couche de sortie doit être une (expliquée plus loin dans cet article).

Il y a quatre opérations principales dans le ConvNet montré dans la **figure III.4** ci-dessus:

- Convolution
- Non linéarité (ReLU)
- Regroupement ou sous-échantillonnage
- Classification (couche entièrement connectée)

Ces opérations sont les éléments de base de *chaque* réseau de neurones convolutionnels, donc comprendre comment ces travaux sont une étape importante pour développer une bonne compréhension des ConvNets. Nous allons essayer de comprendre l'intuition derrière chacune de ces opérations ci-dessous. Les images sont une matrice de valeurs de pixels essentiellement, chaque image peut être représentée comme une matrice de valeurs de pixels.

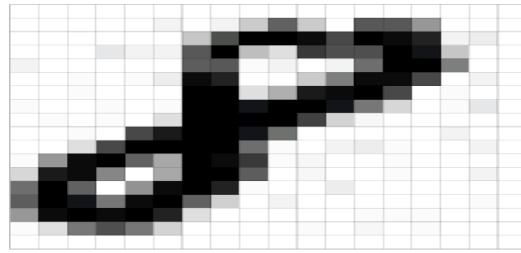


Figure III.5 : chaque image est une matrice de valeurs de pixels.

Le canal est un terme conventionnel utilisé pour désigner une certaine composante d'une image. Une image provenant d'un appareil photo numérique standard aura trois canaux - rouge, vert et bleu - vous pouvez les imaginer comme trois matrices 2d empilées les unes sur les autres (une pour chaque couleur), chacune ayant des valeurs de pixels dans la plage de 0 à 255.

Une image en niveaux de gris, en revanche, n'a qu'un seul canal. Aux fins de cet article, nous ne considérerons que les images en niveaux de gris, nous aurons donc une seule matrice 2d représentant une image. La valeur de chaque pixel de la matrice sera comprise entre 0 et 255 - zéro indiquant le noir et 255 indiquant le blanc.

III.5.1 L'étape de convolution :

Les ConvNets tirent leur nom de l'opérateur de «convolution». Le but principal de Convolution dans le cas d'un ConvNet est d'extraire des caractéristiques de l'image d'entrée. La convolution préserve la relation spatiale entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés de données d'entrée. Nous n'entrerons pas dans les détails mathématiques de la convolution ici, mais nous essaierons de comprendre comment cela fonctionne sur les images.

Comme nous l'avons vu ci-dessus, chaque image peut être considérée comme une matrice de valeurs de pixels. Considérons une image 5 x 5 dont les valeurs de pixels ne sont que de 0 et 1 (notez que pour une image en niveaux de gris, les valeurs de pixels vont de 0 à 255, la matrice verte ci-dessous est un cas spécial où les valeurs de pixels ne sont que de 0 et 1):

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Envisagez également une autre matrice 3 x 3, comme indiqué ci-dessous:

1	0	1
0	1	0
1	0	1

Ensuite, la convolution de l'image 5 x 5 et de la matrice 3 x 3 peut être calculée comme indiqué dans l'animation de la **Figure III.6** ci-dessous:

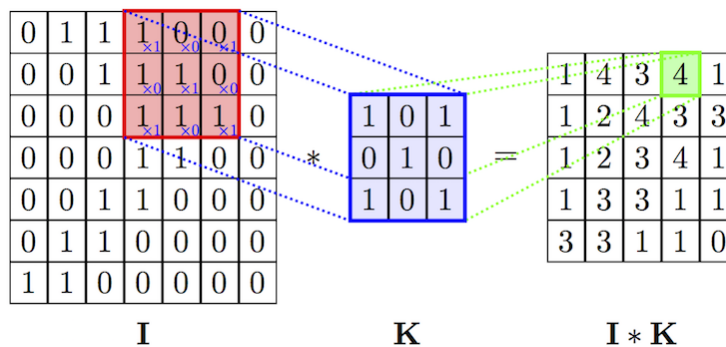


Figure III.6 :L'opération de convolution. La matrice de sortie est appelée entité convolutive ou Carte d'entités. [27]

Prenez un moment pour comprendre comment se fait le calcul ci-dessus. Nous glissons la matrice orange sur notre image d'origine (verte) de 1 pixel (également appelée "foulée") et pour chaque position, nous calculons la multiplication par élément (entre les deux matrices) et ajoutons les sorties de multiplication pour obtenir l'entier final qui forme un seul élément de la matrice de sortie (rose). Notez que la matrice 3 x 3 ne «voit» qu'une partie de l'image d'entrée dans chaque foulée.

Dans la terminologie CNN, la matrice 3 x 3 est appelée « filtre » ou «noyau» ou «détecteur de caractéristiques» et la matrice formée en faisant glisser le filtre sur l'image et en calculant le produit scalaire est appelée «fonction convolutive» ou «activation». Carte "ou" Carte des entités ". Il est important de noter que les filtres agissent comme des détecteurs de caractéristiques de l'image d'entrée d'origine.

Une autre bonne façon de comprendre l'opération de convolution est de regarder l'animation de la **Figure III.7** ci-dessous:



Figure III.7 :L'opération de convolution.

Un filtre (avec contour rouge) glisse sur l'image d'entrée (opération de convolution) pour produire une carte d'entités. La convolution d'un autre filtre (avec le contour vert), sur la même image, donne une carte des fonctionnalités différente, comme indiqué. Il est important de noter que l'opération de convolution capture les dépendances locales dans l'image d'origine. Notez également comment ces deux filtres différents génèrent des cartes d'entités différentes à partir de la même image d'origine. N'oubliez pas que l'image et les deux filtres ci-dessus ne sont que des matrices numériques comme nous l'avons vu ci-dessus.

En pratique, un CNN apprend les valeurs de ces filtres par lui-même au cours du processus de formation. Plus nous avons de filtres, plus les caractéristiques des images sont extraites et mieux notre réseau parvient à reconnaître les motifs dans les images invisibles.

III .5.2 Profondeur:

La profondeur correspond au nombre de filtres que nous utilisons pour l'opération de convolution. Dans le réseau illustré à la **Figure III.8**, nous effectuons une convolution de l'image originale du bateau à l'aide de trois filtres distincts, produisant ainsi trois cartes de caractéristiques différentes, comme illustré. Vous pouvez considérer ces trois cartes d'entités comme des matrices 2D empilées, donc la «profondeur» de la carte d'entités serait de trois.

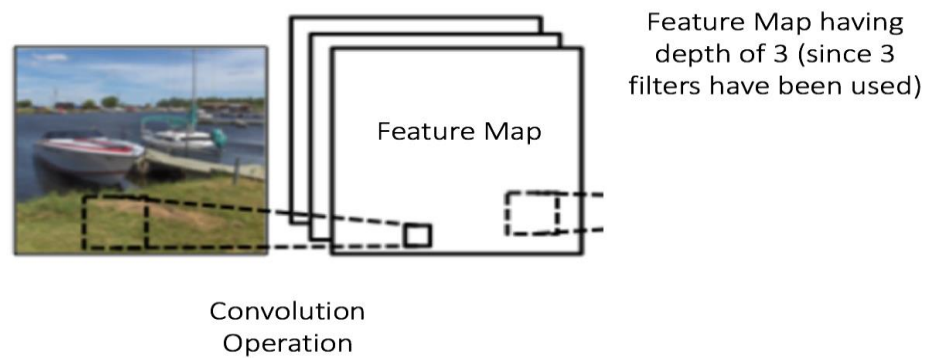


Figure III.8 : nombre de filtres

III .5.3 Stride:

Stride est le nombre de pixels par lesquels nous glissons notre matrice de filtre sur la matrice d'entrée. Lorsque la foulée est de 1, nous déplaçons les filtres un pixel à la fois. Lorsque la foulée est de 2, les filtres sautent de 2 pixels à la fois lorsque nous les faisons glisser. Une foulée plus grande produira des cartes d'entités plus petites.

III .5.4 Zéro-remplissage:

Parfois, il est pratique de remplir la matrice d'entrée avec des zéros autour de la bordure, afin que nous puissions appliquer le filtre aux éléments de bordure de notre matrice d'image d'entrée. Une caractéristique intéressante du remplissage nul est qu'il nous permet de contrôler la taille des cartes d'entités. Ajout de zéro padding est aussi appelé large convolution, et ne pas utiliser zéro-padding serait une convolution étroite. Cela a été expliqué clairement dans [27].

III .5.5 Présentation de la non linéarité (ReLU)

Une opération supplémentaire appelée ReLU a été utilisée après chaque opération de convolution de la **Figure III.9** ci-dessus. ReLU signifie Rectified Linear Unit et est une opération non linéaire. Sa sortie est donnée par:

Output = Max(zero, Input)

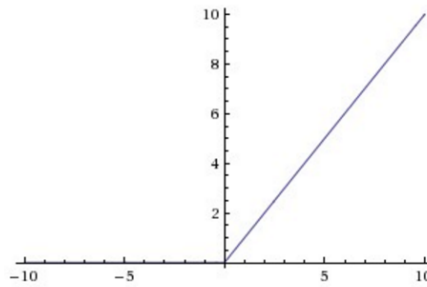


Figure III.9 :l'opération ReLU [27]

ReLU est une opération élément par élément (appliquée par pixel) et remplace toutes les valeurs de pixels négatives dans la carte d'entités par zéro. Le but de ReLU est d'introduire la non-linéarité dans notre ConvNet, car la plupart des données du monde réel que nous voudrions que notre ConvNet apprenne seraient non linéaires (la convolution est une opération linéaire - multiplication et addition matricielles par élément, donc nous tenir compte de la non-linéarité en introduisant une fonction non linéaire comme ReLU).

D'autres fonctions non linéaires telles que tanh ou sigmoïde peuvent également être utilisées à la place de ReLU, mais ReLU s'est révélé plus performant dans la plupart des situations.

III .5.6 L'étape de la mise en commun :

Le regroupement spatial réduit la dimensionnalité de chaque carte d'entités mais conserve les informations les plus importantes. La mise en commun spatiale peut être de différents types: max, moyenne, somme, etc.

Dans le cas de Max Pooling, nous définissons un voisinage spatial (par exemple, une fenêtre 2×2) et prenons le plus grand élément de la carte d'entités rectifiée dans cette fenêtre. Au lieu de prendre le plus grand élément, nous pourrions également prendre la moyenne (mise en commun moyenne) ou la somme de tous les éléments de cette fenêtre. En pratique, Max Pooling s'est avéré la meilleure.

Figure III.11 montre un exemple d'opération de regroupement maximal sur une carte d'entités rectifiées (obtenue après convolution + opération ReLU) à l'aide d'une fenêtre 2×2 .

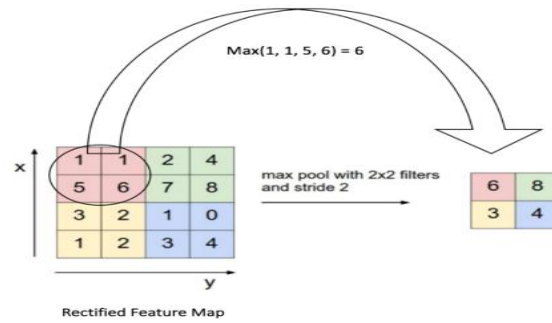


Figure III.10: mise en commun maximale. [27]

Nous glissons notre fenêtre 2 x 2 de 2 cellules (également appelée «foulée») et prenons la valeur maximale dans chaque région. Comme le montre la **Figure III.10**, cela réduit la dimensionnalité de notre carte d'entités.

Dans le réseau illustré à la **Figure III.11**, l'opération de regroupement est appliquée séparément à chaque carte d'entités (notez que, pour cette raison, nous obtenons trois cartes de sortie à partir de trois cartes d'entrée).

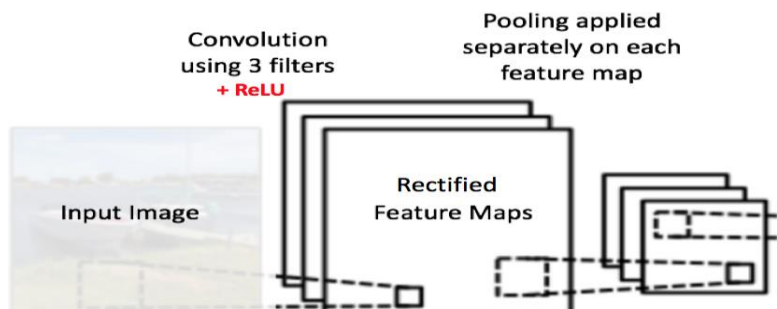


Figure III.11: regroupement appliqué aux cartes d'entités rectifiées

Figure III.12 montre l'effet du regroupement sur la carte d'entités rectifiée que nous avons reçue après l'opération ReLU de la **Figure III.10** ci-dessus.

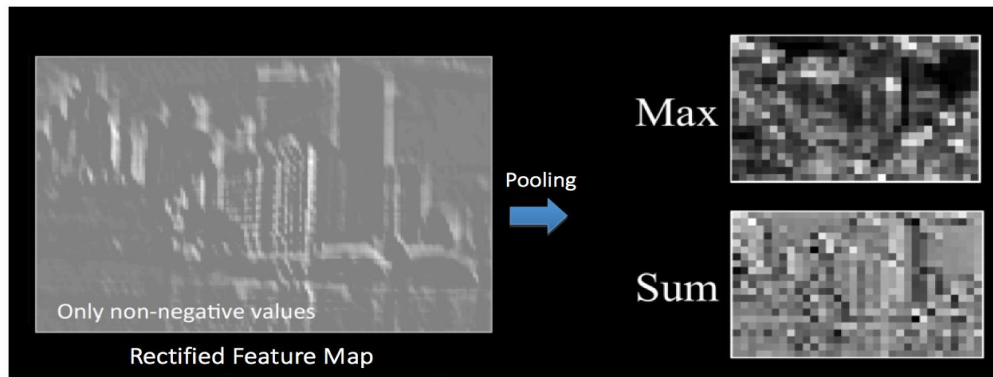


Figure III.12: Mise en commun. [27]

La fonction de Pooling est de réduire progressivement la taille spatiale de la représentation d'entrée [22]. En particulier, la mutualisation

- rend les représentations en entrée (dimension de l'objet) plus petites et plus faciles à gérer.
- réduit le nombre de paramètres et de calculs dans le réseau, contrôlant ainsi le surajustement [22].
- rend le réseau invariant aux petites transformations, distorsions et traductions dans l'image d'entrée (une petite distorsion en entrée ne changera pas la sortie de Pooling - puisque nous prenons la valeur maximale / moyenne dans un voisinage local).
- Nous aide à arriver à une représentation invariante à presque l'échelle de notre image (le terme exact est «équivalent»). Ceci est très puissant car nous pouvons détecter des objets dans une image, peu importe où ils se trouvent (lire [27] et [28] pour plus de détails).

III.6 le modèle CNN complet :

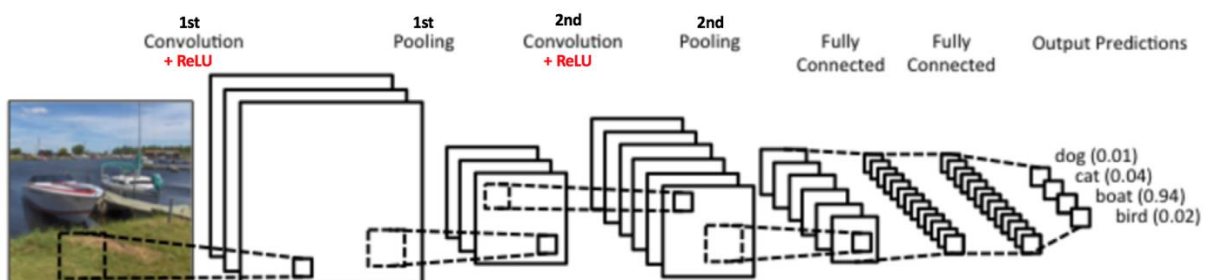


Figure III.13: modèle CNN [28]

Jusqu'à présent, nous avons vu comment fonctionnent la convolution, la ReLU et la mise en commun. Il est important de comprendre que ces couches sont les éléments de base de tout CNN. Comme le montre la **Figure III.13**, nous avons deux ensembles de couches Convolution, ReLU et Pooling - la 2e couche Convolution effectue une convolution sur la sortie de la première couche Pooling en utilisant six filtres pour produire un total de six cartes d'entités. ReLU est ensuite appliqué individuellement sur l'ensemble de ces six cartes d'entités. Nous effectuons ensuite l'opération Max Pooling séparément sur chacune des six cartes d'entités rectifiées.

Ensemble, ces couches extraient les caractéristiques utiles des images, introduisent la non-linéarité dans notre réseau et réduisent la dimension des caractéristiques tout en visant à rendre les caractéristiques quelque peu équivariant à l'échelle et à la traduction [28].

La sortie de la 2e couche de mise en commun agit comme une entrée pour la couche entièrement connectée, dont nous discuterons dans la section suivante.

III .6.1 Couche entièrement connectée :

La couche entièrement connectée est un Perceptron multicouche traditionnel qui utilise une fonction d'activation soft max dans la couche de sortie (d'autres classificateurs comme SVM peuvent également être utilisés, mais s'en tenir à softmax dans ce post). Le terme «entièrement connecté» implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante. Je recommande de lire cet article si vous n'êtes pas familier avec les Perceptrons multicouches.

La sortie des couches convolutionnelles et de mise en commun représente des caractéristiques de haut niveau de l'image d'entrée. Le but de la couche entièrement connectée est d'utiliser ces fonctionnalités pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage. Par exemple, la tâche de classification d'images que nous nous proposons de réaliser a quatre sorties possibles, comme le montre la **Figure III.15** ci-dessous (notez que la figure 14 ne montre pas les connexions entre les nœuds dans la couche entièrement connectée)

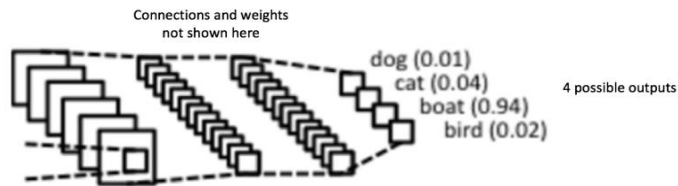


Figure III.14: couche entièrement connectée -chaque nœud est connecté à tous les autres nœuds de la couche adjacente

Outre la classification, l'ajout d'une couche entièrement connectée est également un moyen (généralement) bon marché d'apprendre des combinaisons non linéaires de ces fonctionnalités. La plupart des caractéristiques des couches convolutionnelles et de mise en commun peuvent être bonnes pour la tâche de classification, mais les combinaisons de ces caractéristiques peuvent être encore meilleures [25].

La somme des probabilités de sortie de la couche entièrement connectée est 1. Ceci est assuré en utilisant le Softmax comme fonction d'activation dans la couche de sortie de la couche entièrement connectée. La fonction Softmax prend un vecteur de scores arbitraires en valeur réelle et l'écrase en un vecteur de valeurs comprises entre zéro et un qui totalisent un.

III .6.2 Formation à l'aide de la rétropropagation

Comme indiqué ci-dessus, les couches Convolution + Pooling agissent comme des extracteurs de fonctionnalités à partir de l'image d'entrée tandis que la couche entièrement connectée agit comme un classificateur.

Notez que dans la **Figure III.15** ci-dessous, puisque l'image d'entrée est un bateau, la probabilité cible est de 1 pour la classe de bateau et de 0 pour les trois autres classes, c'est-à-dire

Image d'entrée = bateau

Vecteur cible = [0, 0, 1, 0]

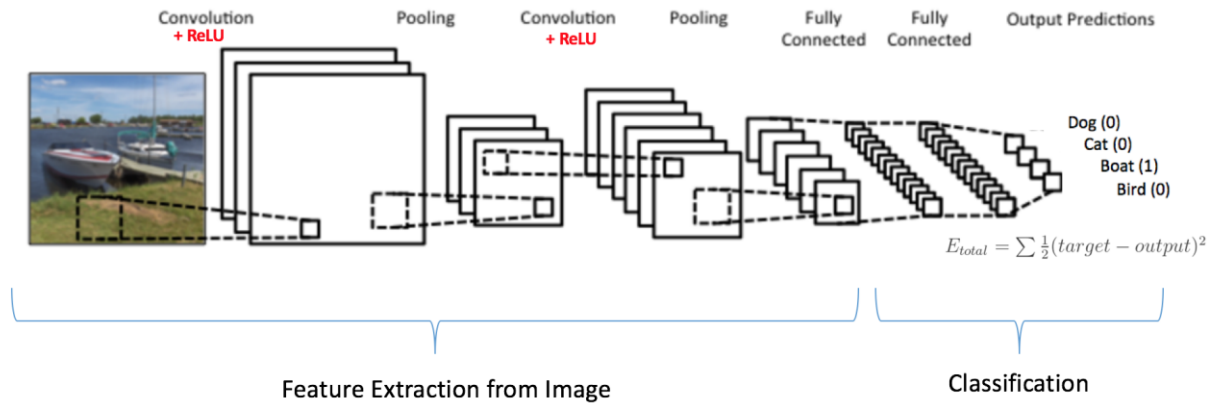


Figure III.15 : Formation du ConvNet

Le processus global de formation du réseau de convolution peut être résumé comme suit:

- **Étape 1:** nous initialisons tous les filtres et paramètres / poids avec des valeurs aléatoires
- **Étape 2:** Le réseau prend une image d'apprentissage en entrée, passe par l'étape de propagation directe (convolution, ReLU et opérations de mise en commun ainsi que la propagation directe dans la couche entièrement connectée) et trouve les probabilités de sortie pour chaque classe.

Disons que les probabilités de sortie pour l'image du bateau ci-dessus sont [0,2, 0,4, 0,1, 0,3]

Étant donné que les poids sont attribués au hasard pour le premier exemple d'apprentissage, les probabilités de sortie sont également aléatoires

- **Étape 3:** Calculez l'erreur totale au niveau de la couche de sortie (somme sur les 4 classes)

$$\text{Erreur totale} = \sum \frac{1}{2} (\text{probabilité cible} - \text{probabilité de sortie})^2$$

- **Étape 4:** utilisez la rétropropagation pour calculer les *gradients* de l'erreur par rapport à tous les poids du réseau et utilisez la *descente de gradient* pour mettre à jour toutes les valeurs / poids de filtre et les valeurs de paramètre pour minimiser l'erreur de sortie.

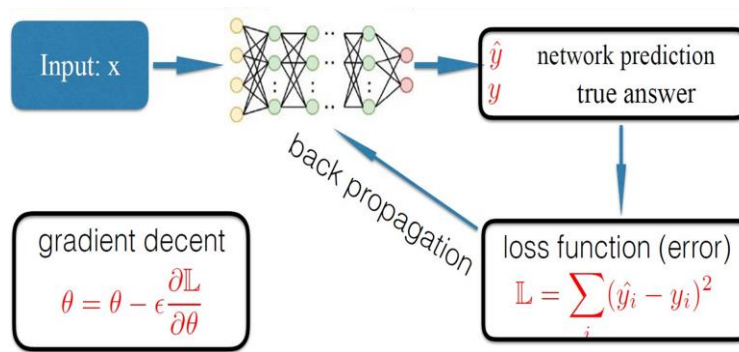


Figure III.16: Backpropagation algorithme (entrainement)[32]

Les poids sont ajustés proportionnellement à leur contribution à l'erreur totale.

Lorsque la même image est à nouveau entrée, les probabilités de sortie peuvent désormais être $[0,1, 0,1, 0,7, 0,1]$, ce qui est plus proche du vecteur cible $[0, 0, 1, 0]$.

Cela signifie que le réseau a *appris* à classer correctement cette image particulière en ajustant ses poids / filtres de sorte que l'erreur de sortie soit réduite.

Des paramètres tels que le nombre de filtres, la taille des filtres, l'architecture du réseau, etc. ont tous été fixés avant l'étape 1 et ne changent pas pendant le processus de formation - seules les valeurs de la matrice de filtre et les poids de connexion sont mis à jour.

➤ **Étape 5:** Répétez les étapes 2 à 4 avec toutes les images du jeu d'entraînement.

Les étapes ci-dessus *entraînent* le ConvNet - cela signifie essentiellement que tous les poids et paramètres du ConvNet ont maintenant été optimisés pour classer correctement les images de l'ensemble d'entraînement.

Lorsqu'une nouvelle image (invisible) est entrée dans le ConvNet, le réseau passe par l'étape de propagation directe et génère une probabilité pour chaque classe (pour une nouvelle image, les probabilités de sortie sont calculées en utilisant les poids qui ont été optimisés pour classer correctement tous les exemples de formation précédents). Si notre ensemble de formation est suffisamment grand, le réseau générera bien des nouvelles images et les classera en catégories correctes.

La Figure est l'organigramme qui comprend toutes les étapes de l'apprentissage du système:

Remarque 1: Les étapes ci-dessus ont été simplifiées à l'excès et les détails mathématiques ont été évités pour fournir une intuition dans le processus de formation. Voir [22] et [25] pour une formulation mathématique et une compréhension approfondie.

III.7 Autres architectures ConvNet :

Les réseaux de neurones convolutifs existent depuis le début des années 1990. Nous avons discuté du LeNet ci-dessus qui était l'un des tout premiers réseaux de neurones convolutionnels. Quelques autres architectures influentes sont énumérées ci-dessous [20] [22].

➤ **LeNet (1990):** déjà couvert dans ce chapitre.

Années 90 à 2012: Entre la fin des années 90 et le début des années 2010, le réseau de neurones convolutionnels était en incubation. Alors que de plus en plus de données et de puissance de calcul devenaient disponibles, les tâches que les réseaux de neurones convolutifs pouvaient entreprendre devenaient de plus en plus intéressantes.

- **AlexNet (2012) :** En 2012, Alex Krizhevsky (et d'autres) a publié AlexNet qui était une version plus profonde et beaucoup plus large du LeNet et a remporté de loin le difficile défi de reconnaissance visuelle à grande échelle ImageNet (ILSVRC) en 2012. C'était une percée importante par rapport aux approches précédentes et l'application répandue actuelle des CNN peuvent être attribuées à ce travail.
- **ZF Net (2013) :** Le gagnant de l'ILSVRC 2013 était un réseau convolutionnel de Matthew Zeiler et Rob Fergus. Il est devenu connu sous le nom de ZFNet (abréviation de Zeiler&Fergus Net). C'était une amélioration sur AlexNet en peaufinant les hyper paramètres de l'architecture.
- **GoogLe Net (2014) :** Le gagnant de l'ILSVRC 2014 était un réseau convolutionnel de Szegedy et al. de Google. Sa principale contribution a été le développement d'un module de démarrage qui a considérablement réduit le nombre de paramètres dans le réseau (4M, par rapport à AlexNet avec 60M).
- **VGGNet (2014) :** Le finaliste d'ILSVRC 2014 était le réseau qui est devenu connu sous le nom de VGGNet. Sa principale contribution a été de montrer que la

profondeur du réseau (nombre de couches) est un élément critique pour de bonnes performances.

- **ResNets (2015)** : Résiduel développé par Kaiming He (et d'autres) a été le vainqueur de l'ILSVRC 2015. ResNets est actuellement de loin des modèles de réseau de neurones convolutionnels de pointe et constitue le choix par défaut pour utiliser ConvNets dans la pratique (en mai 2016).
- **DenseNet (août 2016)** : Récemment publié par Gao Huang (et d'autres), le réseau convolutionnel Densely Connected a chaque couche directement connectée à toutes les autres couches de manière directe. Il a été démontré que le DenseNet obtient des améliorations significatives par rapport aux architectures de pointe précédentes sur cinq tâches de référence de reconnaissance d'objets hautement compétitives.

III.8 Conclusion :

Dans ce chapitre, nous avons essayé d'expliquer les principaux concepts derrière les réseaux de neurones convolutionnels et l'apprentissage par transfert en termes simples. Plusieurs détails ont été simplifiés / ignorés, mais nous avons donné une intuition sur la façon dont ils fonctionnent.

Si la base de données est bien établies, les réseaux de neurones convolutionnels sont capables de surpasser les humains lors des tâches de reconnaissance visuelle . Cependant, ils ne sont toujours pas résistants aux artefacts visuels tels que les reflets et le bruit, que les humains sont capables de faire face.

La théorie des réseaux neuronaux convolutionnels est toujours en cours d'élaboration et les chercheurs travaillent à lui conférer des propriétés telles que l'attention active et la mémoire en ligne, permettant aux réseaux neuronaux convolutionnels d'évaluer de nouveaux éléments très différents de ceux sur lesquels ils ont été formés.



CHAPTER VI
IMPLÉMENTATION

IV.1 Introduction :

Les catastrophes naturelles comme les ouragans, les incendies de forêt, les tremblements de terre, les inondations, les tsunamis et les éruptions volcaniques ont souvent des effets catastrophiques sur la vie humaine et causent des dommages importants aux infrastructures. Souvent, il est difficile de détecter ces catastrophes avant qu'elles ne puissent affecter en grande partie la vie humaine et animale. A cet effet, des applications de télédétection sont utilisées et les images capturées par le satellite sont évaluées pour identifier tout changement anormal dans l'atmosphère, qui pourrait conduire à une catastrophe naturelle. Mais, l'analyse de ces images satellites nécessite des efforts et du temps humains, ce qui entraîne un retard dans la prédiction. Ce défi entraîne une perte substantielle d'infrastructures et de vies. Par conséquent, un système entièrement automatisé est un besoin imminent pour identifier ces catastrophes naturelles aussitôt que possible. Dans cette étude, nous avons développé un système automatisé de détection de calamités en utilisant le deep learning. Ce système peut prévoir les catastrophes en temps réel et envoyer un message d'alerte aux autorités concernées pour prendre les actions adéquates.

Dans ce mémoire, nous verrons comment la vision par ordinateur et les algorithmes d'apprentissage par transfert en profondeur peuvent être utilisés pour détecter automatiquement les catastrophes naturelles dans les images et les flux vidéo en utilisant Keras et son modèle préforme.

Avant de passer directement à la pratique, essayons d'abord comprendre le Logiciels et bibliothèques Utilisés dans l'implémentation.

Est nous pouvons utiliser une technique d'apprentissage automatique pour résoudre les problèmes de classification d'images qui peuvent générer de grands résultats de manière rapide et facile, et cette technique est l'apprentissage par transfert.

IV.2 Logiciels et bibliothèques Utilisés dans l'implémentation :

IV.2.1 Python :

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple,

facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

IV.2.2 TensorFlow :

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des framework les plus utilisés pour le Deep Learning et donc les réseaux de neurones. Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multidimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix.

IV.2.3 Keras :

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur Google.

En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow. Chollet a expliqué que Keras a été conçue comme une interface plutôt que comme un cadre d'apprentissage end to end. Il présente un ensemble d'abstractions de niveau supérieur et plus intuitif qui facilitent la configuration des réseaux neuronaux indépendamment de la bibliothèque informatique de backend. Microsoft travaille également à ajouter un back end CNTK à Keras aussi.

IV.2.4 Scikit-learn :

Est une bibliothèque libre Python dédiée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts

français d'enseignement supérieur et de recherche comme Inria et Télécom ParisTech

Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support.

Elle est conçue pour s'harmoniser avec des autres bibliothèques libre Python, notamment NumPy et SciPy.

IV.2.5 Configuration Utilisé dans l'implémentation :

La configuration du matériel utilisé dans notre implémentation est :

- ❖ Un PC portable HP i3 1.2Ghz
- ❖ RAM de taille 4 GO DDR 3
- ❖ Disque dur de taille 800 GO
- ❖ Système d'exploitation Linux Ubuntu version 19.10

IV.3 Collection de bases de données:

L'ensemble de données que nous utilisons a été organisé par Gautam Kumar qui l'a partagé sur son profil LinkedIn, attirant l'attention de nombreux praticiens du deeplearning.

Gautam a utilisé Google Images pour rassembler un total de 4 428 images appartenant à quatre classes distinctes:

- Cyclone / ouragan: 928 images
- Tremblement de terre: 1350
- Inondation: 1073
- Feu de forêt: 1077

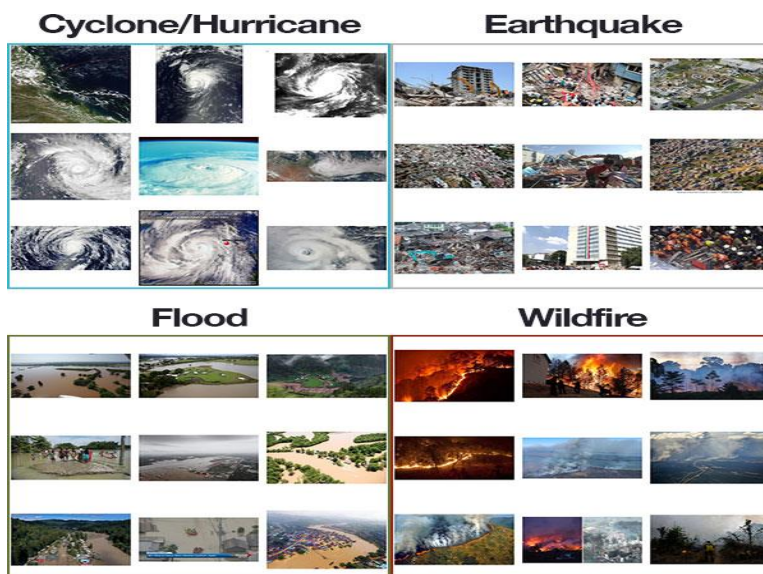


Figure IV.1: ensemble de données d'images de catastrophes naturelles

Il a ensuite formé un réseau neuronal convolutif pour reconnaître chacun des cas de catastrophe naturelle.

IV.4 Formation du modèle:

Au lieu de former le modèle CNN à partir de zéro, nous avons utilisé une technique d'apprentissage par transfert en utilisant des poids ImageNet préformés. Nous avons utilisé ResNet50 comme modèle de base, toutes les couches internes jusqu'à la couche GAP (global average pooling) restent inchangées. Nous avons supprimé la couche FC et ajouté la mise en commun moyenne (pool size = (7, 7)), Aplatisir, la couche dense (512), la couche Dropout (0,5) et la couche dense suivies de l'activation (softmax). L'architecture affinée est illustrée à la figure IV.2

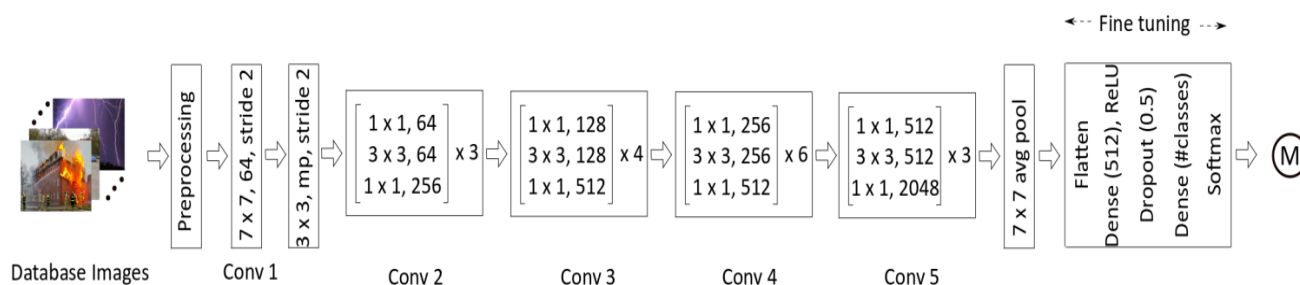


Figure IV.2: réglage fin de ResNet 50.

Un modèle (M) est obtenu par la formation du CNN pour 40 et 80 époques. La performance du modèle est évaluée en deux phases. Dans la première phase, un échantillon de jeu d'images de test de la base de données collectée est utilisé pour évaluer ses performances. La métrique d'évaluation telle que la précision, le rappel et l'exactitude du système est mesurée. Nous traçons également la courbe ROC (Receiver Operating Characteristics) pour montrer le compromis entre le taux de faux positifs et le taux de vrais positifs. Nous avons analysé les performances du modèle en traçant le rappel de précision et l'aire sous la courbe.

Dans la deuxième phase, des vidéos en temps réel sont fournies en entrée au système. À partir d'une source de flux vidéo en direct (S), une image est extraite et prétraitée en la redimensionnant (à la taille de l'image qui a été utilisée pour former le modèle) suivie d'une conversion de l'ordre des canaux RVB. Chaque image extraite et prétraitée de la vidéo en direct est transmise au modèle pour prédire son étiquette de classe. L'algorithme calcule la probabilité de chaque classe et renvoie une étiquette correspondant à la classe pour laquelle la probabilité est maximale. Ce processus est répété pour chaque image d'un flux vidéo. Pour chaque trame entrante, la classe prédite et sa probabilité sont annotées sur la trame de sortie. Par conséquent, nous obtenons la vidéo de sortie sous la forme d'une série de l'image annotée montrant l'étiquette de classe et la probabilité de classe prédite. Cependant, il y a un problème avec cette méthode appelée «scintillement de prédiction». Pour cette raison, l'étiquette dans la vidéo de sortie change fréquemment entre les classes

IV.5 Évaluation :

La base de données est divisée en training (apprentissage) et ensemble de test dans un rapport de 80:20. Le modèle CNN est formé par 80% de l'image de la base de données pendant qu'il est testé avec 20%, et la matrice de confusion est tracée pour l'ensemble de test. Nous calculons True Positive (TP), False Positive (FP), True Négative (TN) et False Négative (FN). À partir de ces paramètres, nous calculons la sensibilité, le rappel, le score F et la précision du système. Nous traçons également la courbe des caractéristiques de fonctionnement du récepteur (ROC) et la courbe de précision-rappel pour analyser le résultat.

IV.6 configuration programme

```
# import the necessary packages
import os

# initialize the path to the input directory containing our dataset
# of images
DATASET_PATH = "Cyclone_Wildfire_Flood_Earthquake_Database"

# initialize the class labels in the dataset
CLASSES = ["cyclone", "Earthquake", "Flood", "Wildfire"]

# define the size of the training, validation (which comes from the
# train split), and testing splits, respectively
TRAIN_SPLIT = 0.75
VAL_SPLIT = 0.1
TEST_SPLIT = 0.25

# define the minimum learning rate, maximum learning rate, batch size,
# step size, CLR method, and number of epochs
MIN_LR = 1e-6
MAX_LR = 1e-4
BATCH_SIZE = 32
STEP_SIZE = 8
CLR_METHOD = "triangular"
NUM_EPOCHS = 20

# set the path to the serialized model after training
MODEL_PATH = os.path.sep.join(["model", "activity_gpu.model"])
INPUT_PATH = os.path.sep.join(["example_clips", "5.mp4"])
OUTOUT_PATH = os.path.sep.join(["outvideo", "25.avi"])
CLR_PLOT_PATH = os.path.sep.join(["Graph", "CLR.png"])
TRAINING_PLOT_PATH = os.path.sep.join(["Graph", "Traning.png"])
LRFIND_PLOT_PATH = os.path.sep.join(["Graph", "LRFIND.png"])
```

IV.6 Résultats obtenus et discussion :

Pour trouver le meilleur modèle, nous apporterons 3 modification au programme

- Modifier le nombre d'époques et fixer le nombre d'image et la résolution
- Modifier le nombre d'images et fixer le nombre d'époque et la résolution
- Modifier le nombre résolutions et fixer le nombre d'image et la d'époque

IV.6.1 Résultats obtenus pour 3 différents Epochs :

➤ Résultats obtenus pour le modèle 1 : Epochs : 20, images : 400

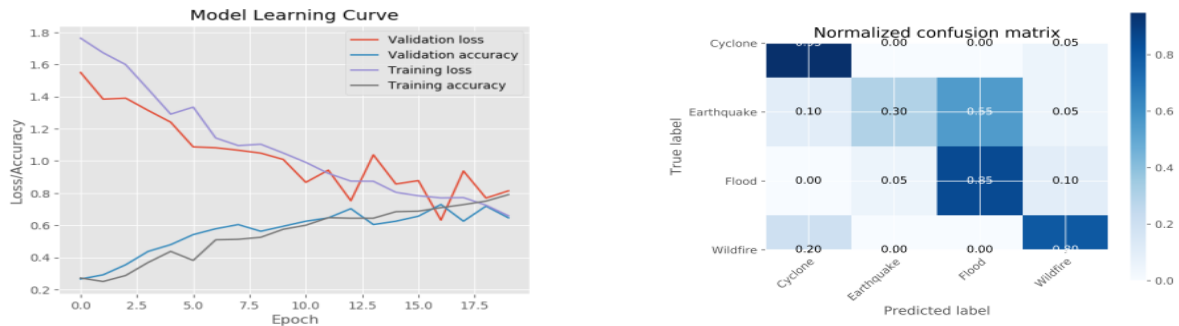


Figure IV.3: Précision et Erreur pour le Modèle & Normalized Confusion Matrix.

Epoch	3	7	10	15	20
Validation acc.	0.4	0.54	0.6	0.62	0.80
Test acc.	0.32	0.6	0.62	0.62	0.7
Validation.loss	1.37	1.08	0.82	0.9	0.9
Training loss	1.54	1.09	1.0	0.8	0.6

Tableau IV.1: a précision et Erreur VS Epochs (400 Images)

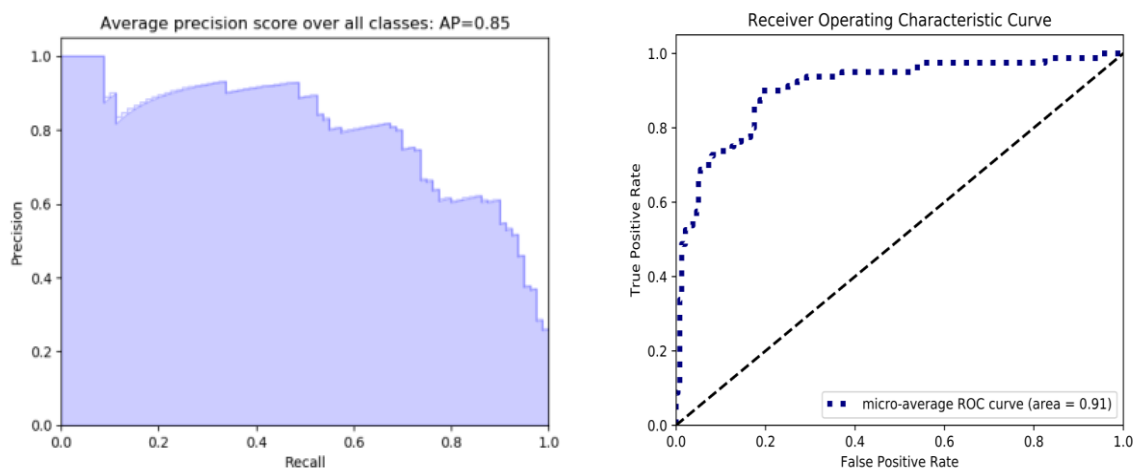


Figure IV.4: Precision-Recall.& courbe ROC.

Après l'analyse des résultats obtenus, On constate les remarques suivantes :

D'après la Figure IV.4 La précision de l'apprentissage et de test augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprend plus d'informations. Si la précision n'est pas satisfaisante alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque et vice versa.

De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époque.

➤ **Résultats obtenus pour le modèle 2 : Epochs : 40 , images : 400**

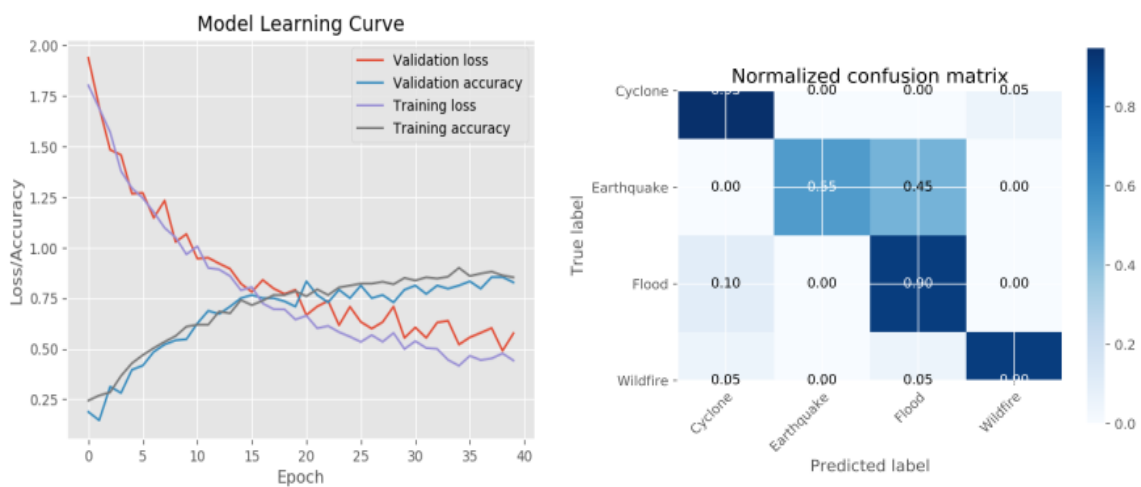


Figure IV.5: Learning accuracy – Losses & Normalized Confusion Matrix..

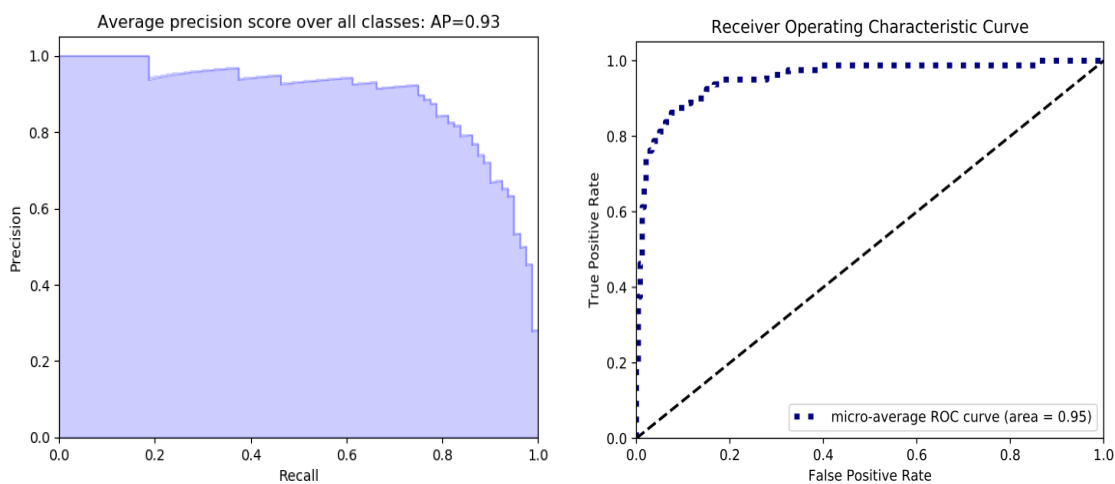


Figure IV.6: Precision-Recall & courbe ROC .

➤ Résultats obtenus pour le modèle 3 : Epochs : 80 , images : 400

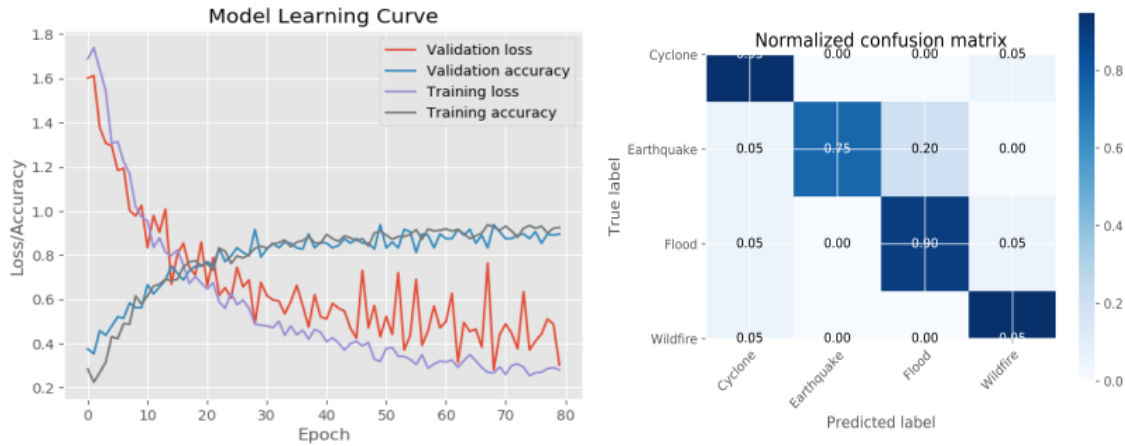


Figure IV.7 : Learning accuracy – Losses & Normalized Confusion Matrix.

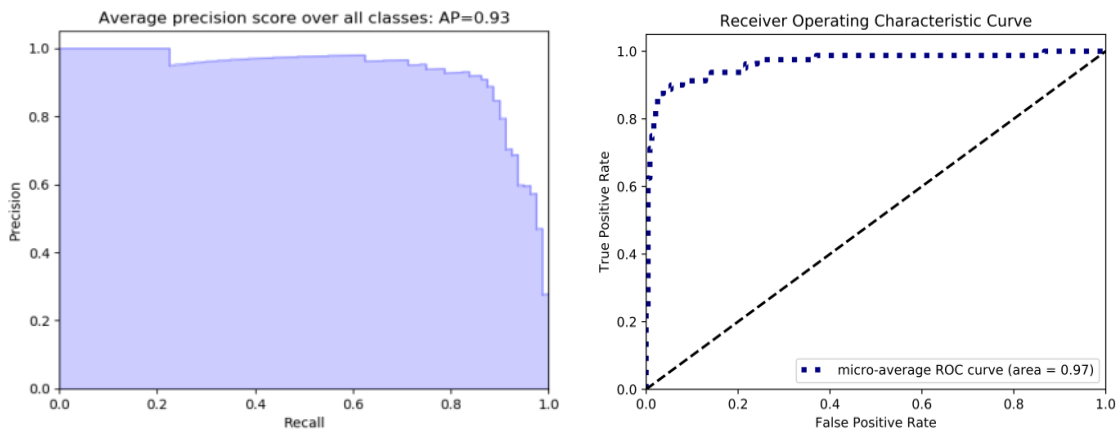


Figure IV.8: Precision-Recall & courbe ROC.

Après l’analyse des résultats obtenus, On constate les remarques suivantes :

La précision de l’apprentissage et de la validation augmente avec le nombre d’époque, ceci reflète qu’à chaque époque le modèle apprend plus d’informations. De même, l’erreur d’apprentissage et de la validation diminue avec le nombre d’époque.

La matrice de confusion permet d’évaluer la performance de notre modèle, puisqu’elle reflète les Métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif.

La **Figure IV.8** illustre de près la position de ces métriques pour chaque classe. De là, nous concluons que dans: Epochs : 80, images : 400 la précision est très satisfaisante.

IV.6.2 Résultats obtenus pour 3 différentes résolutions :

➤ **modèle 1 : Epochs : 80, images : 400, Résolution 80 x 80 filtre de pooling 2x2**

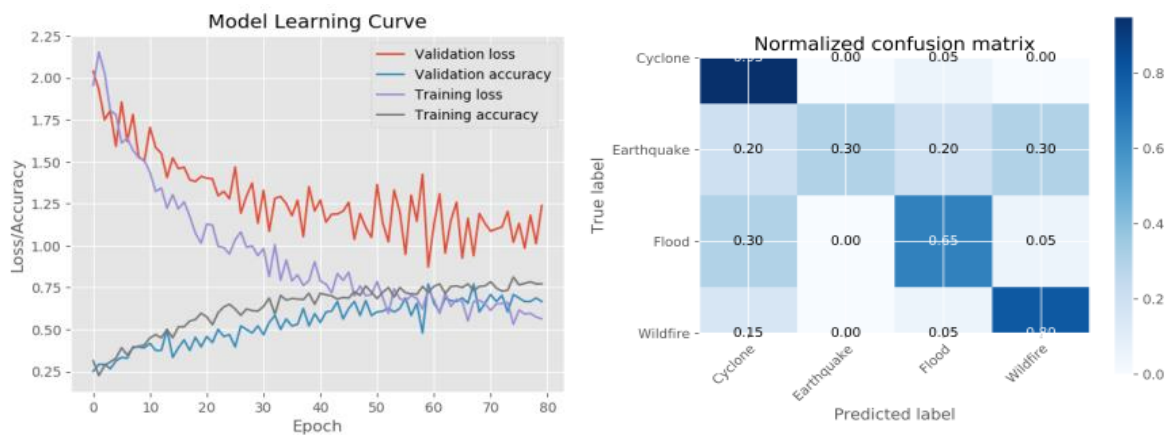


Figure IV.9: Learning Oaccuracy – Losses & Normalized Confusion Matrix..

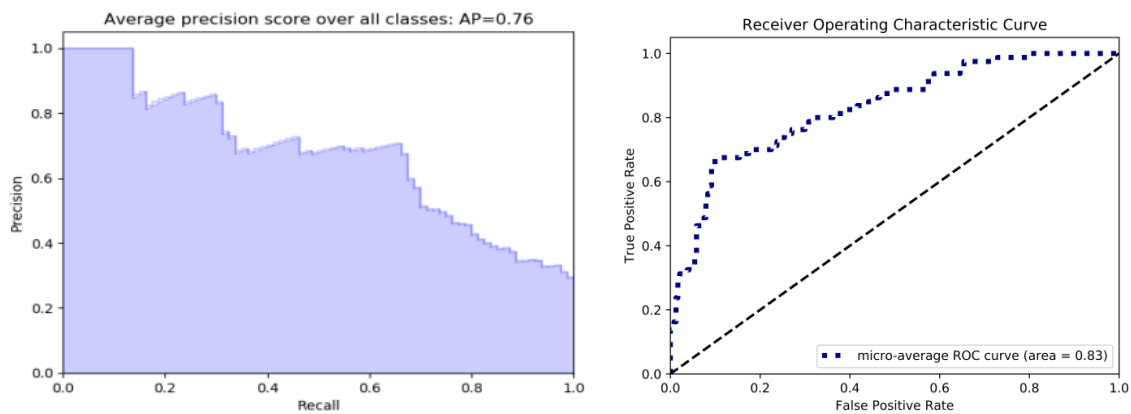


Figure IV.10: Precision-Recall & Courbe ROC.

Après l’analyse des résultats obtenus, On constate les remarques suivantes :

D'après la Figure **Figure IV.10** La précision de l'apprentissage et de la validation augmente avec le nombre d'époques. d'après de confusion, on constate une confusion du modèle entre les inondations et le tremblement de terre.

- **modèle 2 : Epochs : 80 , images : 400 , Résolution 224 x 224 filtre de pooling 7x7**

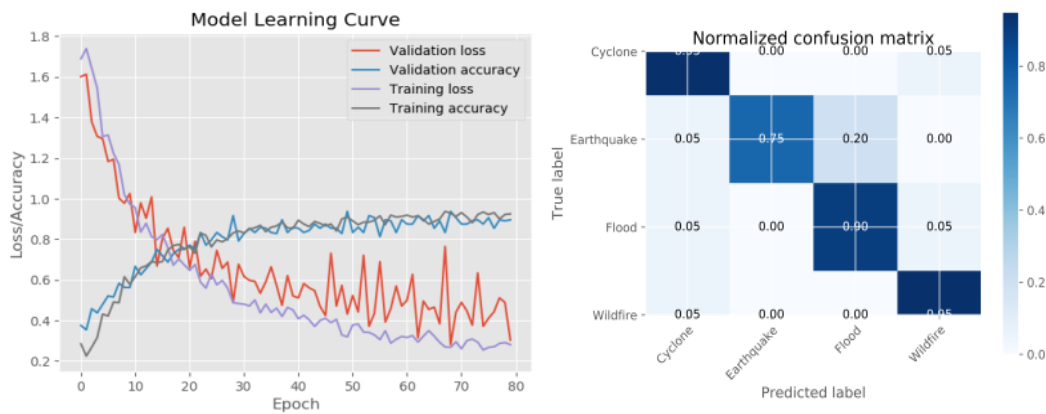


Figure IV.11 : Learning accuracy – Losses & Normalized Confusion Matrix.

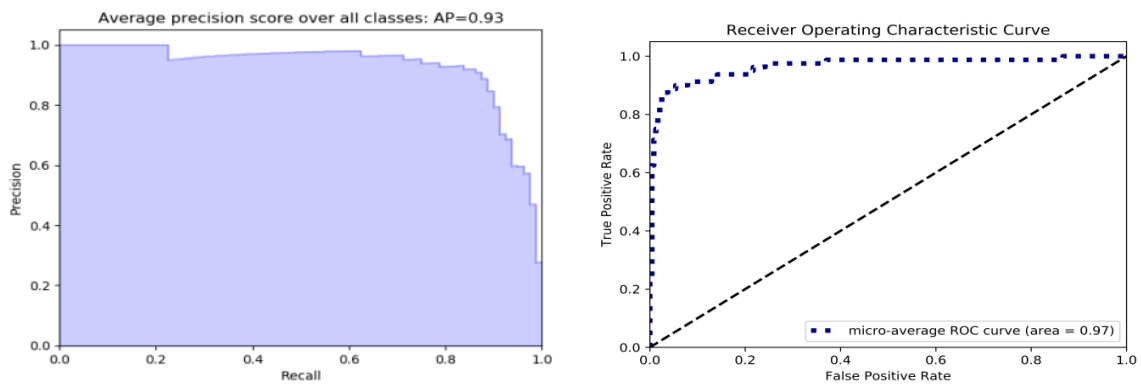


Figure IV.12 : Precision-Recall & Courbe ROC.

IV.6.3 Résultats obtenus pour les 3 différents nombres d'images

➤ modèle 1 : Epochs : 80, images : 100, Résolution 224 x 224 filtre de pooling 7x7

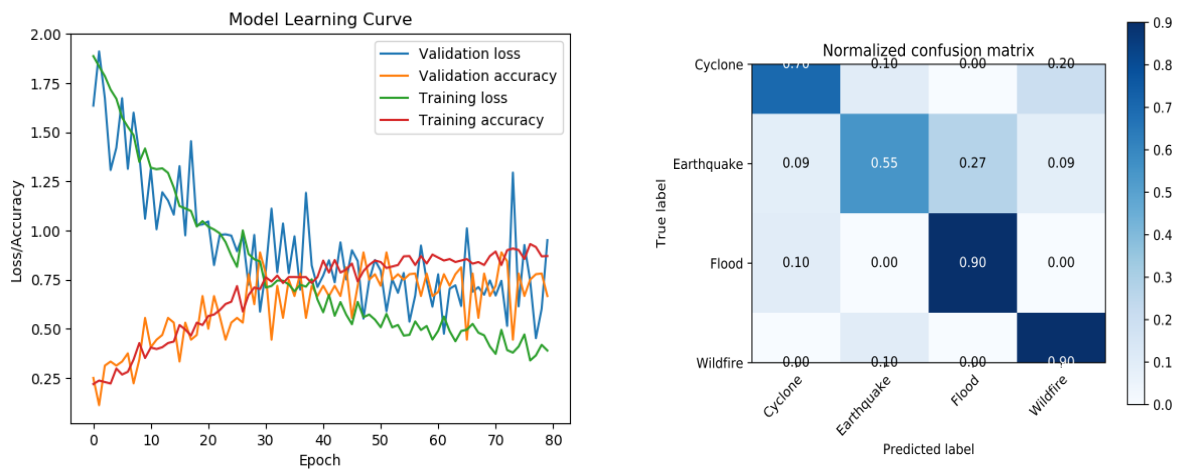


Figure IV.13: Learning accuracy – Losses & Normalized Confusion Matrix.

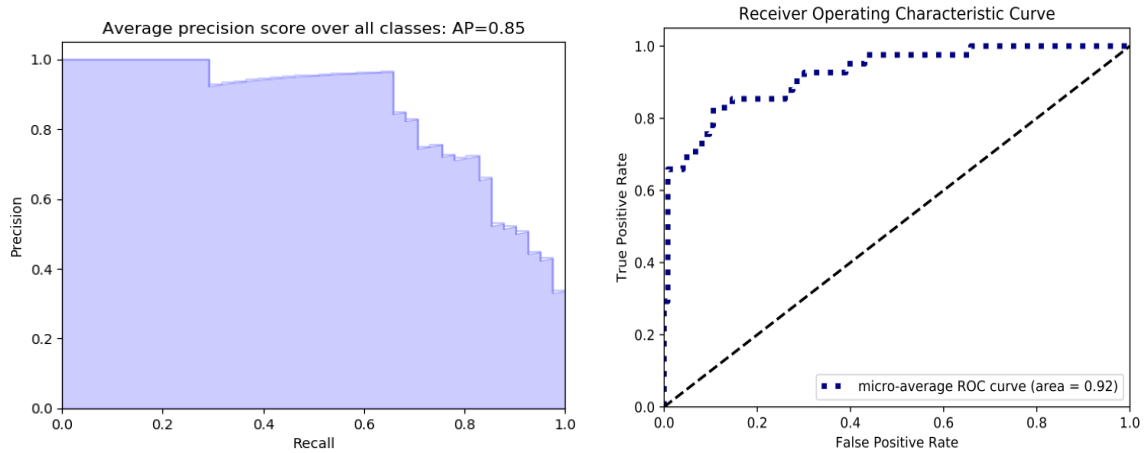


Figure IV.14: Precision-Recall & Courbe ROC.

- modèle 2 : Epochs : 80 , images : 400 , Résolution 224 x 224 filtre de pooling 7x7

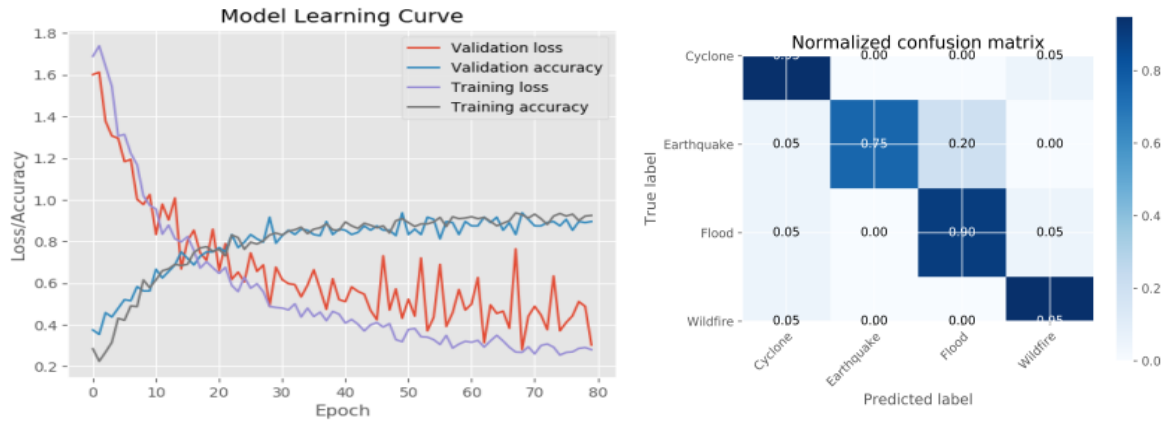


Figure IV.15 : Learning accuracy – Losses & Normalized Confusion Matrix.

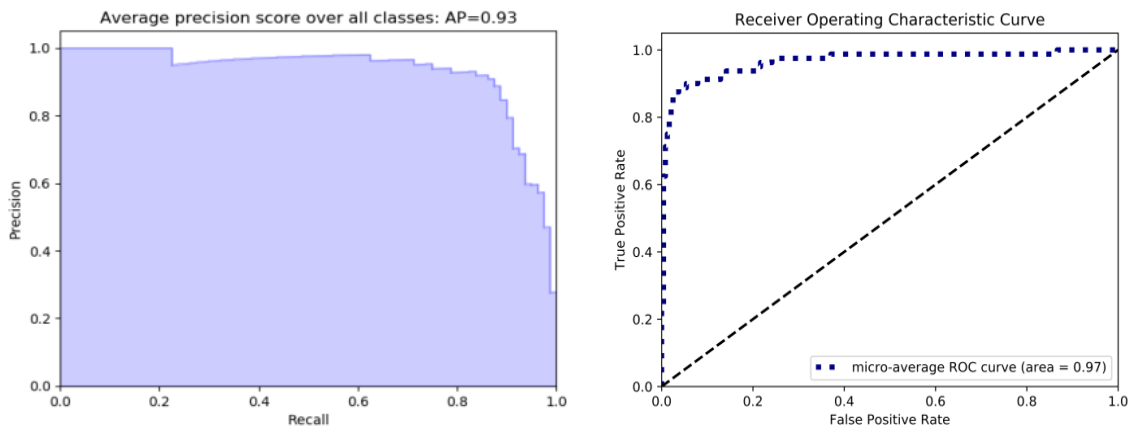


Figure IV.16 : Precision-Recall & Courbe ROC.

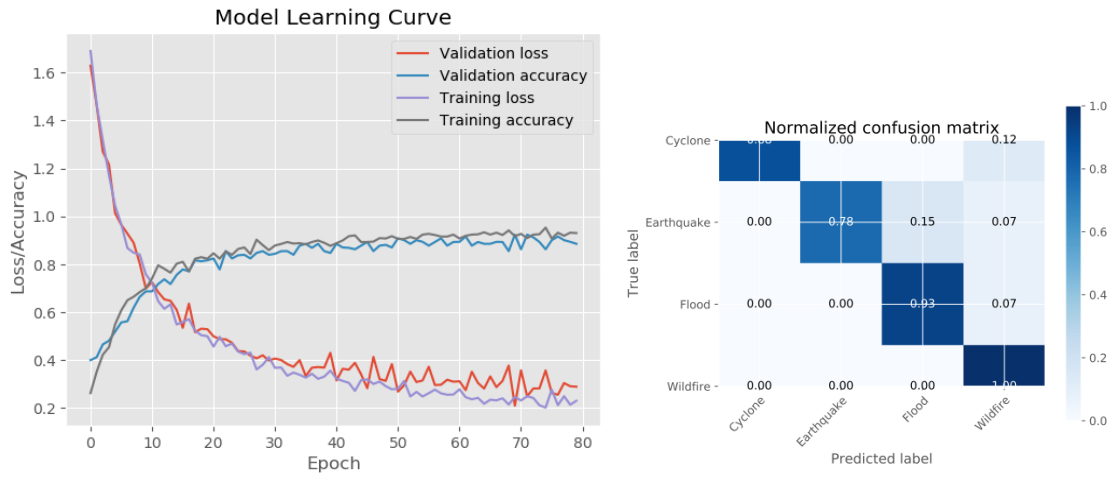


Figure IV.17: Learning accuracy – Losses & Normalized Confusion Matrix.

Epoch	10	20	40	60	80
Validation acc.	0.5	0.8	0.6	0.92	0.9
Test acc.	0.6	0.82	0.62	0.9	0.90
Validation.loss	0.82	0.62	0.82	0.6	0.21
Training loss	0.8	0.61	1.0	0.5	0.23

Tableau IV.2: a précision et Erreur VS Epochs (800 Images)

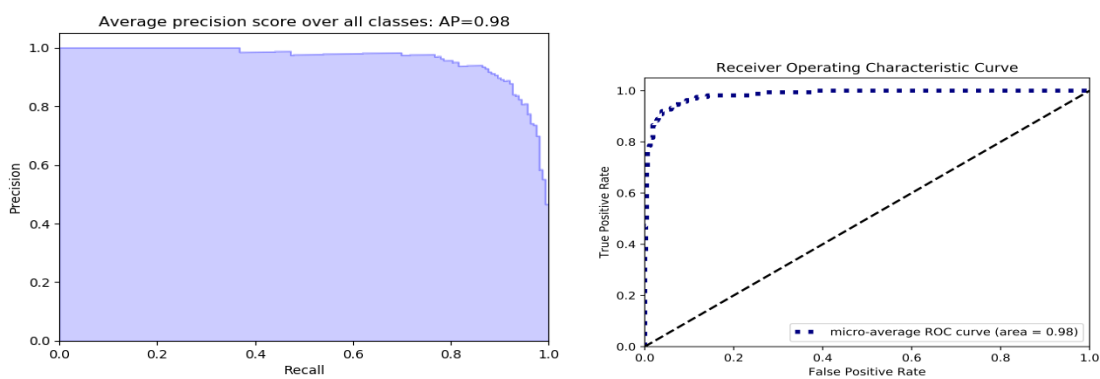


Figure IV.18 : Precision-Recall & Courbe ROC

modèle 3 : Epochs : 80 , images :800 , Résolution 224 x 224 filtre de pooling

IV.6.4 Résultat expérimental:

Le système développé prend une vidéo préenregistrée ou une vidéo en temps réel comme entrée, extrait des images de la vidéo et les alimente au modèle CNN formé. Le modèle CNN prend chaque trame séquentiellement et la traite pour prédire le type de catastrophe et produit une séquence de trames étiquetées en sortie. Chaque trame de sortie est annotée avec le type de calamité prédite ainsi que sa probabilité correspondante. Si aucune catastrophe n'est détectée, la trame est annotée comme «Normal». Un message d'alerte est généré uniquement en cas de modification de l'étiquette de la trame de sortie. Pour les trames annotées comme «Normal», aucun message n'est généré. La sortie du système lors d'un test avec une vidéo enregistrée est illustrée ci-dessous.

IV.7 Détection de calamité en vidéo:

Un exemple de sorties générées par le système après capture d'image est illustré dans les figures suivantes.

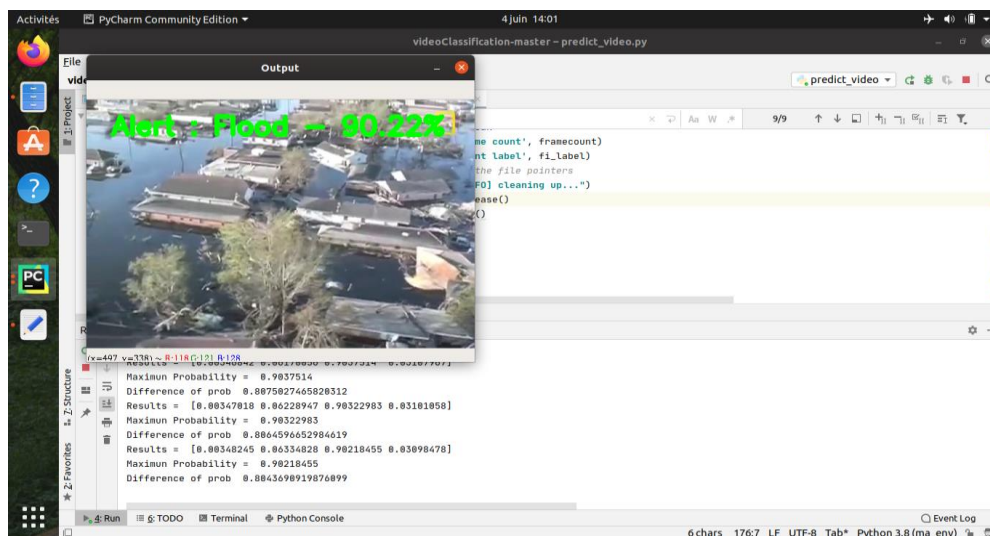


Figure IV.19: Exemples d'images de sortie : inondation

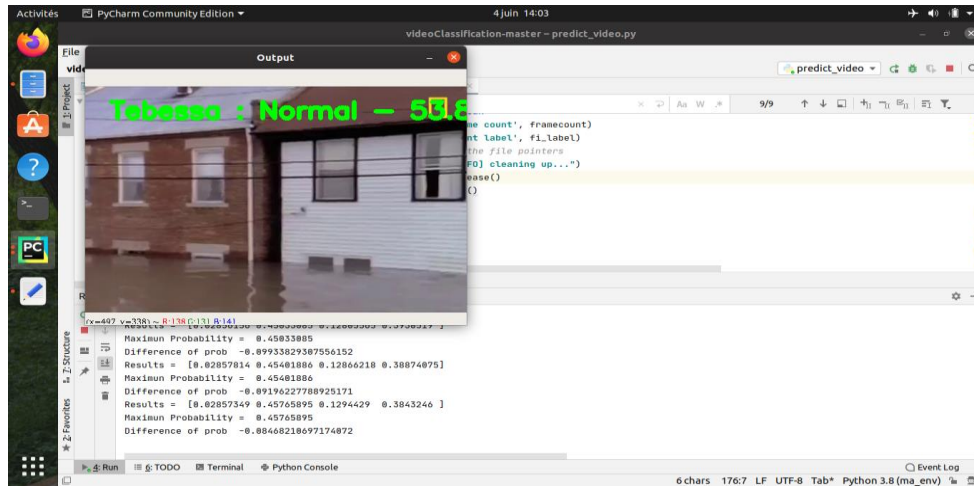


Figure IV.20: Exemples d'images de sortie : inondation ou pluies normale

4-7-1 : Détection de calamité en vidéo: 80 epochs – 400 images

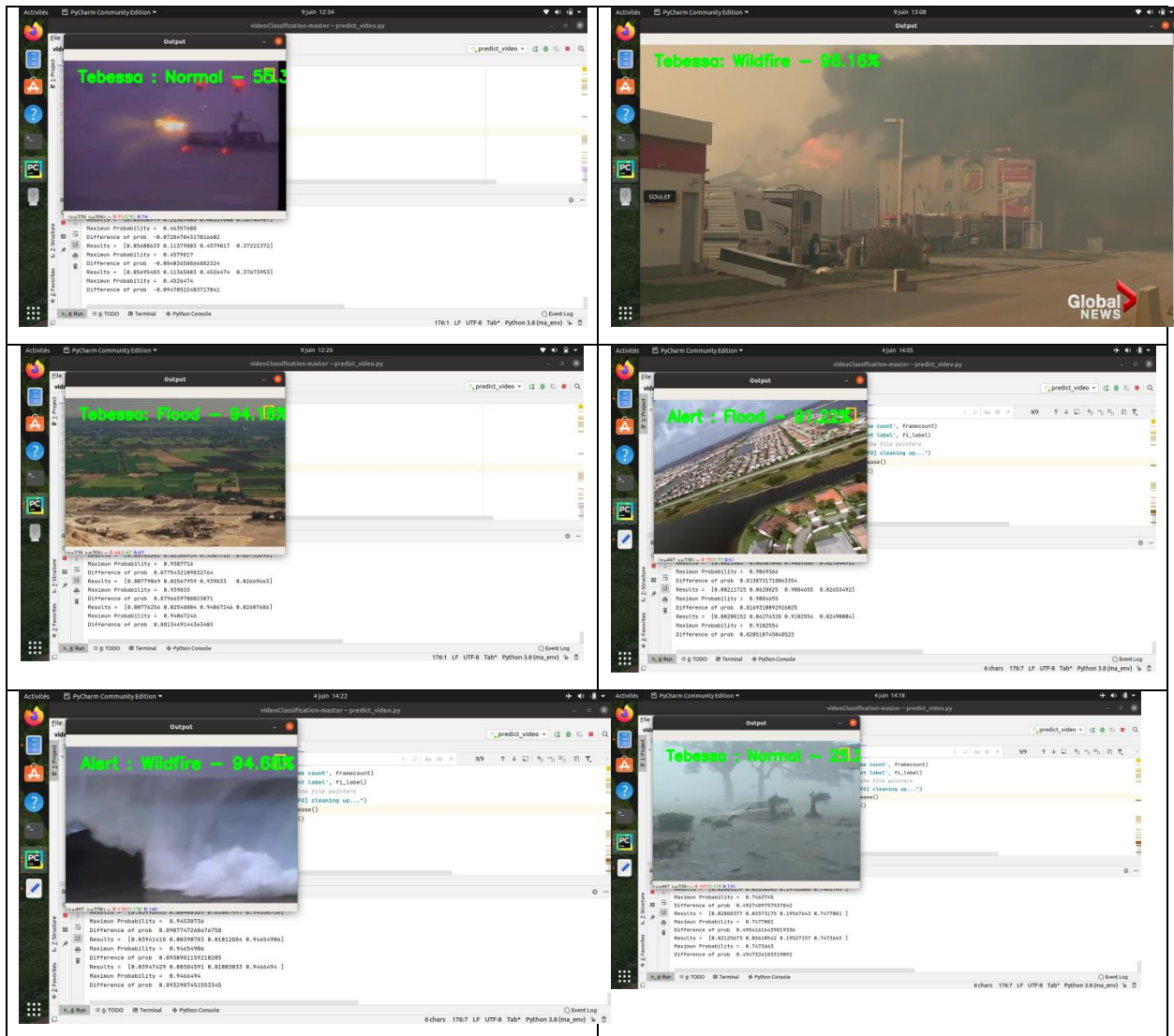


Figure IV.21: Exemples d'images de sortie erronées (80 Epochs, 400 Images)

4-7-2 Détection de calamité en vidéo: 80 epochs – 800 images

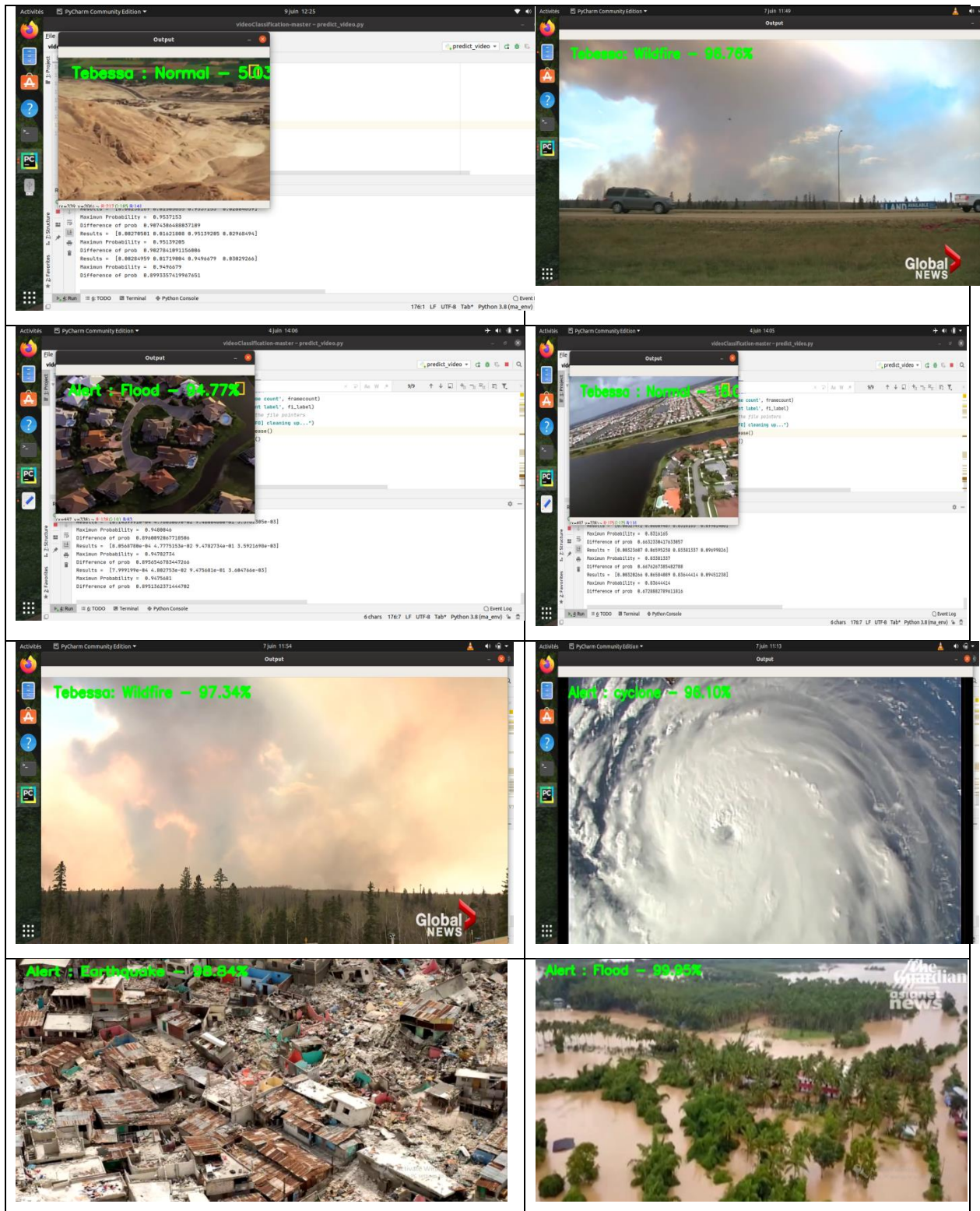


Figure IV.22: Exemples d'images de sortie (80 epochs, 800 images)

Bien que notre modèle d'apprentissage en profondeur peut prédire laquelle des catastrophes naturelles est une tempête, tremblement de terre ou un incendie de forêt présente une précision jusqu'à $\sim 96\%$ qui semble très satisfaisant, le modèle. Naturellement, il peut y avoir une confusion des images pour différentes catastrophes (où une tempête de vent de sable peut ressembler à un feu sauvage), que même un œil humain peut être confus. En augmentant le nombre d'images de formation, on peut y remédier cet handicap en essayant de prévoir la tempête de sable plutôt que d'avoir un mélange dans l'autre classe.

IV.8 Message d'alerte:

Un message d'alerte peut être généré uniquement en cas de modification de l'étiquette de la trame de sortie. Pour les trames annotées comme «Normal», aucun message n'est généré. On peut utiliser la version d'essai de l'API Twilio pour envoyer un message à un numéro de téléphone mobile, comme le montre la **figure IV.23**

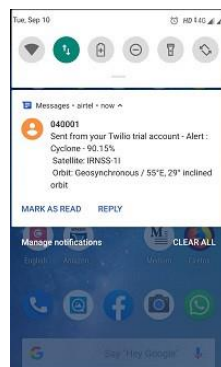


Figure IV.23: Alerte générée par le système [33]

IV.9 Conclusion:

Le système de détection des calamités en temps réel proposé peut être utilisé pour détecter une catastrophe et générer automatiquement une alerte. Ce système peut être utilisé à deux fins différentes. Premièrement, traiter les images satellites pour prévoir et avertir les catastrophes à l'aide d'une caméra satellite. Un satellite géostationnaire situé au-dessus de la région côtière peut détecter un cyclone tandis que le satellite polaire peut facilement capturer un flux en direct de forêts, de montagnes, de rivières, de zones inondables, etc. et s'il trouve un signe de calamité, le système peut automatiquement générer une alerte et réduit considérablement l'effort humain et le temps d'analyse de l'image satellite pour tout type de catastrophe. Pour les tremblements de terre le système peut enregistrer les détails des zones

affectées / détruites. De même, en cas d'inondation, la méthode proposée permet de détecter et de générer rapidement une alerte de manière automatisée qui peut nous aider à prendre les mesures nécessaires contre ces calamités. Par conséquent, le système proposé peut être utilisé comme un «œil dans le ciel» et peut aider la communauté des chercheurs à prévoir à l'avance les catastrophes et à optimiser les dommages lors des calamités naturelles.

Cette idée peut être utilisée pour détecter d'autres calamités naturelles telles que des glissements de terrain, des éruptions volcaniques, etc.



CONCLUSION
GÉNÉRALE

Conclusion Générale

L'idée qui a conduit à ce projet était d'étudier dans quelle mesure les réseaux profonds peuvent être utilisés efficacement pour identifier les catastrophes naturelles à travers des images vidéos envoyées par satellite. Un système précis qui est capable d'identifier le type de catastrophe en utilisant des images satellite en temps réel permettra une meilleure réponse et un meilleur soulagement aux zones touchées.

L'un des points importants que nous avons étudié était l'impact qu'un modèle d'apprentissage profond peut créer en cas de manque de données. Pour s'assurer que notre échantillon de données n'est pas riche en images, nous avons utilisé seulement 200 images, puis 800 images pour tester notre système et obtenu des résultats très satisfaisants avec des mesures de performances presque parfaites.

Dans ce projet nous avons discuté des notions fondamentales des réseaux de neurones en générale et des réseaux de neurones convolutionnels en particulier. Nous avons introduit ces réseaux de neurones convolutionnels en présentant les différents types de couches utilisées dans la classification: la couche convolutionnelle, la couche de rectification, la couche de pooling et la couche fully connected.

Les paramètres du réseau sont difficiles à définir a priori. C'est pour cette raison que nous avons défini différents modèles avec des architectures différentes afin d'obtenir des meilleurs résultats en terme de précision et d'erreur.

Nous avons rencontré quelques problèmes dans la phase d'implémentation, l'utilisation d'un CPU a fait que le temps d'exécution était trop coûteux. À cet effet, nous avons entraîné le modèle ResNet50 CNN, et les performances sont mesurées en calculant la matrice de confusion. Le modèle est également testé avec des vidéos préenregistrées acquises à partir de satellites et de drones. Les résultats expérimentaux donnent une précision de 91% et fonctionnent bien lorsqu'ils sont testés avec des vidéos collectées sur You Tube.

Le système de détection des calamités en temps réel proposé peut être utilisé pour détecter une catastrophe et générer automatiquement une alerte. Ce système peut être utilisé à deux fins différentes. Premièrement, traiter les images satellites pour prévoir et avertir les cyclones et les incendies de forêt. Deuxièmement, pour traiter des images satellites pour une détection rapide des régions touchées lors de calamités naturelles comme les inondations et les tremblements de terre de haute intensité. Cette idée peut être utilisée pour prédire la

Conclusion Générale

catastrophe à l'aide d'une caméra satellite. Par exemple, un satellite géostationnaire situé au-dessus de la région côtière peut détecter un cyclone tandis que le satellite polaire peut facilement capturer un flux en direct de forêts, de montagnes, de rivières, de zones inondables, etc. et s'il trouve un signe de calamité, le système peut automatiquement générer une alerte. Donc, il réduit considérablement l'effort humain et le temps d'analyse de l'image satellite pour tout type de catastrophe. Pour les tremblements de terre, il est difficile de prévoir à l'avance à l'aide d'une image satellite, mais le système peut enregistrer les détails des zones affectées / détruites. De même, en cas d'inondation, la méthode proposée permet de détecter et de générer rapidement une alerte de manière automatisée qui peut nous aider à prendre les mesures nécessaires contre ces calamités. Par conséquent, le système proposé peut être utilisé comme un «œil dans le ciel» et peut aider la communauté des chercheurs à prévoir à l'avance les catastrophes et à optimiser les dommages lors des calamités naturelles. la méthode proposée peut rapidement détecter et générer une alerte de manière automatisée qui peut nous aider à prendre les mesures nécessaires contre ces calamités.

Cette idée peut être utilisée pour détecter d'autres calamités naturelles telles que des glissements de terrain, des éruptions volcaniques, etc.

La méthode proposée peut être étendue pour détecter les accidents sur la route et envoyer une alerte aux hôpitaux voisins, un vol dans les magasins et envoyer une alerte au poste de police le plus proche et un incendie dans les bâtiments. De cette façon, nous pouvons rendre la vidéo surveillance plus intelligente.

Bibliographie

- [01] Giuliani, G. et P. Peduzzi, 2011, « The PREVIEW Global Risk Data Platform : a geoportal to serve and share global data on risk to natural hazards », *Natural Hazards Earth System Sciences*, 11, pp. 53-66.
- [02] X. X. Zhu et al., « Deep Learning in Remote Sensing: A Review », arXiv preprint arXiv:1710.03959, 2017.
- [03] Z. Zhang, H. Wang, F. Xu, et Y.-Q. Jin, « Complex-valued convolutional neural network and its application in polarimetric SAR image classification », *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no 12, p. 7177-7188, 2017.
- [04] Chesner Desir, *Classification Automatique d'Images, Application à l'Imagerie du Poumon Profond*, Université de Rouen, 2013
- [05] Naciri H., Chaoui N, *Conception et Réalisation d'un système automatique d'identification des empreintes digitales*, Mémoire de PFE, Université de Tlemcen, 2003.
- [06] Hadjila F. & Bouabdellah R, *Reconnaissance des visages par les réseaux de neurones*, Mémoire de PFE, Université de Tlemcen, 2003.
- [07] Rafael C.Gonzalez & Richard E.Woods , *Digital Image Processing*, Pearson Education Inc, 2008.
- [08] <https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-Intro-ApprentStat.pdf>
- [09] https://msdn.microsoft.com/big_data_france/2014/06/17/evaluer-un-modle-en-apprentissage-automatique/
- [10] Yann 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.
- [11] <http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>
- [12] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. Deep learning. *Nature*, vol. 521, pages 436–444, 2015. <http://dx.doi.org/10.1038/nature14539>
- [13] Ian Goodfellow, Yoshua Bengio et Aaron Courville. "Deep Learning", MIT Press, 2016. <http://www.deeplearningbook.org/>
- [14] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2, no. 1, pages 1–127, 2009.
- [15] LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" , *Proceedings of the IEEE*. 86 (11): 2278–2324.
- [16] Donahue, Jeff et al. (2014). "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". *Proceedings of Machine Learning Research*.
- [17] [16] <https://deepmind.com/blog/deep-reinforcement-learning/>
- [18] Deep learning : quand l'intelligence artificielle tente d'imiter le cerveau humain, Jérôme Cartegni
- [19] Perceptron simple, Perceptron Multi-couches, Nicolas Rougier
- [20] Victor Roman .Supervised Learning: Basics of Classification and Main Algorithms, Jan 31 2019
- [21] Concepts de réseau neuronal : Le guide complet des réseaux de neurones artificiels: concepts et modèles

- [22] Geoffrey Hinton, Yoshua Bengio et Yann LeCun, Deep Learning NIPS'15 Tutorial, Chap3
- [23] A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning by Dipanjan (DJ) Sarkar
- [24] Architectures de réseaux de neurones , le blog d'Eugenio Culurciello
- [25] Réseaux neuronaux convolutionnels CS231n pour la reconnaissance visuelle, Stanford
- [26] L'apprentissage automatique est amusant! Partie 3: Apprentissage profond et réseaux de neurones convolutifs
- [27] Réseaux de neurones par Rob Fergus, École d'été sur l'apprentissage automatique 2015
- [28] Réseaux de neurones convolutifs, Andrew Gibiansky
- [29] Vincent Dumoulin, et al , «A guide to convolution arithmetic for deep learning», 2015, arXiv: 1603.07285
- [30] Transfer Learning by Andrej Karpathy @ Stanford.
- [31] Keras: Feature extraction on large datasets with Deep Learning by Adrian Rosebrock.
- [32] Fine-tuning with Keras and Deep Learning by Adrian Rosebrock.
- [33] A Gentle Introduction to Transfer Learning for Deep Learning by Jason Brownlee.

المخلص

يجري بحثنا عند تقاطع ثلاث مجالات علمية: الاستشعار عن بعد ، رؤية الآلة وتعلم الآلة، في الأونة الأخيرة، أحرزت أساليب التعلم العميق تقدما كبير في التفسير التلقائي للصورة.

هذه الأطروحة هي في مفترق طرق بين الاستشعار عن بعد ، والرؤية الحاسوبية والتعلم الآلي على وجه الخصوص ، نقترح تنفيذ طرق التعلم العميق للكشف عن الكوارث الطبيعية مؤتمتة باستخدام deep learning ، والتي يمكن التنبؤ بالكوارث في الوقت الحقيقي وإرسال رسالة تنبيه. تحقيقا لهذه الغاية ، قمنا بتدريب نموذج ResNet50 CNN وتم قياس الأداء عن طريق حساب مصفوفة الخلط، تم أيضا اختبار هذا النموذج باستخدام مقاطع فيديو مسجلة مسبقاً تم الحصول عليها من الأقمار الصناعية والطائرات بدون طيار ، النتائج تمنحك التجربة دقة 91٪ وتعمل بشكل مثالي عند الاختبار باستخدام مقاطع الفيديو التي تم جمعها على YouTube

Résumé

Notre recherche se place à l'intersection de trois domaines scientifiques : la télédétection, la vision artificielle et l'apprentissage automatique. La littérature en vision par ordinateur est abondante concernant l'interprétation de données visuelles, y compris pour la télédétection. Récemment, les méthodes dites d'apprentissage profond ont permis de réaliser des avancées considérables en interprétation automatique d'images.

Cette thèse se trouve ainsi au croisement entre la télédétection, la vision par ordinateur et l'apprentissage automatique. En particulier, nous nous proposons de mettre en oeuvre des méthodes d'apprentissage profond permettant la détection de catastrophes naturelles automatisées utilisant le deep learning, qui peut prévoir les catastrophes en temps réel et envoyer un message d'alerte. À cet effet, nous avons entraîné le modèle ResNet50 CNN, et les performances sont mesurées en calculant la matrice de confusion. Le modèle est également testé avec des vidéos préenregistrées acquises à partir de satellites et de drones. Les résultats expérimentaux donnent une précision de 91% et fonctionnent parfaitement lorsqu'ils sont testés avec des vidéos collectées sur You Tube.

Abstract

This research takes place at the intersection of three scientific fields: remote sensing, machine vision and machine learning. Recently, deep learning methods have made a considerable progress in image interpretation.

This thesis is thus at the crossroads between remote sensing, computer vision and machine learning. In particular, we propose to implement a deep learning method for the automatic detection of natural disasters using deeplearning, which can forecast disasters in real time and send an alert message. To this end, we have trained the ResNet50 CNN model, and performances are measured by calculating the confusion matrix and ROC. The model is also tested with prerecorded videos acquired from satellites and drones. The experimental results give 91% accuracy and work perfectly when tested with videos collected from You Tube.