ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3706

THESE

En vue de l'obtention du : ***DOCTORAT***

Centre de Recherche : *Centre de Recherche Rabat Information Technology*

Structure de Recherche : *Laboratoire de Recherche en Informatique et Télécommunications*

Discipline : *Sciences de l'ingénieur*

Spécialité : *Informatique et télécommunications*

Présentée et soutenue le 04/11/2022 par :

EL BIACH Fatima Zahra

Deep Learning for Medical Image, Computer-Aided Diagnosis, and Image Forgery Detection.

JURY

Salma MOULINE	PES	Université Mohammed V, Faculté des Sciences de Rabat.	Présidente
Dounia LOTFI	PH	Université Mohammed V, Faculté des Sciences de Rabat.	Rapportrice/Examinatrice
Ouadoudi ZYTOUNE	PH	Université Ibn Tofail, Ecole Nationale des Sciences Appliquées de Kénitra.	Rapporteur/Examinateur
Benayad NSIRI	PH	Université Mohammed V, École Nationale Supérieure d'Arts et Métiers de Rabat.	Rapporteur/Examinateur
Mohammed KHALIL	PH	Université Hassan II de Casablanca, Faculté des Sciences Techniques de Mohammedia.	Examinateur
Hicham LAANAYA	PH	Université Mohammed V, Faculté des Sciences de Rabat.	Co-Encadrant
Khalid MINAOUI	PH	Université Mohammed V, Faculté des Sciences de Rabat.	Directeur de thèse

Année Universitaire : 2021/2022

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

وَٱلضُّحَىٰ ﴿١﴾ وَٱلَّيْلِ إِذَا سَجَىٰ ﴿٢﴾ مَا وَدَّعَكَ رَبُّكَ وَمَا قَلَىٰ ﴿٣﴾
وَلَلْآخِرَةُ خَيْرٌ لَّكَ مِنَ ٱلْأُولَىٰ ﴿٤﴾ وَلَسَوْفَ يُعْطِيكَ رَبُّكَ فَتَرْضَىٰٓ ﴿٥﴾
أَلَمْ يَجِدْكَ يَتِيمًا فَـَٔاوَىٰ ﴿٦﴾ وَوَجَدَكَ ضَآلًّا فَهَدَىٰ ﴿٧﴾ وَوَجَدَكَ عَآئِلًا
فَأَغْنَىٰ ﴿٨﴾ فَأَمَّا ٱلْيَتِيمَ فَلَا تَقْهَرْ ﴿٩﴾ وَأَمَّا ٱلسَّآئِلَ فَلَا تَنْهَرْ ﴿١٠﴾ وَأَمَّا
بِنِعْمَةِ رَبِّكَ فَحَدِّثْ ﴿١١﴾

Dedication

Praise be to Almighty God, who allowed me to see this long-awaited day

I dedicate this thesis:

To my very dear father Mohamed EL BIACH

You have always been for me an example of the respectful, honest father, of the meticulous person, I want to honor the man that you are.

Thanks to you dad I learned the meaning of work and responsibility. I would like to thank you for your love, your generosity, your understanding... Your support was a light throughout my journey. No dedication can express the love, esteem, and respect I have always had for you. This modest work is the fruit of all the sacrifices you have made for my education and training, may God welcome you to his vast paradise, and to my dear mother for all the sacrifices and unconditional love. I would like to express all my admiration to my husband Mr. IMAD IALA, and to my daughter DOHA for having supported and accompanied me in this great scientific experiment.

TO ALL MY FAMILY

No language can express my respect and appreciation for your support and encouragement. I dedicate this work to you in recognition of the love you offer me daily and your exceptional kindness. May the Almighty God keep you safe and bless you with health and happiness.

Acknowledgments

The presented contributions in this thesis have been performed in Laboratory of Research in Informatics and Telecommunications (LRIT-CNRST 29), Faculty of Sciences Rabat (FSR), University Mohammed V, under the supervision of **Mr. Khalid MINAOUI** and the co-direction of **Mr. Hicham LANNAYA**. Completing my PhD studies and authoring this dissertation was an amazing journey that would not have been possible without the support and encouragement of many people.

My greatest gratitude for my words goes to **Mr. Khalid MINAOUI**, Professor of the Faculty of Sciences of Rabat, and Director of the LRIT Laboratory. I am deeply grateful to him for accepting me into the LRIT lab and for agreeing to supervise my research. I am grateful to him for his encouragement, advice and support from the initial to the final level of my work.

I also thank my co-director, **Mr. Hicham LANNAYA**, Qualified Professor at the Faculty of Sciences of Rabat, Mohammed V University. It was he who allowed me to develop an understanding of this subject. He followed me in the smallest details of my work, which made it possible to carry out this thesis. I thank him for his efforts, his patience, and his cooperation with me.

I am very much honored by the members of my PhD committee. I would like to express my greatest honor and my gratitude to **Mrs. Salma MOULINE**, Professor at the Faculty of Sciences of Rabat, Mohammed V University, for having agreed to chair my doctoral commission.

I would like to thank **Mrs. Dounia Lotfi**, Qualified Professor at the Faculty of Sciences of Rabat, Mohammed V University, for her work in correcting this thesis and for her valuable suggestions.

I address my thanks to **Mr. Ouadoudi ZYTONE**, Qualified Professor at the National School of Applied Sciences of Kenitra, Ibn Tofail University, for his kind availability to correct, assist and judge this thesis.

I express my infinite gratitude to **Mr. Benayad NSIRI**, Qualified Professor at the National School of Arts and Crafts of Rabat, Mohammed V University, for his kind availability to assist and judge this thesis.

Finally, I want to thank **Mr. Mohamed KHALIL**, Qualified Professor at the Faculty of Technical Sciences of Mohammedia, Hassan II University of Casablanca, for his time to examine and attend my thesis.

Abstract

Computer vision is a research axis that allows machines to recognize visual inputs to exploit them in recognition tasks. In recent years, the quantity of images and videos has increased enormously. The exploitation of computer vision systems for analyzing this amount of information becomes important to extract relevant information. Computer vision systems are based primarily on machine learning (ML) and deep learning (DL) approaches. With the increase in the amount of data and the availability of powerful computing infrastructure, DL methods have seen great interest due to their superior performance on large data volumes and their feature extraction capability in the context of unstructured data. These methods were used in different sub domains of computer vision to perform several tasks: classification, localization, detection, and segmentation.

In the context of this thesis, we are interested in the detection of two types of images, falsified images, and computed tomography images by DL methods, precisely by convolutional neural networks (CNN). In this context, we have proposed several approaches to address the various problems related to the application of DL techniques in the segmentation of these types of images. The proposed approaches are based primarily on encoder-decoder architectures, regularization techniques, an adaptive loss function, and transferred learning strategies. It is interesting that image-to-image methods are used to solve various problems related to high variance, over-fitting, and sensitivity of DL networks to database change. In addition, they make it possible to combine the contextual information obtained by the encoder and the spatial information obtained by the decoder by concatenating (adding) two inputs (one from the previous layer of the decoder and the other from the symmetrical layer of the encoder) this generates more robust and stable decisions to changing data. On the other hand, transferred learning and adaptive loss function techniques are used to solve the problem of over-learning on limited volumes of data.

Résumé

La vision par ordinateur est un axe de recherche qui permet aux machines à reconnaitre les entrées visuelles pour les exploiter dans des tâches de reconnaissance. Dans ces dernières années, la quantité des images et des vidéos a énormément augmenté. L'exploitation des systèmes de vision par ordinateur pour l'analyse de cette quantité d'informations devient importante afin d'extraire de l'information pertinente. Les systèmes de vision par ordinateur sont basés principalement sur les approches d'apprentissage automatique (ML) et d'apprentissage profond (DL). Avec l'augmentation de la quantité de données et la disponibilité de l'infrastructure de calcule puissant, les méthodes DL ont connu un grand intérêt en raison de leur bonne performance sur les grands volumes de données et leur capacité d'extraction de caractéristique dans le cadre des données non structurées. Ces méthodes étaient exploitées dans différents sous domaines en vision par ordinateur pour effectuer plusieurs tâches : classification, localisation, détection, et segmentation.

Dans le contexte de la présente thèse, nous nous intéressons à la détection de deux types d'images, les images falsifiées et les images tomographie par les méthodes DL, précisément par les réseaux de neurones convolutifs (CNN). Dans ce cadre, nous avons proposé plusieurs approches pour répondre aux différents problèmes liés à l'application des techniques DL en segmentation de ces types d'images. Les approches proposées sont basées essentiellement sur des architectures d'encodeur décodeur, les techniques de régularisation, une fonction de perte adaptative, et les stratégies d'apprentissage transféré. Il est intéressant de noter que les méthodes image-to-image sont utilisées afin de résoudre les différents problèmes liés à la variance élevée, le sur-apprentissage, et la sensibilité des réseaux DL au changement de base de données. En plus, elles permettent de combiner les informations contextuelles obtenues par l'encodeur et l'information spatiale obtenue par le décodeur en en concaténant (additionnant) deux entrées (une de la couche précédente du décodeur et l'autre de la couche symétrique du codeur) cela génère des décisions plus robustes et stables au changement de données. D'autre part, les techniques d'apprentissage transféré et la fonction de perte adaptatives sont utilisés afin de résoudre le problème de sur-apprentissage sur les volumes limités de données.

Résumé détaillé

Ce mémoire de thèse s'intéresse aux visions par ordinateur. La vision par ordinateur est une branche de l'intelligence artificielle. Il permet à un ordinateur d'analyser, de traiter et de comprendre des images. Les systèmes de vision sont exploités pour extraire des informations pertinentes à partir d'entrées visuelles (image ou vidéo) pour une utilisation dans d'autres tâches de recommandation. La reconnaissance de l'entrée visuelle par les humains nécessite moins d'efforts par rapport aux systèmes automatiques. Avec le développement d'internet et des réseaux sociaux, la quantité d'images a rapidement augmenté. Par conséquent, le traitement de cette quantité d'informations par les êtres humains devient impossible, car ils ne peuvent pas traiter efficacement autant de données. Ces informations sont donc traitées automatiquement à l'aide de systèmes de vision par ordinateur. L'objectif de la vision par ordinateur est de développer des méthodes de reproduction de systèmes ayant une capacité équivalente à la vision humaine. Malgré les efforts déployés, le domaine de la vision par ordinateur présente plusieurs défis liés à la compréhension limitée des systèmes de vision biologique et à leurs complexités très élevées par rapport aux machines actuelles. Les systèmes de prédiction de la vision sont basés sur des algorithmes d'apprentissage automatique (ML) et d'apprentissage profond (DL). Ils permettent d'analyser les entrées visuelles prises par un système d'acquisition. Ces algorithmes sont formés sur des données pour produire des modèles de sortie. Les modèles générés sont ensuite utilisés dans la phase de prédiction. L'apprentissage profond (DL) est une branche de l'apprentissage automatique basée sur les réseaux de neurones artificiels (ANN). Ces réseaux ont été opérés en apprentissage supervisé et non supervisé afin d'optimiser les coûts de stockage et de calcul des réseaux DL classiques, plusieurs types d'architectures ont été proposées pour l'apprentissage supervisé (réseaux de neurones convolutifs (CNN), réseaux de neurones récurrents (RNN), long- mémoire à court terme (LSTM)) et non supervisés (auto-encodeurs empilés (SAE), réseaux de croyances profondes (DBN), réseaux antagonistes génératifs GAN). Dans cette thèse, nous nous intéressons aux CNN. Contrairement aux méthodes ML classiques, les méthodes DL et en particulier les CNN sont plus adaptées aux données complexes car elles intègrent la phase d'extraction de caractéristiques dans le processus d'apprentissage. Ces réseaux sont caractérisés par des couches de convolution et mise en commun par rapport aux réseaux DL conventionnels. Ces couches introduisent des liens partiels pour réduire le nombre de paramètres et renforcer le partage de caractéristiques communes.

Récemment, les réseaux de neurones convolutifs ont été largement exploités en vision par ordinateur grâce à leur stratégie de réduction des paramètres et à la disponibilité de gros volumes de données. De plus, l'évolution de la capacité de stockage et de calcul a encouragé la communauté de la vision par ordinateur à proposer d'autres architectures plus profondes de type CNN. En classification, les architectures proposées optimisent les couches de convolution classiques. L'objectif principal de cette variation est de réduire le nombre de paramètres et d'ajouter des couches supplémentaires qui améliorent la non-linéarité. En détection et segmentation, l'objectif principal était d'adapter les architectures proposées aux applications temps réel. En raison des progrès significatifs des architectures CNN, elles ont été exploitées dans plusieurs applications du monde réel, telles que la classification d'images, la détection et la localisation d'objets, la détection de falsification d'images, la reconnaissance faciale et le traitement d'images médicales. Dans la détection de falsification d'image, nous l'avons définie comme l'opération d'ajout, de suppression et/ou de modification de certaines régions ou caractéristiques significatives d'une image sans traces discernables. Différentes méthodes ont été utilisées pour altérer une image : copie-déplacement, épissage, incohérence d'éclairage, interpolation et transformations géométriques. Les CNN sont formés sur ces images pour

résoudre différentes tâches de détection d'images falsifiées : Classification, localisation, détection et segmentation. En imagerie médicale, les images numérisées ont plusieurs types tels que l'échographie (US), les rayons X, la tomodensitométrie (CT) et l'imagerie par résonance (IRM), la tomographie par émission de positrons (TEP) et les lames histologiques. Les CNN sont formés sur ces images pour résoudre différentes tâches en imagerie médicale : classification, localisation, détection et segmentation.

Les méthodes proposées dans cette thèse s'intéressent à la résolution de deux problèmes par des méthodes d'apprentissage profond. Le premier problème est lié à la détection des images médicales et en particulier des images CT par les réseaux CNN. De plus, le deuxième problème est lié à la détection de falsification d'images par les réseaux CNN. Le chapitre 1 introduit les notions théoriques liées au Deep Learning (DL). Il détaille l'origine et l'histoire des réseaux de neurones ainsi que leur évolution. Ensuite, il présente l'architecture générale d'un réseau de neurones profonds et les techniques utilisées dans l'apprentissage. Enfin, il illustre la configuration de quelques architectures connues en apprentissage supervisé et non supervisé. Le chapitre 2 se concentre sur les réseaux de neurones convolutifs (CNN). Dans un premier temps, il présente les origines et le développement de ce réseau. Ensuite, il présente l'architecture générale ainsi que les différentes techniques utilisées dans l'apprentissage de ces réseaux. Enfin, il explique et compare les différentes architectures connues de type CNN en classification, détection et segmentation. Le chapitre 3 présente l'état de l'art lié à l'application des CNN en vision par ordinateur, tels que la classification d'images, la détection et la localisation d'objets, la segmentation sémantique, la détection d'images falsifiées, la segmentation d'images médicales et la reconnaissance faciale. Le chapitre 4 présente les travaux liés à l'application des réseaux DL dans le domaine de l'imagerie médicale. Dans un premier temps, il introduit un état de l'art sur les méthodes proposées dans ce contexte. Ensuite, ce chapitre détaille la méthode de détection proposée. Enfin, une étude expérimentale est présentée avec les résultats expérimentaux. Les chapitres 5 et 6 de cette thèse présentent la deuxième et la troisième contribution qui visent à segmenter les images falsifiées, ce chapitre présente l'architecture proposée qui vise à segmenter les images manipulées en se basant sur un réseau encodeur-décodeur, et la deuxième contribution vise à résoudre le problème lié au sur-ajustement. Nous introduisons dans les deux chapitres des résultats expérimentaux et des études comparatives avec des approches de l'état de l'art. Chapitre 7 ; dans ce chapitre nous traitons en détail la troisième contribution de cette thèse qui consiste en la détection des infections pulmonaires dans les images médicales, et nous montrons les résultats expérimentaux détaillés qui prouvent l'efficacité de notre méthode. A la fin de ce chapitre nous analysons les conclusions générales de cette contribution. Chapitre 8 Conclusion et perspectives Le dernier chapitre conclut les travaux développés dans cette thèse et propose des orientations futures pour la recherche en apprentissage profond.

Figures list

Tables list

Table of Contents

General Introduction

In this introduction we will present the context of our study, namely deep learning methods. We will focus on the domain of computer vision, which constitutes the core of several automatic systems. We will be particularly interested in the segmentation of falsified images and CT images by different deep learning techniques. We will mainly expose our motivation concerning the optimization of convolutional neural networks to solve the various problems related to the segmentation of these images. We will then move on to a needs analysis justifying our contribution through this thesis work.

Context

Computer vision is a branch of artificial intelligence. It allows a computer to analyze, process, and understand images. Vision systems are exploited to extract relevant information from visual inputs (image or video) for use in other recommendation tasks. Recognition of visual input by humans requires less effort compared to automatic systems. With the development of the internet and social networks, the quantity of images has rapidly increased. Therefore, the processing of this amount of information by human beings becomes impossible, as they cannot efficiently process so much data. This information is therefore processed automatically using computer vision systems. The goal of computer vision is to develop methods for reproducing systems that have a capability equivalent to human vision. Despite the efforts made, the field of computer vision has several challenges related to the limited understanding of biological vision systems and their very high complexities compared to current machines. Computer vision prediction systems are based on machine learning (ML) and deep learning (DL) algorithms. They make it possible to analyze the visual inputs taken by an acquisition system. These algorithms are trained on data to produce output models. The generated models are then used in the prediction phase. Deep learning (DL) is a branch of machine learning based on artificial neural networks (ANN). These networks were operated in supervised and unsupervised learning in order to optimize the storage and computational costs of conventional DL networks, several types of architectures have been proposed for learning supervised (convolutional neural networks (CNN), recurrent neural networks (RNN), long-short term memory (LSTM)) and unsupervised (stacked auto encoders (SAE), deep belief networks (DBN), generative adversarial networks GAN). In this thesis, we are interested in CNNs. Unlike classical ML methods, DL methods and particularly CNNs are more suitable for complex data because they integrate the feature extraction phase into the learning process. These networks are characterized by layers of convolution and pooling compared to conventional DL networks. These layers introduce partial links to reduce the number of parameters and reinforce the sharing of common characteristics.

Recently, convolutional neural networks have been widely exploited in computer vision thanks to their parameter reduction strategy and the availability of large volumes of data. In addition, the evolution of storage and computational capacity has encouraged the computer vision community to propose other deeper CNN-like architectures. In classification, the proposed architectures optimize classical convolution layers. The main purpose of this variation is to reduce the number of parameters and add additional layers that improve nonlinearity. In detection and segmentation, the main goal was to adapt the proposed architectures to real-time applications. Due to the significant progress of CNN architectures, they have been exploited in

several real-world applications, such as image classification, object detection and localization, image forgery detection, facial recognition, and medical image processing.

In image forgery detection, we have defined it as the operation of adding, removing, and/or modifying certain significant regions or features of an image without discernible traces. Different methods have been used to tamper with an image: Copy-move, Splicing, Lighting inconsistency, Interpolation, and geometric transformations. CNNs are trained on these images to solve different tasks in detecting tampered images: classification, localization, detection, and segmentation.

In medical imaging, digitized images have several types such as ultrasound (US), X-ray, computed tomography (CT) and resonance imaging (MRI), positron emission tomography (PET), and histological slides. CNNs are trained on these images to solve different tasks in medical imaging: classification, localization, detection, and segmentation.

The proposed methods in this thesis are interested in solving two problems by deep learning methods. The first problem is related to the detection of medical images and in particular, CT images by CNN networks. Moreover, the second problem is related to image forgery detection by CNN networks.

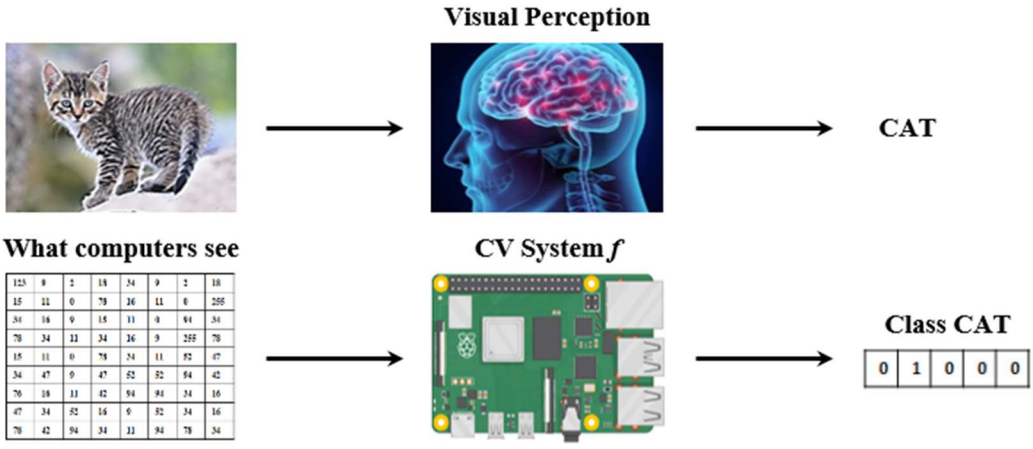

Fig. 1. Image Recognition unlike human visual perception, computer vision.

Problem and Motivation:

Although great progress has been made in image processing, the problem of understanding images is much more complex. Our goal is not just too simply knowing what objects are in the picture; we want to know more about the characteristics of the image, for example, image forgery detection, we aim to know where the manipulated regions are and what is the non-manipulated regions. In other words, we want to partition the image into meaningful regions (manipulated, non-manipulated): pixels in the same region share certain characteristics while adjacent regions are significantly different with respect to the same characteristics. This task is known as image segmentation, one of the great challenges in the long history of computer vision. According to the separation criteria, there are different types of segmentation problems. The most classic is semantic segmentation where the separation criterion is the semantic meaning. Each pixel is classified into a unique class from a set of predefined semantic classes as shown in Figure 2, and a connected region with the same label represents a "manipulated region" in the semantic segmentation task. In this way, we know exactly where the manipulated regions and the non-manipulated regions are in the

image as well as their explicit semantic contours. Another type of image segmentation can also be formed accordingly. For example, foreground-background segmentation aims to segment the most prominent object in the image while instance segmentation solves the problem of segmenting multiple instances of the same object in a scene. Segmentation in time-space is also an important area. For example, in object tracking and motion segmentation problems, pixels need to be classified not only based on spatial information, but also over time.

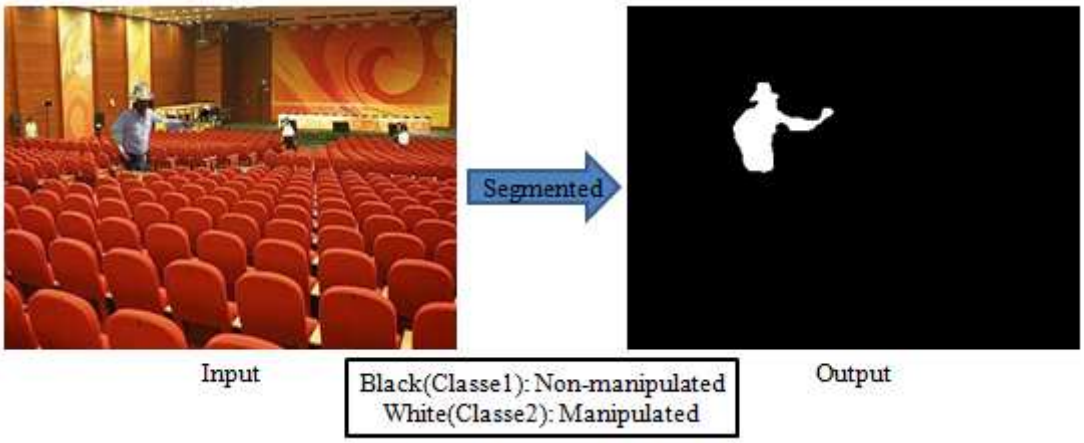

Fig. 2. Semantic Segmentation An illustration of the image forgery detection task. The whole image is partitioned into segments based on semantic meaning. Each pixel in the image is classified into of the two semantic classes and a connected region with the same label represents a 'manipulated-region'.

Despite the advantages of DL methods over ML methods and the success of CNNs in different fields, the latter have several challenges related to overfitting issues on limited data volumes, their requirements in terms of storage and computational capacity, and the considerable inference time in detection and segmentation. It is interesting to note that despite the availability of WSDs and their advantage in data collection, the number of available images remains limited for DL type applications. In addition, the annotation of these data and especially in segmentation requires a considerable effort by the experts, hence the need for automatic methods to solve these problems. Several efforts have been made in the state of the art to adapt DL methods, CNNs, to learning on limited volumes of data:

- **Transfer learning from pre-trained models to the ImageNet learning base:** this technique reduces the various problems related to over-learning, because only a subset of layers is trained on the new task. In addition, this process reduces the considerable time required to learn DL methods.
- **Data augmentation methods**: this technique makes it possible to generate several images from a single original image, which makes it possible to generate a considerable amount of data and therefore reduces over-fitting problems.
- **Regularization methods**: Regularization methods such as L1 and L2 regularization, dropout regularization, data augmentation, and premature termination have been widely exploited in DL applications. These techniques make it possible to reduce the large variance between the performance on the training and validation data and therefore they present a good tool for improving the generalization of the trained models.

Despite the efforts made, several issues remain in semantic segmentation, such as the choice of the optimal architecture for solving the task in question, the choice of suitable hyper-parameters, the choice of the appropriate model among several models recorded during the iterative learning process, the reduction of intra-variability and inter-variability between the decisions of different CNN-like architectures and the link between the ImageNet learning base and the other learning bases either in the field of medical images or in the field of detection of manipulated images.

Contributions and Outline

To answer the problem exposed above, we propose through this thesis three contributions for the semantic segmentation of images.

The first and the second contribution are interested in the problem of the detection of falsified images. In these contributions we have exposed two works, the first contribution; «***Encoder-decoder based convolutional neural networks for image forgery detection*** » provides a novel method based on convolutional neural networks for localization of manipulated regions; we also used the concept of transfer learning to improve the learning performance of our method. The second contribution; ***« Efficient balanced focal loss function for manipulated images detection*** » addresses the problem of overfitting in the case of unbalanced classes; To solve this problem, we used a loss function based on adaptive penalty factors to overcome this problem. For each of the contributions we presented an experimental study to evaluate their performance, show their robustness and their efficiency.

The third contribution; ***« CovSeg-Unet: End-to-End method-based computer-aided decision support system in lung covid-19 detection on CT images***»; is to detect lung infections in medical images, especially in CT images. We have proposed a framework based on coder-decoder convolutional neural networks. We also introduced a new loss function to reduce the high variance problem of CNNs, which helps reduce the learning problem on limited volumes of medical data.

Organization of the thesis

The content of the thesis is organized into chapters as follows:

The first two chapters present the background related to DL methods, to the techniques used in this thesis. The third chapter details the state of the art in the fields of application of DL methods, especially in the processing of medical images and falsified images. The fifth chapter presents the theoretical contribution as well as the state of the art of the work related to our contributions. Finally, the last chapter which designates the body of this thesis presents all the experimental contributions.

- **Chapter 1** introduces the theoretical notions related to deep learning (DL). It details the origin and history of neural networks as well as their evolution. Then, it presents the general architecture of a deep neural network and the techniques used in learning. Finally, it illustrates the configuration of some known architectures in supervised and unsupervised learning.
- **Chapter 2** focuses on convolutional neural networks (CNN). First, it presents the origins and development of this network. Then, it introduces the general architecture as well as the

different techniques used in learning these networks. Finally, it explains and compares the different known CNN-type architectures in classification, detection, and segmentation.

- **Chapter 3** presents the state of the art related to the application of CNNs in computer vision, such as image classification, object detection and localization, semantic segmentation, detection of falsified images, segmentation of medical images, and facial recognition.
- **Chapter 4** presents the work related to the application of DL networks in the field of medical imaging. First, it introduces a state of the art on the methods proposed in this context. Then, this chapter details the proposed method of detection. Finally, an experimental study is presented with the experimental results
- **Chapter 5 and 6** of this thesis presents the second and the third contribution which aim to segment the falsified images, this chapter presents the proposed architecture which aims to segment the manipulated images based on an encoder-decoder network, and the second contribution aims to solve the problem related to overfitting. We introduce in both chapters experimental results and comparative studies with state-of-the-art approaches.
- **Chapter 7**; in this chapter we treat in detail the third contribution of this thesis which consists in the detection of pulmonary infections in medical images, and we show the detailed experimental results which prove the effectiveness of our contribution. At the end of this chapter, we analyze the general conclusions of this contribution.
- **Chapter 8** Conclusion and Perspectives the last chapter concludes the work developed in this thesis and proposes future directions for research in deep learning.

Chapter I: Introduction to artificial intelligence

Artificial intelligence (AI) is a discipline of computer science that aims to create intelligent and autonomous machines capable of making decisions without human intervention. The term AI was first coined by John McCarthy at the Dartmouth Summer Research Project on Artificial Intelligence conference, a summer workshop held in 1956. The term "artificial intelligence" refers to machines and algorithms, or programs that allow a machine to simulate and go beyond human cognitive abilities such as thinking, planning, learning, and understanding to make decisions and solve problems. The AI concept is not new; it dates to 1950 when Alan Turing invented the Turing test. Then the first Chatbot computer program "ELIZA" was invented in the 1960s. Then the IBM computer "Deep Blue" was invented in 1977 that beat the world champion in chess etc. In short, we can define classical AI as a non-biological system that exhibits human-like forms of intelligence.

AI research has progressed faster in recent times thanks to the exponential evolution in the microelectronics field; this evolution has given rise to several disciplines belonging to the AI field such as deep learning (DL). Deep learning is much more powerful than classical algorithms; it is part of the statistical methods family based on deep neural networks and belongs to the large Machine Learning (ML) methods family. Thanks to the superior abilities of artificial neural networks to learn complex behaviours from massive amounts of data, they are at the heart of the new automation wave.

In this chapter, a general machine learning overview is presented. This includes the different types of machine learning, deep learning, and different components of the neural network, the most well-known architectures by detailing the problems that each of them suffers with the proposed solutions.

Contents

1.1 Machine Learning:

Machine learning is a branch of artificial intelligence that allows a machine to learn from a large volume of data. This very powerful technology is widely used in several fields requiring the extraction of

information from data such as search engines, falsified images detection, facial recognition, medical assistance applications, self-driving cars, etc. However, the main purpose of machine learning is to generate a model to automate classification, regression, association, or clustering tasks. These learning processes are categorized into supervised learning (classification and regression), unsupervised (associations and clustering), and reinforcement learning.

Supervised learning is characterized by using a learning database to train models that can predict the desired output. This training dataset D = {x, y} is composed of inputs X and true outputs Y (In regression, output values are continuous (see Figure 3A). In classification, output values are discrete and called classes). There are several types of supervised learning algorithms such as Support Vector Machines (SVM), Nearest Neighbors (KNN), Decision Trees (DT), Linear Regression (LR), etc. Unlike supervised learning which aims to find an estimation function f: X → Y that describes the relationship between the input space X and the space of possible outputs Y. Unsupervised learning only takes input data X without possible output data Y (no variable to predict) $d = \{x\}_N$. This learning process seeks to find a type of structure based only on the input data X, such as data points clustering. However, reinforcement learning is the fact that the model learns behavior from a series of observations that lead to a specific outcome.

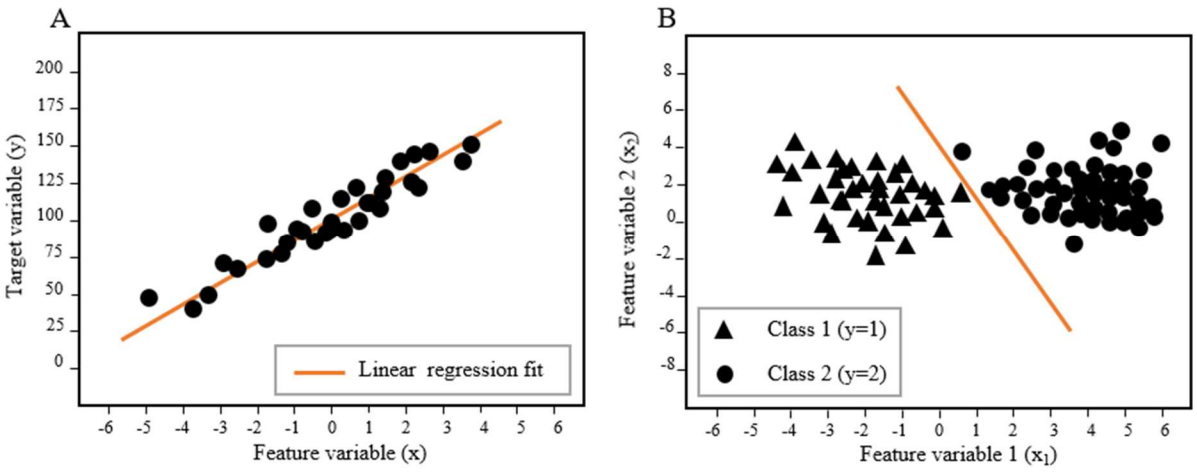

Fig. 3. Illustration of the two main categories of supervised learning, regression (A) and classification (B)

1.1.1 Supervised learning:

Supervised learning focuses on predictive tasks that aim to build a mapping function to acquire information, have predictions, make decisions or build models from related data. For example, a popular application of supervised learning is image classification, a dog or a cat. Here, an image (the data element) should be classified as a cat or a dog. Following the classic computer science approach, one could write a designed algorithm that follows certain rules to decide whether an image is a cat or a dog. Although such an algorithm might logically work well for a while, it has significant drawbacks. As when the characteristics of the image change, it must be rewritten. Programmers may attempt to reverse engineer the algorithm and design algorithms that work around it. And even if it succeeds, it cannot be easily applied to different images.

Supervised learning uses a different approach to design a model that can predict the outputs of related data. The model is learned by itself from the data instead of the programming rules. The learning process works if we have input data for which we know exactly what the exact associated prediction would have. For the

same example, the expert as a cat or a dog annotates the input image. A machine learning model can use this data to train a model, a classifier, to predict the correct label for each piece of annotated data.

We now propose a more formal definition, taken from [254]. We consider that X is the set of inputs and Y the set of outputs or true labels, there is an unknown probability distribution on the set $X \times Y$, which we will denote $p(x_i, y_i)$ with $(x_i, y_i) \in X \times Y$.

The supervised learning process aims to find the best guess function, or mapping function $f : X \rightarrow Y$ that describes the relationship between an input space X and an output space Y. i.e., from known values of X, we come to give a prediction of the values of Y, given a training set of input-output pairs $(x, y) \in X \times Y$, there is an unknown probability distribution on the set $X \times Y$, which we will denote $P(x, y)$ with $(x, y) \in X \times Y$. However, we will denote $\hat{y} = f(x)$ predicted label, and we aim to estimate f such that it provides output predictions only on the input data of a validation set following the same distribution P. We will use a loss function $L(\hat{y}, y)$: $Y \times \hat{Y} \rightarrow R+$ which measures the variance between a predicted label and a y_i truth label as shown in the equation (1)

$$R(f) = \int_{X \times Z} L(\vec{z}, z) p(x, z) dx dz \quad (1)$$

The objective of the learning process is to find the optimal function f^* of a class of functions F to minimize the risk $R(f^*)$ (Equation 1). In practice, it is then necessary to define a subspace of $X \times Y$, which we will call training set T, composed of N elements from $P(x, y)$. Thus, in this set, each xi is associated with an output y_i forming N pairs. This set T makes it possible to minimize the empirical risk $R_n(f)$, which is defined as follows:

$$\mathbb{E}_{(x,y)\sim D} l(f(x), y) \quad (2)$$

$$R_n(f) := \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i) \quad (3)$$

$$f^* = argmin_{f \in F} \{R_n(f)\} \quad (4)$$

The minimization of $R_n(f)$ is not sufficient to find an optimal value of f^*, it will be necessary to add a penalty factor $r(f)$ to reduce the variance without significantly increasing its bias, this regularization factor makes it possible to reduce the complexity of the model.

$$f^* = argmin_{f \in F} \{R_n(f + r(f)\} \quad (5)$$

By formulating $r(f)$ as a prior on f^*, Bayesian learning gives a statistical explanation of the regularization. Several other practical methods and theories have been devoted to this problem (Bishop 2006).

1.1.2 Unsupervised Learning:

The previous section defined supervised learning, which is the most important subclass of machine learning. In the latter, we have the elements of the output space Y, making it possible to model the supervision that we considered that we had annotations or ground truth mask.

This section introduces the second category of machine learning which is unsupervised learning. In which no labelling information is given. Therefore, the model must discover the hidden structure in the data on its own rather than predicting continuous or discrete labels. Tasks of this type of learning are often called data partitioning (clustering). K-means is an example of an unsupervised algorithm, where the output of the algorithm is a group of labels. He attributes to each of the labels a reference point xi as being the barycenter of the groupk_i. Each group of data is thus defined by creating a barycenter for each of these groups. Centroids are like the kernel of the cluster, which snaps the points closest to them and adds them to the

cluster. The centroids are updated relative to the dataset and the chosen distance measure relative to that data. There are many other unsupervised learning algorithms such as those oriented towards data dimension reduction such as PCA Principal Component Analysis, Singular Value Decomposition (SVD) but also those looking for an estimate of the data distribution density, such as kernel estimation (KDE)

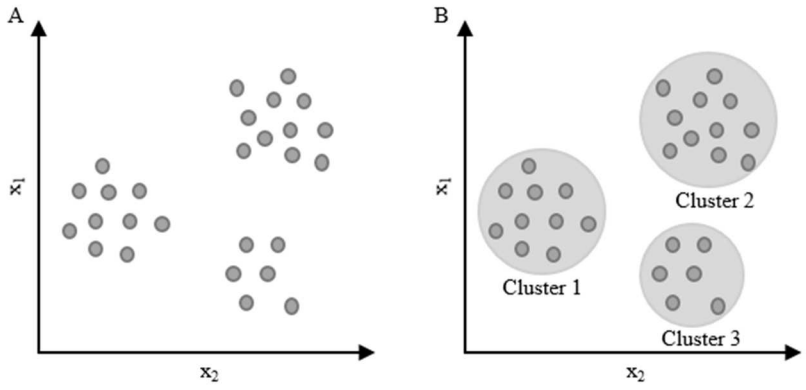

Fig. 4. Illustration of clustering. (A) A two-dimensional, unlabeled dataset. (B) Clusters inferred by a clustering algorithm that groups similar points into the same cluster.

1.1.3 Reinforcement Learning:

The third and most complicated subclass of machine learning is reinforcement learning (Figure 5). Unlike supervised learning, which focuses on predicting a specific outcome, reinforcement learning involves learning a series of actions that lead to a specific outcome. To illustrate reinforcement learning in the context of Figure 5, (1) given a chessboard state S_t and a reward value R_t at iteration t, (2) the reinforcement learning agent selects an A_t action that moves one of the pawns two fields. (3) Next, the environment considers producing the next state S_{t+1}, and the corresponding reward for performing the action, R_{t+1}. This cycle repeats until the end of the episode.

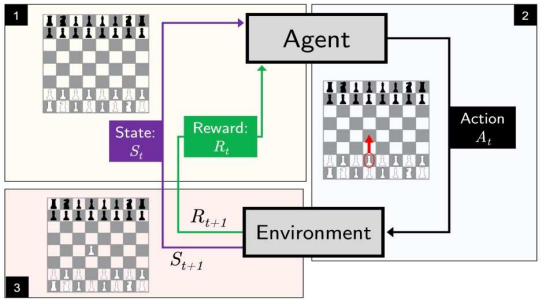

Fig. 5. Illustration of renforcement Learning

In this section, we started with a general introduction to machine learning, and then we briefly presented its different categories such as supervised, unsupervised, and reinforcement learning. In the following section, we deal exclusively with models of neural networks that are the subject of our thesis. We provide more technical operating details of neural networks. Then we present different existing methods in the literature to improve the performance of such models.

1.2 Introduction to Deep learning:

In this section, we process the basic notions of deep learning that we will use in the next chapters of this doctoral thesis. As mentioned in the introduction, neural network approaches belong to Machine Learning (ML) methods family. These approaches have been exploited in supervised and unsupervised learning contexts [1], [2]. Several of these methods have been designed, developed, and implemented to allow machines to evolve through a learning process, and thus perform tasks that are difficult to achieve with classical methods.

The theory behind deep learning is therefore not new, its foundations date back to the 1940s with the work of (McCulloch and Pitts, 1943; Hebb, 1949; Rosenblatt, 1958), which consists of several approaches inspired by the brain functioning of neurons and stack multiple layers of neural networks to transform raw data from one representation space to another with more discriminating features. Although the evolution and massive increase in data and compute infrastructures have revealed the full power of deep learning, it still suffered from several difficult issues such as over-fitting, under-fitting, gradient degradation and the high time complexity [3], [4] and [5], several of the models proposed in the supervised and unsupervised learning literature aim to solve these problems.

Through this section, we detail the different notions related to the deep learning model from the smallest element (neuron or perceptron) to the overall functioning of a multilayer network.

1.2.1 History and inspiration:

Artificial neural networks are inspired by the functioning of biological neural networks in the brain. The human brain contains a considerable number of massively connected biological neurons. These neurons exchange messages by signals transmitted across synapses. These biological systems are characterized by their great complexity compared to artificial systems because of the high number of massively interconnected neurons that cannot be processed by a machine.

Figure 6 illustrates the difference between a biological neuron and an artificial neuron. A biological neuron is composed of several structures like dendrites, nucleus, cell body, and axon. These neurons exchange messages through the signals that are transmitted between the different components. First, the dendrites receive signals from the synapses of neighboring neurons. Then these signals are grouped together and processed by the cell body. The latter generates an impulse that is transmitted to the axon if the received signal exceeds a certain threshold. Finally, the generated signal is transmitted to the other neighbouring neurons through synapses. This signal is modified according to the nature of the synapses. Excitatory synapses increase this signal, while inhibitory ones reduce its value. An artificial neuron mimics a biological neuron in some functions. These neural systems are presented using mathematical notions. The input signals are formulated by the values x_i, $i \in [1, N], i \in N$, the dendrites are the weights w_i, the cell body is the combination function, and the axon is the value output. An artificial neuron can be composed of a set of nodes or neurons which are associated with a value x_i and a weight w_i. These values are combined by a combination function and the result is processed by an activation function f.

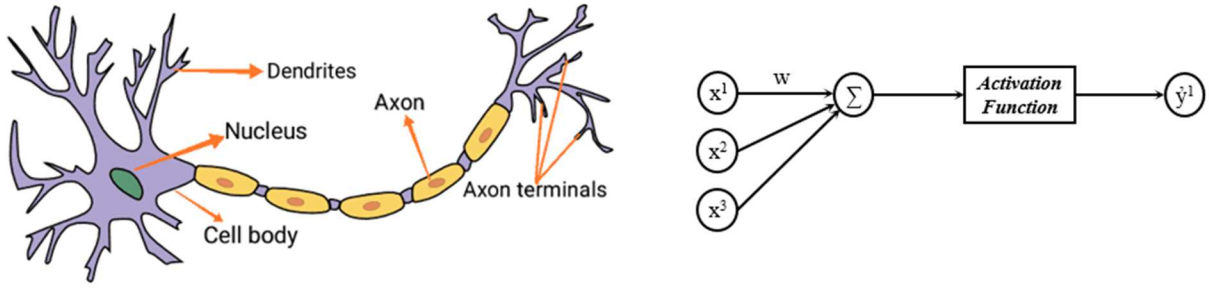

Fig. 6. The difference between (A) a biological neuron and (B) an artificial neuron

ANN research began in the 1940s when developments in neurobiology encouraged researchers to formulate the behavior of biological neurons. In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts developed a mathematical model [6] of brain neurons based on Boolean functions and demonstrated that, when combined, they can calculate weights w_i. They modelled a simple neural network through an electrical circuit. The independence of this approach from learning and its manual processing for the calculation of weights w_i have limited its use in other fields of application. To solve learning problems in ANN, psychologist Donald Olding Hubb proposed seminal work on the learning process [7]. His main idea is that learning in the brain occurs primarily through the formation and signal changes of synapses between neurons, known as synaptic plasticity. In 1957, psychologist Rosenblatt introduced the notion of perceptron for binary classification based on the works of Warren Mcculloch, Walter Pitts [6], and Donald O. Hubb [1]. Unlike the Boolean approach proposed above, this model is based on learning for the adjustment of the weights w_i. In 1968, work on neural networks stagnated after the work of Marvin Minsky and Seymour Papert [8] who showed the limits of the perceptron in solving a simple Boolean XOR function because of its adaptation to linear separations. All these factors have discouraged the AI community from pursuing research in this area. This limitation has been solved by the introduction of the notion of the multilayer perceptron (MLP) which offers more nonlinearity through the additional hidden layers. Then, MLPs were developed into deep learning networks (DNN) which are characterized by more than two hidden layers. Despite the power of DNNs in nonlinear separations, machine-learning algorithms like SVM have seen more success thanks to their optimized time complexity compared to DNNs. Until 2012 when the Alex Net deep learning architecture scored an impressive error rate on the Image Net learning base [9]. The proposed architecture and graphics-processing unit (GPU) technology has encouraged researchers to exploit deep learning methods in other application areas. In the following, we present a formal perceptron description.

1.2.2 Perceptron:

The first formulation dating from 1958 is that of the perceptron [10], initially proposed in the case of linear classification. It makes it possible to predict the values of the weights w_{ij} of an artificial neuron. The goal of perception is to solve linearly separable two-class problems. For a learning base $D = \{X, Y\}_N$ $n = 1, y_n \in \{0, 1\}$ where $X_n = \{x_1, x_2 \ldots x_m\}$ are the instances and y_n are the classes; there is a hyper-plane which makes it possible to separate the different instances into two classes (figure 7).

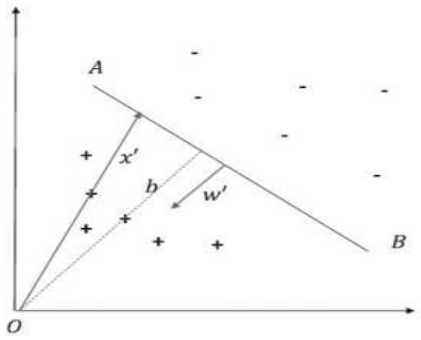

Fig. 7. The hyperplane AB separating two classes

The perceptron is characterized by n inputs and a single output (figure 6: B). The neuron receives the input values which are associated with weights w_i. Then the inputs are linearly combined with the weights through a combination function and the output-weighted sum is provided to the activation function F (equation 6). Equation 7 illustrates the process of calculating the classes.

$$\tilde{y} = f(x) = \Phi(x.w + b) = \Phi(\sum_{i=1}^{D} x_i w_i + b) \quad (6)$$

$$\tilde{y}_j = F(b + \sum_{i=1}^{n} w_{ij} x_i) \quad (7)$$

$$\text{Where } F(Z) = \begin{cases} 1 & si\ z > 0 \\ 0 & si\ non \end{cases}$$

Where $\hat{y}_j$ is the predicted class at iteration j, and b is the bias it has an approximate distance from the origin.

The perceptron training consists of finding a hyper-plane $(W_T X + b) = 0$ separator (Figure 7) that allows to correctly classify the different instances in an iterative process, its objective is to find the best weights and biases that make $\hat{y}$ close to the true label. The learning algorithm consists of updating the weights by increasing or decreasing w if the output $\hat{y}$ is lower or higher than the true label y as indicated in Alog-1:

Algorithm1: Perceptron

Randomly initialize weights w_i and the bias b;

While i ≤ iterations **do**

Predict the y_n classes according to equation 7;

Calculate the new weights W(t) according to equation 8;

$$W^{(t)} = W^{(t-1)} + \alpha(y_n - \tilde{y_n})x_n \quad (8)$$

end

This algorithm continues to learn until the model has been able to classify the instances; the stopping criterion depends on the convergence of the performances or on a defined number of iterations. It has been shown in [11] that in the case of two linearly separable classes, this algorithm converges in a finite number of iterations. However, in the case where the two classes are not linearly separable, the algorithm never converges, consequently, it will loop endlessly.

The perceptron can only work with two classes; it does not solve certain types of problems, proving inefficient for example in the case of non-linear classification. Figure 9 illustrates a dataset with nonlinearly separable classes, which therefore needs to exhibit nonlinear boundaries. One way to solve this problem is to nonlinearly project the data into a space where it will be linearly separable, exhibiting for example more dimensions than in the previous space (kernel trick). This approach type is widely used in other statistical learning approaches such as Support SVM [12]. The idea of MLP [13] is then, as the name suggests, to

bring together several layers of perceptron, separated by non-linearity's, to create a space of representation of the input data, which is linearly separable. For example, for an MLP with one hidden layer, the input x is processed by a first perceptron and gives an intermediate output, called hidden, h. This is injected into a second perception, which makes it possible to obtain a final output y.

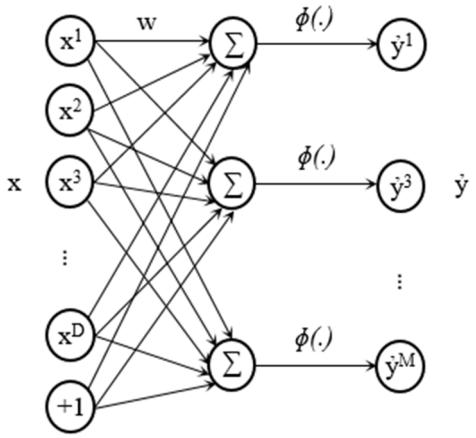

Fig. 8. Multiclass perceptron (Notation x^i is the i^{th} component of x. $\dot{y}^j$ is the j^{th} component of $\dot{y}$

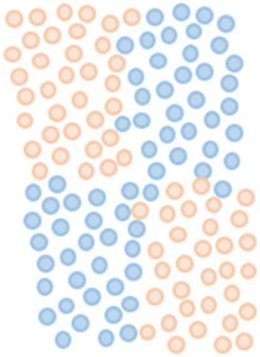

Fig. 9. Example of a nonlinearly separable problem. Circles represent data x, corresponding to their annotation z

In an MLP (figure 8) each node is characterized by a value x_i and contains connections to adjacent layers, which are presented by weights w_{ij}. Each x_i value in the current layer presents an entry to the next layer. The number of nodes in the input layer depends on the number of attributes in the learning base. On the other hand, the topology appropriate to the hidden layers is chosen randomly or according to an optimization procedure to maximize the performance of the addressed problem. The x_i values of the hidden layers are calculated according to equation 9, where $x_i^{(k)}$ is the value of node i of layer k, F is the activation function, w_{ij} are the weights associated with x_i, and m is the number of nodes in the next layer.

$$x_i(k) = F^{(k)}(\sum_{j=1}^{m} w_{ij} x_j^{(k-1)} + b_j \quad (9)$$

Training in an MLP consists of adjusting the weights w_{ij}. Equally by applying for each perceptron the perceptron learning rule described previously to minimize the error between the predicted classes and the true classes.

In the next section, we present the descent gradient method functioning, which is based on error correction between the predicted output and the true output to train a perceptron.

1.2.3 Learning based on Gradient descent

In 1960, Professor Bernard Widrow and his student Ted Hoff proposed a linear model (Adaptive Switching Circuits) at Stanford [14]. The model is an electrical circuit based on a new circuit called a memistor, which is a resistor with memory. It is very similar to a perceptron model with a difference in how it learns w-weights and b-bias from the data. They named their model ADALINE for Adaptive Linear Elements. The main difference between the perceptron and the ADALINE is that the latter works by minimizing the root mean square error of the predictions of a linear function. This means that the learning procedure is based on the result of a linear function rather than the result of a threshold function as in the perceptron. The squared error is as follows: (10)

$$l(f(x), y) = \frac{1}{2}\sum_{j=1}^{M}(y_j - f_j(x))^2 \tag{10}$$

Consequently, the minimization of the mean squared error over all N training bases is expressed as:

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} l(f(x^{(i)}), y^{(i)}) \tag{11}$$

ADALINE solved the optimization problem using the so-called gradient descent algorithm as mentioned in function. However, gradient descent is a well-known and easy-to-use iterative algorithm that minimizes the function L(w), parameterized by the weights W of a neural network. Minimizing the function (i.e., finding the W parameters that minimize the empirical error) is done by calculating the gradient and updating the W parameters in the opposite direction to it. Intuitively, the gradient gives the direction of the slope of the function L at point W, so going in the opposite direction of the gradient is equivalent to descending this slope. This direction is found by calculating the derivative of the total train loss L(W) with respect to the weight vector using the chain rule. Using the linearity of the derivatives of Equation.11, we calculate the derivatives of an instance x_i in Equation 10 and then average over all the samples to obtain $\partial L\ \partial W$, the derivatives of the total loss in Eq.11. For a training sample, deriving the loss in Equation.10 with respect to a parameter weight gives:

$$W \leftarrow W - \alpha \frac{\partial L(W)}{\partial W} \tag{12}$$

A hyper-parameter η, called learning step, determines the length of the step to take in the direction of the gradient. Once the parameters have been updated, the new gradient is calculated, and we start again in the opposite direction. This minimizes the function when stepping in the direction of the slope.

$$\frac{\partial l}{\partial W^{ij}} = \frac{\partial l}{\partial \tilde{y}^j}.\frac{\partial \tilde{y}^j}{\partial W^{ij}} \qquad \forall i = 1, \dots\dots\dots, D+1, \forall j = 1, \dots\dots, M \tag{13}$$

$$\frac{\partial l}{\partial \tilde{y}^j} = \frac{\partial \frac{1}{2}(\tilde{y}^i - y^i)^2}{\partial \tilde{y}^j} = \tilde{y}^j - y^j \Rightarrow \frac{\partial l}{\partial \tilde{y}} = \tilde{y} - y \tag{14}$$

The notation x^i is the i^{th} component of $\tilde{y}$, and w^{ij} is the component at the i^{th} row and j^{th} the column of W. In the case where there is no activation function,

$$\frac{\partial \tilde{y}^j}{\partial W^{ij}} = \frac{\partial(\sum_k x^k W^{kj})}{\partial W^{ij}} = x^i, \quad \forall i = 1, \dots\dots, D+1, \forall j = 1, \dots\dots, M \tag{15}$$

This gives the delta approach learning algorithm,

$$W \leftarrow W - \alpha \frac{\partial L(W)}{\partial W} = W - \alpha x^T . (\hat{y} - y) \quad (16)$$

The delta rule (Eq.16) is like the perceptron learning rule (Eq.12) except for the learning rate α.

$$\frac{\partial \hat{y}^j}{\partial w^{ij}} = \frac{\partial \hat{y}^j}{\partial h^j} \frac{\partial h^j}{\partial w^{ij}} \quad (17)$$

$$= \frac{\partial \Phi(h^j)}{\partial h^j} \frac{\partial \sum_k x^k W^{kj}}{\partial W^{ij}} \quad (18)$$

$$= \frac{\partial \Phi(h^j)}{\partial h^j} . x^i \quad (19)$$

Therefore, the delta rule in this case is,

$$W \leftarrow W - \alpha x^T . \nabla_h \Phi(h) . (\hat{y} - y) \quad (20)$$

In the case of the perceptron, which uses the Heaviside step function, $\nabla_h \Phi(h)$ is not defined at zero and it is equal to zero everywhere else which makes it impossible to apply the delta rule on the perceptron. This led to the use of differentiable functions such as sigmoid functions. The next section gives more details on the other activation functions.

1.2.4 Improving-optimization method

Hyper parameter tuning is one of the important techniques that improve the performance of an MLP. The optimizer is a hyper parameter that has a great influence on the performance and convergence of an MLP.

In the previous section, we detailed the optimization process in an MLP which is mainly based on the gradient descent method. To improve and accelerate training, different variants of GD have been proposed in the literature such as stochastic gradient descent (SGD) and mini-batch gradient descent (BGD). In addition, several methods have been developed to optimize GD such as inertial descent; Nesterov accelerated gradient descent (NAG), Adagrad, AdaDelta, Adam, and RmsProp.

- ***Stochastic Gradient Descent (SGD)***

In the GD method, the gradients are calculated based on the all-training database; this limits its use on large volumes of data because of the limited memory space and the very high training time. To solve these problems, the stochastic gradient descent (SGD) method is exploited. In SGD, the calculation of gradients and the updating of weights are applied to each instance in the learning base. These instances are randomly chosen during the optimization process. Random selection and exploitation of a single instance can speed up training time and improve generalization. Despite these advantages, frequent updates of the weights w^{ij} at each iteration can cause large variance and unstable convergence. To minimize the number of updates, the mini-batch stochastic gradient descent method is proposed. The optimization process of SGD is given in Alg. 1.

Algorithm 1 Stochastic gradient descent (SGD)

Input: Learning rate η.
Input: Initial parameter θ.
1: **while** Stopping criterion is not met **do**
2: Sample a mini-batch of m examples from all training samples $\{x^{(1)},...,x^{(m)}\}$.
3: Compute gradient estimate: $g = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta), y^{(i)})$
4: Apply update: $\theta = \theta - \eta g$
5: **end while**
6: **return** Updated parameter θ.

- ***Mini-batch gradient descent (BGD)***

In mini-batch gradient descent method, parameter update depends on one batch of data. This batch is composed of a defined number of instances chosen randomly at each iteration. Then, the parameter update process is executed based on the selected data set. Mini-lots reduce the gap between old and new weights when updating. This implies a more stable convergence compared to SGD. Currently, this method is frequently used in deep learning works and referenced by SGD instead of BGD. The optimization process in the SGD method presents several challenges related to:

- The choice of the optimal value of the learning rate η and its fixed value during learning. Large values of η can cause rapid convergence to a local minimum. On the other hand, small values lead to slow convergence. In addition, the value of η must vary and depend on the frequency of the input instances.
- The risk of being trapped in a local minimum due to the non-convex nature of the cost function. A function is non-convex (figure 10) if it admits a global minimum as well as other local minima. This problem can trap gradient descent to various local minima.

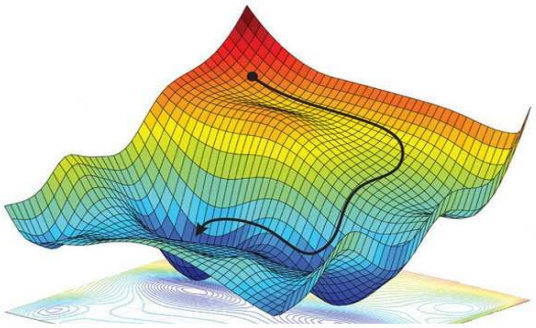

Fig. 10. Non-convex function

- ***SGD with Momentum***

Descent with inertia [15] is an optimized version of the SGD method. This technique makes it possible to accelerate convergence and reduce the various problems related to the non-convex nature of the cost function. This method introduces the notion of speed that depends on the inertia parameter α. This makes it possible to reduce parameter updates when the gradient changes sign and to speed them up if the gradient is in the same direction of v. Equation 21 illustrates the process of calculating parameters where α is the inertia parameter and η is the learning rate. The optimization process of momentum is given in Alg. 2.

$$\begin{cases} v^{(t+1)} = \alpha v^{(t)} - \eta \nabla C(\theta^{(t)} \\ \theta^{(t+1)} = \theta^{(t)} + \vartheta^{(t+1)} \end{cases} \quad (21)$$

Algorithm 2 Stochastic gradient descent (SGD) with momentum

Input: Learning rate η, momentum hyper-parameter α.
Input: Initial parameter θ, initial velocity v.
1: **while** Stopping criterion is not met **do**
2: Sample a mini-batch of m examples from all training samples $\{x^{(1)},...,x^{(m)}\}$.
3: Compute gradient estimate: $g = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta),y^{(i)})$
4: Compute velocity update: $v = \alpha v - \eta g$
5: Apply update: $\theta = \theta + v$
6: **end while**
7: **return** Updated parameter θ.

- ***Nesterov's Accelerated Gradient Descent***

Nesterov's Accelerated Gradient Descent [16] is an optimized version of the inertial descent method. The inertia technique has limitations that are related to large updates of the weights w (Figure 11: 4a). This can prevent the optimization process from detecting the global minimum. To avoid these large jumps, NAG proposes to calculate the speed according to the gradient of the next step instead of the current step. According to equation 22, NAG starts with a partial update (4a) to calculate the intermediate parameter $\theta^{(t+1/2)}$. then, the final parameter$\theta^{(t+1)}$. is adjusted according to the gradient of the intermediate parameter (4b). Changing the sign of the gradient of the cost functions between the parameters $\theta^{(t+1/2)}$ and $\theta^{(t+1)}$. ensures steps backward by reducing the magnitude of the updates.

$$\begin{aligned} \theta^{(t+\frac{1}{2})} &= \theta^{(t)} + \alpha\vartheta^{(t)} \\ \vartheta^{(t+1)} &= \alpha\vartheta^{(t)} - \eta\nabla C(\theta^{(t+\frac{1}{2})} \\ \theta^{(t+1)} &= \theta^{(t)} + \alpha\vartheta^{(t+1)} \end{aligned} \quad (22)$$

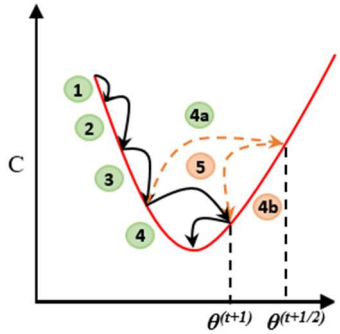

Fig. 11. Nesterov's accelerated gradient descent

- ***Adagrad***

In SGD, the learning rate η is fixed and applied equally to all weights. This is because less frequent features require larger updates. To answer this problem, the Adagrad method [17] proposes to adapt the learning rate η to each parameter. This makes this approach more suitable for sparse learning bases in DL. Equation 23 illustrates the procedure for calculating the parameters η (t+1).

$$\theta^{(t+1)} = \theta^{(t)} - \eta G_{(t)}^{-1} \nabla C\left(\theta^{(t)}\right)$$

$$G_{(t)} = \begin{bmatrix} \sqrt{\sum_{\tau=1}^{t} \theta_1^{(\tau)^2} + \epsilon} & \cdots\cdots\cdots & \cdots\cdots \\ \cdots\cdots & \sqrt{\sum_{\tau=1}^{t} \theta_1^{(\tau)^2} + \epsilon} & \cdots\cdots \\ \cdots\cdots\cdots & \cdots\cdots\cdots\cdots & \sqrt{\sum_{\tau=1}^{t} \theta_1^{(\tau)^2} + \epsilon} \end{bmatrix} \quad (23)$$

- ***AdaDelta***

In Adagrad's method, the learning rate $\eta G_{(t)}^{-1}$ decreases continuously because of the cumulative sum $\sqrt{\sum_{\tau=1}^{t} \theta_1^{(\tau)^2} + \epsilon}$ of the previous iterations. This leads to very slow convergence due to low values of the learning rate. The AdaDelta method [18] is an optimized version of Adagrad that solves the learning rate degradation problem by considering the exponential moving average of the squared gradients. Equation 24 illustrates the procedure for calculating the parameters θ(t+1), where γ is the exponential decay constant, η the learning rate and $g_{ij}^{(t)}$ is the mean square root of the gradients.

$$\begin{cases} g_{ij}^{(t)} = \gamma g_{ij}^{(t-1)} + (1-\gamma)(\nabla C\left(\theta^{(t)}\right))^2 \\ \theta^{(t-1)} = \theta^{(t)} - \frac{\eta}{\sqrt{g_{ij}^{(t)} + c}} \nabla C(\theta^{(t)}) \end{cases} \quad (24)$$

- ***Adam***

Adam [19] belongs to the category of methods that use variable learning rate like Adagrad and AdaDelta. In updating the learning rate, this method uses the exponential moving average of the past square gradients v^t and the past gradients m^t. The variables v^t and m^t present the first and the second moment of the gradient (equation 25) where β1 and β2 are the rates of decay. To avoid the convergence of the moments v^t and m^t towards 0, the bias-corrected moment $\widehat{m_{ij}^{(t)}}$ and $\widehat{v_{ij}^{(t)}}$ are used (equation 26). Finally, the weights $w_{ij}^{(t+1)}$ are updated according to equation 27. The Adam optimizer is particularly suitable for training networks with difficult-to-train networks (e.g., generative adversarial networks [20]) or complex architectures (e.g., U-Net [21]). Adam's optimization process is given in Alg. 3.

$$\begin{cases} m_{ij}^{(t)} = \beta_1 m_{ij}^{(t-1)} + (1-\beta_1)\frac{\partial C^{(t)}}{\partial w_{ij}} \\ v_{ij}^{(t)} = \beta_2 v_{ij}^{(t-1)} + (1-\beta_2)\left(\frac{\partial C^{(t)}}{\partial w_{ij}}\right)^2 \end{cases} \quad (25)$$

$$\begin{cases} \widehat{m_{ij}^{(t)}} = \frac{m_{ij}^{(t)}}{(1-\beta_1^i)} \\ \widehat{v_{ij}^{(t)}} = \frac{v_{ij}^{(t)}}{(1-\beta_2^i)} \end{cases} \quad (26)$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \frac{\eta}{\sqrt{\widehat{v_{ij}^{(t)}} + \epsilon}} \widehat{m_{ij}^{(t)}} \quad (27)$$

Algorithm 3 Adam

Input: Learning rate η, first-order momentum hyper-parameter ρ_1, second-order momentum hyper-parameter ρ_2, constant δ, iteration time t.
Input: Initial parameter $\boldsymbol{\theta}$, initial first-order momentum $\boldsymbol{s}$, initial second-order momentum $\boldsymbol{r}$.

while Stopping criterion is not met **do**
  Sample a mini-batch of m examples from all training samples $\{x^{(1)},...,x^{(m)}\}$.
  Compute gradient estimate: $\boldsymbol{g} = \frac{1}{m}\nabla_{\boldsymbol{\theta}}\sum_i L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}),\boldsymbol{y}^{(i)})$
  Estimate first-order momentum: $\boldsymbol{s} = \rho_1\boldsymbol{s} + (1-\rho_1)\boldsymbol{g}$
  Estimate second-order momentum: $\boldsymbol{r} = \rho_2\boldsymbol{r} + (1-\rho_2)\boldsymbol{g}\odot\boldsymbol{g}$
  Calibrate first-order momentum: $\hat{\boldsymbol{s}} = \frac{\boldsymbol{s}}{1-\rho_1^t}$
  Calibrate second-order momentum: $\hat{\boldsymbol{r}} = \frac{\boldsymbol{r}}{1-\rho_2^t}$
  Apply update: $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\frac{\hat{\boldsymbol{s}}}{\sqrt{\hat{\boldsymbol{r}}}+\delta}$
  $t = t+1$
end while
return Updated parameter $\boldsymbol{\theta}$.

- ***RmsProp***

RmsProp [22] is a variant of the Rprop optimizer [279] that is suitable for mini-batch learning. This method is considered a combination of the Rprop and SGD methods and is known for its similarity to the AdaDelta optimizer. The main goal of this strategy is to solve the degradation problem of learning rate. RmsProp divides the learning rate by the exponential moving average of the squared gradients and sets the decay constant γ of AdaDelta to 0.9. Equation 28 illustrates the process of updating the weights θ (t+1), where η is the learning rate and $g_{ij}^{(t)}$ is the mean square root of the gradients.

$$\begin{cases} g_{ij}^{(t)} = 0.9 g_{ij}^{(t-1)} \\ \theta^{(t+1)} = \theta^{(t)} - \dfrac{\eta}{\sqrt{g_{ij}^{(t)}+\epsilon}} + \nabla C\left(\theta^{(t)}\right) \end{cases} \tag{28}$$

1.2.5 Regularization techniques

A Deep Learning model must both optimize its training and generalize its prediction on training and test data. Although designing a generalized model is one of the great challenges in deep learning because of under-fitting and over-fitting issues. Under-fitting presents a model that has memorized its training data, so it performs well on the training set but not validation. It then makes bad predictions on new ones, because they are not the same as the training data. This leads to a large variance between the two and poor generalization. On the other hand, overfitting represents a model that fails to infer information from the dataset. He, therefore, does not learn enough and makes bad predictions in the training game. It is, therefore, necessary to complicate the network because it does not size well compared to the types of input data. Indeed, it fails to capture the relationship between the input data and their label.

In deep learning, networks are characterized by high variance and low bias. Figure 12 illustrates the overfitting problem, which is mainly related to model complexity, where complex models like DLs have more risk of overfitting. To remedy this, it is necessary to improve the model flexibility; we generally distinguish two notions: structural stabilization and regularization [24]. Structural stabilization controls the complexity of the network according to the variation in the number of parameters. The regularization

method consists of adding a term that penalizes the cost function. This term is used to penalize the cost function in the case of large amplitude parameters. There are several regularization methods: L1 and L2 regularization, drop regularization, data augmentation, and early stopping.

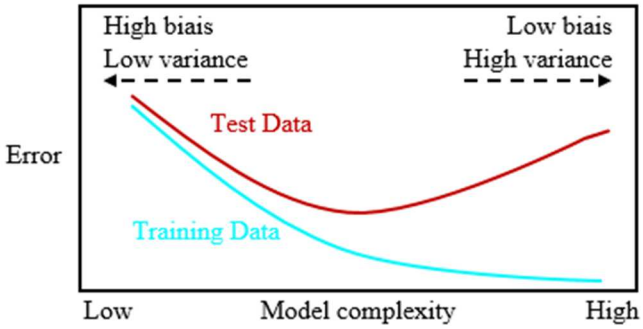

Fig. 12. Modeling the overfitting problem.

- ***L_1 and L_2 regularizations***

L_1 and L_2 regularizations are among the known techniques in regularization. These methods update the cost function by adding a regularization term. Their purpose is to decrease the values of the weights by adding the regularization term to generate simple models and avoid over-fitting problems. Equations 29 and 30 present the regularization terms L_1 and L_2 respectively, where λ is the regularization parameter, $\|\theta\|$ and $\|\theta\|^2$ are the norms l_1 and l_2 of the vector of parameters θ. The L_2 regularization technique is also known as the weight decay method.

$$L_1 = \lambda \sum \|\theta\| \tag{29}$$

$$L_2 = \lambda \sum \|\theta\|^2 \tag{30}$$

- ***Dropout regularization***

Dropout regularization [25] is a regularization method that involves ignoring or "dropping out" of a subset of neurons during training. These neurons are randomly selected based on a dropout rate and belong to the input or hidden layers. This technique reduces many links between neurons and the specialization of some at the expense of others, which will improve generalization and avoid over-fitting. The abort process generates different architectures at each iteration, and then the total performance is defined by the average performance of these architectures. This reduces the variance between these models and improves generalizability.

- ***Data-Augmentation***

Unlike classical machine learning methods, deep learning methods require a large volume of data to avoid the over-fitting problem. Collecting a large amount of data and annotating it presents a challenge in certain fields such as the medical field. To solve these limitations, data augmentation methods are proposed. This increase can be made either before the training stage (offline) or during the training on the mini-batches (online). There are several methods of data augmentation. Among the most used techniques, we have rotation, translation, scaling, gaussian noise, patch division (figure 13), color normalization, enhancement, and the generation of new instances by generative adversarial networks (GANs).

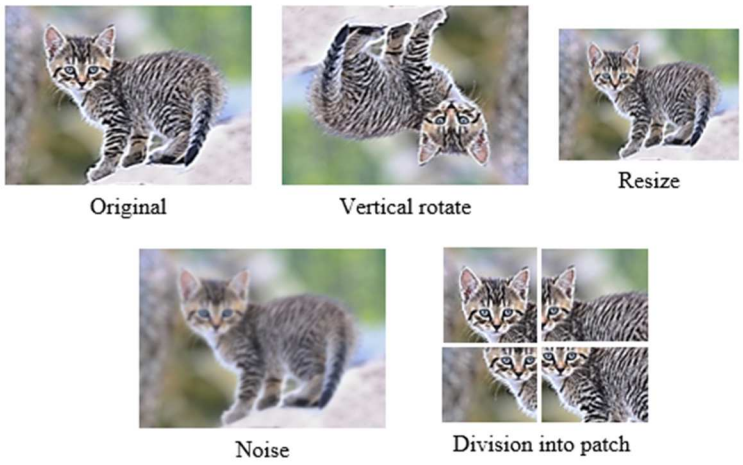

Fig. 13. Data augmentation methods.

- ***Early stopping***

Early stopping is an implicit regularization method [26]. In this method, the dataset is divided into training and validation base. Then, the performance of the model is evaluated during learning on both bases based on an indicator (the error). Generally, learning is stopped when the error value on the validation basis starts to increase (Figure 13) due to the over-learning problem. Finally, the model that minimizes the error rate is stored for the prediction phase.

1.2.6 Activation functions

An activation function is a mathematical function used to convert a signal at the input of a node into an output signal; it will allow the passage of information or not if the stimulation threshold is reached. The term activation function comes from the biological neuron "activation potential", the stimulation threshold that once reached; the function decides whether to activate a neuronal response. There are generally two types of activation functions: linear functions (identity) and nonlinear functions (logistic sigmoid, SoftMax, hyperbolic tangent and rectified linear unit). The nonlinear function helps introduce nonlinear properties into the MLP so that it can solve nonlinear problems and process complex data. Among the functions known to solve nonlinear problems, we have the logistic sigmoid, the hyperbolic tangent, the rectified linear unit and the SoftMax.

Typically, the hidden layers use the same activation function (hyperbolic tangent, sigmoid, or rectified linear unit) and the output layer is based on the SoftMax function or the sigmoid function, depending on the type of classification.

- ***Identity:***

This function is characterized by its linearity (equation 31).

$$F(z) = z \tag{31}$$

- ***The logistic sigmoid:***

The logistic sigmoid is an activation function used by the hidden or output layers (equation 32). It makes it possible to introduce more nonlinearity to the hidden layers and to predict the probabilities of the classes at the output. This function is generally used in binary classification tasks.

$$F(z) = \frac{1}{(1-e^{-z})}, F(z)\ \epsilon\ , z, R \tag{32}$$

- ***The hyperbolic tangent (tanh)***

This function introduces nonlinearity in the hidden layers (equation 33). It is characterized by its good accuracy in recognition compared to the logistic sigmoid function.

$$F(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}, F(z)\epsilon[-1,1[,\ z\ \epsilon\ R \tag{33}$$

- ***The rectified linear unit (ReLu)***

The rectified linear unit is a nonlinear activation function used by the hidden layers. According to equation 34, this function neutralizes negative values to 0. ReLu has proven its efficiency over the sigmoid and tanh functions thanks to its simplicity. In addition, it accelerates the learning time of convolutional neural networks (CNN) [27]. All these criteria have made this function the most used in deep learning [28].

$$F(z) = \max(0, z)\ , F(z) \geq 0, z\epsilon R \tag{34}$$

- ***SoftMax***

The SoftMax function is a nonlinear function used by the output layer. It is used to calculate all the probabilities associated with each class. This function is a generalization of the sigmoid function and is suitable for classification problems with K classes, where K>2.

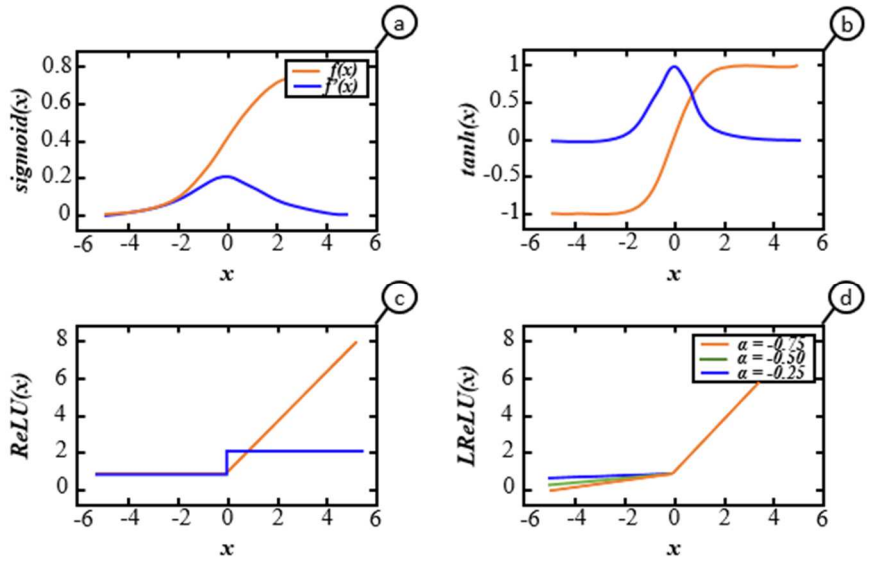

Fig. 14. Activation function graphs examples: (a)- the sigmoid function. (b)-the tanh function. (c)- the rectified Linear Unit (ReLu) function. (d)-the Leaky Relu

1.3 Deep learning architectures:

As we have seen previously, Deep learning is a branch of machine learning which is mainly based on neural networks. An MLP consisting of more than two hidden layers is considered a type of DL network. These layers are used for feature extraction and transformation. Higher-level features are constructed by combining lower-level features involving multi-representation learning. For example, in computer vision, neurons in the first layer represent simple features like terminals. Then, these features become more and

more complex in the deeper layers. This process gives the advantage of deeper networks to solve complex problems. According to Figure 15, the layers in a classical DL network are strongly connected by links (weight w_{ij}). On the other hand, the high number of parameters can quickly lead to over-learning problems and very high time complexity. To reduce these costs and to adapt DL networks to specific application domains, several types of optimized architectures have been proposed in supervised learning (CNN, recurrent neural networks (RNN), and long-term memory). Short-term (LSTM) and unsupervised (stacked auto-encoders (SAE), deep belief networks (DBN), GAN). The following chapter presents the structure of the CNN network in detail.

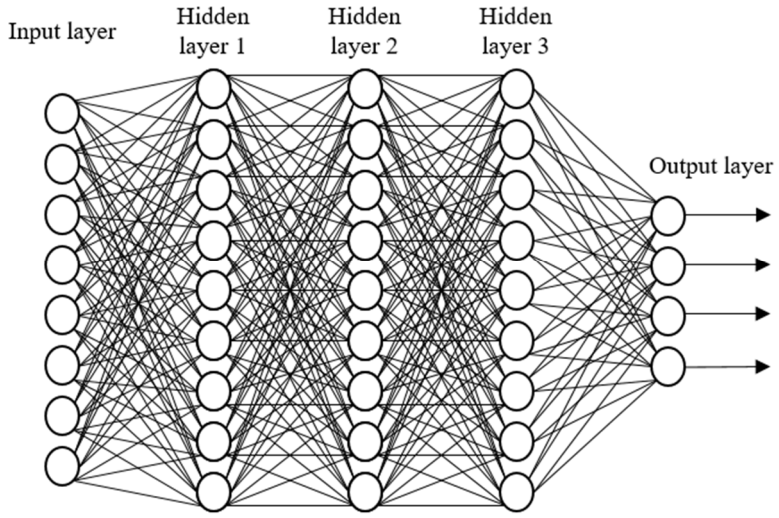

Fig. 15. The architecture of a Fully connected

1.3.1 Supervised learning networks

- ***Recurrent Neural Networks (RNN)***

Recursive neural networks [29] are deep learning networks designed for learning from sequential data. In other words, this deep learning algorithm uses the output of one layer as a new input for another layer. The connection between network layers in the RNN forms directed sequential cycles over time. In general, recurrent neural networks are used to interpret temporal or sequential information.

Unlike MLP, in an RNN the activations in the hidden layers depend on current and past inputs. Any connection makes it possible to consider at the current step one or more pieces of information predicted in a previous step, where the same elements are processed differently depending on the situation. In this way, the results of step t−1 affect the decisions of the next step t. These characteristics have made RNNs a good architecture in text processing tasks because a word can have several meanings depending on its positioning in a sentence.

Figure 16 illustrates the basic structure of an RNN; this network is characterized by cycles between the different connections to process the sequences of variable sizes. The number of layers presents the number of input words x_i. These neurons share the same weights w_{ij} to reduce the total number of parameters. Equation 35 illustrates the process of calculating the outputs o_j, where h_t is the memory, w_{ij} is the weights, and b_0 is the bias.

$$\begin{cases} h_t = (w_{hh}h_{t-1} + w_{xh}x_t \\ o_t = w_{h0}h_t + b_0 \end{cases} \quad (35)$$

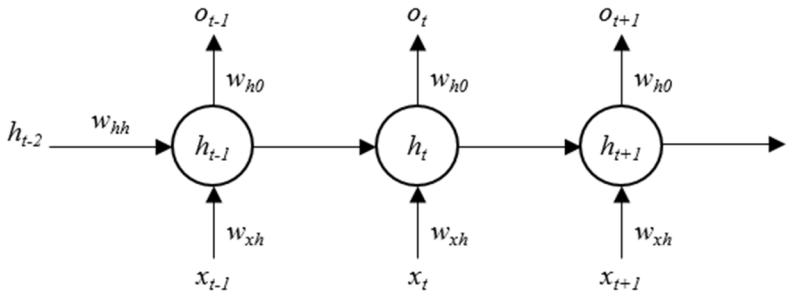

Fig. 16. The structure of a recurrent neural network.

The recursive neural network and its variants have been widely exploited in various word processing applications [30] and in computer vision: action recognition [31], legend generation [32], and image segmentation [33]. Despite the advantages of RNNs in processing temporal sequences, these networks risk the problem of gradient degradation because of the large number of hidden layers. This number is linked to the number of input sequences that varies according to the task processing. Over the iterations, the gradients of the distant sequences tend to converge toward zero, which prevents the model from recognizing the associations between these sequences. To solve these limitations, the long-short-term memory network (LSTM) [34] was developed.

- ***Long-Short Term Memory (LSTM)***

The long-short-term memory network [35] is an optimized version of RNNs. This network makes it possible to memorize the relationships between distant sequences. An LSTM is made up of a set of interconnected LSTM-type units. Figure 17 illustrates the structure of an LSTM unit, where C_t is the cell state. Three gates control this state: entry gate it, forgetting gate f_t and exit gate o_t.

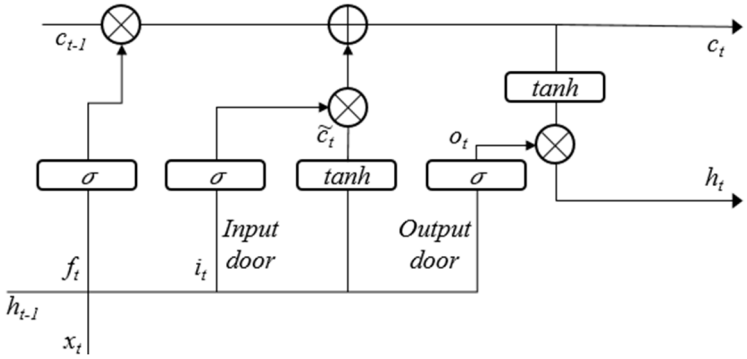

Fig. 17. The structure of a long-short term memory unit.

First, the forgetting gate (equation 36) controls the information to be forgotten from the previous cell state $C_{(t-1)}$, where the value 0 indicates forgetting of the state, while the value 1 keeps its historical. Similarly, according to Equation 37, the front gate decides to update the states of previous cells. Then the memory state C_t and the output gate are updated according to equations 38 and 39 respectively, and finally, the hidden state h_t is calculated based on the gate and the output state (equation 40). σ is the sigmoid activation function. Forget and exit gates have two important parameters that allow you to keep only relevant information.

$$f_t = \sigma(W_f x_t + U_f h_{(t-1)}) \quad (36)$$

$$i_t = \sigma(W_f x_t + U_{if} h_{(t-1)}) \quad (37)$$

$$\begin{cases} \widetilde{C_t} = \tanh(W_c x_t + U_c h_{(t-1)}) \\ C_t = f_t C_{t-1} + i_t \widetilde{C_t} \end{cases} \quad (38)$$

$$o_t = \sigma(W_0 x_t + U_0 h_{(t-1)}) \quad (39)$$

$$h_t = o_t \times \tanh(C_t) \quad (40)$$

LSTM has been used in several fields of application: speech enhancement [36] voice activity detection in real life [37], and automatic subtitling of images [38].

1.3.2 Unsupervised learning networks

- ***stacked auto-encoders (SAE)***

A stacked auto-encoder is a neural network composed of a succession of auto-encoders (figure 18). An auto-encoder consists of two parts: encoder and decoder. The encoder is used to transform the input data x into a compressed representation. Then the decoder uses this representation to reconstruct the input data. The auto-encoder stacking process allows SAE to learn structures that are more complex. Learning is divided into two parts: unsupervised learning and supervised learning based on transferred learning. Unlike an MLP, in each iteration, learning in an SAE is performed only at an auto-encoder unit, where each unit takes as input the outputs of the previous unit. This allows each auto-encoder to minimize the error of the previous layer. Finally, this procedure is followed by transferred learning in the hidden layers.

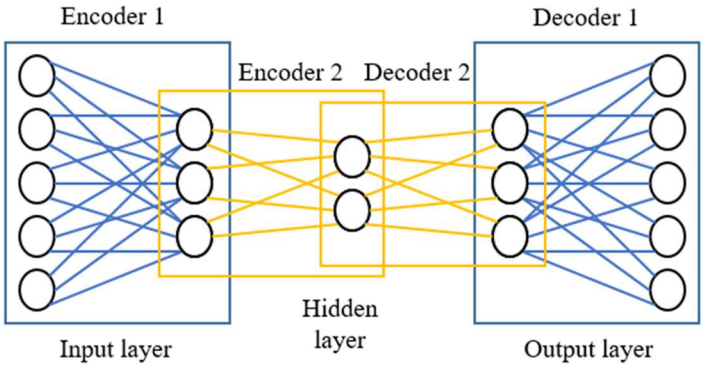

Fig. 18. The structure of a stacked autoencoder.

SAEs have been used in different applications such as data compression [39], dimensionality reduction [40]

- ***Generative adversarial networks (GANs)***

GAN [20] is a generative model based on neural networks. Generative modelling is an unsupervised learning task in machine learning. This task generates new instances from existing instances. GAN is composed of two models: generator G and discriminator D. The generator model makes it possible to generate new instances. Then, these instances are classified into two classes: real or faked. The generator and discriminator models are DL-type neural networks that vary according to the field of application, for example, convolutional neural networks (CNN) are used in image processing. Generative adversarial networks can be used in different fields and applications such as data augmentation [41], semantic segmentation [42], image-to-image translation [43], and the music generation [44].

1.4 Conclusion

In this chapter, we have presented the different principles related to deep learning. Deep learning networks are machine-learning algorithms based on neural networks. The perceptron was the first algorithm proposed for learning an artificial neuron. This algorithm quickly showed its limits in solving nonlinear problems. To solve this limitation, the multilayer perceptron has been proposed. The latter makes it possible to introduce nonlinearity through nonlinear activation functions in the hidden layers. Learning in an MLP is performed using the back probation method based on the SGD optimization technique or its optimized variants. The neurons in a deep MLP are strongly connected, which can lead to an over-fitting problem on the limited data volumes. To address this problem, other variants of DL networks have been proposed in supervised (RNN, LSTM, CNN) and unsupervised (SAE, DBN, GAN) learning, where each network is specialized in specific applications domains and meets the limitations of other networks.

The following chapter presents the principles related to CNN, which is the subject of interest in this thesis. In this context, we will start with the presentation of the general architecture of CNN. Then, we will detail the different common CNN-like architectures in supervised learning.

Chapter II: Convolutional Neural Networks

A convolutional neural network (CNN) is a deep learning algorithm. This network has been widely exploited in computer vision for the classification and detection of objects thanks to its functionalities inspired by the biological processes of the connections between the neurons of the brain. Unlike DL networks, CNNs are characterized by layers of convolution and pooling. These layers introduce partial links to reduce the number of parameters and reinforce the sharing of common characteristics. Despite these advantages, they have several challenges related to over-fitting issues on the limited data volumes and high computational complexity. To solve these problems, known architectures of the CNN type have been proposed. These architectures are based on optimized convolutional blocks, which generate structures that are deeper and less demanding in terms of computing and storage capacity. The purpose of this chapter is to detail the general structure of a CNN and to compare common CNN-like architectures in classification, object detection, and segmentation.

Contents

2.1 Introduction

Convolutional neural networks are deep learning networks inspired by the visual cortex [45]. These networks have been used in recommender systems [46], natural language processing [47], and in computer vision [48]. Their use in computer vision has been very successful thanks to their characteristics inspired by natural visual systems.

In computer vision, the classification process by classical learning methods is based on two main steps: feature extraction and learning. These features are considered handcrafted features because of the manual effort required in the study of discriminating attributes. The methods used to extract these features in an unsupervised way. This separation between the extraction and classification modules can harm the classification task if certain discriminating attributes have been neglected in the extraction phase. Unlike

classical learning methods, CNNs implicitly realize the feature extraction process through the convolution layers, where the first layers represent the single features. Then these features are combined to form others that are more complex in deeper layers. This specificity has made CNNs a good tool for classifying unstructured data like images and text. Despite these advantages, deep CNNs risk the over-fitting problem on limited volumes of data, as they are more suitable for large volumes because of the over-fitting problem [49].

CNNs have proven their efficiency compared to classical deep neural networks in terms of temporal and spatial complexity. These networks are characterized by their parameter sharing strategy. Unlike a DNN, where the adjacent layers are strongly connected, in a CNN, each neuron of the current layer is connected only to a subset of neurons of the previous layer (figure 19). CNNs are considered structural stabilization methods that reduce over-fitting problems by optimizing the number of parameters.

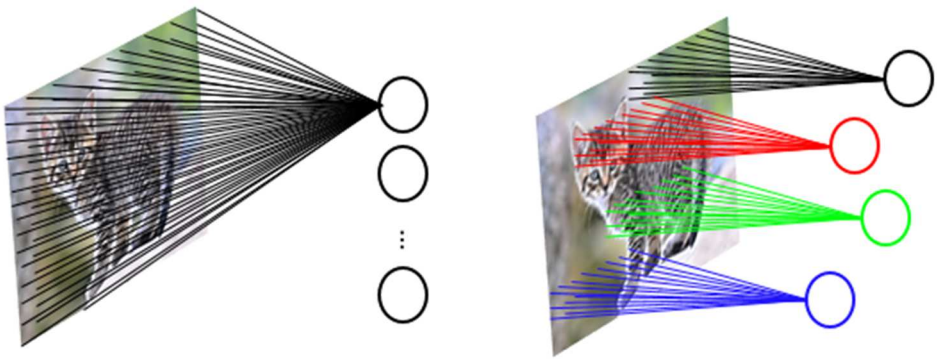

Fig. 19. The difference between the number of connections in a strongly connected layer and a convolution layer.

In the next section, we explain the general architecture of a CNN and compare CNN-type architectures used in classification, object detection, and segmentation.

2.2 History and inspiration:

Convolutional neural networks are inspired by the visual cortex. The visual cortex is the part of the brain responsible for processing information from the eye. In 1962, researchers [45] inserted electrodes into specific parts of a cat's visual cortex to measure activation when the cat observed a few basic shapes. They noticed that single cells respond only to horizontal bars at the bottom of an image. Where complex cells are characterized by spatial invariance, where they can respond to these bars at the different image locations. This invariance is ensured by the combination of the outputs of the simple cells. Based on these assumptions, [50] developed a model (Figure 20) composed of two types of neuronal cells: simple (S) and complex (C) cells. S cells are activated at basic pattern detection, while C cells combine with S cell activations. The idea is to transform previously introduced biological concepts into mathematical ones to model the task of visual pattern recognition. This model has been exploited for pattern recognition in an unsupervised approach.

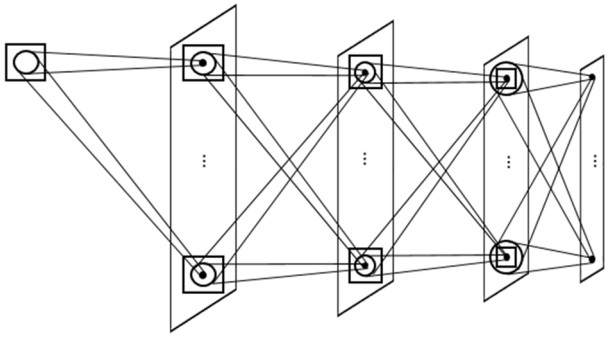

Fig. 20. The structure of the model proposed by Fukushima 1980.

In 1998, [51] introduced the convolutional neural network that is based on the architecture proposed by Fukushima, where they exploited the back-propagation method to accomplish a supervised classification task. Their model was tested on the MNIST learning database specialized in the classification of handwritten characters. In the early 2000s, research on CNN networks stagnated due to insufficient processor power and limited internal memory capacity for the needs of such algorithms. During this period, classical machine learning algorithms have been widely exploited thanks to their reasonable requirements in terms of computational complexity and storage space.

In 2012, the CNN-like AlexNet architecture achieved the best state-of-the-art error rate based on Image-Net learning [9]. This good performance and the ability of GPUs in optimizing time complexity have encouraged the AI community to come up with other optimized variants of the CNN architecture.

2.3 Architecture of a convolutional neural network

A convolutional neural network is composed of four main layers (as shown in Figure 21), a convolutional layer, a pooling layer, a flattening layer, and a fully connected layer. There may be a few repetitions of these layers before the final output. Increasing the number of layers makes the network deeper, which can help to acquire other complex features from the input image [52].

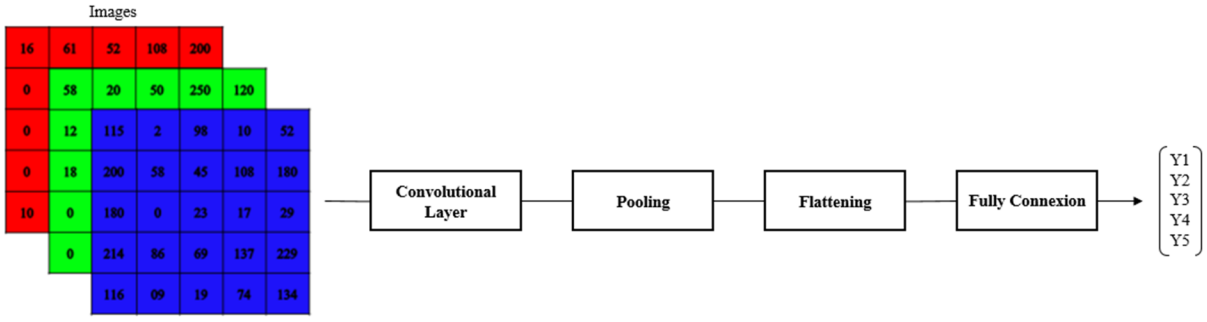

Fig. 21. CNN architecture

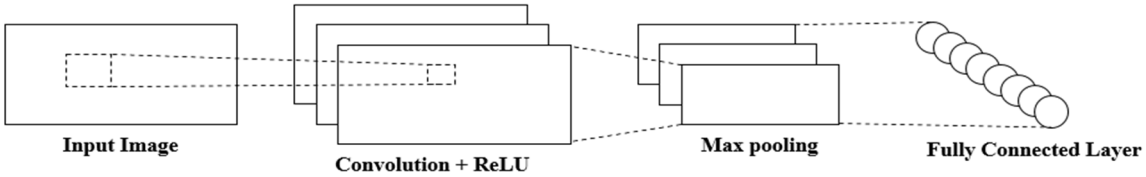

Fig. 22. The general architecture of a convolutional neural network

2.3.1 Convolution layer

The convolution layer is considered the main building block of a CNN. The purpose of this layer is to implicitly extract relevant features from input images during training. This layer performs a convolution operation between two matrices, the first represents a sub-part of the input data (receiver field) and the second represents a filter that contains the learning parameters. A convolution operation generates a third matrix referenced by the feature map. Figure 23 illustrates a convolution operation that is performed by a dot product between the filter and a receiving field. Then the product results are added together to produce a single result presented as a box in the feature map. Finally, the weighting filter is slipped by a step S over the rest of the receiving fields of the input matrix, and this operation is repeated for all the other fields.

In a convolution, the size of the new feature map $N^{(t+1)}$ is calculated according to four hyper-parameters: the size of the old feature map or the input matrix $N^{(t)}$, the size of the filter F, the step value S and the margin value P (equation 40). The margin represents null values that surround the input matrix. This margin prevents the filter from going beyond the scope of this matrix. Applying N^C filters to the input data results in a feature map of size $N^{(t+1)} \times N^{(t+1)} \times N^C$, where N^C is its depth. The concatenation of feature maps forms a convolution layer.

$$N^{(t+1)} = \frac{N^{(t)} - F + 2P}{S} - 1 \tag{40}$$

The process of a convolution illustrates the dimensionality reduction strategy of a CNN, where each bin (neuron) of the current feature map is connected only to a subset of the input neurons (receptive fields). In addition, applying the same filter over the entire feature map allows it to discover previously detected attributes in different areas of the image. At the end of each convolution operation, the activation function ReLu is applied to the resulting convolution layer to improve the generalization.

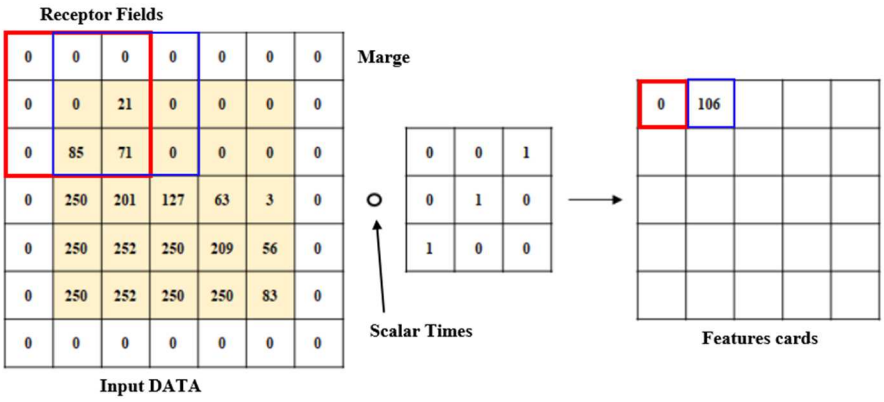

Fig. 23. A convolution operation.

2.3.2 Pooling layer

The pooling layer's role is to reduce the dimensionality of the resulting convolution layers. This reduction aims to improve accuracy by selecting the dominant attributes. In addition, the optimization of parameters number reduces the size of the model and optimizes the time complexity. The resulting matrix size of the pooling operation is calculated by Equation 2.1 with P = 0. There are two types of pooling operations: Max-pooling and Avg-pooling. The Max-pooling operation returns the maximum value of the receptive field while the Avg-pooling operation returns the average of the values. Max-pooling is the most used operation in most CNN-type architectures.

2.3.3 Flattening

It is the operation of converting all the resulting two-dimensional arrays into a single continuous long linear vector, as shown in Figure 24

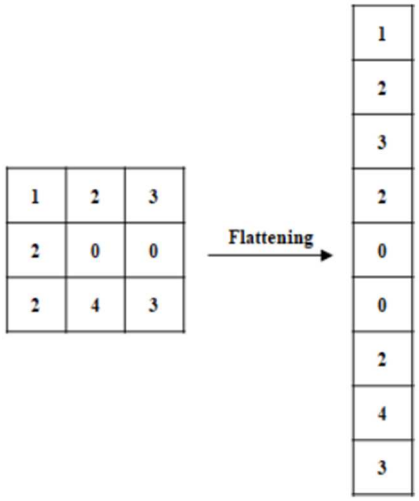

Fig. 24. Flattening operation

The Flattening output will then serve as input to an artificial neural network, most often the fully connected layer, which will do the high-level reasoning.

2.3.4 Fully connected layer

In a CNN, the fully connected (FC) layers have the same structure as an MLP. The purpose of these layers is to learn the nonlinear combinations between the features extracted by the convolution layers. The features, after going through several convolutions and pooling, are stacked together in the FC layer. These characteristics present the input layer to the set of fully connected layers. In supervised classification, the last layer is used for prediction based on the Softmax activation function.

2.4 Training in CNN

We have previously seen the different construction elements of convolutional neural networks, as well as the gradient descent algorithm that allows their training. In this section, we will present the types of learning in a convolutional neural network. We will then review the different most popular network architectures, which have allowed significant advances in terms of image recognition. These architectures owe their popularity to the ImageNet database [9] linked to the ILSVRC competition ("ImageNet Large Scale Visual Recognition Challenge").

2.4.1 Learning types:

Learning in a convolutional neural network can be done in two ways: learning from randomly initialized parameters (training from scratch) or learning by transfer.

- ***Learning from random initializations***

In the previous chapter, we discussed the learning process in an MLP through the back-propagation method. Convolutional neural networks are based on the same method. The main purpose of CNNs is to reduce the error of the cost function by adjusting the filters. These filters represent the learning parameters w. As we

mentioned earlier, CNNs are characterized by parameter sharing. This specificity makes it possible to reduce the number of parameters in a convolution layer and to optimize the temporal and spatial complexity. This sharing and the representation of the parameters in the filters require adapting the back-propagation function on the convolution and pooling layers. Learning in a CNN starts with forwarding propagation to calculate the value of the cost function based on the inputs. Then, the randomly initialized filters are adjusted by a back-propagation process. This process is repeated for several iterations until the stopping criterion is reached. The criterion can depend on a fixed number of iterations, on a premature stop, or on convergence.

- ***Transfer learning and fine-tuning***

Transfer learning is a machine learning method that allows reusing a previously developed model for learning a task A in another task B. These tasks can be similar or different depending on the nature of the process of the transferred learning.

The research in transferred learning started from the years 1995 [54] when they were inspired by the behavior of humans in their method of learning based on the knowledge acquired previously. These methods are categorized into inductive, transductive, and unsupervised transfer learning [53]. In the inductive transfer method, the source and target tasks are different and belong to the same domain. While in translational learning, tasks are similar and different in probability distribution in attribute space. In unsupervised learning, learning data is not categorized into source and target domains.

Transfer learning methods belong to the category of inductive learning. They are used in different fields such as natural language processing [55] and computer vision [56].

In classical machine learning, learning algorithms are characterized by their dependence on the input attribute distribution, where the source and target data must have the same data representation. Changing the distribution of attributes requires starting learning from the beginning. Unlike these algorithms, in deep learning, it is possible to transfer knowledge from previously trained models, where the distribution of parameters (weights) of DL networks provides this possibility.

In computer vision, transfer learning from CNNs has been very successful due to their hierarchical nature. The first layers represent general features like Gabor filters. They can detect basic shapes (curves and borders). On the other hand, the deep layers make it possible to model more complex characteristics related to the field of application of the learning base. The common characteristics of the first layers offer the possibility to perform transfer learning between different tasks.

We have previously seen the various problems of over-fitting related to the lack of data. Transfer learning is one of the methods proposed in the state of the art to solve this limitation. This technique is generally used when the data volume of the target task is limited. It makes it possible to reuse the first layers of models generated from large volumes of data. Transfer learning from models trained on the Image-Net [9] learning base has been widely used in different fields thanks to its large volume of data (15 million) and its large number of categories (22,000).

Transfer learning in CNN can be used in three ways: (a) exploit the source model as a feature extraction module, (b) transfer a subset of layers and readjust the rest, and (c) transfer and readjust all layers.

2.4.2 Feature extraction:

This strategy is seen as a hybridization between machine learning algorithms and DL networks, where CNN-like networks are exploited for feature extraction and ML algorithms for classification. Figure 25 illustrates this process; typically, all layers from the source model are transferred to the target model. Then the last layer is removed. The data from the target-learning base is then passed to the model. This makes it possible to generate a learning base structured in the form of attributes and instances. Finally, this database is classified by an ML-type algorithm.

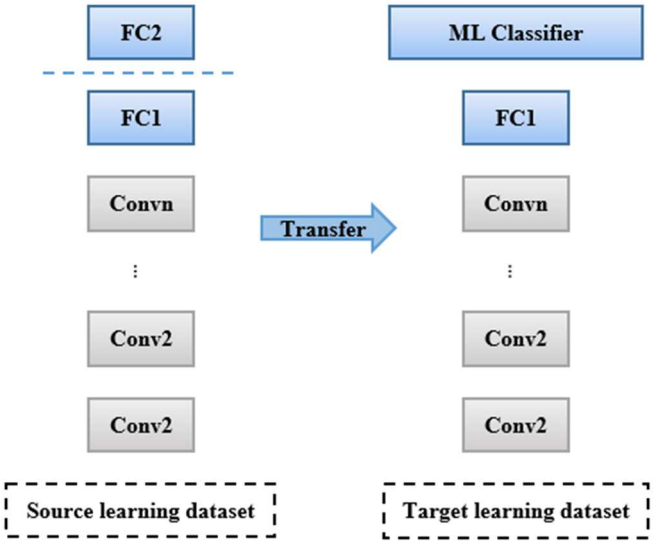

Fig. 25. The use of convolutional neural networks for feature extraction.

2.4.3 Fine Tuning

As we discussed before, the first layers are used to represent general characteristics, while the deeper layers are related to the source application domain. To adapt these layers to the target application domain, a subset of deep layers (convolutions and fully connected) is readjusted by a learning process (figure 26).

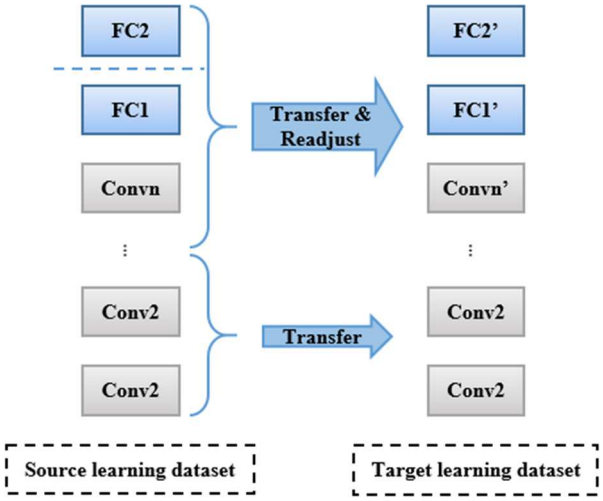

Fig. 26. The fine-tuning process in a convolutional neural network.

The evolution of storage and computing capacity (GPU) encouraged the computer vision community to propose other deeper CNN-like architectures. These architectures optimize conventional convolution layers. The main purpose of this variation is to reduce the number of parameters and add additional layers that improve nonlinearity. This non-linearity is ensured by the activation functions that develop the capacity of the network in solving complex problems

In 2012, the AlexNet network [27] was proposed by Alex Krizhevsky et al. Figure 28 shows the diagram of the AlexNet network. The latter is composed of five convolution layers all followed by maximum-pooling type sub-sampling operators and ReLU type activation functions. The convolutional part of the network is followed by two fully connected layers (multilayer perceptron). To limit overfitting, they use the dropout technique in fully connected layers with a 50% probability of cancellation (see part 2.2.3). This network enabled them to win the ILSVRC 2012 competition (seen in part 2.3.1) with an accuracy of 84.6% (i.e., an error of 15.4%) while the closest competitors obtained an accuracy of around 73.8% (see Table 2.8). This result has encouraged the computer vision community to come up with other CNN-like optimized versions by structure analysis. In 2013, the ZFNet network was developed [18]. This network mimics the structure of AlexNet with a slight reduction in the size of the first filter. The purpose of this modification is to keep more information in the first convolution layer. Increasing the depth of a neural network improves its non-linearity and therefore its ability to recognize complex objects. On the other hand, this increase increases the number of parameters, and this increases the risk of over-fitting and storage requirements. To avoid these problems, other dimensionality reduction strategies have been developed in the convolution layers. In 2014, The VGGNet network [57] offers configurations whose depth varies from 11 to 19 layers. This network suggests reducing the size of the filters to F=3 to avoid the exponential increase of the parameters by adding additional layers. In 2015, the Inception network [58] was developed. This network proposes a parameter reduction strategy through Inception modules. The effectiveness of Inception blocks in reducing dimensionality has encouraged computer vision researchers to propose other optimized versions of Inception modules [59] [61]. Inception-ResNet [60] embeds residual links in inception blocks and Xception [61] uses extreme inception blocks, which have architecture like depth-separable convolutions. These two versions have proven their efficiency compared to the initial Inception architecture [59]. Deep CNNs are characterized by their efficiency in classifying complex objects. Despite this specificity, very deep networks risk gradient degradation. To solve this problem, the ResNet network [62] proposes structures based on residual blocks in configurations composed of 18 to 152 layers. In 2017, the DenseNet network [63] was developed to provide deeper configurations compared to ResNet by reducing the number of parameters. The experimental study on the ImageNet learning base showed that a DenseNet network composed of 201 layers and 20 million parameters has the same performance as a ResNet type network composed of more than 40 million parameters.

Object detection is a technique in computer vision that makes it possible to classify and detect several objects in an image. This task is characterized by its high complexity compared to classification methods because it requires an additional localization step. Localization proposes candidate regions of interest, and then these regions are classified. Several object detection methods have been proposed in the state of the art. The R-CNN network [64] combines the selective search method for region detection and CNN networks for classification. Despite its efficiency, it is not suitable for real-time applications because of its high time complexity. To reduce this complexity, other structures have been proposed: Fast R-CNN [65], Faster R-CNN [66], and YOLO [67].

Semantic segmentation is a technique that classifies each pixel in the input image. This method is characterized by its high complexity with respect to object classification and detection. To optimize its complexity, several CNN-based architectures have been proposed: FCN [68], DeepLab [69], SegNet [70], U-Net [21], and Mask R-CNN [71].

In what follows, we will detail the structure of known CNN-type architectures in classification, object detection, and segmentation.

2.5 Classification

2.5.1 LeNet

LeNet is the first CNN-type supervised classification architecture proposed by LeCun in 1990 [72], this architecture is the best known to the community and is sometimes considered (wrongly) as the first neural network. Figure 2.8 illustrates the structure of the LeNet network. This network is composed of 7 layers in total: 3 convolution layers (Cx), 2 Avg-pooling layers (Sx), and 2 fully connected layers (Fx).

The Cx and Sx layers are composed of a few feature maps of a defined size (number@width × height). This figure shows that the size of the inner layers is reduced compared to the first layers, while they are deeper compared to the input layers. In this architecture, the size of the filters in the convolution layers has been fixed at 5×5 and the inputs are images of size 32 × 32.

Back-propagation-based learning on the MNIST learning basis showed the efficiency of the LeNet deep learning algorithm compared to the classical machine learning algorithms SVM and KNN. The MNIST learning base was designed for the classification of handwritten digits and contains 60,000 instances for learning and 10,000 instances for testing. These instances are normalized and centered in black and white images.

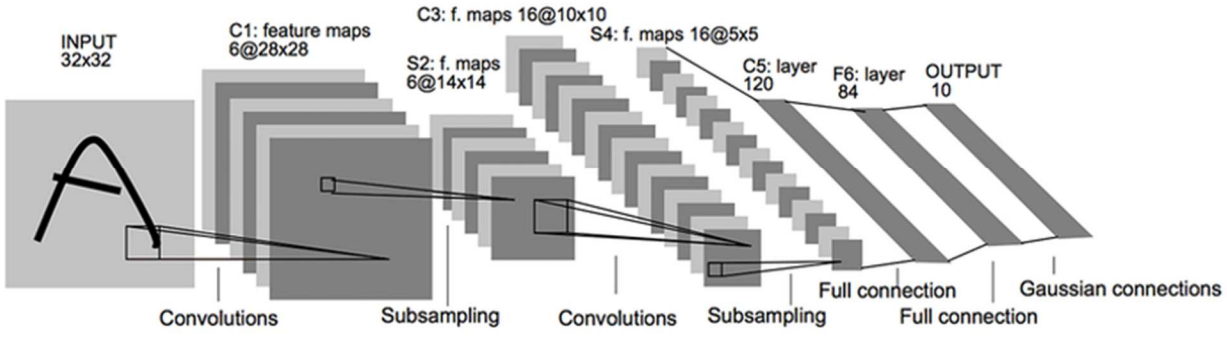

Fig. 27. LeNet network structure

2.5.2 AlexNet

In 2012, the AlexNet network [9] was proposed in the ImageNet Large-Scale Visual Recognition Competition (ILSVRC). This competition uses a subset of the ImageNet training database consisting of 1000 categories, 1.2 million training instances, 50,000 validation instances, and 150,000 test instances. In this competition, the AlexNet network enabled them to win the ILSVRC 2012 [Rus+15] competition (seen in part 2.3.1) with an accuracy of 84.6% (i.e., an error of 15.4%) when the closest competitors obtained an accuracy of around 73.8%.

The AlexNet network has a similar and deeper architecture compared to LeNet (Figure 28). This network is composed of five convolution layers and three fully connected layers. According to Figure 28, the first convolution layer uses 96 filters of size 11 × 11, while the other convolution layers are based on filters of size 5 × 5 and 3 × 3. The first, the second, and the fifth layer of convolution are followed by max-pooling layers and the first two layers are followed by a normalization operation (local response normalization (LRN)). The LRN method improves the generalization and the nonlinearity of the network. This operation is applied to the results of the ReLu activation function. As opposed to LeNet, AlexNet offers the exploitation of the ReLu activation function, because it allows speeding up the learning compared to the tanh function. This leads to a remarkable reduction in time complexity in the case of large models that are trained on data of considerable size.

Despite the considerable size of the training base used, the AlexNet network risks the problem of over-training because of the high number of parameters (60 million). To avoid this problem, the techniques of data augmentation and regularization by dropout (Dropout) have been exploited. The data augmentation method used extracts random patches of size 224 × 224. Then these patches are augmented by horizontal reflections. In the test phase, the decision presents the average of the predictions of all the patches. The Dropout method was used in the first two fully connected layers with a dropout rate of r = 0.5. In learning, the descent with inertia method was used with a data set of size 128 and a learning rate initialized at 0.01. This rate is manually reduced 6 times during learning.

The learning process took five to six days to finish running in 90 epochs on two NVIDIA GTX 580 3GB GPUs. The use of parallel GPUs makes it possible to speed up the execution time and to offer the possibility of loading the entire model in memory, due to the limited memory of the GPUs (3 GB).

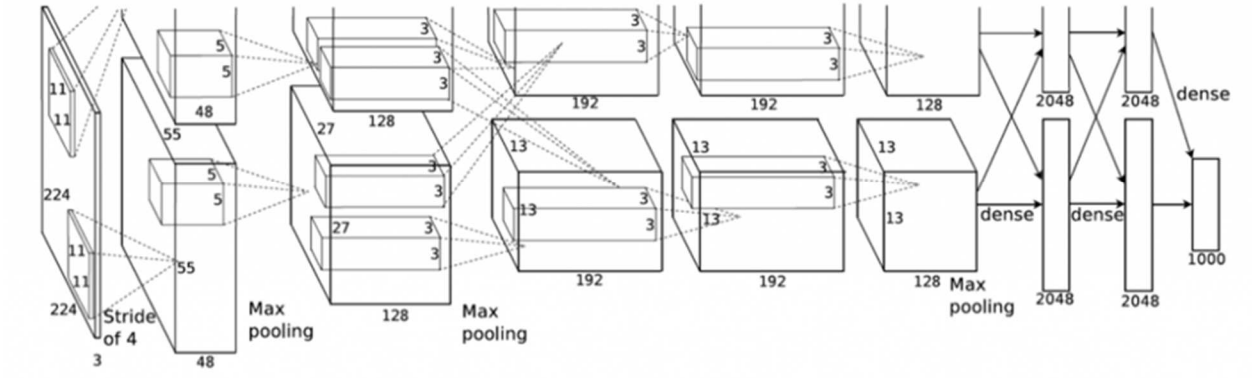

Fig. 28. AlexNet network structure [9].

The good functioning of this network (and therefore its success) can be explained by three factors:

- The use of efficient operators such as ReLU functions and the dropout operator.
- The technical skills of Alex Krizhevsky made it possible to implement the network on a graphics card, making it possible to speed up the training time of the network and therefore to train it longer and more efficiently.
- A large amount of data is used to train the network. These three factors together have enabled an important advance in image recognition and have contributed to the current popularity of convolutional neural networks.

2.5.3 ZFNet

Despite the performance of AlexNet in ILSVRC, [9] did not justify the choice of hyper-parameters (size of filters, number of layers) and how to adjust them to improve CNN performance. In addition, the behavior of the network and its internal functioning remain ambiguous from a scientific point of view.

To understand the behavior of CNNs and improve their performance, [73] proposed a new visualization method that allows deciphering the content of the intermediate layers. The purpose of this visualization is to study the behavior of AlexNet and to propose an optimized version called ZFNet. The visualization technique is based on the multilayer deconvolutional network (Deconvnet) [74]. This network makes it possible to project the maps of the internal characteristics at the entrances to visualize their content.

The Deconvnet network is based on three operations: unpooling, rectification, and filtering. The unpooling method is the reverse of the pooling operation of a CNN. It allows restoring the content of the characteristic cards before the pooling operation. The rectification is based on the ReLu activation function. It eliminates the negative values of the characteristic cards. Filtering is the inverse operation of convolution. It is based on the transposed version of the filters used in the convolution operation. This operation applies the transposed filters on the feature maps to obtain the previous convolution layer. These three operations are successively repeated on the internal feature maps until reaching the input pixel space. To visualize the content of the feature maps, the Deconvnet network is linked to each layer of the CNN network. Figure 29 illustrates the result obtained by Deconvnet on the nine best activations of layers 2 and 5. This result shows the hierarchical nature of the CNN network, where the 2nd layer represents simple features like corners and edges. Then these features become more complex in the deeper layers until the last convolutional layer (layer 5), where the objects are fully visible.

This visualization technique made it possible to detect some problems. The first filters present a mixture of information of varying frequency. In addition, the 2nd layer illustrates the presence of a noise resulting from the large value of the step S = 4 in the convolution operation. To solve these problems, the size of the first filter was reduced from 11 × 11 to 7 × 7 and the value of the step S from 4 to 2. Figure 30 illustrates the new architecture of the proposed ZFNET network. The proposed changes improved the performance of the AlexNet network by 1.7%. This result illustrates the use of visualization techniques in hyper-parameter tuning.

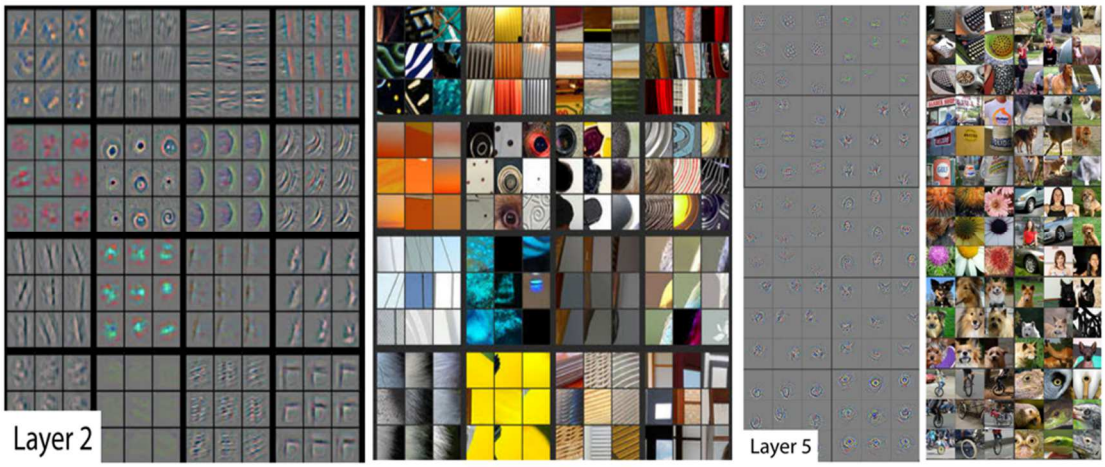

Fig. 29. Result of applying the Deconvnet network on layers 2 and 5 [74].

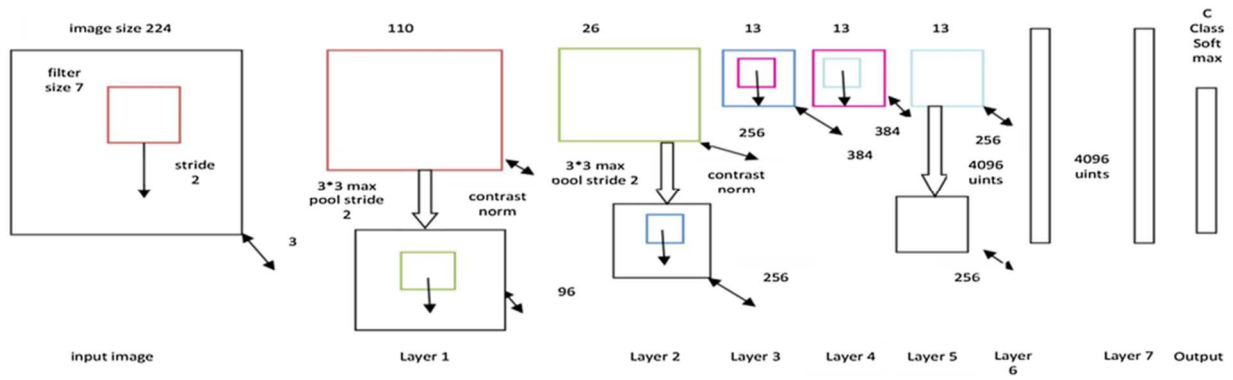

Fig. 30. ZFNet network structure

2.5.4 VGGNet:

VGGNet [75] is a convolutional neural network that is based on the same concept as the AlexNet network [9]. The objective of this version is to offer deep configurations (16 to 19 layers) based on the technique of structural stabilization. This technique makes it possible to control the number of parameters in deep networks to optimize and reduce the risk of over-learning in the AlexNet network. To optimize the number of parameters, the VGGNet network proposes to reduce the size of the filters from 7 × 7 and 5 × 5 to 3 × 3. This change makes it possible to add more intermediate layers without risking an exponential increase in the number of parameters.

The comparative study between the number of parameters in three stacked convolution layers associated with filters of size 3 × 3 and a single convolution layer associated with a filter of size 7 × 7 showed that small filters reduce the number of parameters. If each convolution layer has a depth C, the number of parameters in 3 stacked convolution layers (3 × 3) is 3 (32C2). While the number of parameters in a single layer associated with filters of size 7 × 7 is 72C2. Finally, we summarize the stacking of convolution layers associated with small receptive fields allows to reduce the number of parameters and d Improves network non-linearity through additional activation functions (ReLu).

Figure 31 illustrates the VGGNet network configurations. These configurations vary in the number of convolution layers and the size of the filters. The convF-P annotation expresses a convolution layer associated with filters of size F × F and depth P.

Configuration A is composed of 11 layers (8 convolution layers and 3 fully connected layers). The second configuration A-LRN integrates into A normalization operation (LRN) after the first convolution layer. Configuration B adds two convolution layers to A. Configuration C integrates into B three additional convolution layers associated with filters of size 1 × 1. These filters make it possible to improve the non-linearity of the network through the ReLu functions because they represent a projection in a space of the same dimensionality, where the input layers have the same dimensionality as the output layers.

The experimental study on the ImageNet learning base showed the positive effect of depth on the performance, where the deepest networks perform the best. The comparative study between the A and A-LRN configurations indicates that the normalization operation (LRN) does not improve the performance of model A. In addition, unlike convolution layers associated with 3 × 3 size filters, convolution layers associated with 1 × 1 size filters have a negative effect on network performance.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 31. The VGGNET network configurations [75]

2.5.5 Inception:

The easiest way to improve the performance of a network is to increase its size in terms of width (number of parameters in each layer) and depth (number of layers). Despite the efficiency of deep networks, they have several drawbacks related to the risk of over-fitting on limited volumes of data. In addition, deep networks are more demanding in terms of storage and computing capacity. To solve these problems and adapt the use of deep networks to real-time applications, research has focused on partially connected architectures more than fully connected architectures.

The Inception network [58] is a convolutional neural network that offers the exploitation of Inception modules. These modules feature optimized variants of the classic convolution layers. Inception modules introduce partial connections inside a convolution layer to reduce its dimensionality.

Figure 32 shows the structure of an Inception module. This module uses filters of variable size (1 × 1, 3 × 3, and 5 × 5) which are applied on the same convolution layer. Then, the resulting feature maps are stacked to form the next convolution layer. Varying the size of the filters helps to avoid patch alignment issues. Despite these partial connections, the encapsulation of feature maps rapidly increases the depth of the convolution layers. To solve this problem, 1 × 1 size filters were introduced before 3 × 3 and 5 × 5 size filters. These filters reduce the depth of the convolution layer before the other filters are applied and improve the nonlinearity by the ReLu activation functions. In summary, Inception modules increase the depth of the network by controlling the computational complexity in parallel through dimensionality reduction techniques. Moreover, the variation in the size of the filters makes it possible to process the input information on different scales.

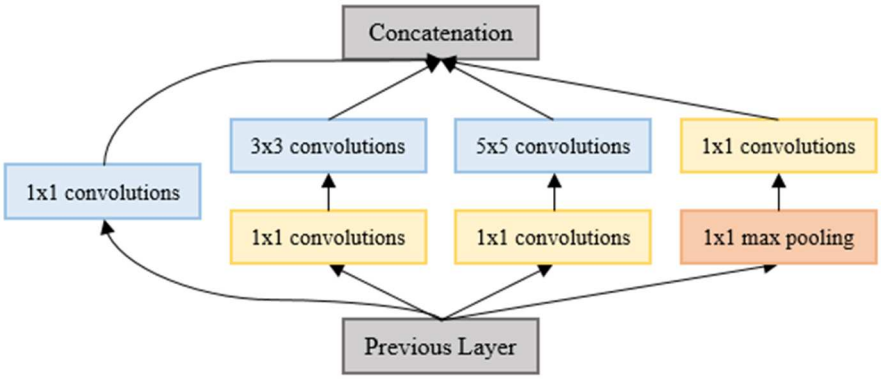

Fig. 32. Structure of an Inception model

Figure 33 illustrates the structure of the Inception network, which is composed of 22 layers in total. This network is formed by a concatenation of classical convolution layers, Inception modules, and Avg-pooling layers. Unlike the architectures proposed previously, Inception proposes the integration of auxiliary classifiers, which are connected to the intermediate layers. This technique introduces the discriminative strength of shallower networks through the intermediate layers, where the error rate is calculated based on a weighted average of the results of the three prediction layers. In summary, the Inception network has shown its effectiveness in dimensionality reduction through Inception modules. These modules offer a deeper and more efficient network by reducing the number of parameters of AlexNet by 12 times [9].

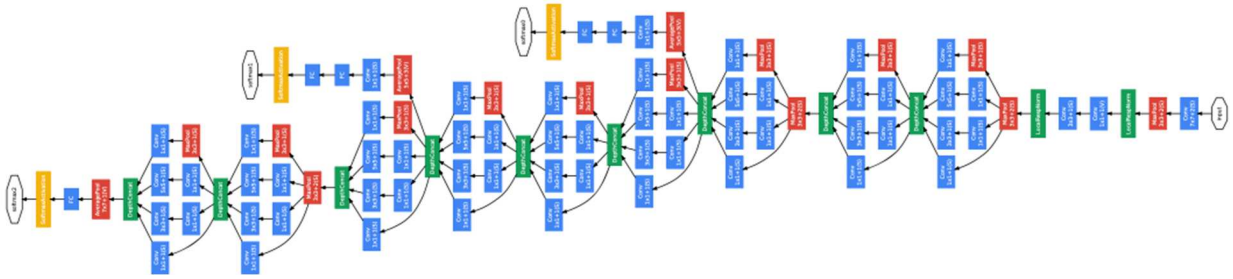

Fig. 33. Structure of Inception model [58].

2.5.6 InceptionV2 and InceptionV3

InceptionV2 and InceptionV3 [59] are optimized versions of the Inception network [58]. These architectures offer variants of Inception blocks to reduce the number of multiplications in convolution and therefore optimize the computational complexity. These variants are based on two factorization techniques: factorization of convolutions associated with large filters and spatial factorization in asymmetric convolutions. The first technique proposes to reduce the size of filters from 5 × 5 to 3 × 3 because filters of size 5 × 5 are 2.78 times more expensive compared to filters of size 3 × 3. Despite the effectiveness of this reduction in the number of parameters, it can generate a loss of information. To avoid this problem, the factorization technique proposes to replace a convolution layer associated with a filter of size 5 × 5 with two convolution layers associated with filters of size 3 × 3. This leads to a reduction in the number of parameters by 25%. Figures 34 and 35 show the block structure of Inception in InceptionV2 and InceptionV3.

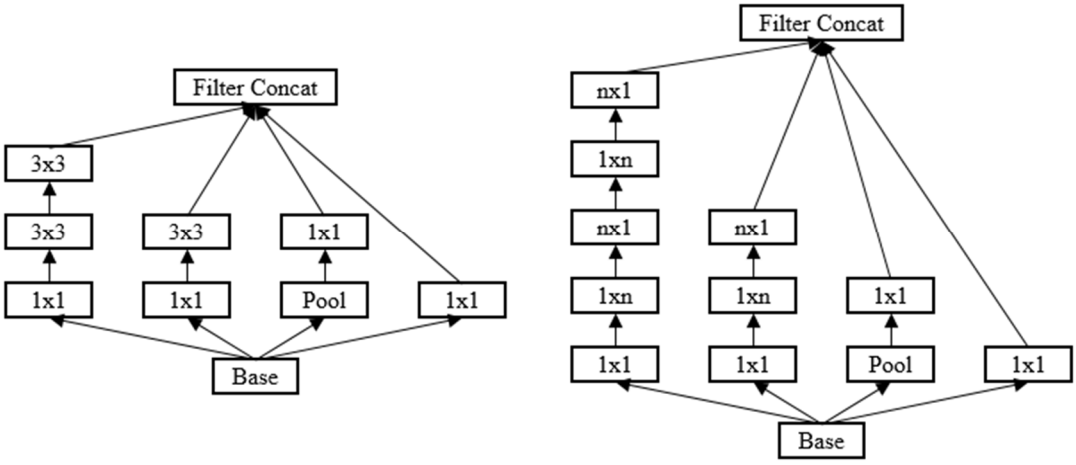

Fig. 34. Structure of Inception blocs in InceptionV2 and InceptionV3

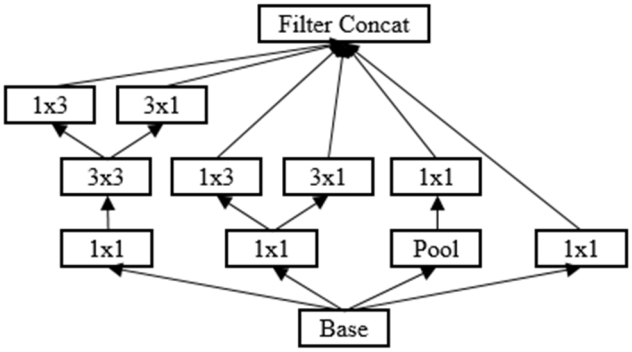

Fig. 35. Structure of Inception blocs in InceptionV2 and InceptionV3

The second technique proposes to replace the classical convolutions (n × n) with asymmetric convolutions (n × 1 and 1 × n). This method reduces the computational complexity by 33% for n = 3 (Figure 2.15 (right block)). In addition to the techniques mentioned above and exploited in InceptionV2, inceptionV3 proposes the use of (a) the method of regularization by smoothing (label smoothing), (b) the auxiliary classifiers where the entire connected layer is normalized by the method of normalization per batch, and (c) factoring filters of size 7 × 7 to asymmetric filters (1 × 7 and 7 × 1). Figure 2.17 illustrates the structure of the InceptionV3 network that is composed of 42 layers in total.

The comparative study between the results of the Inception, InceptionV3, and VGGNet networks demonstrated the effectiveness of InceptionV3 on the ImageNet learning base.

2.5.7 ResNet

ResNet [62] is a convolutional neural network based on residual blocks. The main purpose of this architecture is to solve the gradient degradation problem of vanishing gradient. This problem appears in very deep networks, where the precision starts to be saturated and then degrades rapidly due to the decrease in the values of the gradients. To solve this problem, residual blocks were introduced.

The residual blocks (Figure 36) represent residual connections between the output of the previous layer and the output of the current layer. These connections are formulated by equation 2.2. To perform the addition, a linear projection Ws is performed to obtain equivalent dimensions (x and F(x)).

To demonstrate the effectiveness of residual blocks in solving the gradient degradation problem, [62] performed a comparison between the performance of CNNs with and without residual blocks (Figure 37). They used two networks inspired by VGGNet and composed of 18 and 34 layers in total. The comparative study shows that in the classic version of the CNN networks, the shallower networks (18 layers) are the most efficient. While in ResNet networks the deeper networks are the most efficient. These results proved the effectiveness of residual blocks in solving the gradient degradation problem. To analyse the effect of depth on ResNet networks, a comparative study was carried out between ResNet-50, ResNet-101, and ResNet-152 networks. These networks contain residual connections between three convolution layers (Figure 36) instead of two layers to speed up the training time. The results obtained illustrate the advantages of depth on the performance and efficiency of residual blocks in solving gradient degradation problems in very deep networks.

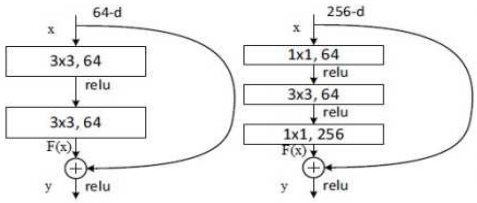

Fig. 36. Structure of residual blocs

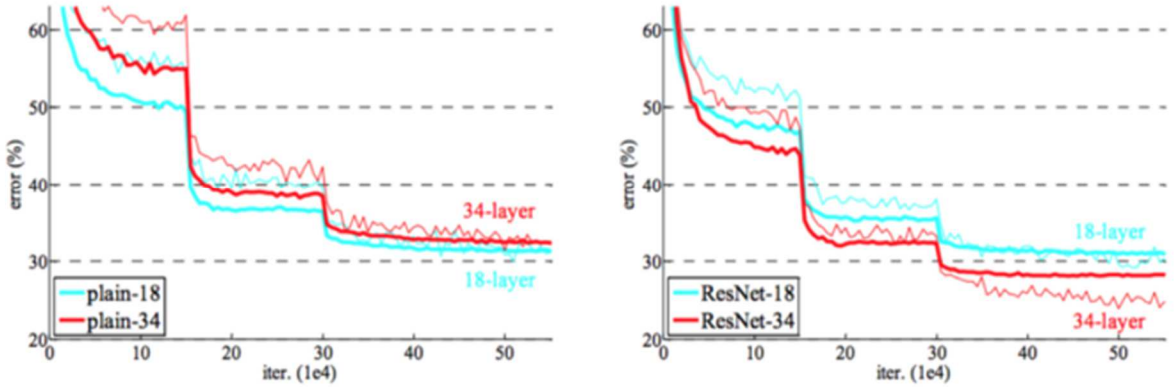

Fig. 37. Comparison between the performance of CNNs with and without residual blocks

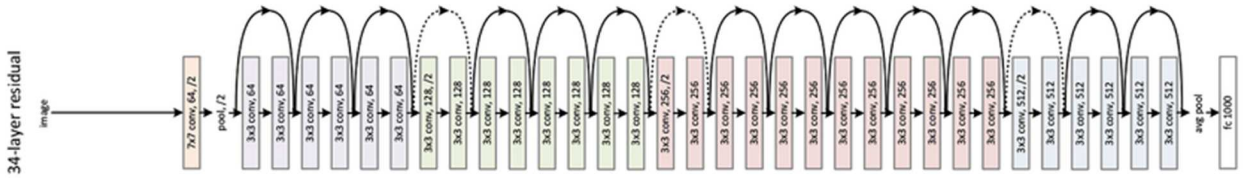

Fig. 38. Structure of ResNet Network [62].

To optimize the structure of ResNet networks, several variants have recently been proposed: ResNet−CutMix [76], ResNet+SWA [77], AA−ResNet [78], and ResNeXt [79].

2.5.8 Inception-v4 and Inception-ResNet

Szegedy and al. [60] proposed two variants of the inceptionV3 network [59]: inceptionV4 and Inception-ResNet. The InceptionV4 network is a deeper, uniform and simplified version of the InceptionV3 network. Figures 39, 40, and 41 illustrate the structures of the Inception modules used (Inception-A,

Inception-B, and Inception-C), respectively, and Figure 39 presents the architecture of the InceptionV4 network

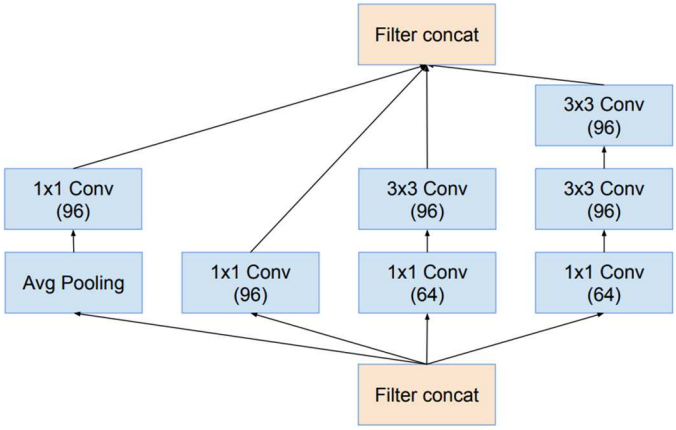

Fig. 39. The structure of the Inception modules (B) of the InceptionV4 network

The Inception-ResNet network proposes hybridization between the modules introduced into the InceptionV3 network [59] and ResNet [62]. The purpose of this combination is to speed up the training time of the Inception network and avoid the gradient degradation problem of very deep networks. In inception layers, filters of size 1 × 1 are used before the residual link to obtain the same dimensionality of the input data and perform the addition.

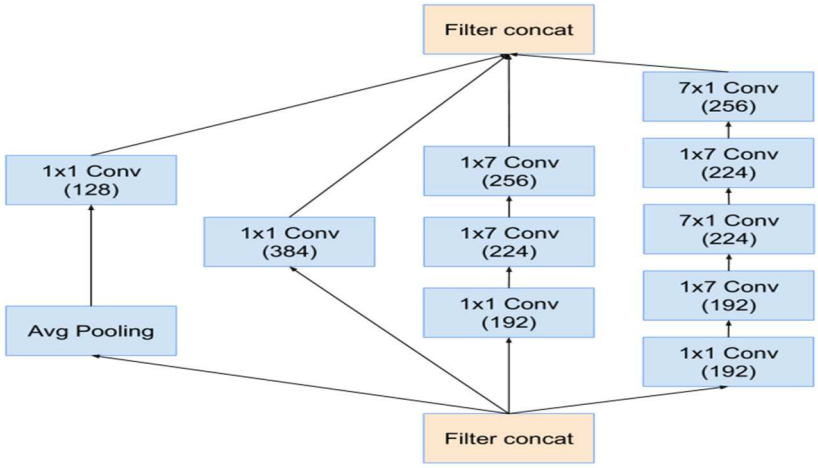

Fig. 40. The structure of the Inception modules (B) of the Inception V4 network

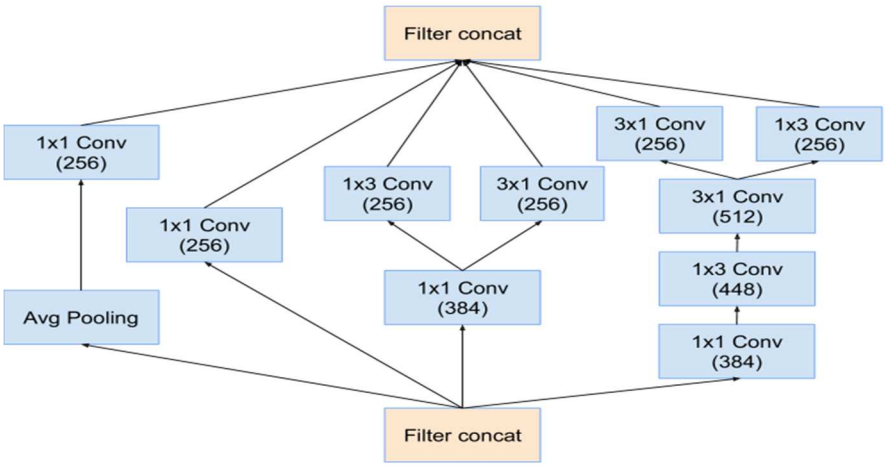

Fig. 41. The structure of the Inception (C) modules of the Inception V4 network

The experimental study on the ImageNet learning base showed that the residual blocks are not necessary in some cases. For example, the InceptionV4 network performs better compared to Inception-ResNetV1. Where the Inception-ResNetV2 network was the most efficient compared to all the other networks (InceptionV3, InceptionV4, and Inception-ResNetV1).

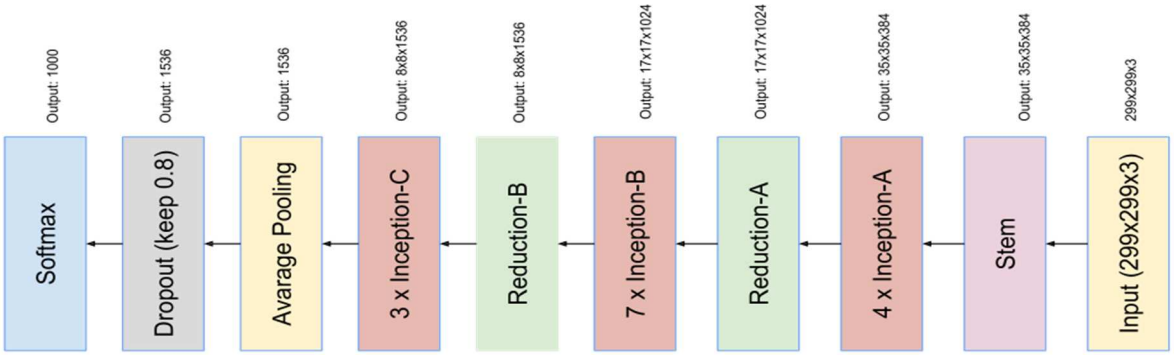

Fig. 42. The structure of the Inception V4 network

2.5.9 DenseNet

DenseNet [80] is a convolutional neural network based on dense connections between convolution layers. According to Figure 43, this network is composed of a set of dense blocks, which are linked by transition layers. Each block contains a set of convolution layers, where each layer is connected to all subsequent layers belonging to the same block. This introduces L (L+1) 2 connections in a block containing L layers in total. Unlike classical CNNs, each layer receives L inputs which represent the feature maps of the previous L − 1 layer. These connections establish direct links between the gradient of the cost function and the original inputs. In addition, they improve the regularization and therefore reduce the problem of over-fitting and gradient degradation.

The number of layers in a dense block depends on the growth rate k. This rate specifies the number of input feature maps and regulates the amount of information added to each layer in the network. To minimize the total number of parameters, DenseNet uses layers of down-sampling and rate compression. These layers are presented by transition layers and composed of a batch normalization layer, a convolution associated

with a 1 × 1 size filter, and an Avg-pooling layer. The transition rate θ ∈ {0, 1} allows reducing the number of resulting feature maps of a dense block. Figure 43 illustrates the structure of the different DenseNet network configurations. These configurations vary in network depth (from 121 to 201). Each block is composed of a set of convolution layers, where each is associated with filters of sizes 1 × 1 and 3 × 3.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Fig. 43. Configurations of DenseNet network [80].

The comparative study between the results of the DenseNet and ResNet networks showed their equivalence. This indicates the efficiency of DenseNet due to its reduced number of parameters compared to ResNet. In summary, the DenseNet network has several advantages related to its efficiency in feature extraction, reduction in the number of parameters, and reduction in gradient degradation problems.

2.5.10 Discussion and comparison

The purpose of this section is to compare the results of the deep networks detailed previously. Most of the architectures were submitted to the ImageNet competition to validate the results obtained. AlexNet is the first DL-type network that achieved an interesting error rate (15.3%) compared to the results of classical ML methods. In 2013, the ZFNet variant [73] of the AlexNet network achieved an error of 14.8%. In 2014, this error rate decreased to 6.8% based on the Inception architecture [58]. This network proposes a structure which reduces the number of parameters by 12 times that of the AlexNet network. In the same year, the VGGNET network achieved an error rate of 7.3%. In 2015, the deep network ResNet [62] composed of 152 layers reduced the error rate to 3.57%. In 2016, the Inception-ResNetV2 network, which offers hybridization between Inception blocks and residual links, reached a rate of 3.7%.

Table I compares these architectures in terms of depth, number of parameters, and accuracy.

The purpose of these architectures is to provide high-performance networks by reducing storage capacity and computation time. Optimizing the structure of classical convolution layers was one of the main strategies to achieve this treatment. For example, the Inception deep network reduces the number of AlexNet parameters from 60 to 6.8 million and improves performance from 63.3% to 69.8%. This reduction is achieved by Inception blocks through dimensionality reduction techniques.

TABLE I. COMPARISON BETWEEN CNN ARCHITECTURES IN TERMS OF DEPTH, NUMBER OF PARAMETERS, AND ACCURACY.

Architecture	Depth	Parameters (Million)	Top 1 (Accuracy)	Top 5 (Accuracy)
AlexNet [1]	8	60	63.3%	84.6%
ZFNet [73]	8	-	64%	85.3%
Inception [58]	22	6.8	69.8%	89.9%
VGG-19 [57]	19	144	74.5%	92.0%
InceptionV2 [59]	-	11.2	74.8%	92.2%
ResNet-152 [62]	152	21.8 to 60.2	78.57%	94.29%
Incep-ResNetV[70]	-	55.8	80.1%	95.1%

The results obtained also indicate the efficiency of Incep-ResNetV, where it reached a rate of 80% with 55.8 million parameters. As we mentioned before, deep networks improve nonlinearity through the additional activation (ReLu) functions. This improves performance in most cases on large volumes of data. This strategy has been exploited by the ResNet-152 network, where it increases the depth of AlexNet from 8 to 152 while maintaining the same number of parameters (60 million). The techniques used in ResNet-152 improved performance to 78.57%, which reinforces assumptions about the power of deep networks. Finally, the performance of the other variants of Inception (InceptionV3, Xception, and Inception-ResNetV2) varies by 78.8% and 80.1% with a reduced number of parameters compared to ResNet-152.

Figure 44 presents a more informative view of performance compared to the previous table. It indicates that the VGGNet network is the most demanding in terms of storage and computing capacity. The ResNet and Inception architectures form a straight line and are organized according to their depth. Generally, within the same category, the deepest networks perform best.

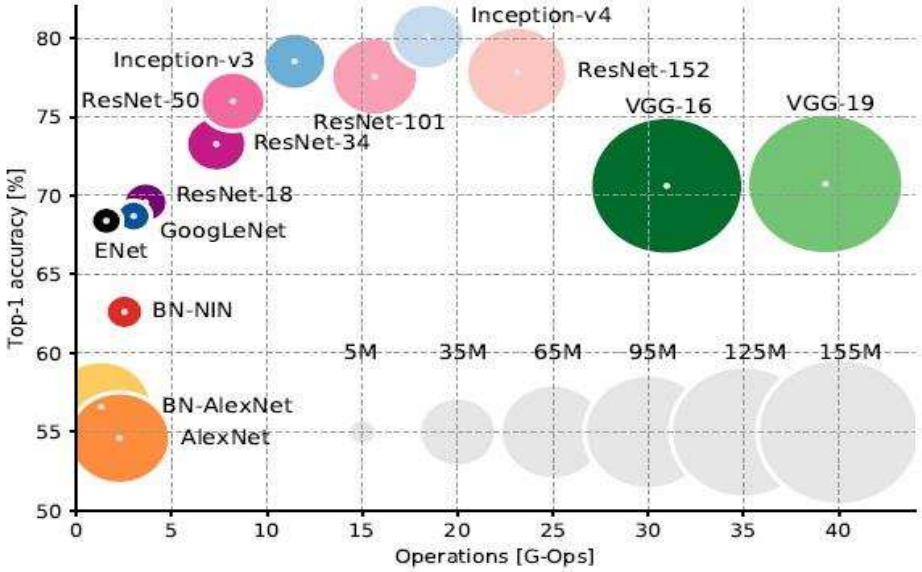

Fig. 44. Representation of architectures in terms of Top-1 accuracy, depth, number of operations, and number of parameters

2.6 Object detection

2.6.1 Regions with convolutional neural networks (R-CNN)

Regions with convolutional neural networks (Regions with CNN features (R−CNN)) [64] is a deep learning architecture designed for object detection. This architecture combines regions proposal methods and CNNs.

Figure 45 illustrates its structure which is composed of 3 modules: ROI extraction, feature extraction, and classification. The first module is based on the selective search method [1] which offers 2000 independent input regions. Then, these regions are adjusted to achieve CNN-compliant dimensionality.

R−CNN exploits the AlexNet model [9] pre- trained on the ImageNet database and followed by fine-tuning on the PASCAL target learning base. The purpose of this transfer is to avoid the problem of overfitting the limited data volumes of the PASCAL database. This step extracts 4096 attributes from each region. Finally, these vectors are provided to the SVM algorithm for the classification task and to the bounding-box regressors algorithm to adjust the bounding box of the proposed region.

The experimental study on the PASCAL VOC learning base showed that R-CNN improved the average error (MAP) by 30% compared to the results obtained previously. Despite these performances, RCNN cannot be exploited in real-time applications because of the high processing time for 2000 regions of interest (47 s/image).

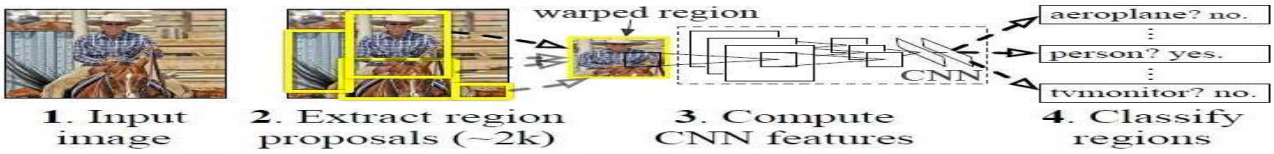

Fig. 45. The Structure of Regions with Convolutional Neural Networks (R-CNN) [64].

2.6.2 Fast R-CNN

Fast R-CNN [65] is an optimized version of the R-CNN architecture [64]. Its main purpose is to speed up the learning and testing time of R-CNN. As we detailed earlier, R-CNN performs the feature extraction task for each proposed region. This requires making 2000 passes in the CNN network, which slows down the test time. To solve this problem, Fast R-CNN takes as input the entire image and the coordinates of the regions of interest, so only one pass is performed for each image instead of 2000.

Figure 46 illustrates the Fast R-CNN network structure. This network passes the input image to the CNN network to generate output feature maps. Then, the proposed regions are identified in these maps and resized by the RoI-pooling layer. The output vectors are then passed to the fully connected layers. Finally, the outputs are used to predict the class by the Softmax classifier and to readjust the bounding box by the bounding-box regressor.

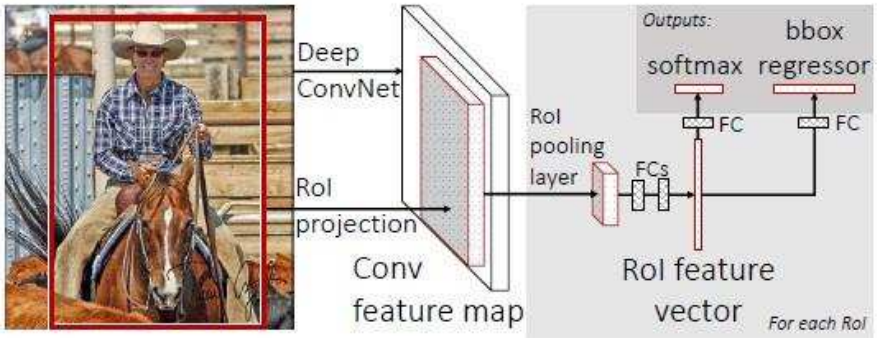

Fig. 46. The structure of the Fast R-CNN network [65].

Like R-CNN, Fast R-CNN uses pre-trained models based on ImageNet learning and fine-tuning technique. To adapt these models on the Fast R-CNN architecture, the last max-pooling layer is replaced by a RoI-pooling layer and the last fully connected and Softmax layer are replaced by two other FC type layers. This structure shows that learning in a Fast R-CNN is performed in a single step instead of three separate steps (SVM, Softmax and the regression algorithm). All these factors have helped to speed up the learning and testing time of RCNN and improve its performance.

The experimental study based on the VGG16 model on the PASCAL VOC 2012 learning database showed that Fast R-CNN is 9 times faster than R-CNN in learning and 213 times in test.

2.6.3 Faster R-CNN

Faster R-CNN [66] is an optimized version of the Fast R-CNN network. Unlike the methods explained previously, this algorithm eliminates the selective search and integrates the process of selecting regions of interest inside the network. This strategy automates the task of selecting regions of interest and speeds up processing time. First, Faster R-CNN passes the input image to convolution layers to generate output feature maps. Then, Region proposal network (RPN) generates the selection frames of regions of interest from these feature maps (Figure 47). The regions proposed by RPN are then resized by the RoIpooling layer. Finally, the output vectors are passed to fully connected layers to classify the object and optimize bounding boxes.

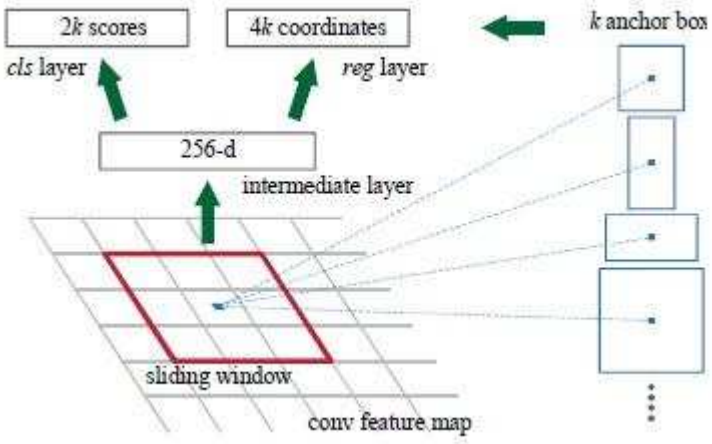

Fig. 47. The process of a Region Proposal Network (RPN)

2.6.4 YOLO

You Only Look Once (YOLO) [67] is a convolutional neural network designed for object detection. Unlike the methods explained previously, this architecture treats the object detection problem as a regression problem to predict object classes and bounding boxes in parallel. The learning and prediction process from the whole image allows the network to encode the contextual information, and thus reduce the rate of false positives. In addition, the use of a single network for detection has accelerated processing 1000 times compared to R-CNN and 100 times compared to Fast R-CNN, where it can process 25 frames per second.

Figure 48 illustrates the detection process performed by YOLO. First, this grating divides the input image into S × S grids. Each grid predicts B selection frames and the probabilities of belonging of the object to the different classes C. A selection frame is characterized by 5 parameters: 4 coordinates (x, y, h, w) and a confidence score P. This score represents the probability of an object belonging to this frame and its accuracy. To summarize, each input image is associated with a prediction encoded as a 3D tensor of size S × S × (5B + C).

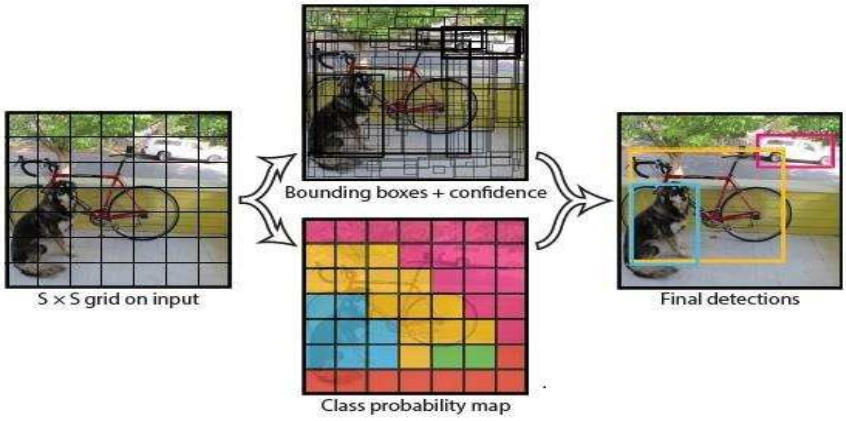

Fig. 48. YOLO's object detection process [67].

YOLO's architecture is inspired by the Inception network [60]. This architecture is composed of 24 convolution layers and 2 fully connected layers. In learning, the first 20 layers are initialized by pre-processing on the ImageNet learning base, while the remaining layers are initialized randomly. Despite the efficiency of YOLO and its faster processing compared to previously introduced systems, this network has some disadvantages related to spatial constraints such as: the number of selection frames in a grid. This limits YOLO's ability to detect small objects organized in groups. In addition, this network has a high localization error compared to other systems based on regions of interest.

2.6.5 Comparison and discussion

The structures detailed above were evaluated on object detection benchmarks: PASCAL VOC 2007 and PASCAL VOC 2012. The PASCAL Visual Object Classification (PASCAL VOC) learning base is composed of 20 classes, including humans, animals, vehicles and interior objects. This training database contains about 10,000 images for training and validation, and it was used in 8 challenges in the period 2005-2012, where each challenge had its own specificities.

2.7 Semantic segmentation

2.7.1 Convolutional Neural Networks

Semantic segmentation consists of assigning a class to each pixel belonging to the input image. In such applications, CNNs are used as pixel classifiers. The network takes as input segments of the image and classifies its center. This operation is repeated for all the pixels of the image, where each pixel is considered as a center of the proposed segment. The main drawback of this method is its very high processing time due to the dense classifications of all the input pixels. This limits their exploitation in real-time applications. In addition, the input patches of neighboring pixels overlap, and therefore the same convolutions are calculated several times.

2.7.2 Fully convolutional networks

Unlike CNN, the FCN network [68] performs the segmentation phase in a single pass (Figure 49). This network is composed of two main parts: down-sampling and resampling. Down-sampling captures semantic and contextual information. Then resampling restores the spatial information. The FCN network can handle variable size entries. This is achieved by eliminating fully connected layers, as they are bound to the fixed input size. This network replaces fully connected layers with convolutional layers to produce spatial maps. Then, these maps are passed to the deconvolution layers [74] to restore the input size and produce per-pixel classified outputs. FCN is characterized by its end-to-end learning process compared to region-based semantic segmentation methods. This network has achieved good performance in segmentation compared to other classical methods on the PASCAL VOC learning base. Despite its efficiency, FCN is characterized by some limitations related to spatial invariance and lack of contextual information. In addition, the resolution of the input image decreases because of its passage through a succession of convolution and pooling layers.

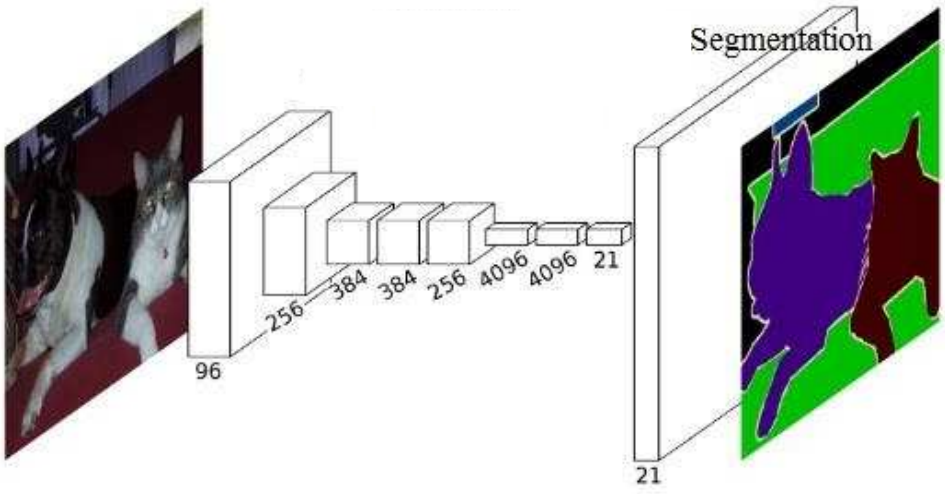

Fig. 49. The structure of the fully convolutional network [68].

2.8 Conclusion

In this chapter, we have detailed the general structure of a CNN network and analysed some known architectures in classification, object detection, and segmentation. In classification, the architectures submitted to the ImageNet competition have been of great interest thanks to their remarkable performance. The main goal of these architectures was to propose deep variants while reducing in parallel the total number of parameters. These variants are characterized by new basic blocks like the Inception blocks in the Inception network, and the DSCs in the MobileNet network. In addition, other variants offer to hybridize between different blocks, such as the Inception-ResNet network which combines between Inception blocks and residual links. In detection and segmentation, the objective was to propose structures which make it possible to detect and segment objects in real time such as YOLO and FCN networks. YOLO speeds up detection time by integrating the RoI generation phase into the network, and FCN performs segmentation in the prediction phase in a single pass.

The following chapter presents the fields of application of the architectures defined in this chapter in classification, detection, and segmentation. In this context, we will start with a general description of a few fields of application. Then, we will detail the field of medical imaging and the field of detection of falsified images which have the area of interest of this thesis.

Chapter III: Convolutional neural networks in vision by computer, image forgery detection, and medical imaging

In recent years, deep learning has been exploited in several areas. In computer vision, CNNs are known for their good accuracy in solving real-world problems. The objective of this chapter is to present an overview of some application domains of CNNs in computer vision, such as image classification, object detection and localization, semantic segmentation, object recognition, image tampering detection, and facial recognition. We also presented some application areas of CNNs in medical image processing based on classification, detection, and segmentation techniques.

Keywords: Computer Vision, Deep Learning, Convolutional Neural Network, Fields of Application, Medical Imaging, image tampering detection.

Contents

3.1 Introduction

Computer vision is a branch of artificial intelligence. It allows a computer to analyze, process, and understand images. Vision systems are exploited to extract relevant information from visual inputs (image or video) for use in other recommendation tasks. Computer vision prediction systems are based on machine learning algorithms. They make it possible to analyze the visual inputs taken by an acquisition system. These algorithms are trained on data to produce output models. The generated models are then used in the prediction phase. Traditional machine learning methods require a formal representation of complex data (images, video, or text). This representation is produced in the feature extraction phase (the handcrafted features). The main drawback of these approaches is their negative influence on the results. Unlike classical ML methods, DL methods and especially CNNs are suitable for complex data, because they merge the feature extraction phase into the learning process. Several factors have contributed to the success of CNNs in computer vision, such as the first GPU implementation [81], the first application of Max pooling [82], and Massive Amounts of Data. CNNs are composed of a set of convolution and pooling layers, which are grouped into blocks, and one or more fully connected layers. These blocks are stacked to form a deep learning network. In recent years, several optimized architectures have been proposed to improve classification accuracy and reduce the computational cost of CNNs. Therefore, in the category of DL networks, CNNs have become the core algorithms in computer vision. Due to their efficiency in handling

large volumes of data, deep learning techniques present powerful tools for processing and analyzing big data. Given the significant progress in computer vision methods, these techniques have been exploited in several real-world applications such as the detection of falsified images [217], R [83], the medical field [84], robotics [85], and autonomous cars [86].

The main objective of this chapter is to detail some known applications of convolutional neural networks in computer vision such as image classification, object detection, and localization, semantic segmentation, object recognition, image tampering detection, and medical image processing (Figure 50).

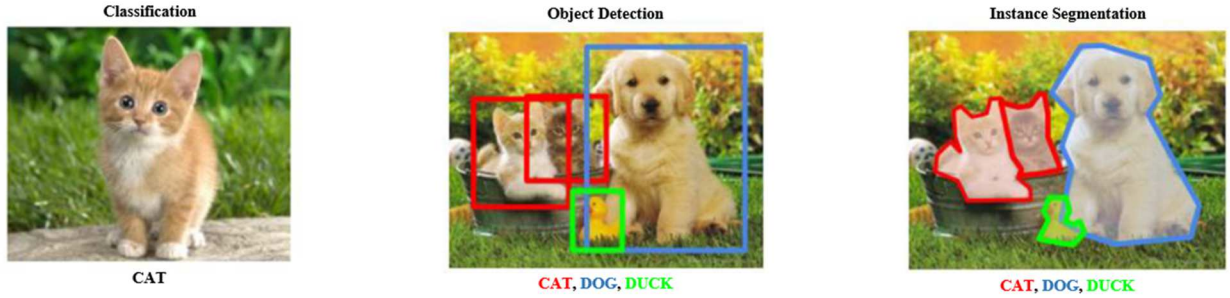

Fig. 50. Applications of convolutional neural networks in computer vision

The proposed methods in this thesis are interested in solving the problems related to falsified images and medical images and exactly CT images. For this, we have presented a section that explains the different types of medical images and some methods proposed in the state of the art for the processing of these images.

3.2 Fields of application

3.2.1 Classification of images

Image classification consists of classifying an image into one or more classes. This problem is also defined by object classification or image recognition. It is considered a basic problem in computer vision. It forms the basis for other computer vision tasks such as localization, detection, or segmentation. In recent years, deep learning techniques have advanced considerably in computer vision, especially in the field of object recognition. Deep learning methods are known for their good performance on large volumes of data and their over-fitting problem on limited data. Therefore, the ImageNet Database composed of 15 million annotated images has attracted a lot of attention [87]. As we mentioned in the previous chapter, the performance of CNNs can be controlled by adjusting the depth and width, and by sharing the weights. It involves a short learning process. The CNNs tested on the ImageNet basis have achieved satisfactory performance; this has encouraged the computer vision community to use them in other fields (Table. II).

TABLE II. CLASSIFICATION METHODS

Methods	Task
Karpathy and al. [88]	Large Scale Video Classification
Lawrence and al. [89]	Facial recognition
Ciresan and al. [90]	Classification of handwritten characters
Tajbakhsh and al. [91]	Analysis of medical images

Hu and al. [92]	Classification of hyper-spectral images
Spanhol and al. [93]	Classification of breast cancer images
Lakhani and Sundaram.[94]	Classification from radiological images

3.2.2 Object detection and localization

Image classification consists of assigning a class to an image while object detection involves surrounding one or more objects in an image with bounding boxes. Object detection is a more difficult task compared to classification because it combines the notions of classification and localization. It allows precisely locating and classifying target objects in an image. For example, it is possible to use object detection methods to identify cells or tissues in medical images [95]. Object detection is among the known areas in computer vision, which have received a lot of interest [96], [97]. Standard object detection methods were based on handcrafted features. These methods are known for their lack of generalization because the attributes extracted depend on the domain of the task being processed. In addition, their evolution was very slow between 2010 and 2012 in the PASCAL VOC challenge. Recently, several efforts have been made to solve these problems based on CNNs. The convolutional neural network as a DL model has achieved great success in several fields in computer vision. In 2012, [9] exploited this network for image classification and they succeeded in reducing the error rate of classical methods from 26.2% to 15.3%.

This progress encouraged the computer vision community to use CNNs in object detection. In 2014, [64] proposed R-CNN, which is based on selective search and CNN and SVM algorithms. This method achieved good performance and reduced the detection time compared to methods based on sliding windows for proposing regions of interest. Despite the efficiency of this method in object detection, its processing time is not suitable for real-time applications. To speed it up, several CNN-based structures have been proposed (Fast R-CNN [6] and Faster-RCNN [66]). [66] have developed an RPN that can almost detect objects in real-time. This network makes it possible to simultaneously predict the bounding boxes and their accuracies in each position. The Faster R-CNN structure [66] combines CNN and RPN networks to perform end-to-end detection. However, Faster R-CNN does not always meet the requirements of real-time object detection. The YOLO method [67] is one of the strategies proposed to adapt the detection time to the requirements of real-time applications. This approach turns the object detection problem into a regression problem.

In the works proposed in object detection, a variety of CNN-type architectures have been proposed: weakly supervised cascaded CNN [98], subcategory-aware CNN [99], Alexnet [64], and an architecture inspired by the Inception network [67]. CNN methods for object detection have been used in several fields: remote sensing [100], medical diagnosis [101], and video surveillance [102].

3.2.3 Semantic segmentation

Over the past decades, semantic segmentation has presented one of the great challenges in computer vision. It consists of segmenting an image into different parts and objects. Its purpose is to assign a class to each pixel of the input image. For a set of k classes $L = \{l_1, l_2 ... l_k\}$ and N variables $X = \{x_1, x_2, ..., x_N\}$, each entry x_i is associated with a class l_j. The class space is composed of k possible states, which are generally extended to k + 1 to deal with the background class of the image. In general, X is a 2-D image of $W \times H = N$ pixels. In segmentation, the processing is more complicated compared to object recognition and detection. Classification assigns a class to each image and detection classifies objects and defines their

bounding boxes, while a segmentation algorithm can also segment new objects. Classical image segmentation algorithms are usually based on clustering methods and additional information on contours and edges [103], [104]. Several approaches have been proposed to improve clustering performance. Modelling based on the Markov process [105] and the combinations of edge detection in a hierarchical approach [106] are among the known methods. Despite the popularity of classical methods, the new success of deep learning techniques in various tasks has made these methods very popular in computer vision including segmentation. The DL methods present the alternative, which makes it possible to automatically learn the characteristics of the problem treated instead of extracting them by the extraction methods, because this process requires expertise in the domain, efforts, and often too much adjustment to adapt them to the problem being addressed. In deep learning, the performance of CNNs in classification [9] [58] and in object detection [64], [65], [66] encouraged researchers to exploit them in pixel classification problems like semantic segmentation. These networks have been used as components in several architectures of segmentation.

Image segmentation methods in DL are classified into three types: object segmentation, semantic segmentation based on FCNs, and weakly supervised segmentation [107].

The object segmentation methods by region start with the extraction of regions of interest, then these regions are classified by classification techniques. R-CNN [64] is one of the DL-type architectures exploited in object detection and semantic segmentation. It allows the segmentation phase according to the results of the detection. Despite the effectiveness of this method, it can cause a loss of information related to the field, as the attributes used to come from fully connected layers, while the intermediate layers contain more specific information. In addition, the generation phase of the proposed segments has a high temporal complexity, which can affect the final performance.

The main idea of semantic segmentation methods based on FCN is to carry out a pixel classification, without the need to go through the proposal stage of the interest regions. FCN [68] is among the most used networks in semantic segmentation. It is considered an extension of CNN networks, where known architectures (Alexnet [9] VggNet [75], Inception [23], and Resnet [62] are transformed into FCN. Despite its effectiveness, FCN is characterized by certain limitations related to spatial invariance, lack of contextual information, and poor resolution of output images. U-Net [21] has demonstrated an abolishing result in the segmentation of biomedical images. It is a fully convolutive network (FCN) from start to finish that it contains only convolutive layers and contains no dense layer thanks to which it can accept an image of N 'No matter size. Deeplab [69], [108] presents one of the solutions that improve output resolution. This method uses a fully Connected Pairwise CRF [109] as a separate module to perform post-processing and refine the result of the segmentation. Other works propose to improve segmentation by exploitation of contextual information. For example, [110] used the overall AVG-Pooling layer to obtain the overall context. Other research has solved the problem of multi-seller by the proposal of a network composed of N FCN which uses different scales [111].

Segmentation in weakly supervised learning is another area of interest in semantic segmentation [112]. The purpose of this method is to speed up the annotation of images in the learning base because the generation of segmentation masks for learning is a difficult and time-consuming task. Segmentation in weakly supervised learning proposes the use of bounding boxes instead of segmentation masks to reduce overhead. For example, [113] used a bounding box-based annotation for learning, and they iteratively obtained the

segmentation masks. The PASCAL VOC learning base [114] is among the known bases in segmentation and has been widely used for the validation of the methods proposed in semantic segmentation. To improve this base, several extensions have been developed: PASCAL Context [115] and PASCAL Part [116]. Microsoft COCO [117] is another segmentation database composed of more than 80 classes.

DL methods in semantic segmentation have been used in several fields of application: autonomous cars [118], medical imaging [21], and urban remote sensing [119].

3.2.4 Human Pose Estimation:

The estimation of the human pose is a known problem in computer vision. It makes it possible to determine the position of the human joints from images or image sequences. The estimate of human installation is a very difficult task because of the great dimensionality of input data and the high variation of human poses. Recognition of action and the estimate of human installation are two related problems but are generally treated differently in literature. [120] were the first to offer a multitasking CNN network that allows you to manage the two problems at the same time. The estimate of the human pose presented an important role in different applications of the real world motivated by current technological advancements. Among the known applications are surveillance videos [121]: Surveillance makes it possible to follow and monitor movements in particular circumstances, for example in airports or supermarkets. Men machine interaction [122], in systems; computers can be checked for example by human gestures or sign language. The human-robot interaction [123] in certain assisted living situations, robots must estimate human positions to ensure good interaction, and medical imagery [124]: The estimate of the human pose can be used to assist doctors in remotely checking patient activities.

Previously, the human pose estimation problem was handled using pictorial structures [125]. In recent years, deep learning has proven its effectiveness in this area, where different architectures have been exploited. These methods are classified into holistic and part-based methods [126]. Holistic processing methods accomplish the task holistically without the need to explicitly define a model for each part and its spatial relationships. In contrast, part-based methods start by detecting individual parts of the human body and then a graphical model is used to integrate the spatial information.

DeepPose [127] is the first DL model proposed for human pose detection. This model belongs to the class of holistic methods. Several works have used CNNs to accomplish this task. For example, [128] proposed the exploitation of local and background patches for the training of a CNN to predict the probabilities of the presence of the parts and their spatial relationships. In another contribution, [129] used multiple CNNs to independently classify multiple body parts. [130] proposed hybridization between a CNN and a Markov random field. To improve CNN learning, [131] combined CNN with a deformable mixture of parts model to perform end-to-end learning.

3.2.5 Convolutional neural networks for image forgery detection:

Tampered image detection is considered an important application area given the number of huge images around us. Several research studies have been conducted for this purpose. Generally, one will find two types of methods, classical methods, and deep learning methods. The classical method is based on the classical process of classification; it starts with the extraction of the descriptors and then classifies them. The authors of [49] presented an image splicing technique using visual artifacts. In [51], the authors used local binary

pattern (LBP) and pyramidal transformation (SPT) to detect counterfeit images. In [32], the authors highlight the

Recent advances in image manipulation and discuss the process of restoring damaged or missing areas of an image. The authors of [4] presented a review of the different counterfeit detection techniques. In deep learning methods as we have already presented, it is the network that takes care of the descriptor extraction and classification phase.

3.2.6 Convolutional neural networks for analysis medical images

Diagnostic assistance systems (CAD) are used to help medical specialists in their decisions. These systems make it possible to reduce inter-variety between decisions of different experts and to avoid subjectivity. In recent decades, the evolution of the Deep Learning fields on the one hand and the other hand the development of the equipment and the good quality of the scanned images have encouraged the computer vision community to improve the efficiency of the CADs. Despite the efforts made, CADs have several limitations related to the collection and annotation of the data necessary for their design. Collection requires many patients and tests to ensure the necessary amount of data. In addition, annotation is a costly task in terms of time and effort, especially segmentation which requires annotating each pixel in the entry image. Generally, the annotation is ensured by a group of experts to guarantee the validity of the classes.

At the end of the 1990s, classic learning techniques (ML) experienced great interest from the medical imaging community [132], [133]. The critical step in these methods is the extraction phase of discriminating characteristics from the image. This process requires an in-depth study by experts in medicine and computer vision. In addition, the characteristics extracted greatly depend on the medical task treated. This limits the application of the methods offered to other types of medical images. To solve this problem, the logical solution is to automate the characteristics extraction task to adapt it to any type of application. This process is carried out by several types of deep learning algorithms.

Convolutional neural networks are among the most used networks in image processing [134]. The development in the structure of CNNs in computer vision has encouraged the medical imaging community to exploit these architectures to design powerful CADs. In medical imaging, scanned images have several types like ultrasound (US), x-ray, computed tomography (CT) and resonance imaging (MRI), positron emission tomography (PET), and slides histopathology (Figure 3.2). DL algorithms use these images to solve different tasks in medical imaging: classification, localization, detection, and segmentation.

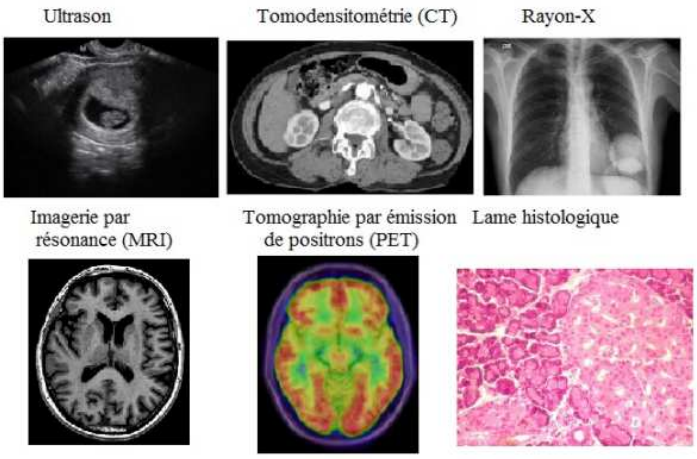

Fig. 51. Scanned images types

From a medical point of view, a classification consists for example of checking whether disease exists or not or of distinguishing between different types of cancer. Localization makes it possible to locate and identify regions of interest. For example, locate tumor areas on an image to classify them. Detection makes it possible to detect several similar regions of interest on the same image. For example, it is possible to detect tumors in CT images. Finally, segmentation allows precise identification of regions of interest, for example, the segmentation of brain tumors on MRI images.

We will detail what follows some methods (Table 3.2) are proposed in the state of the art for the processing of different types of medical images. For more information [84] and [134] detailed syntheses on the application of DL networks in medical imaging.

- ***Classification***

The classification of objects in medical images consists in classifying parts previously extracted in two or more classes. To carry out this task, the location of the lesion is an important pre-treatment to ensure good precision. Several works have been proposed in the state of the art to automate the classification of different medical image modularity: radius-X [135], [136], CT [137], MRI and PET [138], histological blades [139], [140], and US [141], where the exploitation of CNNs has experienced great success. [135] used a modified version of the GoogleNet pre-entry network for the classification of pulmonary radiography images (X-ray). In another contribution, [136] proposed a modified version of DenseNet made up of 121 layers of convolution (Chexnet). The purpose of this network is to automate the classification of 14 diseases from pulmonary radiography images (X-ray). The proposed system has achieved the same performance as radiologists. In the MRI and PET scans category, [138] used a 3-D CNN to rebuild missing PET images. The proposed system allows you to assist radiologists in the diagnosis of Alzheimer's disease (See Table. III).

TABLE III. SUMMARY OF SOME WORKS PROPOSED FOR THE PROCESSING OF MEDICAL IMAGES BY CONVOLUTIONAL NEURAL NETWORKS.

Tasks	Author	Modularity	network	Application
Classification	Rajkomar and al. [135]	X-ray	GoogLeNet	Classification of chest X-ray images

Rajpurkar and al. [136]	X-ray	ChexNet	Disease classification from chest x-rays
Li and al. [137]	MRI and PET	3-D CNN	Assist in the diagnosis of Alzheimer's
Spanhol and al. [138]	Histological slides	AlexNet	Breast cancer tissue Classification
Xu and al. [140]	Histological slides	CNN	CNN Classification of epithelial (EP) and stromal (ST) Tissue
Byra et al. [141]	Ultrasound	Inception−ResNet−v2	Assess fat in the liver

- ***Location and detection***

The detection of objects of interest in an image is an important step in diagnosis that requires a lot of effort by clinicians. For example, a CT image may contain hundreds to thousands of cancer cells to be detected [142]. Several works in the state of the art have automated the task of detection in medical images. The main purpose of the proposed systems is to improve accuracy and reduce manual processing time. Localization of organs or landmarks has received great interest, as it presents an important pre-processing step in several segmentation tasks.

The first CNN-based detection system was proposed to detect nodules in X-ray images [143]. In another study, [144] exploited the R-CNN network for the detection and classification of malignant or benign lesions on a mammogram. In the category of CT images, [145] proposed an optimized version of the Faster R-CNN network for the detection of lung nodules in CT images. In another contribution, [146] identified landmarks on the surface of the distal femur on MRI images based on three CNN arrays. In another study, [147] exploited a 3-D CNN network to detect microbleeds in brain MRI scans.

In this thesis, we have dedicated a section that details the recent works proposed for the detection of tumors due to the complexity of this task and the challenges it presents.

- ***Segmentation:***

Segmentation makes it possible to identify the set of voxels or pixels that constitute the outline or the interior of the objects of interest. This task is an important step in CAD, as it allows precise identification of regions of interest in medical images. The automation of tumor segmentation in the brain has been of great interest [148], as it reduces the manual effort made by specialists on MRI and CT volumes. This treatment is necessary to precisely direct the surgical resection. For example, [149] used a CNN composed of 11 convolutional layers for the segmentation of glioma in the brain from MRI scans.

In the category of electron microscopy images, [21] proposed a new U-net architecture based on the CNN network for the segmentation of neural structures. This architecture is composed of the same number of up-sampling and down-sampling layers. It allows us to perform the segmentation task in a single pass and to consider the contextual information. In another contribution, [150] applied pixel-based segmentation using the sliding window strategy and the CNN network.

In the category of CT images, [151] exploited a DCNN composed of encoding-decoding structures for kernel segmentation. To speed up the processing time of CT images, [152] used the FCN network for cancer cell segmentation.

3.3 Conclusion

Computer vision systems are based on machine learning algorithms. These systems are used to analyze visual inputs like images and videos. In recent years, the exploitation of DL networks in computer vision and especially CNNs has been of great interest in several real-world applications, such as image classification, object detection and localization, semantic segmentation, segmentation of biomedical images, falsified images detection, and facial recognition.

CNNs are characterized by their efficiency in feature extraction, and they are also stable to transformations. These characteristics have made CNNs a good use case in several fields of computer vision applications. Several CNN-like variants have been proposed in the literature. The main goal was to adapt the classic architecture to the nature of the application. In addition, the nature and complexity of the architecture vary according to the complexity of the problem treated, for example, object segmentation is a more difficult task compared to detection and classification.

The processing of medical images and falsified images are among the known problems in computer vision. Several CNN-type architectures previously proposed have been exploited and adapted for the processing of different modularity of this type of image.

In conclusion, despite the promising results of the architecture proposed in the state of the art, significant challenges remain. These challenges are related to the choice of the optimal architecture to solve a defined task. In addition, some architecture is characterized by their high complexity, and this limits their real-time executions in real-world applications.

The following chapters present the state-of-the-art contributions of this thesis. In this context, we will start with a comprehensive overview of the DL methods proposed for the detection of biomedical images. At the beginning of this chapter, we detail the work related to segmentation methods. Then, we will explain the techniques for pre-processing these images and the structure of some public CT learning bases. Finally, we present the proposed contribution with the experimental results. The second chapter details the steps of the segmentation of the falsified images and the experimental results which also present the fields of interest in this thesis.

Chapter VI: Encoder-decoder based convolutional neural networks for image forgery detection

The semantic manipulation of images has become easier thanks to the enormous evolution of image editing software and increasingly efficient computer infrastructures. As result, the identification of these modifications becomes a very difficult task since the modified regions are not visually apparent. In this chapter, a novel encoder/decoder-based convolutional neural network method called Fals-Unet is proposed to localize manipulated regions. The encoder of our method uses architecture topologically identical to that of the Resnet50 method; its main objective is the exploitation of spatial maps to analyze the discriminating characteristics between manipulated and non-manipulated regions. The decoding network learns the mapping from low-resolution feature maps to pixel-level predictions to locate tampered regions. Finally, the predicted binary mask (0: tamper, 1: do not tamper) is generated by the final layer (Softmax).

This chapter is organized as follows: Section 4.1 introduces image manipulation. Then we present an art study in section 4.2. Then, the method proposed in this context is contributed to section 4.3. Finally, we end with the results obtained and the qualitative and quantitative studies in section 4.4.

Contents

4.1 Introduction

Nowadays, we live in a world where we are exposed to a huge digital image flow. Therefore, the manipulation of these images has become easier, thanks to the fast evolution of image editing software. In addition, it becomes very difficult to differentiate between a manipulated and original image because of such software, which leaves no visible trace to the human eye. Falsified images are used in several application fields such as police investigations as forensic evidence, journalism, etc. Falsified images for malicious purposes will have dangerous consequences in our society.

Copy-move and image splicing are considered the most common methods for image manipulation. These methods involve replacing one or more fragments of the image with one or more fragments of the same

image or from other different images to create falsified images (some examples of this manipulation are shown in Fig. 52). Therefore, falsified image detection is the path that we have chosen to provide efficient solutions using deep learning approaches. In literature, generally, we have two detection categories for digital image falsification, in the first category we find the classical methods, Fridrich et al. [176] proposed a method based on a discrete cosine transform of the superimposed blocks (DCT). Popescu and Farid [211] used a method based on the analysis of principal components (PCA) to represent the image segments. Luo et al. [200] proposed an algorithm that divides the image into small, overlapped blocks, then, this algorithm applies the comparison of the similarity between these blocks. Finally, it uses intensity characteristics to identify double regions. The method based on bi-spectral analysis was proposed by Farid [182] to detect unnatural higher-order correlations introduced into the signal by the falsification process. Thus, Fu et al. [182] used the Hilbert-Huang transform (HHT) to generate classification characteristics and a statistical natural image model, this method is based on moments of characteristic functions with wavelet decomposition (DWT). Shi et al. [220] utilized the natural image model multi-size blocks (MBDCT) to detect splicing images. Johnson and Farid [190] presented a method for detecting the composition of two or more people in a single image as a function of the intrinsic estimation parameters of the camera. Dirik and Memon [174] calculated the estimate of CFA number patterns and analyzed noise based on CFA, and then used a simple threshold classifier. The second category includes deep learning methods; these methods have shown better results in the classification and detection of manipulated images. The authors of [217] proposed a fully convolutional network (FCN) to detect manipulated regions. FCN is based on replacing fully connected layers with convolutional layers having a 1×1 kernel, which allows each image pixel to be classified as falsified or non-falsified. In [158], the authors exploited the spatial and frequency domains using an LSTM network and an encoder to detect the manipulated regions. Finally, they used a decoder network that learns mapping from low-resolution feature maps for pixel-wise classification to generate binary masks. The authors of [76] have proposed a course to refined convolutional neural network (C2RNet) method for locating the tampered regions. This approach uses C-CNN to predict suspect coarse tampering regions and R-CNN to refine C-CNN detection results. In [79] the authors use the SRM (Spatial Rich Model) filters to exploit the residual characteristics of the manipulated and non-manipulated pixels. They use a U-Net encode based on DenseNet architecture to integrate the spatial characteristics to predict the binary mask of the falsified Regions. In our contribution, we propose a new method that aims to segment the forgeries images based on U-Net [187] Encoder/Decoder network, followed by a pixel-wise classification layer. The encoder network consists of convolutional layers, which correspond to the ResNet50 network [64] designed to classify objects. We start by initializing the training process with the weights that are already training on the gigantic ImageNet dataset [65]. Then, we consider that each decoder layer has its own symmetrical layer in the encoder. Finally, we use the two-class SoftMax classifier to generate binary masks and classify the decoder output.

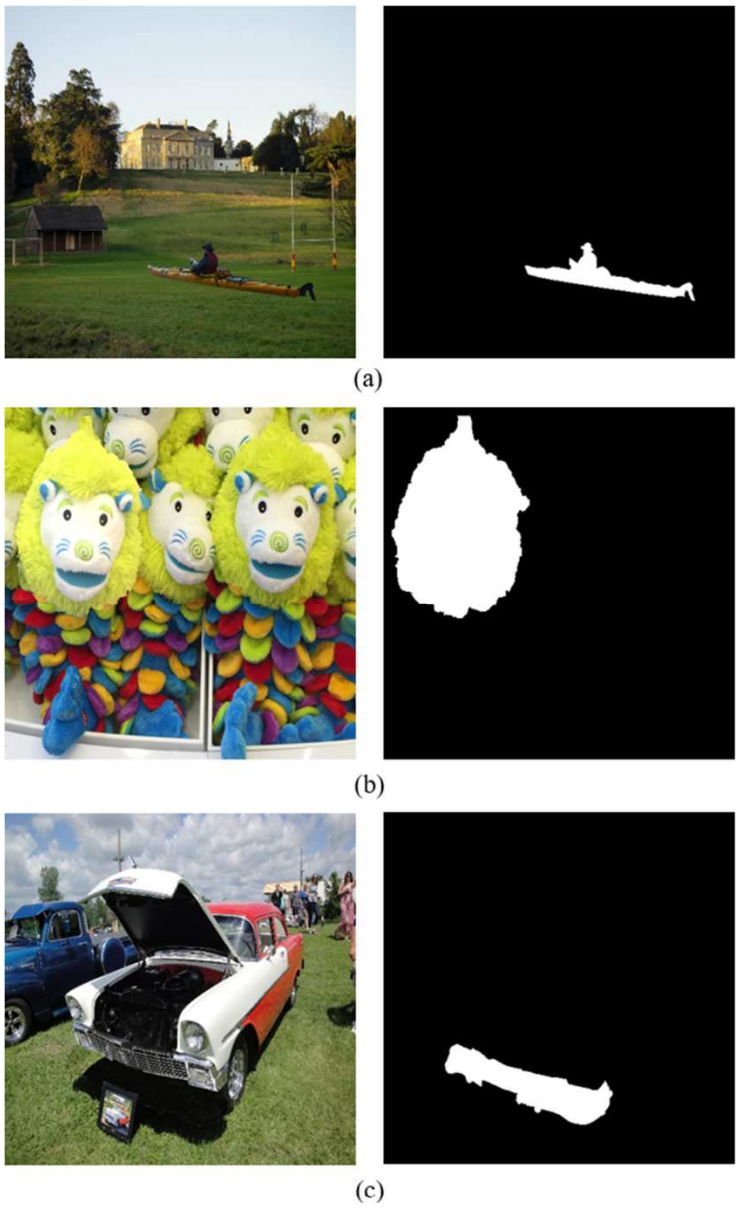

Fig. 52. Images manipulation examples. a, b, and c respectively illustrate splicing, copy-move, and object removal techniques on a tempered image. The first column corresponds to falsified images while the second column represents their ground-truth masks

4.2 Related work

In recent years, several research studies have been proposed to detect different types of image manipulation such as JPEG artifacts, resampling, and content modification. In this section, we will give a brief overview of some of the existing solutions for detecting image forgeries. In literature, several approaches aim to detect the resampling of digital images [179, 202,207, 2012, 220]. Most of these methods use cubic or linear interpolation. The authors of [216] used the periodic properties of interpolation by the second derivative of the transformed image to detect manipulation in an image. The authors of [207] proposed an approach based on adding noise to identify resampling on compressed JPEG images, where we add noise before the image passes through the resampling detector; this method has proved that adding

noise makes it easier to detect resampling. In [178, 179], the authors generated a feature from the normalized energy density, and then they used an SVM classifier to robustly detect resampled images. The authors of [184, 193] have proposed approaches to optimize the JPEG artifacts produced by compression techniques. Feature-based methods have been presented by the authors of [154, 226] for detecting image manipulation.

Several approaches have been proposed to detect the carving of the seams [181, 199, and 222] and the removal of paint-based objects [169, 196, and 229]. Many methods use the JPEG blocking artifacts to detect changed regions [163, 164, 177, 198, and 201]. The authors of [153, 189, 191, and 194] focused their efforts to identify and localize copy-move manipulation. The authors of [194] used an approach based on segmentation to detect copy-movement manipulations. This approach consists of dividing an image into semantically independent patches and then performing a correspondence of key points between these patches. In [173], the authors used an algorithm of patch matching to calculate the approximate nearest neighbour block on an image. They also used circular harmonic transforms as an invariant characteristic that showed efficiency on duplicated blocks that underwent geometric transformations. The authors of [203] presented an image splicing technique using visual artifacts. In [205], the authors used the LBP (Local Binary Pattern) and the SPT (Steerable Pyramid Transform) to detect the forgeries images. In [185], the authors highlight the recent advances in image manipulation and discuss the restoring process of damaged or missing areas in an image. The authors of [155] presented a review of the different techniques of forgeries image detection.

Recently, many types of research have been done to detect image manipulation by applying different machine learning and computer vision algorithms [56, 62]. Several learning architectures [156, 69, 81] have been proposed in the semantic segmentation field, which outperform previous classical state-of-the-art methods in terms of accuracy. The deep network methods [4, 69] are based mainly on CNN (Convolutional Neural Networks), where different layers exploit hierarchical characteristics to learn the spatial map of the semantic region. The authors of [219] transformed a CNN classification into a fully convolutional classification by replacing fully connected layers to produce spatial heat maps. Finally, they used a deconvolution layer to up-sample the heat maps for generating dense per-pixel labelling. SegNet [156] exploits a decoder to effectively learn low-resolution heat maps for pixel-wise predictions for segmentation. The authors of [78] proposed a bidirectional vector allowing the full exploitation of the contextual semantic relationships between pixels and the construction of a GCN under the attention mechanism, whose characteristics that have high probability weights are linked to each other. In [223], the authors use a sequential joint deep learning algorithm for improving small sample classification. This algorithm uses Bi-LSTM and AML to integrate the multi-scale convolution features under an attention mechanism. The authors of [19, 40] used the fully connected pairwise CRF as a step of post-processing to refine the segmentation result. In [210], the authors used skip connection to perform a late fusion of feature maps to make independent predictions for each layer and merge the results. Recent studies such as [7, 8, 15, 52, and 63] use deep learning-based models for manipulation detection. These works include generic manipulations detection [7, 8], bootleg [166], splicing [215], and up-sampling [161]. The authors of [213] proposed GNCNN (Gaussian-Neuron CNN) for steg-analysis. In [162] the authors proposed an approach of deep learning to identify facial manipulation. The authors of [80] proposed an image region forgery detection using a model of stacked auto-encoder. In [159], the authors proposed a new form of the convolutional layer to learn the manipulated features from an image. Deep learning has improved performance in different visual recognition tasks such as semantic segmentation [219], image identification [168], and image

classification [82]. Deep learning is based on the extraction of hierarchical characteristics to represent the visual concept, which is useful in the segmentation of objects. Most architecture depends on CNN, which provides spatial maps relevant to the manipulated regions. In [197], the authors propose a CNN-based forensic analysis framework for the detection of image operator's chains. They used the architecture of two-stream CNN; the first stream is used for extracting local residual features, while the second is used for detecting tampering artifacts. The authors of [183] propose a theoretical framework of information to detect tampering operations, such as median filtering, resizing, and contrast enhancement in multiple chains. This method uses conditional probability criteria to quantify specific operation identification.

4.3 Proposed algorithm

The main objective of our work is to locate the falsified regions in the images. To achieve this objective, we used a CNN architecture composed of an encoder/decoder called Fals-Unet as shown in Fig. 2. The main role of the encoder is to extract high-level feature vectors by performing a series of convolution, activation, and normalization functions to obtain context information. However, the decoder is used to locate the spatial information location (up-sampling) by concatenating (addition) two inputs (one from the previous layer of the decoder and the other from the symmetrical layer of the encoder). Typically, the Fals-Unet network consists of several layers, where each data layer is considered a three-dimensional array of size $h \times w \times p$, where h and w represent respectively the height and width of the data, and p indicates the depth size. Each convolutional layer involves varying size filters. These filters generate spatial maps of entities connected to the local region of the previous layer; thus the weights of these filters are initialized with weights already performed on a large dataset (such as ImageNet) which gives better results. The architecture of Fals-Unet is illustrated in Fig. 53. The framework of this architecture is divided into two parts (1) encoder and (2) decoder.

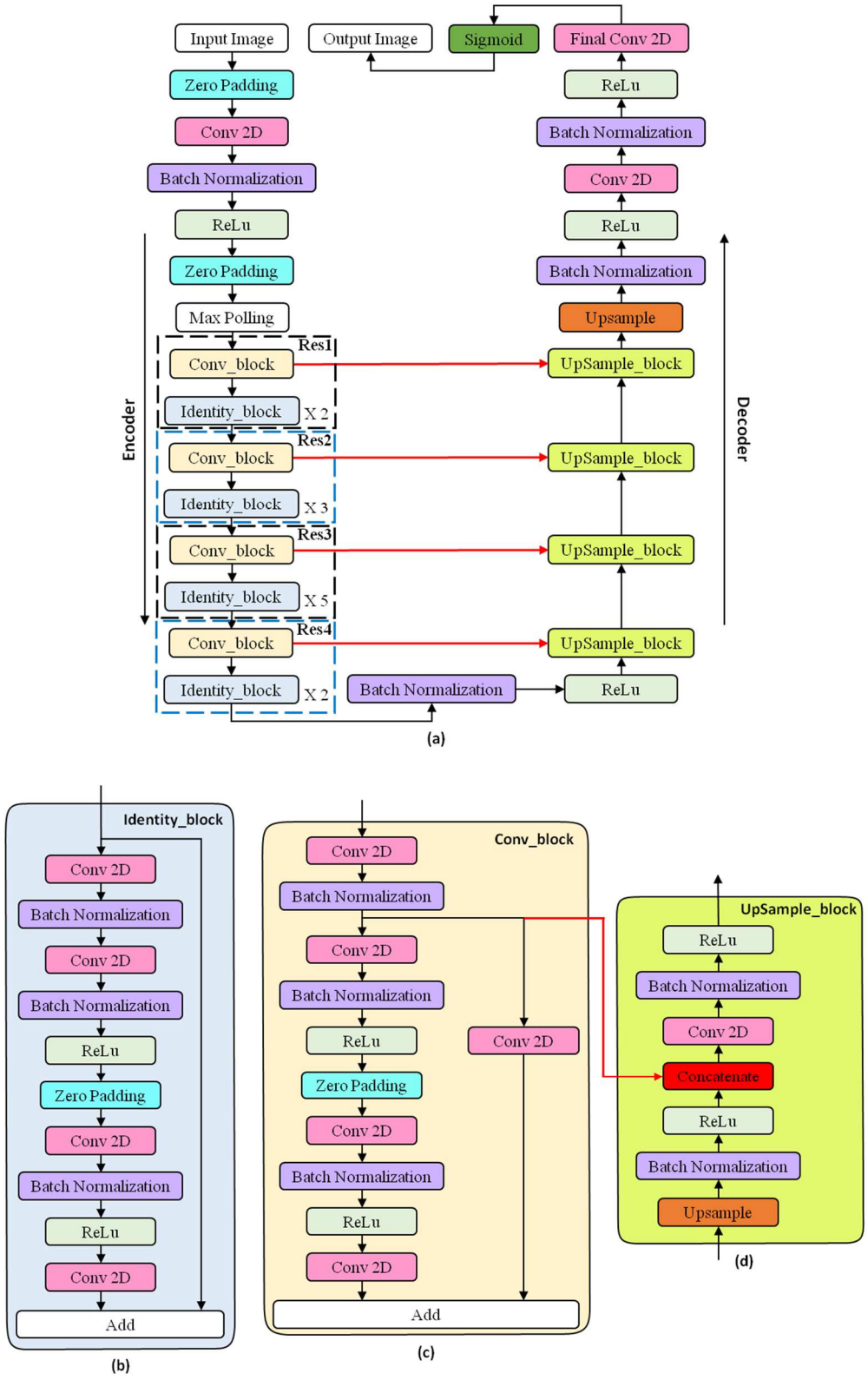

Fig. 53. Overview of the proposed framework for localization of manipulated images regions. a is the main architecture of Fals-Unet, b represents the identity block which will be repeated n times in each encoder stage. c corresponds to a convolutional block that will be concatenated to each upsample block of the same decoder stage level. d is the upsample block that forms the decoder stages

4.3.1 Encoder:

In this article, we used the ResNet [64] encoder as a basic method; this method showed high performance in the image classification process, its main advantages are easy optimization of residual cards, the optimal formation of several layers, and the accuracy improvement by the network depth increase. This encoder consists of convolutional blocks, and it uses an identity shortcut connection that can ignore one or more convolutional layers. This optimizes performance in the simple convolutional block. Among several ResNet variants, we have chosen ResNet50 with some adaptation changes. Consider x as an input to ResNet, thus the mapping from input to output of the unit is M (.). Therefore, the residual unit output would be M(x) + x in the front pass.

Fals-Unet approach takes as input an image of size 256 × 256 × 3; the encoder initial block makes the convolution with 7 × 7 kernel size and a stride equal to two. This block is followed by Max-Pooling with a stride size of two for reducing the input size. Subsequently, we apply four residual stages Res1, Res2, Res3, and Res4 in the cascade; each residual stage of the network consists of the conv and identity blocks (see Fig. 2). We use at each step three successive convolution products whose filter sizes are respectively 1 × 1, 3 × 3, and 1 × 1. The stage convolution operations Res1, Res2, Res3, and Res4 respectively use the kernel size (64, 64, and 256), (128, 128, and 512), (256, 256, and 1024), and (512, 512, and 2048). In each conventional layer, we use batch normalization [188], which is characterized by its robustness to the covariance shift. In addition, we use an activation function ReLU [206] represented by max (zero, x). Using zero-padding preserves information at the edges of filter when its size greater than 1 × 1. Generally, the input size will be reduced to half in terms of width and height, but the depth size will be doubled. As we progress from one stage to another, the depth size is doubled, and the input size is halved. Therefore, each residual unit in the encoder yields a set of spatial feature maps.

4.3.2 Decoder

The Fals-Unet network decoder is composed of several decoder blocks, these blocks are concatenated with corresponding coder blocks. Each decoder block performs an up-sampling of the feature cards received from the previous layer, and it includes a conventional operation whose kernel size is 3 × 3 followed by Batch Normalization and ReLu. This up-sampling procedure is repeated four times, as mentioned in Fig. 2. In the decoder part, four Up-sample bloc blocks are respectively mapping with Res4, Res3, Res2, and Res1 of the encoder, where each UpSample blocks are respectively composed of (16 × 16 × 256), (32 × 32 × 128), (64 × 64 × 64), and (128 × 128 × 32) feature maps. At the end of the decoder network, we use the Softmax pixel-wise classification function to predict the manipulated regions. We pose $P(Y_k)$ as the probability distribution on the two classes (0: falsified, 1: non-falsified) which is calculated by the Softmax function, the prediction of labels is obtained by maximizing $P(Y_k)$ by reporting to k, with $Y= \text{argmax}P(Y_k)$. Finally, we will produce binary masks that indicate the regions manipulated in the image.

4.3.3 Unbalanced classes issue

One of the most encountered problems in the falsified image segmentation field is the unbalance of the classes; this problem presents a very active axis of research [18, 34]. It occurs when the pixels of the non-falsified regions greatly exceed the pixels of the falsified regions, which leads to poor pixel classification. To solve this problem, we will calculate the weights of the spliced (W_m) and non-spliced (W_n) regions based

on the percentage of statistics on the truth mask of the training set. The weights are inversely proportional to the frequency of appearance of each class, in other words, the higher appearance frequencies produce lower weights. During the learning, we use Weighted Cross-Entropy to learn the parameters, so the weight loss function (40) is:

$$L(p,q) = \sum_{i=0}^{n} W_n\,(1-q_i)\log(1-p_i) + W_m q_i \log(p_i) \quad (41)$$

Where W_n and W_m respectively represent the weights of the training set non-manipulated and manipulated regions. Weighting the weights makes our network more sensitive to the falsified region, n is the number of pixels in each image, q_i represents the ground-truth mask and p_i is the predicted mask.

4.4 Experiments

4.4.1 Datasets preparation

To evaluate our model, we use four datasets (see Table IV). NIST'16 [209] (a dataset that contains the three main manipulation types - (a) copy-clone, (b) remove, and (c) spliced), CoMod [221], CoVerage [222], and CASIA [175]. NIST, CoMod, and Coverage datasets contain truth masks. Unlike the CASIA v1.0 and CASIA v2 datasets, in which we generate truth masks based on the information provided by the authentic images. We divide each dataset into three randomly chosen subsets: training (70%), validation (10%), and tests (20%).

4.4.2 Performance metrics evaluation

In this section, we define the different performance metrics (ROC, AUC, F1-score, MCC, and IoU) used to evaluate and verify the efficiency and classifying correctness of our method.

- **Receiver Operator Characteristic** (ROC) curve represents the trade-off between specificity (1 − False Positive Rate) and sensitivity (or True Positive Rate). A curve close to the 45 degrees diagonal of the ROC space indicates less accurate tests. Generally, the closer the curve is to the top-left corner, the more accurate the test is.
- **Area Under the ROC Curve** (AUC) measures the entire two-dimensional area under the ROC curve (by integral computation) from (0,0) to (1,1). This metric provides an aggregate measure of performance for all possible classification thresholds.
- **F1-score** metric represents the weighted average of recall and precision. Therefore, this metric takes both false negatives and false positives into account. F1-score is usually more significant and more useful than accuracy, especially, in the unbalanced classes' case.

$$F1 - score = \frac{2TP}{2TP+FN+FP} \quad (42)$$

TABLE IV. TAMPERED IMAGES DATASETS

Datasets	Images number	Ground-Truth	Image size
NIST'6 [209]	564	YES	–
CoMoD[221]	260	YES	512 × 512
COVERAGE [22]	100	YES	400 × 486
CASIA v1.0 [175]	921	No	384 × 256
CASIA v2.0 [175]	5123	No	384 × 256

- **Matthews Correlation Coefficient** (MCC) [165] (43) metric is also used because it is well suited to the classification of unbalanced class problems due to its integration of true negatives. This metric only produces a high score if the prediction performed good results in all four categories of the confusion matrix (true negatives, true positives, false positives, and false negatives). MCC returns a value between −1 and +1, where 0 is a random prediction, +1 indicates a perfect prediction, and −1 represents a total disagreement between observation and prediction.

$$MCC = \frac{TP*TN-FP*FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (43)$$

Finally, Jaccard Similarity Index or Intersection over Union (IoU) (4) metric is measured to represent the similarity between two classes of data A and B. It is within the range from 0 to 100%, a high percentage indicates more similarity between the two classes.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (44)$$

Where normalization condition takes place: $0 \leq J(A, B) \leq 1$. Every image is consisting of pixels, for adapting it to discrete objects, we can write the last expression in the following way:

$$J = \frac{1}{n}\sum_{i=0}^{n} \frac{\tilde{y}_i y_i}{y_i + \tilde{y}_i - y_i \tilde{y}_i} \quad (45)$$

Where y_i represents a binary value (label) of the corresponding pixel i and $\tilde{y}_i$ is estimated probability for the pixel i.

4.5 Results

In this section, we carry out several simulations to demonstrate our proposed method. We prove the robustness and effectiveness of our method by evaluating, on the first hand, the performance metrics on several datasets, on the other hand, by comparing the obtained results by our method with those obtained by the state-of-the-art methods and recent methods.

4.5.1 Fals-Unet results on different datasets

Table V shows Jaccard Index and F1-score results obtained by the proposed method on different datasets. We can see that Fals-Unet has correctly classified the falsified pixels. The receiver operating characteristic curves (ROC curves) are illustrated in Figs. 54 and 55. Our proposed method presents good performances on both CASIA v2.0 and NIST'16 datasets. From this figure, we calculate the AUC of ROC and show this measure in Table V. We clearly see that our method shows advantages and better abilities to locate splicing forgery. We give some examples of output from the Fals-Unet on the CASIA v1.0 [175], CASIA v2.0 [175], NIST'16 [209], and CoMoD [221] datasets. In Figs. 5, 6, 7, and 8, the first column (1) shows the manipulated images (input images), the second column (2) represents the ground-truth mask, and the last column (3) corresponds to the binary mask predicted by Fals-Unet method (output images).

TABLE V. PERFORMANCE EVALUATION OF FALS-UNET ON NIST'16, COMOD , COVERAGE , AND CASIA DATASETS

Datasets	Jaccard Index	F_1-score
NIST'16 [57]	62.56%	63.89%
CoMoD [71]	97.46%	81.56%
COVERAGE [74]	88.64%	79.52%
CASIA v1.0 [23]	92.76%	73.62%
CASIA v2.0 [23]	91.33%	69.53%

Fig. 56 represents the localization results on the CASIA v1.0 dataset. Thus, the localization results on the CASIA v2.0 dataset are illustrated in Fig. 57. We can clearly see in Figs. 56 and 57 that the predicted binary mask locates the manipulated regions from an image with a high probability.

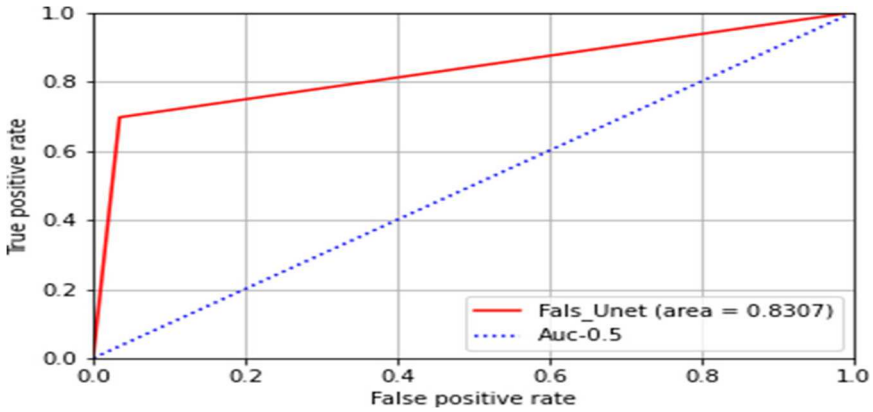

Fig. 54. ROC curve on NIST'16 dataset

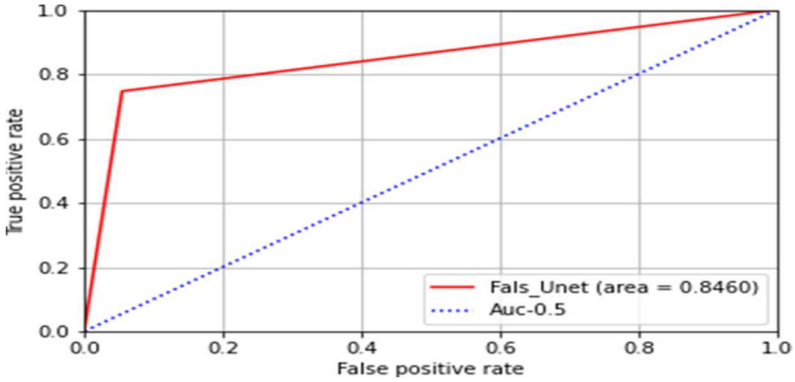

Fig. 55. ROC curve on NIST'16 dataset

TABLE VI. AUC COMPARISON AGAINST BASELINE ARCHITECTURES ON CASIA [175] AND NIST'16 [209] DATASETS

Methods	CASIA v2.0	NIST'16
UNet	0.5936	0.4321
SegNet	0.3286	0.4121
Fals-Unet	**0.8460**	**0.8307**

Fig. 58 shows that the Fals-Unet method locates the manipulated regions with great precision, especially, if it was the copy-move manipulation. From Fig. 59, we notice that the manipulated objects' boundaries are

not well predicted in the segmentation results (third column), this is justified by the fact that the boundary of images is smooth (blurred) for the dataset NIST'16. However, the Fals-Unet method always locates the manipulated regions but overlaps with respect to the truth mask.

4.5.2 Comparison with baseline methods

Since our proposed method, Fals-Unet, combines the state-of-the-art U-Net and ResNet architectures, we compare its performance to U-Net and SegNet architectures as the baseline.

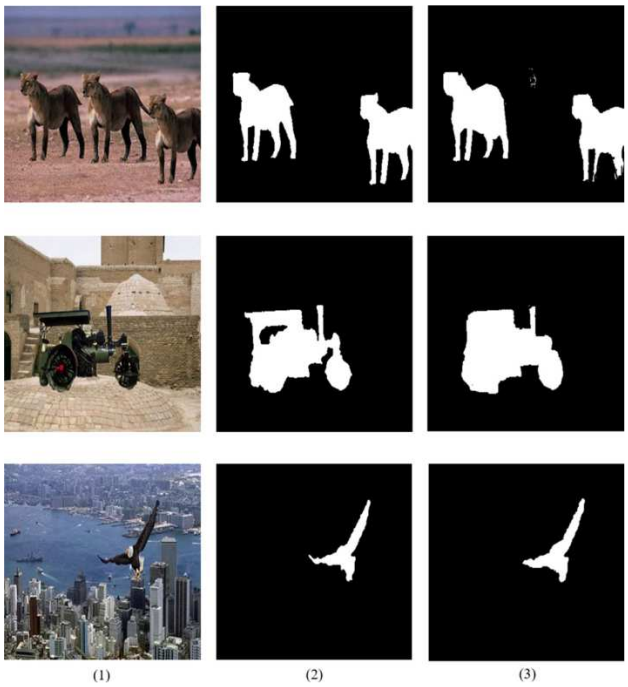

Fig. 56. Results examples obtained by False-Unet on CASIA v1.0 [175] dataset

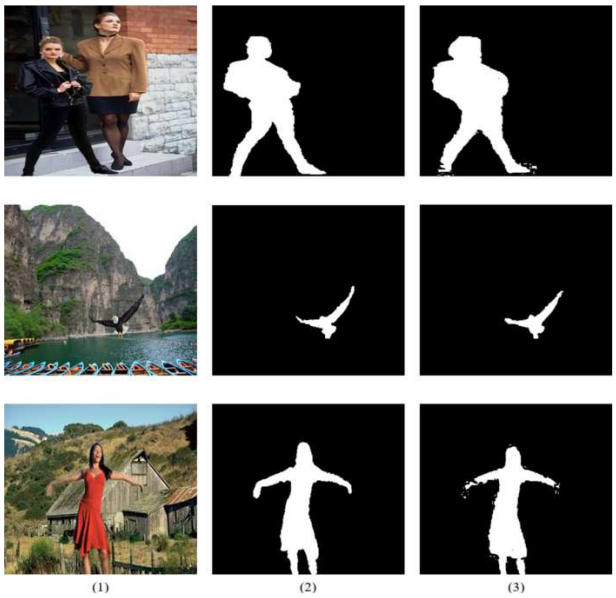

Fig. 57. Results examples obtained by False-Unet on CASIA v2.0 [175] dataset

Table VI summarizes the results obtained by the different methods on the CASIA v2.0 and NIST'16 datasets. From Table VI, we notice that the results obtained by our proposed method slightly outperform the obtained results of the other methods. Fals-Unet presents high rates of AUC equal to 84% on the CASIA dataset and 83% on the NIST'16 dataset.

4.5.3 Comparison with existing methods

To prove the robustness and the efficiency of our method, we compare the results obtained by Fals-Unet with those of the state-of-the-art classical methods such as ADQ1 [198], BLK [195], CFA1 [180], DCT [77]. These methods are provided in an accessible Matlab toolbox written by Zamboglou et al. [165]. In addition to these methods, we compare FalsUnet with deep learning methods such as Jawadul H. Bappy [157], Bappy et al. [158], Zhang et al. [79], Xiao et al. [76], and MFCN [217]. The comparative study of this contribution is carried out with the results obtained by the different approaches mentioned above on the datasets summarized in their articles. The quantitative results of the comparison are presented in Tables VII and VIII.

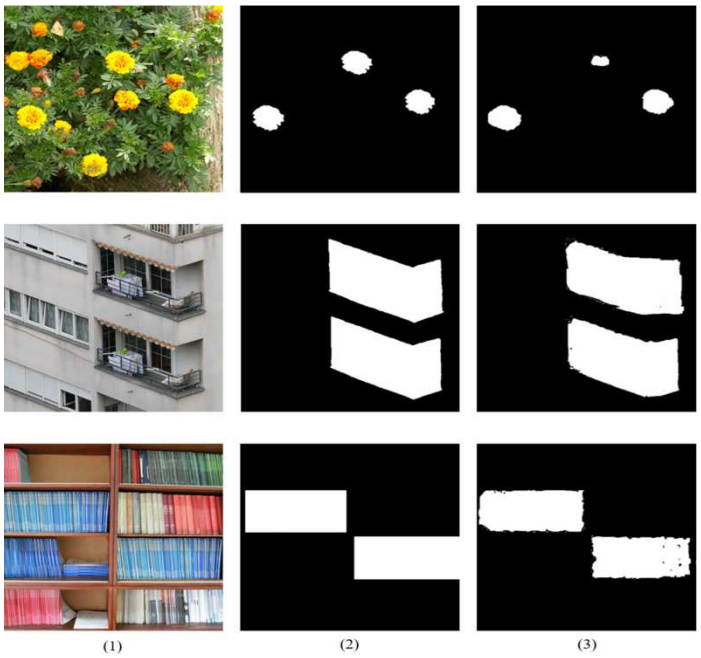

Fig. 58. Results examples obtained by False-Unet on CoMoD [221] dataset

We demonstrate that the Fals-Unet method outperforms benchmarking approach. The advantage of our model compared to MFCN BAYAR is that Fals-Unet can learn a larger context thanks to the initialization of the filter weights with weights already trained on the ImageNet dataset [65]. Further, the exploitation of spatial functionalities using the ResNet encoder helps to better learn the manipulations. In addition, we note that the Fals-Unet method gives better results compared to the other approaches (the results are highlighted in bold in Table VII).

Table VIII summarizes the obtained results by our method on the different datasets. From this table, we notice that Fals-Unet presents high rates of ACCU equal to 84% on the CASIA dataset and 83% on the NIST'16 dataset. Further, through our obtained F1-score, we are sure that both the recall and precision of the classifier indicate good results. There are explained by the efficiency of the ResNet architecture that

well locates spatial information from the falsified region, another important reason why our method gives better results is the use of the class balancing procedure, which allowed high precision classification of pixels.

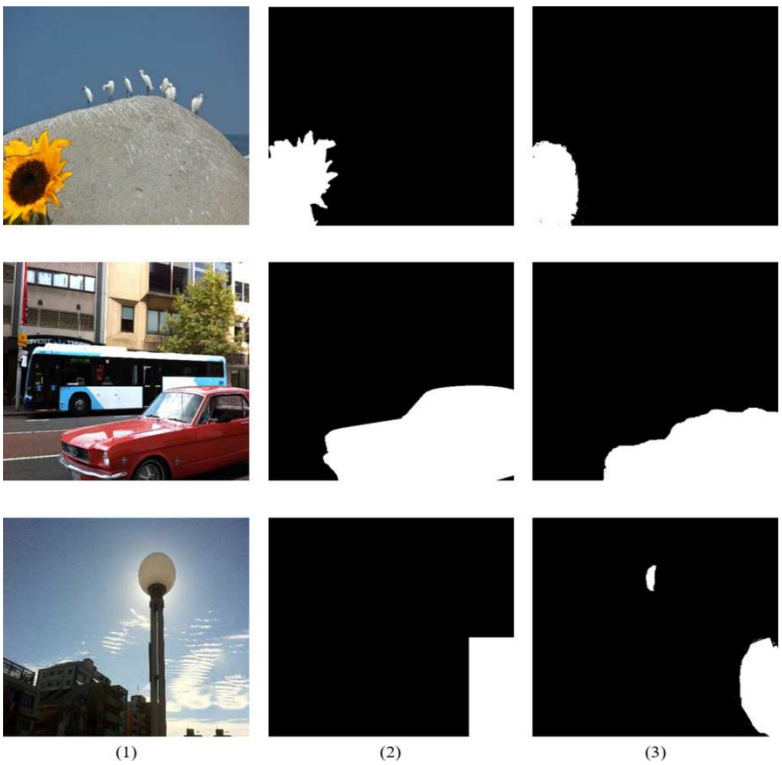

Fig. 59. Results examples obtained by False-Unet on NIST'16 [209] dataset

TABLE VII. MCC AND F1-SCORE COMPARISON AGAINST EXISTING APPROACHES ON CASIA [175] AND NIST'16 [209] DATASETS

Approaches	CASIA v1.0		CASIA v2.0	NIST'16	
	AUC	**F1-score**	**F1-score**	**AUC**	**F1-score**
ADQ	0.1262	0.2053	0.1752	0.1880	0.4220
DCT	0.2516	0.3005	0.1752	0.2600	0.5199
CFA1	0.1521	0.2073	0.2581	0.1408	0.4667
BLK	0.1769	0.2312	0.1852	0.2657	0.5234
MFCN	0.5201	0.5410	0.2316	0.5703	0.6117
FCN	NF	0.6236	0.6675	NF	NF
Bappy et al.	NF	NF	NF	0.6257	0.6242
Fals-Unet	**0.6238**	**0.7362**	**0.6953**	**0.6365**	**0.6389**

TABLE VIII. AUC AND F1-SCORE COMPARISON AGAINST RECENT APPROACHES ON CASIA [175] AND NIST'16 [209] DATASETS

Approaches	CASIA v1.0		NIST'16	
	AUC	**F1-score**	**AUC**	**F1-score**
Zhang et al. [79]	---	0.5722	---	0.5140
Xiao et al. [76]	---	0.6758	---	---
Bappy et al. [158]	---	---	0.7936	---
Fals-Unet	**0.8325**	**0.7362**	**0.8463**	**0.6389**

4.6 Conclusion

In this chapter, we present an approach called Fals-Unet based on deep learning for the manipulated regions' localization in a falsified image. More precisely, we use a network of the U-Net family as encoder/decoder, the encoder used is identical to that of the ResNet method, and its aim is to provide maps of spatial characteristics of manipulated objects. On the other hand, the decoder's objective is to learn the mapping of spatial maps to generate the binary mask. From the obtained simulations results, we prove that Fals-Unet presents a real solution of the manipulated regions localization in terms of F1-score, MCC, AUC, and Jaccard index thanks to the initialization of the encoder weights by weight values already trained on a large dataset, which optimizes the network training time and increases the detecting precision. Extensive experiments on NIST'16, CASIA, COVERAGE, and COMOD datasets have demonstrated the superiority of our Fals-Unet has exceeded other methods by a large margin on the manipulated regions localization and has obtained very competitive results on the three images manipulation types (copy-move, objects removing and splicing). Although our proposed method has demonstrated excellent performance, it still has some shortcomings. For example, our approach can only process images of fixed size. It also presents a high computational complexity. Additionally, there are still a few poorly detected images, especially on the NIST'16 dataset. Future work will consider improving the Fals-Unet method to process different size images, thereby reducing its computational complexity. CNN networks have seen great interest in detecting falsified images due to their efficiency in image processing. Despite their advantages, these networks risk the problem of overfitting on limited volumes of data due to their high number of parameters. On the other hand, the manipulated regions can be small regions and the difficulty of their localization has limited the exploitation of these techniques for the analysis of these images. Several works have suggested the use of the transferred learning technique on the one hand. It allows transferring knowledge from a model trained on the large ImageNet learning base to a new model. Then, the new model is readjusted for the new detection task. The high amounts of data belonging to the ImageNet database and many categories have made this database a good tool for transferred learning. On the other hand, the hierarchical nature of DNNs favors the exploitation of the transferred learning technique over other classical machine learning methods.

Chapter IV: Efficient balanced focal loss function for manipulated images detection

We mentioned earlier that loss functions play an important role in the learning process of deep neural networks, especially the problem that suffers from a strong bias in the data, while the inappropriate choice of a loss function can cause an over-fitting problem. While designing an appropriate loss function can be beneficial in learning. In the last years, BCE is the most used loss function for semantic segmentation tasks. However, it suffers from several drawbacks. In this chapter, we focus on the loss function for semantic segmentation. After carefully analyzing the disadvantages of BCE for segmentation, we propose a loss function that solves these problems. Inspired by the focal loss, we propose the balance factor based on a focal loss function to tackle this data imbalance problem. Through extensive experimental validation on CASIA1 and CASIA2 datasets, we show that the method consistently improves segmentation accuracy with negligible additional computational overhead and can be added to all popular segmentation networks in a training framework end to end. The work of this chapter is accepted in the conference

Contents

5.1 Introduction

In this chapter, we address the problem of falsified image segmentation from a DL perspective and mainly address the problem of imbalanced classes. Unbalanced classes usually refer to a classification problem where the ratio of observations in a class to all observations is very low. This class imbalance clearly increases learning difficulty and introduces strongly biased predictions in favor of high precision but low recall. Most research has focused on the data imbalance problem, mainly by modifying the network structure and introducing different modules to make better use of spatial and contextual information. Instead, we believe that the network can also learn spatial and contextual information through an appropriate loss function. Current state-of-the-art methods mainly rely on end-to-end architectures (Long et al. 2015) that are trained by optimizing a loss per pixel between predictions and ground truth labels. Neural networks in end-to-end networks intelligently inherit the idea of classification from convolutional neural networks (CNNs) and classify each pixel. From these methods, popular classification networks can be adapted into a convolutional network and their learned representations can be transferred to segmentation

tasks by training them with loss of cross-entropy per pixel (BCE). This loss is a natural fit as a spatial extension of the cross-entropy loss, which is the classification standard. However, BCE loss has some drawbacks when applied to semantic segmentation. In this contribution, we propose a new function of loss expected from the focal loss to put an end to this problem.
The chapter is organized as follows. We start with a presentation of the work in the state of the art in section 5.1. Then, in the section, we address the function problem in a specific learning framework of the falsified image segmentation problem that uses only image-level labelled images to train the model segmentation. Thereafter, the experimental results are presented in section 5.3. And we end with a conclusion

5.2 Related work:

Deep convolutional neural networks have shown superior performance in tampered image detection compared to classical methods. In the literature, several methods have been proposed in this context such as [159, 160, 167,204, 161], these studies use models based on deep learning for tampered image detecting. These works include the detection of generic manipulations [159, 160], oversampling [215], splicing [161], and bootleg [166]. In [213], the authors proposed Gaussian-Neuron CNN (GNCNN) for the steg-analysis. Deep learning has a great effect in terms of improving performance in different visual recognition tasks such as identification [168], semantic segmentation [233], and classification [232] of images. Deep Learning consists of extracting hierarchical characteristics to represent the visual concept. Most of the architectures are based on CNN for generating relevant spatial maps to the manipulated regions. A CNN-based forensic analysis framework is proposed by the authors of [197] for detecting image operator's chains. They used the two-stream CNN architecture; they apply the first stream to extract the local residual features, while the second stream is used to detect the tampered artifacts. In [183] the authors proposed a theoretical framework of information for detecting the tampering operations, such as contrast enhancement, resizing, and median filtering in multiple chains. These approaches quantify the operation identification using the conditional probability criteria. Usually, the class imbalance problem means the difference in the data amount between different classes. This problem makes the model trained with an unbalanced dataset biased in favor of the majority classes. To deal with this problem, it is always necessary to use a correction mechanism capable of balancing the different classes. In general, there are two types of solutions to avoid the bias effect: Algorithm-level and data level [234].
The most efficient solution at the data level is based on the resampling of training data either by sub-sampling the majority classes or over-sampling the minority classes or a combination of both [235], [236], [237]. This type of solution changes the underlying data distributions, leads to increased computation, and can present the risk of overfitting. Another solution for resampling training data is the classifier set, where each classifier is induced by different samples from the original dataset [238], [239]. To increase the dataset for model training [240, 241]. This type of solution uses the Generative Adversarial Network or other techniques such as iterative sampling [242] and incremental rectification of mini-batches [234] to learn the distribution of the dataset and better train the model. Unlike data-based solutions, algorithm-based solutions introduce a weight for each class or each sample. This type of solution puts more emphasis on minority classes. Moreover, it consists in defining the weight parameter as a class hyper parameter [247]. Usually, hyper parameters are defined as cost-sensitive matrices based on training parameters [245], [246], or statistics [243], [244]. However, the model becomes more complex and the cost of training increases considerably [234].

5.3 Focal Loss method

5.3.1 Problem statement

The classes unbalance is considered the major problem of the falsified image segmentation because the distribution of the manipulated/non-manipulated regions is strongly asymmetric (the manipulated regions vary between [0.001%, 35%] of the manipulated image pixels, the non-manipulated regions vary between [0%, 100%]). This problem leads the network to converge towards local minima, also it generates probability maps strongly biasing in favor of non-manipulated regions, which negatively influences the network learning. To deal with this problem, we use an LBF balancing loss function inspired by the focal loss function [247] with two penalty factors α and σ to control the FP and FN. Generally, the focal loss function is used for segmenting widely unbalanced data; its efficiency focuses on the down-weighting of the well-classified example and thus enabling learning of harder examples.

$$LF = \alpha(1 - p_t)log(p_t) \tag{46}$$

Where p_t is the prediction probability obtained by applying Softmax function (Eq), p_t is defined as bellow:

$$p_t = \begin{cases} p, & y = 1 \\ 1 - p, & y = 0 \end{cases} \tag{47}$$

γ is the degree of down-weighting of easy samples, α is the control parameter of the class weights, and y is the ground truth mask. The main idea of our method is to establish a relation between class imbalance and hard samples; we define this relation as follows:

$$L_{BF}(p) = -\alpha_t(1 - p_t)^{1/\sigma}log(p_t) \tag{48}$$

$$\alpha_t = \begin{cases} {}^{w_m}/_{w_n}, & y = 1 \\ 1, & otherwise \end{cases} \tag{49}$$

Wherew_m and w_n are respectively the pixel weights of the manipulated and non-manipulated regions, which means that the loss of the small-manipulated region will be amplified, and the loss will focus on this class. α_t is calculated from the dataset. σ is the restricted amplification factor, it is greater than 1, its goal is like γ in FL. Generally, we can say that the more difficult the classification is, the more the σ is greater. As opposed to FL [247], which has two parameters α_t and γ in the loss function, to adjust, L_{BF} has only one parameter. L_{BF} allows precise amplification of small class loss and simultaneously considers class unbalance and hard sample, which avoids over-amplification of small class loss.

5.3.2 Architecture

To test and evaluate the performance of our loss function we use the U-Net [21] architecture which is characterized by a fully convolutional image-to-image network. The U-Net architecture is composed of two paths, a contraction path, and an expansion path (see figure 60). The strong point of this architecture is the concatenation of the high-resolution features with the overall low-resolution features. Thanks to this concatenation, the network will be ready to learn and use the local information.

The U-Net contraction path consists of four residual blocks already performed on the Image-Net dataset [215]. Each residual block is formed by two blocks (Conv block and identity block) and expressed by N(I) + I, where I is the input vector, and N(.) represents the input mapping I at the residual unit outlet. These residual blocks allow us to improve the precision by increasing the depth of the network, preserving the

local information, avoiding the disappearance/explosion gradient, and optimizing the formation of the layers. In the expanding path of U-Net we apply the transposed convolution operation to generate the resulting feature map, which will then be concatenated to the corresponding feature map from the contraction path. In the final layer, we use convolution with the function of Softmax to generate the feature map having a depth equal to the number of classes.

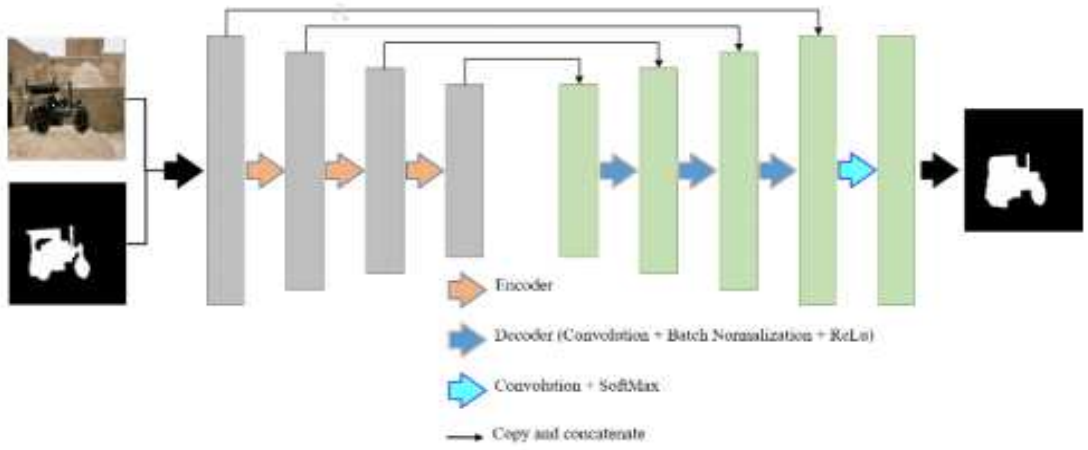

Fig. 60. The overview of the used framework

5.4 Experiments

In this section, we provide a detailed experimental analysis, thus the qualitative and quantitative results of the proposed method.

5.4.1 Implementation details:

The proposed solution is implemented in the Keras simulation environment with the TensorFlow backend and is executed on an infrastructure equipped with an NVIDIA Tesla P100 GPU card and 16 GB of RAM memory. Dataset is augmented with scaling, random flips, and crops during the fine-tuning process. We trained our network using the Adam optimizer [248], with a learning rate of 0.00001, and a mini-batch size of 32. Concerning the loss function, we empirically define its hyper parameters, after several experiments, we fixed the values of the hyper-parameters of the balanced focal loss function at $\alpha = 0.25$ and $\sigma = 4.5$.

5.4.2 Datasets

To evaluate our method, we use the CASIA dataset [175] (CASIA v1.0 and CASIA v2.0). We generate the ground-truth mask based on the information provided by the authentic images. We divide the dataset into two sets: training (80%), and testing (20%). To assess the degree of imbalance classes, we calculated the percentage of pixels of the manipulated region and the non-manipulated regions for datasets see Table. IX.

TABLE IX. PERCENTAGE OF PIXELS OF THE MANIPULATED/NON-MANIPULATED ON CASIA DATASETS

Datasets	Total of images	Manipulated	Non-Manipulated

CASIA v1.0	921	2.3%	97.7%
CASIA v2.0	5123	3.5%	96.5%

5.4.3 Comparison with baseline methods

In the first part of this section, we vary the σ parameter to find an optimal value that allows precise detection of the manipulated regions, Table. X shows that the optimal value of σ is 4.5.

TABLE X. DIFFERENT VALUES OF THE Σ PARAMETER

U-Net	Accuracy
L_{BF} σ = 1.5	69.74%
L_{BF} σ = 2.0	58.71%
L_{BF} σ = 4.5	**84.54%**
L_{BF} σ = 5.0	64.63%

Then, we compare the performances obtained by the proposed loss function with those of some reference methods such as CE [250], BCE [251], and FL [247]. From Table. XI, we prove that the proposed method surpasses the other methods in terms of F-Measure.

TABLE XI. COMPARISON OF LOSS FUNCTION F-MEASURE

Methods	F-Measure
CE [250]	0.6902
BCE [251]	0.6856
LF [247]	0.7194
L_{BF}	**0.7397**

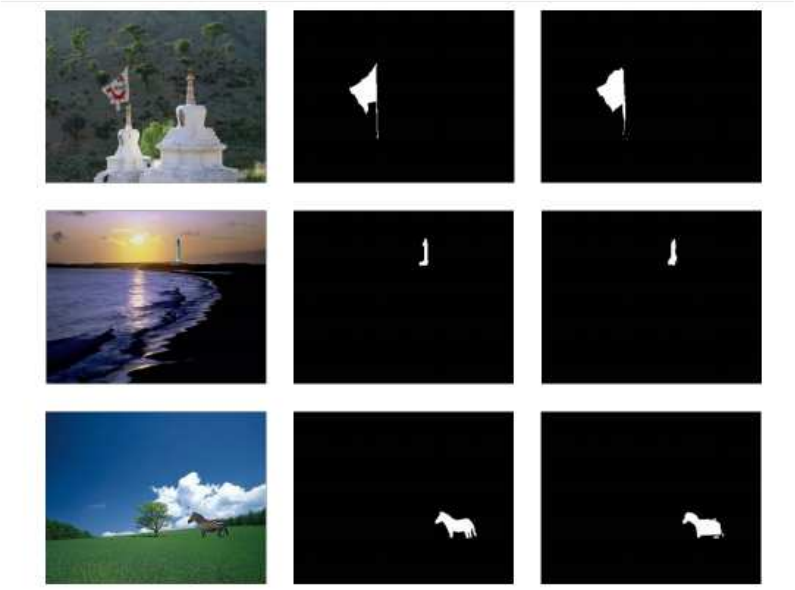

Fig. 61. Result examples obtained on CASIA datasets

Figure 61 shows the qualitative results of our method on the CASIA dataset, the first column (1) represents the image manipulated, the second column (2) shows the ground-truth mask, and the last column

(3) corresponds to the predicted mask. These qualitative results prove the efficiency and the robustness of our loss function. Our proposed method allows a good to distinguish the pixels of the manipulated region from the pixels of the non-manipulated region and obtain from a clear segmentation the small regions manipulated.

5.4.4 Comparison with existing methods

Several solutions have been proposed in the context of manipulated image detection. To prove the robustness of our method we carried out a comparative study with some methods such as Xiao B et al. [226], Salloum et al. [217], and Zhang et al. [252]. The results presented in Table.XII show that our method is more efficient in terms of F-Measure and Mcc, the proposed method provides a clear segmentation of the manipulated regions, its F-Measure is 73.94% greater than Xiao B of 6.36%, MFCN of 19.53%, and Zhang of 16.72%.

TABLE XII. COMPARISON OF OUR METHOD WITH EXISTING METHODS

Methods	F-Measure	Mcc
MFCN [217]	0.5410	0.5201
Zhang [252]	0.5722	---
Xia B [226]	0.6758	---
OURS (L_{BF})	**0.7397**	**0.6421**

5.5 Conclusion

The most notable problem for the segmentation of the manipulated images is the problem of the imbalanced pixels classification. To solve this problem, we propose a function of focal loss based on the weight factor w_n and w_m to amplify the small-manipulated regions, we introduced the σ restriction factor to control the amplification of these regions. The experimental results on the CASIA dataset show the robustness of our method for the segmentation of the manipulated regions, especially the localization of the small, manipulated regions. Our future work is to extend the idea of this article to process different size images, thereby reducing its computational complexity. Through this work, we believe that our proposed method is a good solution that improves the detection of manipulated images and can be used in different contexts such as classification, segmentation, etc.

Chapter VII: CovSeg-Unet: End-to-End method-based computer-aided decision support system in lung covid-19 detection on CT images

In this chapter, we address the problem of biomedical image segmentation, in which we need to ensure a robust system that can help specialists in the diagnosis of COVID-19 infections. To face this challenge, we propose a method based on a deep neural network to process and analyze chest X-ray or CT scan images for the diagnosis of COVID-19. This approach is mainly to identify the regions of interest in these images, such as affected areas or lesions by infections in the bronchi-pulmonary parts, lungs, and lobes for further quantification and evaluation.

The chapter is organized as follows: After having introduced the segmentation of the biomedical image by deep learning. The proposed method is presented in section 6.2. Section 6.3 presents the simulation environment, the evaluation metrics, and the database. While Section 6.4 shows the experimental results. Finally, we end this chapter with a conclusion in section 6.5.

Contents

6.1 Introduction:

Deep learning methods are characterized by their ability to extract features compared to classical machine learning algorithms. This encourages the scientific community to propose various segmentation methods based on deep learning networks to process and analyze chest X-ray or CT scan images for the diagnosis of COVID-19 [261]. These methods mainly consist of identifying regions of interest in these images, such as lobes, bronchi-pulmonary parts, lungs, affected areas, or lesions for subsequent quantification and evaluation. However, these methods have a problem of over-fitting on limited volumes of medical data. In addition, they are characterized by their high variance and low bias. The main objective of this contribution is to reduce these problems by optimizing the trained models in terms of generalization. In this context, we have proposed a framework based on different regularization techniques: image pre-processing, CovSeg-UNet learning models, selection of the appropriate optimization method in learning (Adam, SGD), and the proposal of the loss function appropriate. As part of the pre-processing phase, we have proposed a signal normalization method that processes the intensity of each CT scan voxel. In this step, we use the lung window to separate the lungs from other organs. Since each CT scanner has its own Hounsfield Unit (HU), therefore data collected from different hospitals will have different HUs. For this reason, we use a multi-valued window (WL is the middle value of the window and WW is the width of the window). Then we exploited a static selection strategy (best and last models) to select the models to be combined from several learning points. Next, we combined the models selected by the voting and un-

weighted average techniques. To evaluate the proposed framework, we tested it on the CT COVID-19 Learning Base. The results obtained confirm the efficiency of the CovSeg-UNet model based on the Res-Net architecture and the proposed loss function. The best results were obtained by combining the two best models U-Net and ResNet. In summary, these results prove the efficiency of the technique of combining models generated from several learning points, to localize regions infected by covid-9 in CT images.
The experiments were performed on an infrastructure equipped with a Tesla P-100 GPU card and 16 GB RAM and the Keras deep learning library.

6.1.1 Problem statement:

In December 2019, a viral pneumonia epidemic of unknown etiology emerged in Wuhan city, Hubei province, China [1]. On January 9, 2020, the World Health Organization (WHO) and Chinese Health Authorities officially announced the discovery of a new coronavirus. This pneumonia is an infectious disease caused by a virus identified under the name SARS-CoV-2 (Severe Acute Respiratory Syndrome CoronaVirus-2) by the ICTV (International Committee on Taxonomy of Viruses) [254], and causing a disease called COVID-19 (Corona Virus Disease 2019). SARS-CoV-2 belongs to the coronavirus family. The reservoir of this virus is animals. Although SARS-CoV-2 closely resembles a virus detected in a bat, the animal that transmits it to humans has yet to be identified with certainty. Several research studies suggest that the pangolin, a small mammal eaten in southern China, could be involved as an intermediate host between bats and humans. The new coronavirus has been confirmed to be transmitted between humans [255], and this is done by air or by close contact with a contagious subject. Smaller particles can also be emitted in the form of aerosols during speech or during coughing efforts, which would explain that the virus could persist suspended in the air in an unventilated room. Finally, the virus can retain infectivity for a few hours on inert surfaces from where it can be transported by the hands. According to data from the World Health Organization, updated up to 24 hours on June 18, 2021, COVID-19 has affected 220 countries and territories, causing 178,584,744 people to be infected and 3,866,607 deaths worldwide. The overall number of people recovered is 163,102,134. Currently, the active cases are 11,616,003 of which 99.3% are in mild condition and 0.7% are in serious or critical condition, which poses a great threat to international human health. Due to the vaccination process's slowness, the high rate of virus contamination, and the appearance of new dangerous COVID-19 mutations, it is essential to detect and identify the disease at an early stage so that suspected patients do not infect the healthy population. As a result, new requirements for the prevention and control strategy must be put in place. Reverse Transcription Polymerase Chain Reaction (RT-PCR), gene sequencing for respiratory, or blood samples confirm the diagnosis of COVID-19. However, the false negatives of the RT-PCR [256], the delay in obtaining the results, and the tests carried out on people not strongly suspected of being infected with COVID-19 imply that numerous COVID-19 patients would not be identified quickly to isolate them from others. In addition, given the rapid and contagious spread of the virus, they present a real threat of infecting a larger population, especially in areas with high epidemics. On the other hand, chest examinations quickly established themselves as an interesting diagnostic tool, given the characteristic presentation of COVID-19 lesions [257]. These tests can identify lesions, underlying conditions, and complications associated with acute airway conditions. Consequently, the use of CT in particular high resolution computed tomography (HRCT) could provide enormous help to radiologists [259] for the diagnosis, follow-up, or investigation of pulmonary complications in patients suspected or confirmed

of COVID-19. Thus, the development of artificial intelligence (AI) method based on deep learning could help them enormously to assess the degree of lung damage caused by COVID-19.
According to [258], the authors indicated that CT images can be used to detect COVID-19 even before certain clinical symptoms are observed. Typical signs of COVID-19 appear in CT images as unilateral, multifocal, and terminal Ground Glass Opacity (GGO). Which is a hazy cloud above the lungs that indicates a variety of problems, and may mean that the lungs are partially filled with inflamed material, and there is thickening in lung tissue or partial breakdown of the alveoli and tiny air sacs of the lungs? Pleural effusion, lymphadenopathy, and condensation [255], which are air spaces in the lungs filled with a substance, usually pus, blood, or water, surrounded by an opaque edge of frosted glass, and although this is a common feature of lung disease, it may be more characteristic of COVID-19. To detect COVID-19 disease at an early stage, it is necessary to detect and locate these pathological changes in a short time. The growing number of patients and the limited number of well-trained expert radiologists in most hospitals prevent and slow down the process of early detection. Indeed, the use of deep learning methods for the automatic segmentation of the COVID-19 CT model has become paramount and may offer an effective solution to identify and locate signs of COVID-19 in CT images [260].

6.1.2 Motivation:

Our main objective through this contribution is to diagnose COVID-19 lung infection in chest CT scan images. In this report, we propose architecture like that of the U-Net approach [275] called the CovSeg-Unet method; U-Net has known a great success for the segmentation of medical images. U-Net is a symmetric encoder/decoder architecture composed of several stages. Each stage of the encoder performs a set of operations such as convolution, normalization, maximum pooling, activation, concatenation, etc. In parallel, each stage of the decoder performs de-convolution operations. The U-Net method uses jump connections allowing the exploitation of local and global information. These connections concatenate the under-sampling characteristics of the contraction path with those of the oversampling of the expansion path.

6.2 State-of-the-art of proposed methods

In the literature, numerous methods of segmentation based on deep learning networks have been used to process and analyze chest X-ray or CT images for the COVID-19 diagnosis [261]. These methods mainly consist of delineating the regions of interest in these images, such as lobes, broncho-pulmonary segments, lung, and infected regions or lesions for further quantification and evaluation. CT provides detailed and high-definition three-dimensional images to detect COVID-19. Among the segmentation methods used for the diagnosis of COVID-19 we cite, U-Net [275], UNet++ [262], V-Net [263]. The authors of [261] have proposed a 3-D architecture of U-Net using inter-slice information; this method consists in replacing the conventional layers of U-Net with a 3-D version. In [263], the authors proposed the VNet architecture, in which they used the residual blocks as the basic convolutional block and optimized the network by a loss of dice. In [264], the authors proposed the Attention U-Net method, which captures fine structures to locate lesions and pulmonary nodules in medical images. Generally, the large number of well-labelled images is the key to forming an efficient and robust segmentation network. In the case of COVID-19 image segmentation, the data used during the training phase is limited and often unavailable because manual lesion delineation is a difficult operation and requires a lot of time. Several other research works obtaining

reasonable segmentation results have been proposed in this context. The lung segmentation field is experiencing a lack of labelled medical images, as a result, semi-supervised and unsupervised methods are very favorable and recommended in studies on COVID-19, as in [275], the authors used an unsupervised method to generate pseudo- segmentation masks for the images. In [265], the authors proposed a new COVID-19 lung infection segmentation network called Inf-Net to detect infected regions from chest CT images. This method uses a parallel decoder for aggregating high-level features and generating a global map. Then, it uses a semi-supervised segmentation framework based on a propagation strategy chosen at random to overcome the lack of labelled data. In [266], the authors proposed a computer-assisted diagnosis (CAD) system based on the YOLO predictor to detect and diagnose COVID-19. The CAD method calls the data balancing regularizations, augmentation, and transfer learning to improve the overall diagnostic performance for COVID-19. The authors of [267] proposed a synergistic approach based on deep meta-learning to accelerate the detection of COVID-19 cases. This approach uses contrastive learning with a pre-trained ConvNet encoder for the classification of COVID-19 cases. In [268], the authors proposed a computer-aided detection (CAD) method to assist radiologists in automatically detecting COVID-19 on the chest X-ray images. The proposed method uses the DLs: the Discrimination-DL to extract lung features from chest X-ray images, and the Localization-DL to localize and assign the infected lung region. In [269], the authors-built prognosis models to predict the patients' severity outcomes. The proposed method is based on deep learning in the CT image segmentation process for COVID-19 pneumonia, and it uses datasets from multiple institutions worldwide to validate the proposed models. In [270], the authors propose a CovFrameNet framework to detect COVID-19 cases using CT images, which incorporates an image pre-processing mechanism and a deep learning model for feature extraction, classification, denoising, smoothing, and performance measurement.

6.3 The proposed method:

The general idea of the CovSeg-Unet approach is described as follows. Let S be a learning space that contains a set of n images $X = X_1, \ldots, X_n$ and n corresponding ground truth masks $Y = Y_1, \ldots, Y_n$. From the Y ground truth masks, the network learns the lung infection distribution of the X training images to establish an image-to-image mapping relationship between X and Y, this map is defined as:

$$\Phi = f_{dec} \circ f_{enc} \tag{50}$$

Where f_{enc} denotes the function encoder which learns characteristic vectors of infected lung regions to establish a functional space, f_{dec} denote the decoder function, which learns the spatial location of features to better locate the infected/uninfected region, and Φ is the set of probabilities extracted from an input biomedical image and is represented by $\Phi(f) = P_1, P_2, \ldots, P_n$, where $P_i = \{0, 1\}$ are the probabilities given to the last convolutional layer of classification by the nonlinear function SoftMax.

The main idea here concerns the spatio-temporal modelling of characteristic points that can represent pulmonary infections. The architecture of the CovSeg-UNet approach is shown in Figure 62. Generally, the architecture of the CovSeg-UNet approach is composed of two blocks; the first is to pre-process the CT images. While the second performs all encoder/decoder operations to learn high-level features from training sets and localize spatial information. Details of the CovSeg-UNet network are shown below.

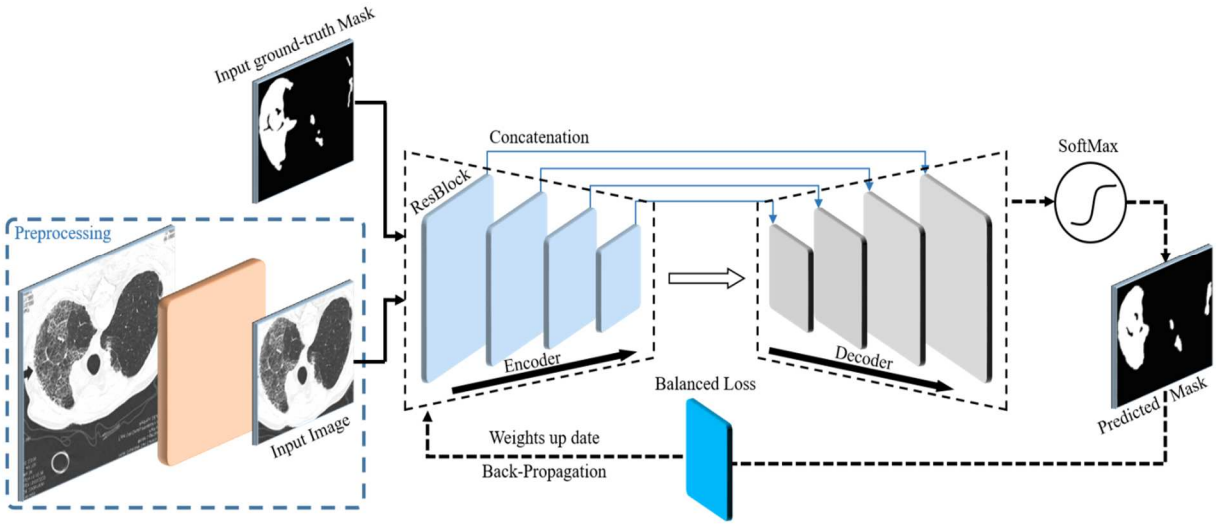

Fig. 62. Overview of the proposed framework for diagnosing COVID-19 from CT images.

6.3.1 Pre-processing:

The major challenge of biomedical image segmentation with DL is the processing of biomedical images coming from several machines with different acquisition parameters. To face this challenge, we apply a pre-processing step to these images to improve the training of the COVID-19 suspect recognition model despite the heterogeneity of the data. This step of pre-processing consists of two steps; the first concerns the normalization of the signal, which processes the intensity of each CT scanner voxel. In this step, we use the lung window to separate the lungs from other organs. Since each CT scanner has its own Hounsfield Unit (HU), therefore data collected from different hospitals will have different HUs. For this reason, we use a multi-valued window (WL is the middle value of the window and WW is the width of the window), the WL value is randomly assigned from -600 to -500, while the WW value is set to 1200. In the second pre-processing step, we normalize the CT images to be within [0.255]. Through this process, we normalize multiple images from different CT scans to the same standard, separating the lungs from other organs, removing unnecessary information (CT features, etc.) or/and noise, enhancing the images and enhancing the precision.

TABLE XIII. DIFFERENTS HOUNSFIELD VALUES OF DIFFERENT SUBSTANCE

Substance	Hounsfield Unit (HU)
Air	-1000
Bone	+700 to +3000
Lungs	-500
Water	0
Kidney	30
Blood	+30 to +45
Grey matter	+37 to +45
liver	+40 to +60

White	matter +20 to +30
Muscle	+10 to +40
Soft Tissue	+100 to +300
Fat	-100 to -50
Cerebrospinal fluid(csf)	15

6.3.2 CovSeg-UNet architecture:

The main objective of our contribution is to detect COVID-19 lung infections using CT images, to this end we propose the CovSeg-UNet approach, which is characterized by an end-to-end architecture based on one of the most robust approaches in biomedical image segmentation that is U-Net. The network of the CovSeg-UNet approach is supplied as input by pre-processed CT images and their ground-truth masks. Our proposed method relies on an encoder to extract contextual feature maps from pre-processed images to reduce the dimensions of CT images, and a decoder to locate feature map information in the image. Table XIV shows the detailed architecture of our encoder/decoder.

Generally, the encoder is made up of four residual blocks (ResBlock), already pre-formed on the ImageNet database, the use of these blocks allows us to avoid the disappearance/explosion gradient, preserve the local information, to improve precision by increasing the depth of the network, and to optimize the formation of layers. Each residual block is made up of two blocks (Conv block and identity block) and is expressed by R (I) + I, where I is the input vector and R (.) represents the mapping from input I to the output of the residual unit. However, the decoder has two inputs, one input from the parallel layer of the encoder and a second from the previous layer of the decoder. Finally, the decoder output is sent to the SoftMax activation function (see equation. (51)) for predicting the region infected with COVID-19. Algorithm. 1 describes the training steps of the proposed model.

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{51}$$

Where $\vec{Z}$ is the input vector of the Softmax function ($z_0, \ldots, z_K$), all z_i values can take any real value. E Z_i is applied to have a positive value for each element of the input vector. The term at the bottom of the formula is the normalization term. This makes it possible to have a sum = 1 of all the output values (are each in the range (0, 1)) of the function, thus constituting a valid probability distribution, k represents the number of classes in the multiclass classifier.

TABLE XIV. DETAILLED ARCHITECTURE OF THE PROPOSED METHOD. WHERE THE ENCODER IS THE RESIDUELLE BLOCK SIMILAIRE TO RESNET50[271] ARCHITECTURE, AND THE DECODER BLOCK IS A SUCCESSION OF BATCHNORMALISATION, RELU, CONV2-D, BATCHNORMALISATION, AND RELU OPERATIONS.

Stages	Layers	Output size
Initial	Conv2-D (7 × 7), stride = 2 BatchNormalisation ReLu MaxPoling	(64 × 64 × 64)
Encoder	Stage 1: ×2 ResBlock(64 × 64 × 256) Stage 2: ×3 ResBlock(128 × 128 × 512) Stage 3: ×5 ResBlock(256 × 256 × 1024) Stage 4: ×2 ResBlock(512 × 512 × 2048)	(64 x 64 x 256) (32 x 32 x 512) (16 x16 x 1024) (8 x 8 x 2048)

Decoder	Stage 1: UpSampling(16 × 16 × 256)	(32 × 32 ×128)
	Stage 2: UpSampling(32 × 32 × 128)	(64 × 64 × 64)
	Stage 3: UpSampling(64 × 64 × 64)	(128 × 128 × 32)
	Stage 4: UpSampling(128 × 128 × 32)	(256 × 256 ×16)
Final	Conv2-D (256 × 256 × 16)	(256 × 256 × 1)
	Softmax Activation (256 × 256 × 1)	(256 × 256 × 1)

Algorithm 1: Training procedure for CovSeg-UNet method

input : X: Image CT; Y: Label; N: Batch-size; λ_1; λ_2;
Lr: Learning-rate; W_0: Initial weights;
output: P_L: Predicted mask;

begin
  $(X_{train}, Y_{train},)(X_{valid}, Y_{valid}) \leftarrow$ split((X, Y), split-size=0.2)
  while epoch ≤ 200 **do**
    for mini batch sample $x_k\{x_{train}, x_{valid}\}_{k=1}^{N}$ **do**
      $z_1 = f_{encoder}(x_k)$
      $z_2 = f_{decoder}(z_1)$
      predict(z_2) with SoftMax equation
      Compute Δ_w the stochastic gradient by minimizing the loss function eq. 2
      Update weights
      $w \leftarrow w + \alpha.AdamOptimiser(w, \Delta_w)$

6.3.3 Loss

The imbalanced class problem is considered the major challenge in the detection process of lung infections because the distribution of infected/uninfected regions is highly skewed (the infected regions vary between 0% and 20% of the pixels of the lung image). Therefore, if the loss function does not consider this problem, the model will classify most pixels as uninfected regions, and become over-fit. For this reason, we have proposed in our contribution a class-balanced cross-loss function with the penalty factor λ to avoid over-fitting problems on unbalanced volumes of medical data.

The loss function is defined as a weighted sum of two-loss functions; the balanced cross loss L_{BCE} and the inverse cross loss L_{ice}:

$$L_B = \lambda_1 \cdot L_{BCE} + \lambda_2 \cdot L_{ICE} \tag{52}$$

Respecting the cross-loss, we use the balanced cross-entropy to overcome noise in biomedical images; we also use the balance parameter W to balance the pixel distribution of infected/uninfected regions in L_{BCE}:

$$L_{BCE} = -w_c \sum_{i \in y_c} q(y_i = 1 \mid s) \log(p(y_i = 1 \mid s) \quad -w_{cn} \sum_{i \in y_{nc}} q(y_i = 0 \mid s)\, log(p(y_i = 0 \mid s) \tag{53}$$

Where $q(y_i = i \mid s)$ the ground-truth mask of the sample is S. $Y_i = 0, 1$, $p(y \mid s)$ is the probability map produced by the function softmax, w represents the balancing parameter $w(k) = S/KS(k)$. S is the sample number in the training set. And S(K) represents the number of samples in class K. However, cross-entropy relies heavily on the accuracy of the annotation. When the data is mislabeled, $q(k \mid s)$ will not be able to represent the true class distribution, which will lead the cross-entropy $p(k \mid x)$ to learn this incorrect

distribution type. To deal with this problem, we use reverse learning to know which classes the input x does not belong to. Inverse cross-entropy is defined as follows:

$$L_{ICE} = \sum_{i=1}^{k} p(y_i = 1 \mid s) log\, q(y_i = 1 \mid s) \quad (54)$$

The weighted combination of entropies (L_{ICE} and L_{BCE}) in the loss function allowed a good convergence of the gradient and relevant learning. The L_{BCE} term efficiency exhibits in the distribution balance of infected and uninfected classes, while the L_{ICE} term strength appears in the resistance against noise caused by scanner settings. As a result, the balanced symmetric entropy function obtained high performance in the COVID-19 lung infections localization on the test data set.

6.4 Experimental study

We explain in this part, the different experiments carried out to evaluate the performance of the proposed contribution in different simulation scenarios. We start by describing the dataset used and the experimental environment of the simulation. Subsequently, we perform a quantitative study of the hyper-parameters to show their effects on model learning. Finally, we present the different performance measures used to evaluate the efficiency and robustness of our approach.

6.4.1 Datasets:

In this contribution, we have applied our method on two Datasets:

Dataset-1: this dataset contains two versions [272]. The first version is published on April 2, 2020, comprising 100 CT images (of 40 covid-19 patients) of size 512 × 512 labelled by radiologists, these radiologists have defined three tags: pleural effusion in frosted glass (= 3), consolidation (= 2), and mask value (= 1). The second version is released on April 14, 2020, comprising 829 CT images (of 9 covid-19 patients) sized 630 × 630. Radiologists have tagged 373 images with covid-19 pulmonary symptoms and the rest of the images as normal cases.

Dataset-2: this dataset is publicly available and contains 20 CT volumes with more than 1800 slices collected from 40 different COVID patients [273]. Each CT slice is of size 512 × 512 and labelled by expert radiologists to mark regions for infections.

Figure 63. The first column images represent the original images, while the second column images represent their corresponding ground-truth masks. Two examples of images of normal people are shown in the third and fourth rows. The first and second rows include two COVID-19 images, where the COVID-19 regions are the white and gray regions of the ground-truth masks, while the healthy regions are the black pixels (note that if a person is in good health his ground-truth mask will be completely black).In our simulations, we merged the two versions of dataset-1 to form a new dataset, while keeping 60% of these images for the training, 20% for the validation, and 20% for the test

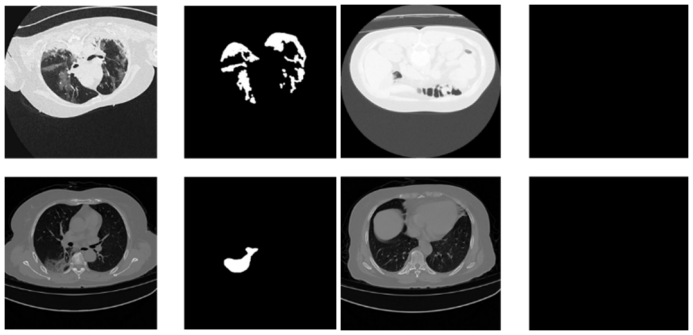

Fig. 63. Example of images belong to Dataset-1 and Dataset-2.

6.4.2 Performance metrics evaluation:

To assess the efficiency and robustness of our proposed approach, we use the following performance metrics: Accuracy [273], Sensitivity [273], Matthews Correlation Coefficient [273], and Dice [274]. Higher values of these metrics imply a better segmentation quality. The mathematical formulas of these metrics are respectively expressed below:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (55)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (56)$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (57)$$

$$Dice = \frac{2 * TP}{2 * (TP + FN + FP)} \quad (58)$$

6.4.3 Hyper-parameters setting

In deep learning, hyper-parameters of the deep neural network crucially influence the performance of the network. In this part, we carried out several experiments to choose the best values of the Hyper-parameters allowing improving the performance of our method. In this regard, we fixed the number of epochs and the batch size respectively at 100 and 64. We simulated and compared the different precision obtained values with different optimizers, different learning rates, and different values of λ_1, λ_2 of the L_B loss function.

TABLE XV. THE CHOICE OF THE BEST COMBINATION FOR INPUT HYPERPARAMETERS, BEST ACCURACY SHOWN IN BOLD FONT

Optimizer	Learning- rate	Loss (LB)	Accuracy
ADAM			**0.9445**
SGD	Lr = 0.0001	$\lambda_1 = 0.5, \lambda_2 = 0.5$	0.8921
Adadelta			0.9032
	Lr = 0.001		0.944
ADAM	Lr = 0.00001	$\lambda_1 = 0.5, \lambda_2 = 0.5$	0.924
	Lr = 0.000001		**0.964**
		$\lambda_1 = 0.4, \lambda_2 = 0.2$	0.957
ADAM	Lr=0.000001	$\lambda_1 = 0.5, \lambda_2 = 0.5$	0.964
		$\lambda_1 = 0.3, \lambda_2 = 0.5$	**0.991**

From Table. XVI we observe that the values of the hyper-parameters λ1 = 0.3, λ2 = 0.5, and the use of the ADAM [23] optimizer with a learning rate equal to 0.000001 show the best performance in term of accuracy.

6.4.4 Ablation study:

To show the importance of each component of our contribution, we did an ablation study on dataset 1 by evaluating the following performance metrics, Accuracy, Dice, Sensitivity, and Precision. The study of ablations was subdivided into three possible cases. In the first case, we added the pre-processing block without using the loss function (LB) during the learning phase. Whereas in the second case, we used the loss function without the pre-processing block. In the latter case, we used the pre-processing block and the loss function (LB). From these simulation cases, we notice that the data pre-processing step has a remarkable effect on the model performance. The first case shows the over-fitting phenomenon that occurred when it exceeded epoch 40, which led to degradation in the performance of the model. The results of this study are illustrated in Table XVII.

TABLE XVI. ABLATION STUDY ON THE DATASET 1

Cases	Accuracy	Dice	MCC	Sensitivity
Preprocessing, BinaryCrossEntroy	0.890	0.514	0.432	0.741
L_B, without Preprocessing,	0.934	0.632	0.590	0.842
Preprocessing, L_B	**0.991**	**0.833**	**0.851**	**0.982**

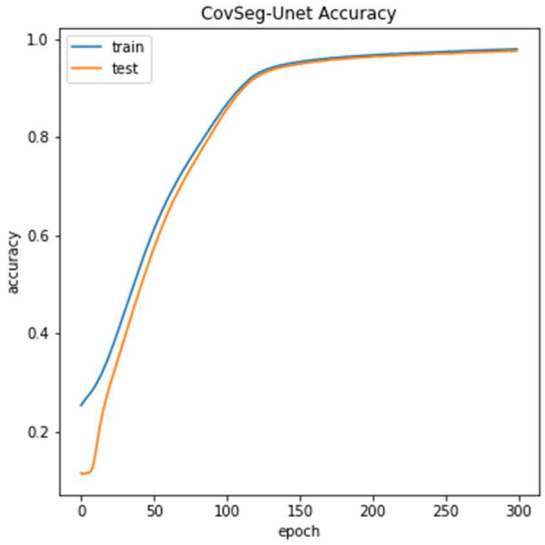

Fig. 64. Accuracy of simulation without data preprocessing.

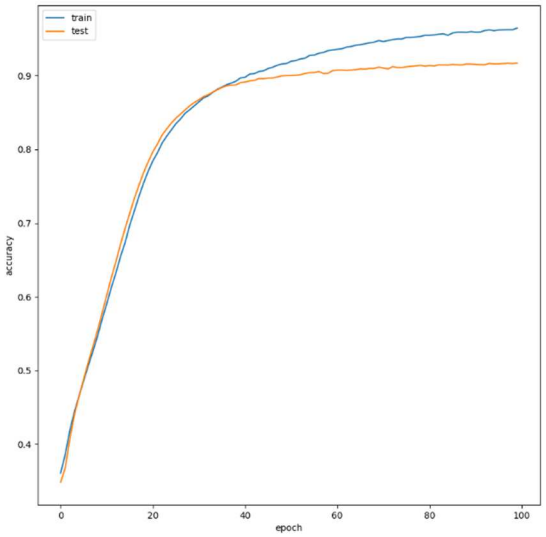

Fig. 65. Accuracy of simulation without data preprocessing step.

On the other hand, the result of the third case shows that the use of the loss function with the pre-processing block helps to avoid the over-fitting problem, and consequently, improves the performance of the model and has good results. Figure 66 and Figure 67 show the qualitative results of our method on different test samples, the first column (1) represents the lung images, the second column (2) shows the ground-truth mask, and the last column (3) corresponds to the predicted lung infection mask of covid-19. These qualitative results prove the robustness and the efficiency of our diagnostic method for the region infected by covid-19.

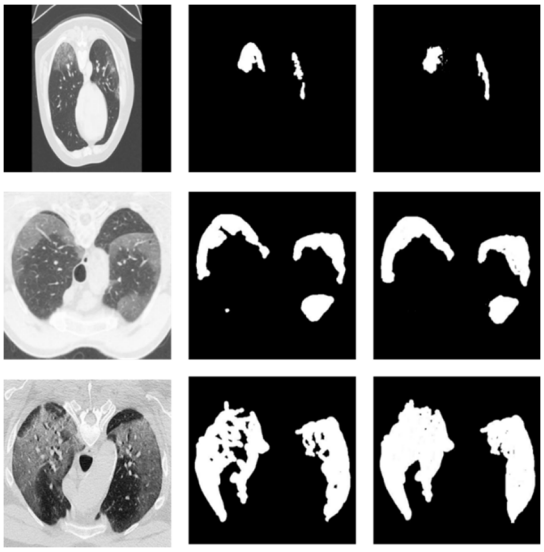

Fig. 66. Results examples obtained by CovSeg-Unet on dataset-1.

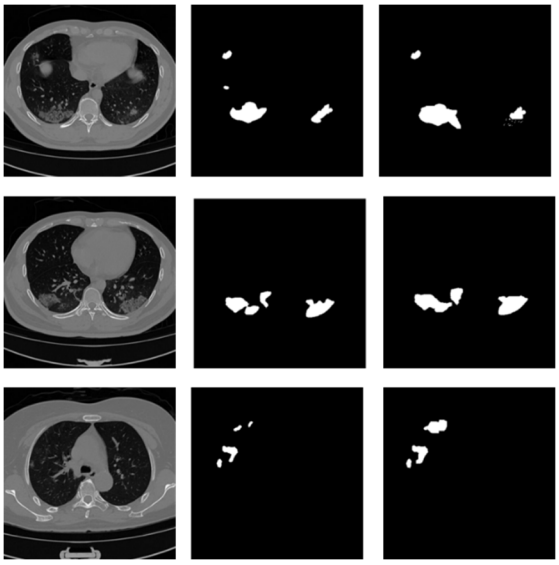

Fig. 67. Results examples obtained by CovSeg-Unet on dataset-2.

6.4.5 Comparison with baseline methods:

In this part, we compare our method of segmentation of CT images with the reference segmentation methods such as U-NET basic [275], DenseUNet [2], Attention U-Net [264], and UNet++ [262]. Table. XVIII represents obtained results by the different methods on dataset-1 and dataset-2. As result, the proposed method outperforms other methods in terms of performance measures. Dice, Sensitivity, and Accuracy metrics reach 83.3%, 98.2% and 99.1% respectively on dataset-1, and 83.4% 89.1% 98.3% on dataset-2.

TABLE XVII. PERFORMANCES COMPARISON AGAINST BASELINE ARCHITECTURES ON DATASET-1 AND DATASET-2.

	Methods	Dice	Sensitivity	Accuracy
Dataset-1	U-Net [275]	0.708	0.678	0.865
	DenseUNet [2]	0.660	0.607	0.651
	Attention U-Net [264]	0.560	0.623	0.632
	UNet++ [262]	0.815	0.857	0.903
	CovSeg-Unet	**0.833**	**0.982**	**0.991**
Dataset-2	UNet [275]	0.712	0.665	0.747
	DenseUNet [2]	0.610	0.607	0.715
	Attention U-Net [264]	0.631	0.723	0.890
	UNet++ [262]	0.815	0.887	0.968
	CovSeg-Unet	**0.834**	**0.891**	**0.983**

6.4.6 Comparison with other methods

Many studies have been done to diagnose COVID-19. To prove the robustness of our method, we carried out a comparative study with different approaches such as InfNet [265] and Automatic [268]. We simulated these approaches using their open-source implementation. The quantitative results obtained by the different methods are shown in Table. VI. Dice, Sensitivity, and Accuracy are the performance metrics to be evaluated in this benchmarking study. From Table. XIX we notice that the CovSeg-Unet method reaches an Accuracy = 0.991 on Datasets-1, and an Accuracy = 0.983 on Datasets-2. The obtained values

in these simulations prove that the CovSeg-Unet method outperforms other approaches in terms of Dice, Sensitivity, and Accuracy.

TABLE XVIII. PERFORMANCES COMPARISON AGAINST EXISTING APPROACHES ON DATASETS 2.

Methods	Dice	Sensitivity	Accuracy
Inf-Net [265]	0.579	0.877	—
Automatic [268]	0.714	0.733	0.739
CovSeg-Unet	**0.834**	**0.891**	**0.983**

6.5 Conclusion

In this contribution, we address the most difficult task of segmenting limited and unbalanced biomedical images. To cope with this task, we proposed an end-to-end architecture like U-Net, the network learns the discriminating characteristics of lung infections from CT images to establish an image-to-image mapping relationship; We used the architecture ResNet50 to preserve local information and avoid the problem of fading gradients. To improve the learning of discriminating network characteristics we have introduced a pre-processing block to remove noise and unnecessary information that affects network performance. To strengthen the model to be learned from the non-equilibrium data, we proposed a loss function LB. The experimental results on two datasets show the effectiveness of the CovSeg-Unet method in locating the infected regions with COVID-19. The promising results obtained by our method can effectively help radiologists to diagnose and locate regions infected with covid-19.

General conclusion

This thesis deals with image processing by deep learning methods. The choice of deep learning (DL) methods was motivated by their advantages in processing unstructured objects, such as images. In this context, we are interested in convolutional neural networks (CNN) because of their adaptation to image classification and segmentation. These are characterized by built-in feature extraction modules (convolution layers) compared to traditional machine learning methods that require handcrafted feature extraction pre-processing.

The first CNN-type architecture in supervised classification finds its roots in the years 1998. The limited capacity of computers at that time and the reduced number of available images limited the exploitation of these architectures. At the same time, ML methods have attracted great interest due to their reduced requirements in terms of processing capacity and data volumes. Until 2012, when the better error rate of the AlexNet architecture based on ImageNet learning, and the ability of GPUs to reduce computation time, encouraged the image processing community to reuse and optimize these networks to exploit them in computer vision applications.

In the state of the art, several works have been proposed for the classification and segmentation of images by DL methods, and CNNs.

The main objective of this research was to propose networks based on deep learning methods for different tasks. In this context, although our training is focused on deep learning methods, we have chosen to apply these methods in two areas. In the field of medical image segmentation, more specifically, we were interested in the location of pulmonary infections by the Covid-19 virus in CT images; we are interested in optimizing the performance of CNN networks for the segmentation of these images. The analysis of CT images is a significant task in the diagnosis of several types of todometric images. A manual review of these images presents several problems related to the subjectivity of radiologists' decisions and the intervariability of images collected from different scanner machines. In the field of the detection of falsified images, we are concerned with the localization of the regions manipulated by CNN networks. The analysis of these images is a very difficult task because we are confronted with a great variety of digital images. In addition, the manipulation of these images has become increasingly simple and very easy, on the one hand, thanks to the technological evolution of GPU cards, which has allowed us to use very efficient infrastructures to process these images. On the other hand, thanks to the rapid development of image processing software.

This thesis work is part of proposing models based on DL methods for two major issues in the fields already mentioned.

The first problem concerns diagnostic aid systems. The objective of these systems is to assist radiologists in their decisions and therefore to avoid subjective decisions in the case of pulmonary infections. The main objective of the proposed approaches was to solve the different problems related to the localization of regions infected by the COVID-19 virus from CT images by DL methods. In our work, the medical field has been a new experience since this thesis is located at the intersection of computer science and the medical field. The 2019 coronavirus pandemic prompted our research work to propose solutions to help radiologists

speed up the process of diagnosing COVID-19. To understand the structure and processing techniques of todometric images, we have done an in-depth study on the manual procedure performed by radiologists and the whole process performed in, the acquisition, pre-processing, and localization of CT images.

In this context, we propose a contribution based on a deep neural network to process and analyze CT images for the diagnosis of COVID-19 [EL BIACH]. This approach mainly involves identifying regions of interest in these images, such as areas affected by the COVID-19 virus. This method consists of a first step in pre-processing the CT images to eliminate the noise and render all the images to the same standard. Then we use an end-to-end architecture to train the network. Since CT images are not balanced, we propose a loss function to balance the pixel distribution of infected/uninfected regions. Our approach has achieved high performance in localizing COVID-19 lung infections compared to other methods.

The second problem concerns the detection of falsified images because it is very difficult to differentiate a manipulated image from an original image because of image editing software, which leaves no trace visible to the human eye. Whereas these images are used in several fields of application such as police investigations as forensic evidence, journalism, etc. Images tampered with for malicious purposes will have dangerous consequences in our society. The main objective of the proposed approaches is to locate falsified regions in an image using deep learning methods. In this context, we have proposed two approaches.

In the first contribution, we proposed a new convolutional neural network method based on an encoder/decoder called Fals-Unet is proposed to localize the manipulated regions. The encoder of our method uses architecture topologically identical to that of the Resnet50 method; its main objective is the exploitation of spatial maps to analyze the discriminating characteristics between manipulated and non-manipulated regions. The decoding network learns the mapping from low-resolution feature maps to pixel-level predictions to locate tampered regions. Finally, the predicted binary mask (0: tamper, 1: do not tamper) is generated by the final layer (Softmax)

In the second contribution [EL BIACH], we focused on the problem of overfitting because falsified image segmentation methods based on convolutional neural networks are confronted with the problem of imbalanced classes; unbalanced classes usually refer to a classification problem where the ratio of observations in a class to all observations is very low. This class imbalance clearly increases learning difficulty and introduces strongly biased predictions in favor of high precision but low recall. In this contribution, we propose the balance factor based on a focal loss function to solve this data imbalance problem.

Future Work

Based on the results obtained in the different approaches, we can conclude that the use of methods based on the "encoder-decoder" architecture allows combining the spatial information and the local information and reduces the high variance in the problems of image segmentation, and therefore improves the relevance of the results. In addition, the learning transferred between large databases like ImagNET improves the performance of the segmentation, reduces the requirements of the models used in terms of computation time, and reduces the various problems related to overfitting. Besides, the loss function plays a very important role in the learning process, so the use of a penalty factor in the case of strongly biased classes can improve the result. Despite the efficiency of the methods based on the encoder-decoder architecture, we found that there is a considerable drop in performance in the case of the detection of smaller infected regions with blurred noisy boundaries. Because the “encoder-decoder” network increases as one deepens into the network. This increase in the size of the receptive field influences the learning of the characteristics that correspond to small fine regions such as the edges and their texture. This prevents any network with the standard sub-complete architecture from producing sharp predictions around edges in tasks such as segmentation. As a result, in future contributions, we are interested in overcoming this problem.

- The implementation of a per-pixel consistency regularization technique based on the augmentation of CutMix data to encourage the Fals-UNet network to focus more on small regions and to disentangle between non-manipulated and manipulated images. This can improve the training of the U-Net discriminator, further improving the quality of the generated samples.
- We propose hybrid architecture between the U-NET network and another network like Transformer to capture spatial information from small structures. Hybridization keeps spatial and local information to train the network on the different information. We are designing an over-complete architecture of U-NET which consists in projecting information on higher dimensions to improve learning in the case of small regions.
- It would be interesting to exploit the U-NET model trained on medical databases as a feature extraction module on medical databases instead of using the classic models trained on ImageNet. Recall that this model has proven its effectiveness compared to models trained on the ImageNet database in transferred learning.

References

1. Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., 2013. S
2. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W. and Heng, P.A., 2018. H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. IEEE transactions on medical imaging, 37(12), pp.2663-2674.
3. Bishop, C.M. and Nasrabadi, N.M., 2006. Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
4. Nielsen MA. Neural networks and deep learning. San Francisco, CA, USA: Determination press; 2015 Sep 25.
5. Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. MIT press.
6. McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), pp.115-133.
7. Hebb, D.O., 1949. The organization of behavior: a neuropsychological theory. Science editions.
8. Minsky, M. and Papert, S., 1969. Perceptron: an introduction to computational geometry.
9. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.
10. Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), p.386.
11. Novikoff, A.B., 1963. On convergence proofs for perceptrons. STANFORD RESEARCH INST MENLO PARK CA.
12. Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.
13. Xu, J., Schwing, A.G. and Urtasun, R., 2015. Learning to segment under various forms of weak supervision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3781-3790).
14. Widrow, B., 1960. Adaptive" adaline" Neuron Using Chemical" memistors.".
15. Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural networks, 12(1), pp.145-151.
16. Nesterov, Y.E., 1983. A method for solving the convex programming problem with convergence rate $ Obigl (frac {1}{k^ 2} bigr) $. In Dokl. Akad. Nauk SSSR, (Vol. 269, pp. 543-547).
17. Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).
18. Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.
19. Kingma, D.P., 2015. &Ba J.(2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
20. Goodfellow Ian, J., Jean, P.A., Mehdi, M., Bing, X., David, W.F., Sherjil, O. and Courville Aaron, C., 2014, December. Generative adversarial nets. In Proceedings of the 27th international conference on neural information processing systems (Vol. 2, pp. 2672-2680).
21. Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

22. Tieleman, T. and Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2), pp.26-31.
23. Riedmiller, M. and Braun, H., 1993, March. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In IEEE international conference on neural networks (pp. 586-591). IEEE.
24. Bishop, C.M., 1995. Neural networks for pattern recognition. Oxford university press.
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.
26. Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O., 2021. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), pp.107-115.
27. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.
28. Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for activation functions. arXiv preprint arXiv:1710.05941.
29. Pineda, F., 1987. Generalization of back propagation to recurrent and higher order neural networks. In Neural information processing systems.
30. Pineda, F., 1987. Generalization of back propagation to recurrent and higher order neural networks. In Neural information processing systems.
31. Du, Y., Wang, W. and Wang, L., 2015. Hierarchical recurrent neural network for skeleton-based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1110-1118).
32. Vinyals, O., Toshev, A., Bengio, S. and Erhan, D., 2015. Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).
33. Xie, Y., Zhang, Z., Sapkota, M. and Yang, L., 2016, October. Spatial clockwork recurrent neural network for muscle perimysium segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 185-193). Springer, Cham.
34. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
35. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
36. Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Roux, J.L., Hershey, J.R. and Schuller, B., 2015, August. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In International conference on latent variable analysis and signal separation (pp. 91-99). Springer, Cham.
37. Valstar, M., Schuller, B., Smith, K., Eyben, F., Jiang, B., Bilakhia, S., Schnieder, S., Cowie, R. and Pantic, M., 2013, October. Avec 2013: the continuous audio/visual emotion and depression recognition challenge. In Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge (pp. 3-10).
38. Bai, C., Zheng, A., Huang, Y., Pan, X. and Chen, N., 2021. Boosting convolutional image captioning with semantic content and visual relationship. Displays, 70, p.102069.
39. Tan, C.C. and Eswaran, C., 2011. Using autoencoders for mammogram compression. Journal of medical systems, 35(1), pp.49-58.

40. Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Du, P. and Marshall, S., 2016. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. Neurocomputing, 185, pp.1-10.
41. Antoniou, A., Storkey, A. and Edwards, H., 2018, October. Augmenting image classifiers using data augmentation generative adversarial networks. In International Conference on Artificial Neural Networks (pp. 594-603). Springer, Cham.
42. Luc, P., Couprie, C., Chintala, S. and Verbeek, J., 2016. Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408.
43. Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
44. Dong, H.W., Hsiao, W.Y., Yang, L.C. and Yang, Y.H., 2018, April. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).
45. Hubel, D.H. and Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1), p.106.
46. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L. and Leskovec, J., 2018, July. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 974-983).
47. Thomas, A.A., Zheng, C., Jung, H., Chang, A., Kim, B., Gelfond, J., Slezak, J., Porter, K., Jacobsen, S.J. and Chien, G.W., 2014. Extracting data from electronic medical records: validation of a natural language processing program to assess prostate biopsy results. World journal of urology, 32(1), pp.99-103.
48. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.
49. Keshari, R., Vatsa, M., Singh, R. and Noore, A., 2018. Learning structure and strength of CNN filters for small sample size training. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9349-9358).
50. Fukushima, K., 1980. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol, Cybern, 36, pp.193-202.
51. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.
52. Albawi, S., Mohammed, T.A. and Al-Zawi, S., Understanding of a convolutional neural network Proceedings of the 2017 International Conference on Engineering and Technology (ICET) August 2017Antalya..
53. Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
54. Smola, A. and Vishwanathan, S.V.N., 2008. Introduction to machine learning. Cambridge University, UK, 32(34), p.2008.
55. Piccinini, G., 2004. The First computational theory of mind and brain: a close look at mcculloch and pitts's “logical calculus of ideas immanent in nervous activity”. Synthese, 141(2), pp.175-215.
56. Hebb, D.O., 1949. The organization of behavior: a neuropsychological theory. Science editions.
57. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

58. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
59. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
60. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-first AAAI conference on artificial intelligence.
61. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
62. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
63. Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
64. Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
65. Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
66. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
67. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
68. Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
69. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.
70. Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495.
71. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
72. Denker, J. and LeCun, Y., 1990. Transforming neural-net output levels to probability distributions. Advances in neural information processing systems, 3.
73. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.
74. Zeiler, M.D., Taylor, G.W. and Fergus, R., 2011, November. Adaptive deconvolutional networks for mid and high level feature learning. In 2011 international conference on computer vision (pp. 2018-2025). IEEE.

75. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
76. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J. and Yoo, Y., 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6023-6032).
77. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. and Wilson, A.G., 2018. Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407.
78. Bello, I., Zoph, B., Vaswani, A., Shlens, J. and Le, Q.V., 2019. Attention augmented convolutional networks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3286-3295).
79. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A. and Van Der Maaten, L., 2018. Exploring the limits of weakly supervised pretraining. In Proceedings of the European conference on computer vision (ECCV) (pp. 181-196).
80. Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
81. Chellapilla, K., Puri, S. and Simard, P., 2006, October. High performance convolutional neural networks for document processing. In Tenth international workshop on frontiers in handwriting recognition. Suvisoft.
82. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).
83. Muhammad, K., Ahmad, J. and Baik, S.W., 2018. Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing, 288, pp.30-42.
84. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B. and Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical image analysis, 42, pp.60-88.
85. Turan, M., Almalioglu, Y., Araujo, H., Konukoglu, E. and Sitti, M., 2018. Deep endovo: A recurrent convolutional neural network (rcnn) based visual odometry approach for endoscopic capsule robots. Neurocomputing, 275, pp.1861-1870.
86. Tian, Y., Pei, K., Jana, S. and Ray, B., 2018, May. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th international conference on software engineering (pp. 303-314).
87. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
88. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).
89. Lawrence, S., Giles, C.L., Tsoi, A.C. and Back, A.D., 1997. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1), pp.98-113.
90. ClaudiuCiresan, D., Meier, U. and Gambardella, L.M., 2011. JurgenSchmidhuber,"Convolutional Neural Network Committees for Handwritten Character Classification". In 2011 International Conference on Document Analysis and Recognition, IEEE.

91. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J., 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), pp.1299-1312.
92. Hu, W., Huang, Y., Wei, L., Zhang, F. and Li, H., 2015. Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015.
93. Spanhol, F.A., Oliveira, L.S., Petitjean, C. and Heutte, L., 2016, July. Breast cancer histopathological image classification using convolutional neural networks. In 2016 international joint conference on neural networks (IJCNN) (pp. 2560-2567). IEEE.
94. Lakhani, P. and Sundaram, B., 2017. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology, 284(2), pp.574-582.
95. Xu, J., Luo, X., Wang, G., Gilmore, H. and Madabhushi, A., 2016. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. Neurocomputing, 191, pp.214-223.
96. Huang, K.Q., Ren, W.Q. and Tan, T.N., 2014. A review on image object classification and detection. Chinese Journal of Computers, 37(6), pp.1225-1240.
97. Zhang, X., Yang, Y.H., Han, Z., Wang, H. and Gao, C., 2013. Object class detection: A survey. ACM Computing Surveys (CSUR), 46(1), pp.1-53.
98. Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
99. Chen, T., Lu, S. and Fan, J., 2017. S-CNN: Subcategory-aware convolutional networks for object detection. IEEE transactions on pattern analysis and machine intelligence, 40(10), pp.2522-2528.
100. Long, Y., Gong, Y., Xiao, Z. and Liu, Q., 2017. Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 55(5), pp.2486-2498.
101. Cireşan, D.C., Giusti, A., Gambardella, L.M. and Schmidhuber, J., 2013, September. Mitosis detection in breast cancer histology images with deep neural networks. In International conference on medical image computing and computer-assisted intervention (pp. 411-418). Springer, Berlin, Heidelberg.
102. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X. and Ouyang, W., 2017. T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology, 28(10), pp.2896-2907.
103. Weinland, D., Ronfard, R. and Boyer, E., 2011. A survey of vision-based methods for action representation, segmentation and recognition. Computer vision and image understanding, 115(2), pp.224-241.
104. Ilea, D.E. and Whelan, P.F., 2011. Image segmentation based on the integration of colour–texture descriptors—A review. Pattern Recognition, 44(10-11), pp.2479-2501.
105. Sacco, M., 1990. Stochastic relaxation, gibbs distributions and bayesian restoration of images.
106. Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J., 2010. Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5), pp.898-916.
107. Sinha, R.K., Pandey, R. and Pattnaik, R., 2018. Deep learning for computer vision tasks: a review. arXiv preprint arXiv:1804.03928.

108. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), pp.834-848.
109. Krähenbühl, P. and Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems, 24.
110. Liu, W., Rabinovich, A. and Berg, A.C., 2015. Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579.
111. Bian, X., Lim, S.N. and Zhou, N., 2016, March. Multiscale fully convolutional network with application to industrial inspection. In 2016 IEEE winter conference on applications of computer vision (WACV) (pp. 1-8). IEEE.
112. Papandreou, G., Chen, L.C., Murphy, K.P. and Yuille, A.L., 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In Proceedings of the IEEE international conference on computer vision (pp. 1742-1750).
113. Dai, J., He, K. and Sun, J., 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In Proceedings of the IEEE international conference on computer vision (pp. 1635-1643).
114. Everingham, M., Eslami, S.M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), pp.98-136.
115. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R. and Yuille, A., 2014. The role of context for object detection and semantic segmentation in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 891-898).
116. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R. and Yuille, A., 2014. Detect what you can: Detecting and representing objects using holistic models and body parts. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1971-1978).
117. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
118. Levi, D., Garnett, N., Fetaya, E. and Herzlyia, I., 2015, September. StixelNet: A Deep Convolutional Network for Obstacle Detection and Road Segmentation. In BMVC (Vol. 1, No. 2, p. 4).
119. Kampffmeyer, M., Salberg, A.B. and Jenssen, R., 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-9).
120. Luvizon, D.C., Picard, D. and Tabia, H., 2018. 2d/3d pose estimation and action recognition using multitask deep learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5137-5146).
121. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X. and Ouyang, W., 2017. T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology, 28(10), pp.2896-2907.

122. Wang, K., Zhao, R. and Ji, Q., 2018, May. Human computer interaction with head pose, eye gaze and body gestures. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) (pp. 789-789). IEEE.
123. Liu, H. and Wang, L., 2018. Gesture recognition for human-robot collaboration: A review. International Journal of Industrial Ergonomics, 68, pp.355-367.
124. Kügler, D., Sehring, J., Stefanov, A., Stenin, I., Kristin, J., Klenzner, T., Schipper, J. and Mukhopadhyay, A., 2020. i3PosNet: instrument pose estimation from X-ray in temporal bone surgery. International journal of computer assisted radiology and surgery, 15(7), pp.1137-1145.
125. Popa, A.I., Zanfir, M. and Sminchisescu, C., 2017. Deep multitask architecture for integrated 2d and 3d human sensing. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 6289-6298).
126. Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E., 2018. Deep learning for computer vision: A brief review. Computational intelligence and neuroscience, 2018.
127. Toshev, A. and Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1653-1660).
128. Chen, X. and Yuille, A.L., 2014. Articulated pose estimation by a graphical model with image dependent pairwise relations. Advances in neural information processing systems, 27.
129. Jain, A., Tompson, J., Andriluka, M., Taylor, G.W. and Bregler, C., 2013. Learning human pose estimation features with convolutional networks. arXiv preprint arXiv:1312.7302.
130. Tompson, J.J., Jain, A., LeCun, Y. and Bregler, C., 2014. Joint training of a convolutional network and a graphical model for human pose estimation. Advances in neural information processing systems, 27.
131. Yang, W., Ouyang, W., Li, H. and Wang, X., 2016. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3073-3082).
132. Shamir, L., Orlov, N., Mark Eckley, D., Macura, T.J. and Goldberg, I.G., 2008. IICBU 2008: a proposed benchmark suite for biological image analysis. Medical & biological engineering & computing, 46(9), pp.943-947.
133. Kather, J.N., Weis, C.A., Bianconi, F., Melchers, S.M., Schad, L.R., Gaiser, T., Marx, A. and Zöllner, F.G., 2016. Multi-class texture analysis in colorectal cancer histology. Scientific reports, 6(1), pp.1-11.
134. Ker, J., Wang, L., Rao, J. and Lim, T., 2017. Deep learning applications in medical image analysis. Ieee Access, 6, pp.9375-9389.
135. Rajkomar, A., Lingam, S., Taylor, A.G., Blum, M. and Mongan, J., 2017. High-throughput classification of radiographs using deep convolutional neural networks. Journal of digital imaging, 30(1), pp.95-101.
136. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K. and Lungren, M.P., 2017. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225.
137. Shen, W., Zhou, M., Yang, F., Yang, C. and Tian, J., 2015, June. Multi-scale convolutional neural networks for lung nodule classification. In International conference on information processing in medical imaging (pp. 588-599). Springer, Cham.

138. Li, R., Zhang, W., Suk, H.I., Wang, L., Li, J., Shen, D. and Ji, S., 2014, September. Deep learning based imaging data completion for improved brain disease diagnosis. In International conference on medical image computing and computer-assisted intervention (pp. 305-312). Springer, Cham.
139. Spanhol, F.A., Oliveira, L.S., Petitjean, C. and Heutte, L., 2016, July. Breast cancer histopathological image classification using convolutional neural networks. In 2016 international joint conference on neural networks (IJCNN) (pp. 2560-2567). IEEE.
140. Xu, J., Luo, X., Wang, G., Gilmore, H. and Madabhushi, A., 2016. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. Neurocomputing, 191, pp.214-223.
141. Byra, M., Styczynski, G., Szmigielski, C., Kalinowski, P., Michałowski, Ł., Paluszkiewicz, R., Ziarkiewicz-Wróblewska, B., Zieniewicz, K., Sobieraj, P. and Nowicki, A., 2018. Transfer learning with deep convolutional neural network for liver steatosis assessment in ultrasound images. International journal of computer assisted radiology and surgery, 13(12), pp.1895-1903.
142. Ker, J., Wang, L., Rao, J. and Lim, T., 2017. Deep learning applications in medical image analysis. Ieee Access, 6, pp.9375-9389.
143. Lo, S.C., Lou, S.L., Lin, J.S., Freedman, M.T., Chien, M.V. and Mun, S.K., 1995. Artificial convolution neural network techniques and applications for lung nodule detection. IEEE transactions on medical imaging, 14(4), pp.711-718.
144. Ribli, D., Horváth, A., Unger, Z., Pollner, P. and Csabai, I., 2018. Detecting and classifying lesions in mammograms with deep learning. Scientific reports, 8(1), pp.1-7.
145. Fan, W., Jiang, H., Ma, L., Gao, J. and Yang, H., 2018, August. A modified faster R-CNN method to improve the performance of the pulmonary nodule detection. In Tenth International Conference on Digital Image Processing (ICDIP 2018) (Vol. 10806, pp. 1469-1476). SPIE.
146. Yang, D., Zhang, S., Yan, Z., Tan, C., Li, K. and Metaxas, D., 2015, April. Automated anatomical landmark detection ondistal femur surface using convolutional neural network. In 2015 IEEE 12th international symposium on biomedical imaging (ISBI) (pp. 17-21). IEEE.
147. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V.C., Shi, L. and Heng, P.A., 2016. Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. IEEE transactions on medical imaging, 35(5), pp.1182-1195.
148. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L. and Erickson, B.J., 2017. Deep learning for brain MRI segmentation: state of the art and future directions. Journal of digital imaging, 30(4), pp.449-459.
149. Pereira, S., Pinto, A., Alves, V. and Silva, C.A., 2016. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE transactions on medical imaging, 35(5), pp.1240-1251.
150. Ciresan, D., Giusti, A., Gambardella, L. and Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. Advances in neural information processing systems, 25.
151. Mercadier, D.S., Besbinar, B. and Frossard, P., 2019, May. Automatic Segmentation of Nuclei in Histopathology Images Using Encoding-decoding Convolutional Neural Networks. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1020-1024). IEEE.

152. Peng, B., Chen, L., Shang, M. and Xu, J., 2018, October. Fully convolutional neural networks for tissue histopathology image classification and segmentation. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 1403-1407). IEEE.
153. Al-Qershi, O.M. and Khoo, B.E., 2013. Passive detection of copy-move forgery in digital images: State-of-the-art. Forensic science international, 231(1-3), pp.284-295.
154. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A. and Serra, G., 2011. A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE transactions on information forensics and security, 6(3), pp.1099-1110.
155. Ansari, M.D., Ghrera, S.P. and Tyagi, V., 2014. Pixel-based image forgery detection: A review. IETE journal of education, 55(1), pp.40-46.
156. Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495.
157. Bappy, J.H., Roy-Chowdhury, A.K., Bunk, J., Nataraj, L. and Manjunath, B.S., 2017. Exploiting spatial structure for localizing manipulated image regions. In Proceedings of the IEEE international conference on computer vision (pp. 4970-4979).
158. Bappy, J.H., Simons, C., Nataraj, L., Manjunath, B.S. and Roy-Chowdhury, A.K., 2019. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing, 28(7), pp.3286-3300.
159. Bayar, B. and Stamm, M.C., 2016, June. A deep learning approach to universal image manipulation detection using a new convolutional layer. In Proceedings of the 4th ACM workshop on information hiding and multimedia security (pp. 5-10).
160. Bayar, B. and Stamm, M.C., 2017. Design principles of convolutional neural networks for multimedia forensics. Electronic Imaging, 2017(7), pp.77-86.
161. Bayar, B. and Stamm, M.C., 2017, March. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2152-2156). IEEE.
162. Bharati, A., Singh, R., Vatsa, M. and Bowyer, K.W., 2016. Detecting facial retouching using supervised deep learning. IEEE Transactions on Information Forensics and Security, 11(9), pp.1903-1913.
163. Bianchi, T., De Rosa, A. and Piva, A., 2011, May. Improved DCT coefficient analysis for forgery localization in JPEG images. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2444-2447). IEEE.
164. Bianchi, T. and Piva, A., 2012. Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Transactions on Information Forensics and Security, 7(3), pp.1003-1017.
165. Boughorbel, S., Jarray, F. and El-Anbari, M., 2017. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PloS one, 12(6), p.e0177678.
166. Buccoli, M., Bestagini, P., Zanoni, M., Sarti, A. and Tubaro, S., 2014, December. Unsupervised feature learning for bootleg detection using deep learning architectures. In 2014 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 131-136). IEEE.
167. Bunk, J., Bappy, J.H., Mohammed, T.M., Nataraj, L., Flenner, A., Manjunath, B.S., Chandrasekaran, S., Roy-Chowdhury, A.K. and Peterson, L., 2017, July. Detection and localization of image forgeries using resampling features and deep learning. In 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW) (pp. 1881-1889). IEEE.

168. Cai, W. and Wei, Z., 2020. PiiGAN: generative adversarial networks for pluralistic image inpainting. IEEE Access, 8, pp.48451-48463.
169. Chang, I.C., Yu, J.C. and Chang, C.C., 2013. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. Image and Vision Computing, 31(1), pp.57-71.
170. Chawla, N.V., Japkowicz, N. and Kotcz, A., 2004. Special issue on learning from imbalanced data sets. ACM SIGKDD explorations newsletter, 6(1), pp.1-6.
171. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), pp.834-848.
172. Chollet, F., 2015. keras, GitHub. GitHub repository.
173. Cozzolino, D., Poggi, G. and Verdoliva, L., 2015. Efficient dense-field copy–move forgery detection. IEEE Transactions on Information Forensics and Security, 10(11), pp.2284-2297.
174. Dirik, A.E. and Memon, N., 2009, November. Image tamper detection based on demosaicing artifacts. In 2009 16th IEEE International Conference on Image Processing (ICIP) (pp. 1497-1500). IEEE.
175. Dong J, Wang W (2011) Casia tampered image detection evaluation (tide) database v1.0 and v2.0. http:// forensics.idealtest.org/
176. Farid, H., 1999. Detecting digital forgeries using bispectral analysis.
177. Farid, H., 2009. Exposing digital forgeries from JPEG ghosts. IEEE transactions on information forensics and security, 4(1), pp.154-160.
178. Feng, X., Cox, I.J. and Doërr, G., 2011, July. An energy-based method for the forensic detection of re-sampled images. In 2011 IEEE International Conference on Multimedia and Expo (pp. 1-6). IEEE.
179. Feng, X., Cox, I.J. and Doerr, G., 2012. Normalized energy density-based forensic detection of resampled images. IEEE Transactions on Multimedia, 14(3), pp.536-545.
180. Ferrara, P., Bianchi, T., De Rosa, A. and Piva, A., 2012. Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Transactions on Information Forensics and Security, 7(5), pp.1566-1577.
181. Fillion, C. and Sharma, G., 2010, January. Detecting content adaptive scaling of images for forensic applications. In Media Forensics and Security II (Vol. 7541, pp. 359-370). SPIE.
182. Fu, D., Shi, Y.Q. and Su, W., 2006, November. Detection of image splicing based on Hilbert-Huang transform and moments of characteristic functions with wavelet decomposition. In International workshop on digital watermarking (pp. 177-187). Springer, Berlin, Heidelberg.
183. Gao, S., Liao, X. and Liu, X., 2019. Real-time detecting one specific tampering operation in multiple operator chains. Journal of Real-Time Image Processing, 16(3), pp.741-750.
184. Golestaneh, S.A. and Chandler, D.M., 2014. Algorithm for JPEG artifact reduction via local edge regeneration. Journal of Electronic Imaging, 23(1), p.013018.
185. Guillemot, C. and Le Meur, O., 2013. Image inpainting: Overview and recent advances. IEEE signal processing magazine, 31(1), pp.127-144.
186. He, H. and Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), pp.1263-1284.
187. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

188. Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.
189. Jaberi, M., Bebis, G., Hussain, M. and Muhammad, G., 2014. Accurate and robust localization of duplicated region in copy–move image forgery. Machine vision and applications, 25(2), pp.451-475.
190. Johnson, M.K. and Farid, H., 2007, June. Exposing digital forgeries through specular highlights on the eye. In International Workshop on Information Hiding (pp. 311-325). Springer, Berlin, Heidelberg.
191. Kakar, P. and Sudha, N., 2012. Exposing postprocessed copy–paste forgeries through transform-invariant features. IEEE Transactions on Information Forensics and Security, 7(3), pp.1018-1028.
192. Kendall, A., Badrinarayanan, V. and Cipolla, R., 2015. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680.
193. Kwon, Y., Kim, K.I., Tompkin, J., Kim, J.H. and Theobalt, C., 2015. Efficient learning of image super-resolution and compression artifact removal with semi-local Gaussian processes. IEEE transactions on pattern analysis and machine intelligence, 37(9), pp.1792-1805.
194. Li, J., Li, X., Yang, B. and Sun, X., 2014. Segmentation-based image copy-move forgery detection scheme. IEEE transactions on information forensics and security, 10(3), pp.507-518.
195. Liang, Z., Yang, G., Ding, X. and Li, L., 2015. An efficient forgery detection algorithm for object removal by exemplar-based image inpainting. Journal of Visual Communication and Image Representation, 30, pp.75-85.
196. Liao, X., Li, K., Zhu, X. and Liu, K.R., 2020. Robust detection of image operator chain with two-stream convolutional neural network. IEEE Journal of Selected Topics in Signal Processing, 14(5), pp.955-968.
197. Lin, Z., He, J., Tang, X. and Tang, C.K., 2009. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition, 42(11), pp.2492-2501.
198. Liu, Q. and Chen, Z., 2014. Improved approaches with calibrated neighboring joint density to steganalysis and seam-carved forgery detection in JPEG images. ACM Transactions on Intelligent Systems and Technology (TIST), 5(4), pp.1-30.
199. Luo, W., Huang, J. and Qiu, G., 2006, August. Robust detection of region-duplication forgery in digital image. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 4, pp. 746-749). IEEE.
200. Luo, W., Huang, J. and Qiu, G., 2010. JPEG error analysis and its applications to digital image forensics. IEEE Transactions on Information Forensics and Security, 5(3), pp.480-491.
201. Mahdian, B. and Saic, S., 2008. Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security, 3(3), pp.529-538.
202. Manu, V.T. and Mehtre, B.M., 2015, November. Visual artifacts based image splicing detection in uncompressed images. In 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS) (pp. 145-150). IEEE.
203. Mohammed, T.M., Bunk, J., Nataraj, L., Bappy, J.H., Flenner, A., Manjunath, B.S., Chandrasekaran, S., Roy-Chowdhury, A.K. and Peterson, L.A., 2018. Boosting image forgery detection using resampling features and copy-move analysis. Electronic Imaging, 2018(7), pp.118-1.

204. Muhammad, G., Al-Hammadi, M.H., Hussain, M. and Bebis, G., 2014. Image forgery detection using steerable pyramid transform and local binary pattern. Machine Vision and Applications, 25(4), pp.985-995.
205. Nair, V. and Hinton, G.E., 2010, January. Rectified linear units improve restricted boltzmann machines. In Icml.
206. Nataraj, L., Sarkar, A. and Manjunath, B.S., 2010, January. Improving re-sampling detection by adding noise. In Media Forensics and Security II (Vol. 7541, pp. 177-187). SPIE.
207. Nguyen, H.H., Tieu, T.N.D., Nguyen-Son, H.Q., Nozick, V., Yamagishi, J. and Echizen, I., 2018, August. Modular convolutional neural network for discriminating between computer-generated images and photographic images. In Proceedings of the 13th international conference on availability, reliability and security (pp. 1-10).
208.
209. Nist(2016)Nimble. https://www.nist.gov/sites/default/files/documents/2016/11/30/shouldibelieveornot. pdf
210. Pinheiro, P.O., Lin, T.Y., Collobert, R. and Dollár, P., 2016, October. Learning to refine object segments. In European conference on computer vision (pp. 75-91). Springer, Cham.
211. Popescu, A.C. and Farid, H., 2004. Exposing digital forgeries by detecting duplicated image regions.
212. Popescu, A.C. and Farid, H., 2005. Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on signal processing, 53(2), pp.758-767.
213. Qian, Y., Dong, J., Wang, W. and Tan, T., 2015, March. Deep learning for steganalysis via convolutional neural networks. In Media Watermarking, Security, and Forensics 2015 (Vol. 9409, pp. 171-180). SPIE.
214. Rahmouni, N., Nozick, V., Yamagishi, J. and Echizen, I., 2017, December. Distinguishing computer graphics from natural images using convolution neural networks. In 2017 IEEE workshop on information forensics and security (WIFS) (pp. 1-6). IEEE.
215. Rao, Y. and Ni, J., 2016, December. A deep learning approach to detection of splicing and copy-move forgeries in images. In 2016 IEEE international workshop on information forensics and security (WIFS) (pp. 1-6). IEEE.
216. Ryu, S.J. and Lee, H.K., 2014. Estimation of linear transformation by analyzing the periodicity of interpolation. Pattern Recognition Letters, 36, pp.89-99.
217. Salloum, R., Ren, Y. and Kuo, C.C.J., 2018. Image splicing localization using a multi-task fully convolutional network (MFCN). Journal of Visual Communication and Image Representation, 51, pp.201-209.
218. Sarkar, A., Nataraj, L. and Manjunath, B.S., 2009, September. Detection of seam carving and localization of seam insertions in digital images. In Proceedings of the 11th ACM workshop on Multimedia and security (pp. 107-116).
219. Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
220. Shi, Y.Q., Chen, C. and Chen, W., 2007, September. A natural image model approach to splicing detection. In Proceedings of the 9th workshop on Multimedia & security (pp. 51-62).
221. Tralic, D., Zupancic, I., Grgic, S. and Grgic, M., 2013, September. CoMoFoD—New database for copy-move forgery detection. In Proceedings ELMAR-2013 (pp. 49-54). IEEE.

222. Wen, B., Zhu, Y., Subramanian, R., Ng, T.T., Shen, X. and Winkler, S., 2016, September. COVERAGE—A novel database for copy-move forgery detection. In 2016 IEEE international conference on image processing (ICIP) (pp. 161-165). IEEE
223. Verdoliva, L., Cozzolino, D. and Poggi, G., 2014, December. A feature-based approach for image tampering detection and localization. In 2014 IEEE international workshop on information forensics and security (WIFS) (pp. 149-154). IEEE.
224. Wang, Z., Zou, C. and Cai, W., 2020. Small sample classification of hyperspectral remote sensing images based on sequential joint deeping learning model. IEEE Access, 8, pp.71353-71363.
225. Wu, Q., Sun, S.J., Zhu, W., Li, G.H. and Tu, D., 2008, July. Detection of digital doctoring in exemplar-based inpainted images. In 2008 international conference on machine learning and cybernetics (Vol. 3, pp. 1222-1226). IEEE.
226. Xiao, B., Wei, Y., Bi, X., Li, W. and Ma, J., 2020. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. Information Sciences, 511, pp.172-191.
227. Ye, S., Sun, Q. and Chang, E.C., 2007, July. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In 2007 IEEE International Conference on Multimedia and Expo (pp. 12-15). Ieee.
228. You, H., Tian, S., Yu, L. and Lv, Y., 2019. Pixel-level remote sensing image recognition based on bidirectional word vectors. IEEE Transactions on Geoscience and Remote Sensing, 58(2), pp.1281-1293.
229. Zhang, R. and Ni, J., 2020, May. A dense u-net with cross-layer intersection for detection and localization of image forgery. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 2982-2986). IEEE.
230. Zhang, Y., Goh, J., Win, L.L. and Thing, V.L., 2016. Image region forgery detection: A deep learning approach. SG-CRC, 2016, pp.1-11.
231. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C. and Torr, P.H., 2015. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 1529-1537).
232. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. and Oliva, A., 2014. Learning deep features for scene recognition using places database. Advances in neural information processing systems, 27.
233. Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
234. Dong, Q., Gong, S. and Zhu, X., 2018. Imbalanced deep learning by minority class incremental rectification. IEEE transactions on pattern analysis and machine intelligence, 41(6), pp.1367-1381.
235. Kong, D., Tang, J., Zhu, Z., Cheng, J. and Zhao, Y., 2017, July. De-biased dart ensemble model for personalized recommendation. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 553-558). IEEE.
236. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A. and Togneri, R., 2017. Cost-sensitive learning of deep feature representations from imbalanced data. IEEE transactions on neural networks and learning systems, 29(8), pp.3573-3587.
237. Maciejewski, T. and Stefanowski, J., 2011, April. Local neighbourhood extension of SMOTE for mining imbalanced data. In 2011 IEEE symposium on computational intelligence and data mining (CIDM) (pp. 104-111). IEEE.

238. Fernandes, E., Rocha, R.L., Ferreira, B., Carvalho, E., Siravenha, A.C., Gomes, A.C.S., Carvalho, S. and de Souza, C.R., 2018, July. An ensemble of convolutional neural networks for unbalanced datasets: A case study with wagon component inspection. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE.
239. Qian, Y., Liang, Y., Li, M., Feng, G. and Shi, X., 2014. A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing, 143, pp.57-67.
240. Rezaei, M., Yang, H., Harmuth, K. and Meinel, C., 2019, January. Conditional generative adversarial refinement networks for unbalanced medical image semantic segmentation. In 2019 IEEE winter conference on applications of computer vision (WACV) (pp. 1836-1845). IEEE.
241. Liu, S., Zhang, J., Chen, Y., Liu, Y., Qin, Z. and Wan, T., 2019, May. Pixel level data augmentation for semantic image segmentation using generative adversarial networks. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1902-1906). IEEE.
242. Morales, R.R., Domínguez, D., Torres, E. and Sossa, J.H., 2012. Image segmentation through an iterative algorithm of the mean shift. In Advances in Image Segmentation. IntechOpen.
243. Xu, J., Schwing, A.G. and Urtasun, R., 2015. Learning to segment under various forms of weak supervision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3781-3790).
244. Cui, Y., Jia, M., Lin, T.Y., Song, Y. and Belongie, S., 2019. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9268-9277).
245. Rota Bulo, S., Neuhold, G. and Kontschieder, P., 2017. Loss max-pooling for semantic image segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2126-2135).
246. Li, Y. and Vasconcelos, N., 2019. Repair: Removing representation bias by dataset resampling. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9572-9581).
247. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).
248. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
249. Fernandez-Moral, E., Martins, R., Wolf, D. and Rives, P., 2018, June. A new metric for evaluating semantic segmentation: leveraging global and contour accuracy. In 2018 IEEE intelligent vehicles symposium (iv) (pp. 1051-1056). IEEE.
250. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).
251. Mostajabi, M., Yadollahpour, P. and Shakhnarovich, G., 2015. Feedforward semantic segmentation with zoom-out features. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3376-3385)..
252. Zhang, R. and Ni, J., 2020, May. A dense u-net with cross-layer intersection for detection and localization of image forgery. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 2982-2986). IEEE.

253. Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X. and Cheng, Z., 2020. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The lancet, 395(10223), pp.497-506.
254. Zhu, N., Zhang, D., Wang, W., Li, X., Yang, B., Song, J., Zhao, X., Huang, B., Shi, W., Lu, R. and Niu, P., 2020. A novel coronavirus from patients with pneumonia in China, 2019. New England journal of medicine.
255. Wu, J.T., Leung, K. and Leung, G.M., 2020. Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. The Lancet, 395(10225), pp.689-697.
256. Liang, T., 2020. Handbook of COVID-19 prevention and treatment. The First Affiliated Hospital, Zhejiang University School of Medicine. Compiled According to Clinical Experience, 68.
257. Pan, F., Ye, T., Sun, P., Gui, S., Liang, B., Li, L., Zheng, D., Wang, J., Hesketh, R.L., Yang, L. and Zheng, C., 2020. Time course of lung changes on chest CT during recovery from 2019 novel coronavirus (COVID-19) pneumonia. Radiology.
258. Salehi, S., Abedi, A., Balakrishnan, S. and Gholamrezanezhad, A., 2020. Coronavirus disease 2019 (COVID-19): a systematic review of imaging findings in 919 patients. Ajr Am J Roentgenol, 215(1), pp.87-93.
259. Kanne, J.P., 2020. Chest CT findings in 2019 novel coronavirus (2019-nCoV) infections from Wuhan, China: key points for the radiologist. Radiology.
260. Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., Shen, D. and Shi, Y., 2020. Lung infection quantification of COVID-19 in CT images with deep learning. arXiv preprint arXiv:2003.04655.
261. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T. and Ronneberger, O., 2016, October. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In International conference on medical image computing and computer-assisted intervention (pp. 424-432). Springer, Cham.
262. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N. and Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation. In Deep learning in medical image analysis and multimodal learning for clinical decision support (pp. 3-11). Springer, Cham.
263. Milletari, F., Navab, N. and Ahmadi, S.A., 2016, October. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 fourth international conference on 3D vision (3DV) (pp. 565-571). IEEE.
264. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B. and Glocker, B., 2018. Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.
265. Fan, D.P., Zhou, T., Ji, G.P., Zhou, Y., Chen, G., Fu, H., Shen, J. and Shao, L., 2020. Inf-net: Automatic covid-19 lung infection segmentation from ct images. IEEE Transactions on Medical Imaging, 39(8), pp.2626-2637.
266. Al-Antari, M.A., Hua, C.H., Bang, J. and Lee, S., 2021. Fast deep learning computer-aided diagnosis of COVID-19 based on digital chest x-ray images. Applied Intelligence, 51(5), pp.2890-2907.
267. Shorfuzzaman, M. and Hossain, M.S., 2021. MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients. Pattern recognition, 113, p.107700.

268. Wang, Z., Xiao, Y., Li, Y., Zhang, J., Lu, F., Hou, M. and Liu, X., 2021. Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays. Pattern recognition, 110, p.107613.
269. Gong, K., Wu, D., Arru, C.D., Homayounieh, F., Neumark, N., Guan, J., Buch, V., Kim, K., Bizzo, B.C., Ren, H. and Tak, W.Y., 2021. A multi-center study of COVID-19 patient prognosis using deep learning-based CT image analysis and electronic health records. European journal of radiology, 139, p.109583.
270. Oyelade, O.N., Ezugwu, A.E.S. and Chiroma, H., 2021. CovFrameNet: An enhanced deep learning framework for COVID-19 detection. Ieee Access, 9, pp.77905-77919.
271. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
272. Covid-19 ct segmentation dataset. available: https://medicalsegmentation.com/covid19/,"2020.
273. Jun, M., Cheng, G., Yixin, W., Xingle, A., Jiantao, G., Ziqi, Y., Minqing, Z., Xin, L., Xueyuan, D., Shucheng, C. and Hao, W., 2020. COVID-19 CT lung and infection segmentation dataset.
274. Fernandez-Moral, E., Martins, R., Wolf, D. and Rives, P., 2018, June. A new metric for evaluating semantic segmentation: leveraging global and contour accuracy. In 2018 IEEE intelligent vehicles symposium (iv) (pp. 1051-1056). IEEE.
275. Trongtirakul, T., Oulefki, A., Agaian, S. and Chiracharit, W., 2020, April. Enhancement and segmentation of breast thermograms. In Mobile Multimedia/Image Processing, Security, and Applications 2020 (Vol. 11399, pp. 96-107). SPIE.

Résumé

Avec l'augmentation de la quantité de données et la disponibilité de l'infrastructure de calcule puissant, les méthodes Deep Learning (DL) ont connu un grand intérêt en raison de leur bonne performance sur les grands volumes de données et leur capacité d'extraction de caractéristique dans le cadre des données non structurées. Ces méthodes étaient exploitées dans différents sous domaines en vision par ordinateur pour effectuer plusieurs tâches : classification, localisation, détection, et segmentation.

Dans le contexte de la présente thèse, nous nous intéressons à la détection de deux types d'images, les images falsifiées et les images Computed Tomographie (CT) par les méthodes DL, précisément par les réseaux de neurones convolutifs (CNN). Dans ce cadre, nous avons proposé plusieurs approches pour répondre aux différents problèmes liés à l'application des techniques DL en segmentation de ces types d'images. Les approches proposées sont basées essentiellement sur des architectures d'encodeur décodeur, les techniques de régularisation, une fonction de perte adaptative, et les stratégies d'apprentissage transféré. Il est intéressant de noter que les méthodes image-to-image sont utilisées afin de résoudre les différents problèmes liés à la variance élevée, le sur-apprentissage, et la sensibilité des réseaux DL au changement de base de données. En plus, elles permettent de combiner les informations contextuelles obtenues par l'encodeur et l'information spatiale obtenue par le decodeur en en concaténant (additionnant) deux entrées (une de la couche précédente du décodeur et l'autre de la couche symétrique du codeur) cela génère des décisions plus robustes et stables au changement de données. D'autre part, les techniques d'apprentissage transféré et la fonction de perte adaptatives sont utilisées afin de résoudre le problème de sur-apprentissage sur les volumes limités de données.

Mots-clefs : Deep Learning, Réseaux de Neurones Convolutifs, Computed Tomographie, Images falsifiées.

Abstract

With the increase in the amount of data and the availability of powerful computing infrastructure, DL methods have seen great interest due to their good performance on large data volumes and their feature extraction capability in the context of unstructured data. These methods were used in different sub domains of computer vision to perform several tasks: classification, localization, detection, and segmentation.

In the context of this thesis, we are interested in the detection of two types of images, falsified images and computed tomography images by DL methods, precisely by convolutional neural networks (CNN). In this context, we have proposed several approaches to address the various problems related to the application of DL techniques in the segmentation of these types of images. The proposed approaches are based primarily on encoder-decoder architectures, regularization techniques, an adaptive loss function, and transferred learning strategies. It is interesting to note that image-to-image methods are used in order to solve the various problems related to high variance, over-fitting, and sensitivity of DL networks to database change. In addition, they make it possible to combine the contextual information obtained by the encoder and the spatial information obtained by the decoder by concatenating (adding) two inputs (one from the previous layer of the decoder and the other from the symmetrical layer of the encoder) this generates more robust and stable decisions to changing data. On the other hand, transferred learning and adaptive loss function techniques are used to solve the problem of over-learning on limited volumes of data.

Key Words: Deep Learning, Convolutional Neural Network, Computed Tomography images, Falsified images.