République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Master Thesis

Major: Intelligent Model and Decision

Theme

Fully convolutional Networks for Semantic Segmentation

Realized by:

- **BENSAAD Meriem**
- **DERDARINE Chaimaa**

Defended on: 04 - 07 - 2022 In front of the jury:

- **Mr. BENAZZOUZ Mortada** **(President)**
- **Mr. HADJILA Fethallah** **(Supervisor)**
- **Mr. BENZIAN Yaghmorasen** **(Examiner)**

2021/2022

Acknowledgment

“**Praise to the Almighty, the light of heavens and earth**”

We first give thanks to Allah, who gave us power, courage, and guidance to accomplish this work.

Words cannot express our gratitude to our professor and supervisor Mr. Hadjila Fethallah, who generously provided knowledge and expertise, and for his invaluable patience and feedback.

We would like to extend our sincere thanks to the members of the Jury, who accepted to evaluate our modest work, as well as all the professors who guided us through our university studies.

Finally, it would be ungrateful, not to acknowledge our families, for their endless support and encouragement.

Thank you.

Dedication

I dedicate this work to the Almighty, the master, and the ultimate guide.

A special feeling of gratitude, to my beloved "**Oumi**" and "**Walidi**", for their constant support and uncontrollably love, for the years they spend and the efforts they did to see me here, and for the prayers that were answered, I am forever thankful and appreciative.

For my brothers Ismail, Said and, Taha, their reliable presence and backup, and my cousins for being more than sisters.

For Chaimaa Derdarine, the journey partner, it would have been senseless without your company, a brave and compassionate spirit that I was fortunate to meet.

For my friends throughout my entire academic career, especially high school and college friends, it has been a memorable time.

Finally, to our supervisor Mr.Hadjila, for his enormous giving and his appreciation for the profession.

Bensaad Meriem

Dedication

To my mom « Aisha »: No dedication, dearest mother, could express the depth of feelings to you, your innumerable sacrifices and your devotion have been an encouragement to me. You have watched over my steps, and have covered me with tenderness, your prayers were great help to me to carry out my studies.

To my dad « Mohammed »: Who has been patient, understanding and encouraging, his paternal warmth was and will always be a great comfort to me. I implore the Almighty, to grant you good health, a long life and much happiness

To my brother « Yousef »: Your kindness, your precious support and encouragement through my years of study, your love was for me the example of perseverance. I find in you the advice of a brother and the support of a friend. May God protect you, grant your health, success and happiness.

To mama « Samira »: I would like to thank you, no words could describe your effort and support for me, you are truly an amazing person. God to bless you, and I love you**.**

To my sisters « Rahma » « Ghofrane» and **« Meriem** »: my backup, my strength, you're so special to me, may Allah keep our bond under his blessings and protection and to my little one **« Mohammed »** I love you little brother.

To my partner and best friend « Meriem »: We went through ups and downs, and in every step and challenge I made sure you were the right friend for never failing me, best choice that I ever made thank you.

To my best friends « **Amal »** and « **Aya »,** for all the happy moments spent together, with my sincere wishes for happiness, health and prosperity.

For all the family **Derdarine** and **Moulay el boudkhili.**

Finally, I would like to thank me for believing on me, being patient and persistent, this would be considered as my first achievement for more success Inchallah.

Derdarine Chaimaa

Content

List of Figures

List of Tables

List of Equations

General Introduction

Context

Recently, humans live in an advanced environment using the evolution of techniques and knowledge in many fields, such as medicine, physics, mathematics, and more recently computer vision.

One of the particularities of humans is to be able to acquire images, via the eye, to be able to interpret them via the brain. The challenge here of artificial vision is to allow a computer to “see” like humans, to retrieve information. Thus, the machine will then be able to recognize shapes or to separate an image into different distinct and coherent zones.

In the field of computer vision, computers or machines are designed to gain a high level of understanding of input digital images or videos, in order to automatize tasks that the human visual system can perform. Image processing can be considered as a subset of Computer Vision that performs low-level operations on images like resizing, lighting, histogram correction etc. It aims to get an improved version and/or to extract some useful information from it, this information is usually stored, processed, indexed by computer systems, which makes their recovery a very rapid task.

In order to realize the automatic understanding of an image, segmentation turns out to be one of the key steps of image processing, which is part of the major research themes.

It consists of extracting objects present in an image by dividing it into multiple components, such that each of them is meaningful. The process of semantic segmentation (SS) can be seen as a labeling operation, which points to assigning a label to each pixel. SS allows to establish a compact and representative description of the image content.

Problematic

Despite the multitude of methods proposed for image segmentation, it's still considered as challenging problem that face a lot of obstacles due to the diversity of images, noise present in images, low contrast, occlusions, complexity of the regions surrounding the target, and the variability of the shapes to be segmented. In addition, the same image can have various possible segmentations.

When it comes to semantic segmentation, A number of vision applications like recognition tasks require segmentation of objects in images. The presence of overlap between objects of the same image poses a major challenge; identifying the hidden boundaries and grouping contours that belong to same objects are some of the complexities that overlap introduces. Another challenge that faces semantic segmentation is the presence of imbalanced objects, where the surface of some objects of interest is less than the surface of the background, all those challenges could truly effect the result of segmentation.

Contribution

The existing methods for semantic image segmentation can be classified according to the target to be achieved. For instance, we can distinguish types such as thresholding, clustering, region growing, and pixel classification.

Usually, traditional segmentation methods (i.e., region growing) return a low performance when we process situations that require large database of images or even detecting high-level features.

In this work, we are dealing with a huge dataset, termed PASCAL VOC 2012. So basically our objective is to get the appropriate and meaningful segmentation for most of objects in images of this dataset by implementing an efficient method of semantic segmentation, based on fully convolutional neural networks; more precisely, we propose 03 models, namely: Standard U-Net, Inception-based U-Net, and an Inception-based U-Net with changes in terms of channels.

Manuscript Plan:

Our work is constituted of three chapters:

Chapter One: Semantic Segmentation

This chapter presents an overview of the field of semantic image segmentation; we will take a view on various current challenges, in addition to the different methods used in it; before concluding with use cases of semantic segmentation.

Chapter Two: Machine Learning, Traditional and Deep Learning

This chapter aims to present the concept of Machine Learning and Deep Learning, as well as the main points of each, accompanied by a brief comparison and ends with an introduction to the FCNs and more specifically U-Net architecture that will be realized in our work.

Chapter Three: Proposed Model, design, and Implementation

In this last section, we discuss first the environment of development, the used datasets as well as the design of our proposed models, along with their implementation and a discussion of the obtained results.
At last, we conclude with a general conclusion.

Chapter I Semantic Segmentation

I.1 Introduction

Image processing is a branch of computer science that focuses on digital images and their transformations to improve their quality or extract information from them without human assistance**.**

Image segmentation can be therefore, considered an important step in image processing and computer vision.

According to H. Boveiri & V. Parihar [1]: Image Segmentation is the process of dividing a digital image into a group of pixels or regions. Thus, the main target of segmentation is to change the image representation into something more significant and easier to extract information. Meaning that in segmentation a value is given to every pixel in an image such that pixel with the same values Sharing specific features such as color, intensity or texture in an in a specific area.

In most cases, Image segmentation is used to locate objects and boundaries (lines, curves, etc.) in images. To get a set of segments that cover the whole image combined, or a set of contours extracted from the image

In this chapter, we will focus on fundamental notions of semantic segmentation, its approaches, and use cases.

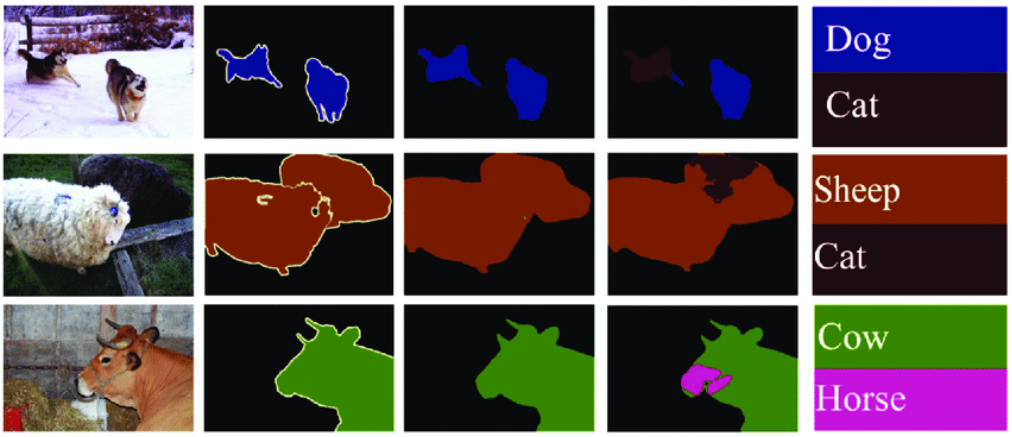

Figure I. 1 Segmentation of a color image. [2]

In the figure above, it is obvious that each object in the image (dog, sheep, and cow) is defined by a color (homogeneous regions). However, we notice that there may be defects in object recognition, so there can be confusion between areas, as it is shown in the figure, it considered a part from the cow as a different class and gave it a different color (horse).

It is worth noting that image segmentation is a critical component of image analysis and pattern recognition, and still known as one of the most difficult tasks in image processing [1].

I.2 Link between recognition and segmentation

To get the right segmentation of each object and then simply send it to deal with recognition, seems perfect but it is so far from reality, as we all know that sometimes segmentation algorithms do not produce a correct result. Therefore, the question remains as to whether this affects recognition.

According to Rabinovich & al [3] [4] that, « Recognition based on segmentation, where they stated that when it's preceded by segmentation is much better for multi-class and single-object recognition».

In their approach, the image is segmented using low-level signs of brightness, texture, color, or motion. These segments are passed to the recognition engine.

This approach has several gaps based on that a limited number of segments can catch only the necessary information and that is false, due to different appearances of objects (colors, shapes, sizes, and texture, it can be in many poses).

Another model where recognition and segmentation go together is Textonboost [5], developed at Microsoft Research Center, Cambridge, this model involves shape, appearance, and context at the same time to recognize and segment an image. In a contradictory way, all those operations are located on the pixel level (recognition and segmentation), they introduce new features called texture layout filters that capture texture, spatial information, and textural context altogether.

The main goal is to detect objects that belong to various categories and also those which belong to the same category but with different appearances, Where the difficulty lies in finding category descriptions that can represent these differences, and that's what are we looking for.

I.3 Challenges

Naturally, humans know very well how to separate objects in an image, based on high-level knowledge that allows them to detect in the image what seems interesting to them. This is not the case for artificial systems.

Faced with the multitude of methods proposed for segmentation, it is still considered a problem that it does not have a universal solution, it can only be applied unless it adheres to a set of standards including precision and robustness. Without forgetting, that it is important to take in consideration the context of use envisaged to design a method of segmentation, which will lead to a good interpretation.

The obstacles encountered during segmentation are numerous due to the diversity of images and the large number of possible applications [6] [7], these difficulties can be stated as follows:

The noise present in the image, the low contrast, the occlusions, the complexity of the regions surrounding the target, and the variability of the shapes to be segmented.

Moreover, the same image can have several possible segmentations [8], according to Tu and Zhu [9]:

- It is quite difficult to model a large number of visual patterns of images.
- The perception is not clear. Quite often, we can propose different logical interpretations for the same image.

Those challenges could be automatically removed through segmentation methods that are based on machine learning techniques which enable to learn different representations of objects from given images; pixel-level illustrations are crucial to reach the accuracy of the best performing methods which for many applications are restricted and even unavailable. [10] [11].

I.4 Definitions

Semantic segmentation is an approach that aims to assign a categorical label to each pixel in an image [12]. There is no distinction between two instances of the same object. It treats multiple objects of the same class as a single entity [12] where all pixels that belong to a particular class hold the same value.

Nevertheless, it can be easily confused with instance segmentation where the only difference is that in instance segmentation, each instance of the same object within the image is assigned to a unique label.

For example, in the figure below, all sheep are segmented as one object, which means that the same label is given to all the pixels of them in the image. As for Instance Segmentation, each sheep in the figure is segmented as an individual object, unlike semantic segmentation, which defines the same color for all sheep that belong to the same class.

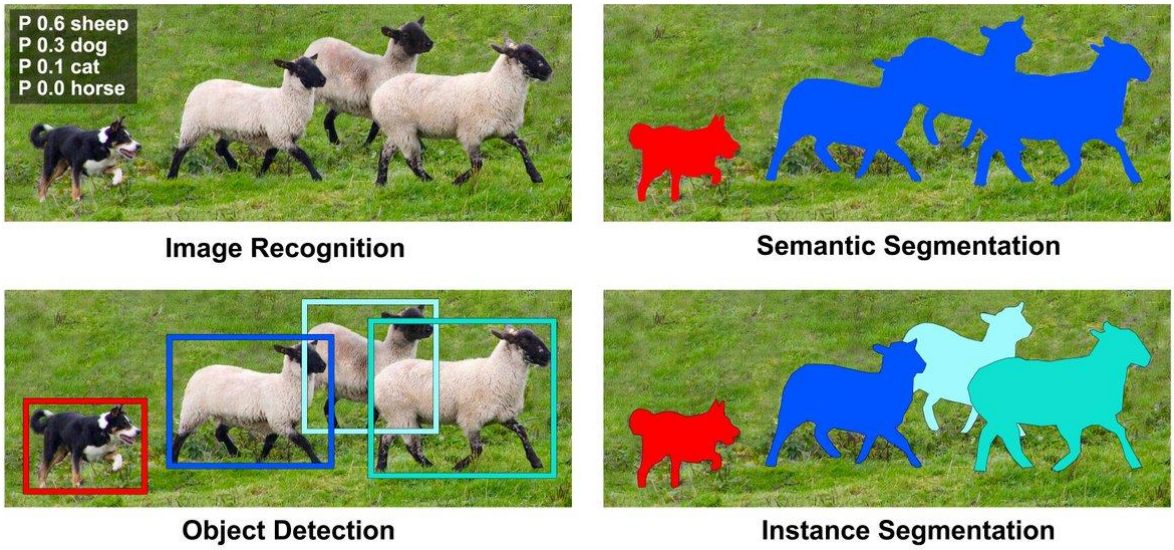

Figure I.2 Semantic segmentation vs instance segmentation [14]

Segmentation is different from object detection, it not only determines if an object exists or not in the image but also learns the spatial information of the image, it can also identify the shape of every object present in the image. An object can also include sky or background, which means boundary localization is critical in the process of semantic segmentation [15].

Segmentation can be used in applications that involve a particular kind of object recognition such as face recognition, fingerprint recognition, traffic control systems, locating objects in satellite images, and industry applications (Airport security systems, Automatic car assembly in robotic vision).

I.5 Image Segmentation approaches

There are many image segmentation methods, which divide the image into multiple parts based on some image features like pixel intensity value, color, texture, etc.

According to Goutam Sen & al 2012 [16], segmentation could be divided into spectral and spatial segmentation. The spectral segmentation method assigns a pixel to a region according to spectral similarities. It only depends on spectral features and considers only one pixel at a time.

The Spatial segmentation method classifies an image pixel based on a relationship with the pixels surrounding them.

Therefore, we can consider that Thresholding, clustering, neural networks segmentation, fuzzy segmentation, and edge-based segmentation belong to the spectral segmentation, whereas the rest of them belong to the spatial segmentation (region growing, splitting and merging, and object-based approach). [16]

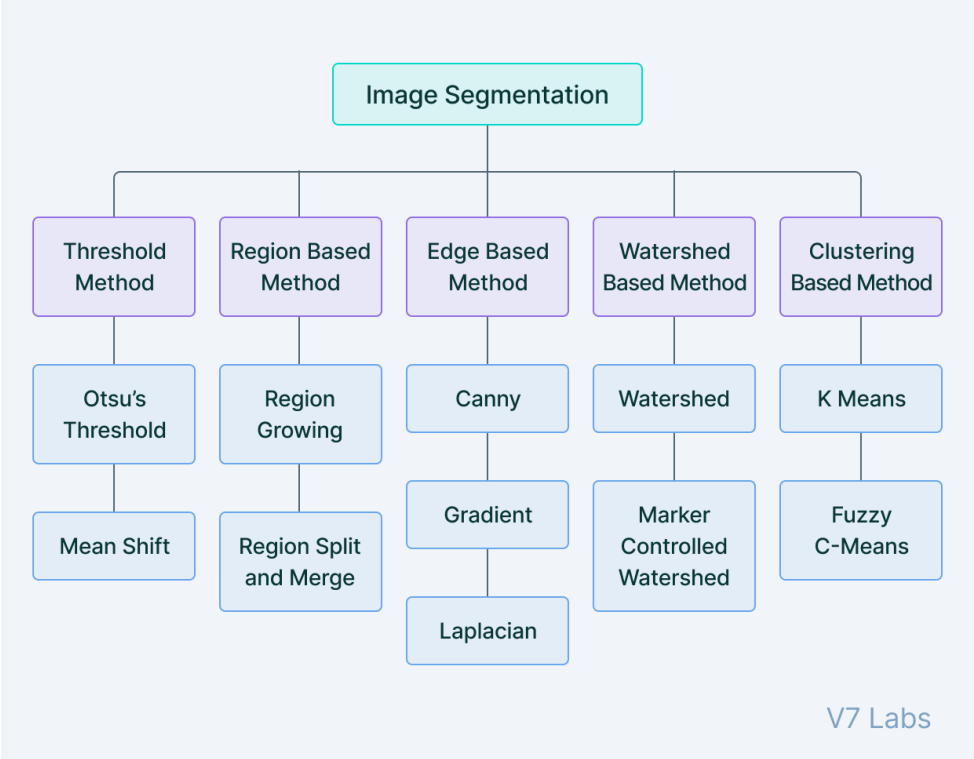

Figure I.3 Image segmentation techniques [V7 Labs] [17]

I.5.1 Region-Based Segmentation

Region-based segmentation algorithms divide the image into sections with similar features; these components are a group of pixels.

The algorithm finds these groups by first locating a starting point, which could be a small section or a large part of the input image. Then it would add more pixels to them or shrink them so it can merge them with other seed points. [18]

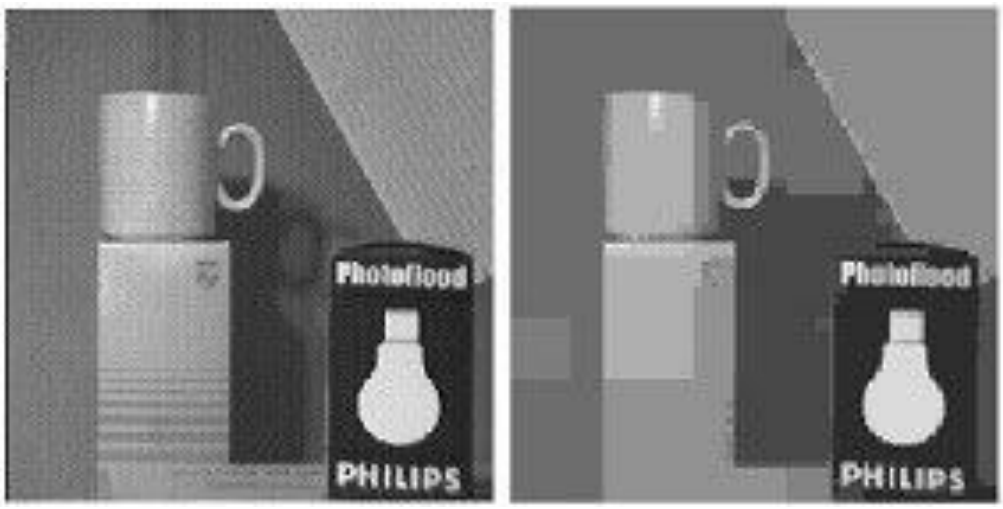

Figure I.4 Examples of region-based segmentation [19]

We can classify region-based segmentation into the following categories:

Region Growing

A method that groups pixels in the whole image into sub-regions or larger regions based on predefined standards [20]. We first start with a smaller set of pixels, then we iteratively merge based on certain predefined similarity constraints, this algorithm would pick an arbitrary seed pixel in the image, compare it with neighboring pixels and start increasing the region by finding matches with the original point.

When a particular region cannot grow anymore, the algorithm will pick another seed pixel that might not belong to any existing region. One region can have too many attributes causing it to take over most of the image.

This algorithm is useful for images that have a lot of noise, as the noise would make it difficult to find edges.

Region Splitting and Merging

This method (a region splitting and merging) would perform two actions together, splitting and merging portions of the image.

First, starting with splitting the image into regions that have similar attributes and merge the adjacent parts which, are similar to each other, In region splitting, the

algorithm considers the entire image while in region growth, it would focus on a particular point.

The region splitting and merging method divides the image into different portions and then matches them according to its predetermined conditions.

Approach based on mathematical morphology

The most known segmentation method based on mathematical morphology is the watershed technique (LPE) proposed by DIJABEL et LANTUEJOUL [21][22].

This technique consists of enlarging together all the regions until the image is completely segmented; it handles an image as if it was a topographic map. It considers the brightness of a pixel as its height and finds the lines that run along the top of those ridges, in this case, the image can be represented as a three-dimensional terrain.

The principle then is to progressively fill each basin of the terrain with water, each basin represents a region, and when the water rises and when two basins meet the meeting line (watershed) is marked as a borderline between the two regions.

This method does not apply to the original image but to its morphological gradient, over-segmentation is considered one of the main problems of this method.

1.5.2 Edge-Based Segmentation

In image processing, Contour-based segmentation is one of the most popular applications of segmentation; it is based on identifying the borders of various objects in an image.

It allows finding the features of the different objects present in the image as edges contain a lot of information that can be useful and remove unwanted and unnecessary information from the image. It significantly reduces the size of the image, making it easier to analyze. Edge-based image segmentation algorithms aim to detect the edges of an image, based on different discontinuities in gray level, color, texture, saturation, brightness, and contrast. To improve the results obtained all edges should be connected in edge chains that more accurately match the borders of the image [18].

Figure I.5 An example of edge-based segmentation**.** [23]

In what follows we present the different methods suitable for edge detection in grayscale images:

- **Derivative methods**

Derivative methods are most commonly used to detect intensity transitions by using the first and second derivatives. At each position, an operator is applied to detect significant transitions at the chosen discontinuity attribute. The result is a binary image consisting of edge and non-edge points [24]

Several contour extraction techniques can be classified as follows [25]:

- **Gradient-based algorithm**

The first derivative is used to calculate the gradient. it's a vector characterized by an amplitude and a direction. There are several gradient operators among them: Roberts, Prewitt, and Sobel masks. [24]

$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial Z}dz \qquad \textbf{Eq. I.1}$$

- **Sobel and Prewitt operators**

The 'Sobel' and 'Prewitt' operators are used to estimating the norm of the two-

dimensional gradient of a greyscale image. These operators consist of a pair of 3×3 convolution masks. The separate application of each of the masks gives an estimate of the horizontal and vertical components of the gradient by simple linear filtering with a 3×3 mask.

- **Roberts' operator (1965)**

The Roberts detector allows the calculation of the two-dimensional gradient of an image in a simple and fast way. The principle does not differ much from that of the operators of "Prewitt" and "Sobel" operators. [24]

The following figure shows the edges detected by these filters:

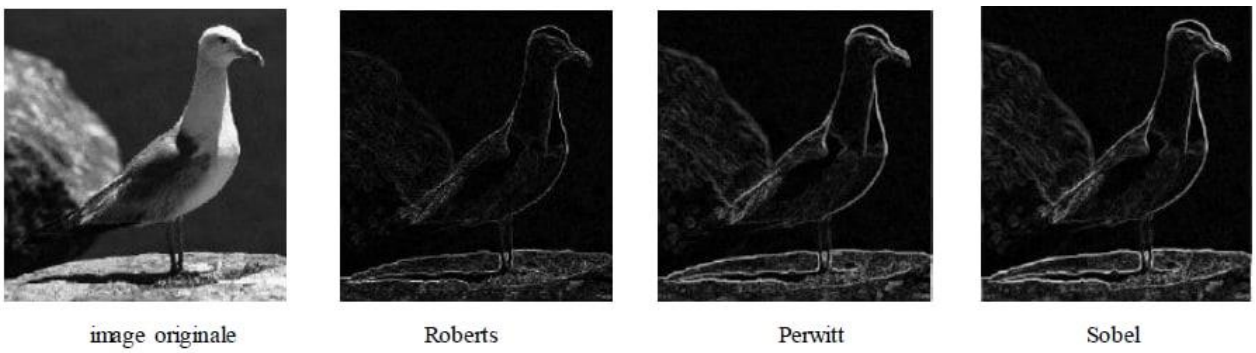

Figure I.6 Edge detection with different filters [24]

- **Algorithms based on the Laplacian**

In this approach, the extraction of the contours is based on the calculation of the second derivatives; it is defined as the divergence of the gradient. We distinguish the scalar Laplacian:

$$(f) = div\big(grad(f)\big) = \Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial Z^2} \qquad \textbf{Eq. I.2}$$

Figure I.7 Original image (left), Edge detected by Laplacian filter (Right). [24]

❖ **Analytical methods**

Canny and Deriche's approach

It consists to find an optimal filter satisfying the following 3 constraints:

- **Good detection**: low probability of forgetting a real edge point and detecting the wrong one.
- **Good location**: edge points should be as close as possible to their actual position in the image.
- **A unique answer**: a filter should detect an edge point only once.

The Canny contour detector is the most widely used. It is based on three norms: detection (robust to noise), localization (precision of the localization for contour point), and unicity [27].

Deriche proposed another filter that allows a simplification of its implementation, which meets the same quality criteria as Canny's. [28]

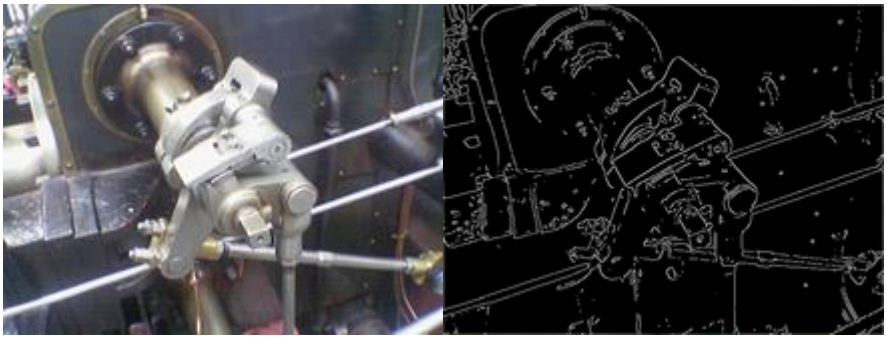

Figure I.8 Applying the Canny filter to an image. [28]

❖ **Deformable method**

Deformable models are introduced by Kass. Known as “snakes” or “active contours”. The objective behind them is to detect objects in an image using curve evolution techniques. The idea is to start from an initial curve, usually a square or a circle, and through an iterative process it will deform the curve at each iteration until its final position.[29]

The main advantage of segmentation methods based on the edge approach is that they minimize the number of operations required when the process is iterated on a series of images that are not very different from one another. Indeed, once the contours of the regions have been found in the first image, the application of the deformable model to the next image is more efficient than recalculating everything, if the difference between the images is small.

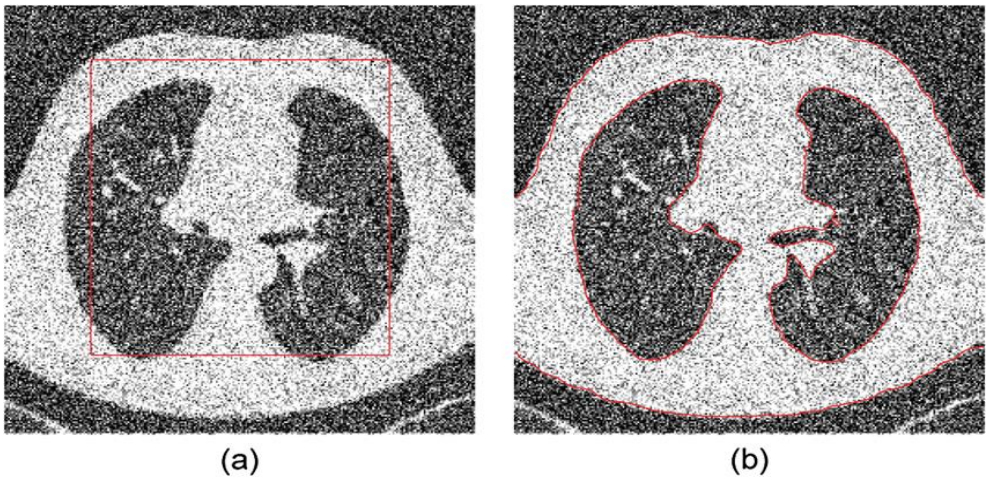

Figure I.9 Active contour segmentation [30]

I.5.3 Clustering-Based Segmentation Algorithms

These methods are based on unsupervised clustering algorithms; they help to find hidden data in the image that might not be visible to normal vision. This hidden data includes information such as clusters, structures, shades, etc. [16]
So basically, a clustering algorithm segment the image into clusters (disjoint groups) of pixels that have similar features by separating the data elements into clusters where the elements are more similar in a cluster than the elements present in others.

Fuzzy C-means (FCM), k-means, and improved k-means algorithms, are some of the popular clustering algorithms.

In image segmentation, the most used is k-means because of its simplicity and efficiency, as for Improved K-Means minimizes the number of iterations used in

the K-Means algorithm. In contrast, the FCM algorithm places pixels into different classes based on their different membership degrees.

Here are the most crucial clustering algorithms for image segmentation:

K-means Clustering

K-means is a simple unsupervised Machine Learning algorithm. It classifies an image through a specific number of clusters. It starts the process by dividing the image space into k pixels that represent k group centroids.

Then they assign each object to the group based on the distance between them and the centroid. When the algorithm has assigned all pixels to all the clusters, it can move and reassign the centroids.

When the algorithm converges, we will have areas in the image segmented into K groups where the constituent pixels represent levels of similarity.

Fuzzy C Means

With the fuzzy c-means clustering method, the pixels in the image can get clustered in multiple clusters. This means a pixel can belong to more than one cluster. However,

every pixel would have varying levels of similarities with every cluster. The fuzzy C-means algorithm has an optimization function that affects the accuracy of the results, and the convergence of the latter depends on the minimization. In the case of convergence, the areas in the image will be segmented into C groups, where the constituent pixels of a cluster represent certain levels of similarity, as well as a certain degree of association with other clusters.

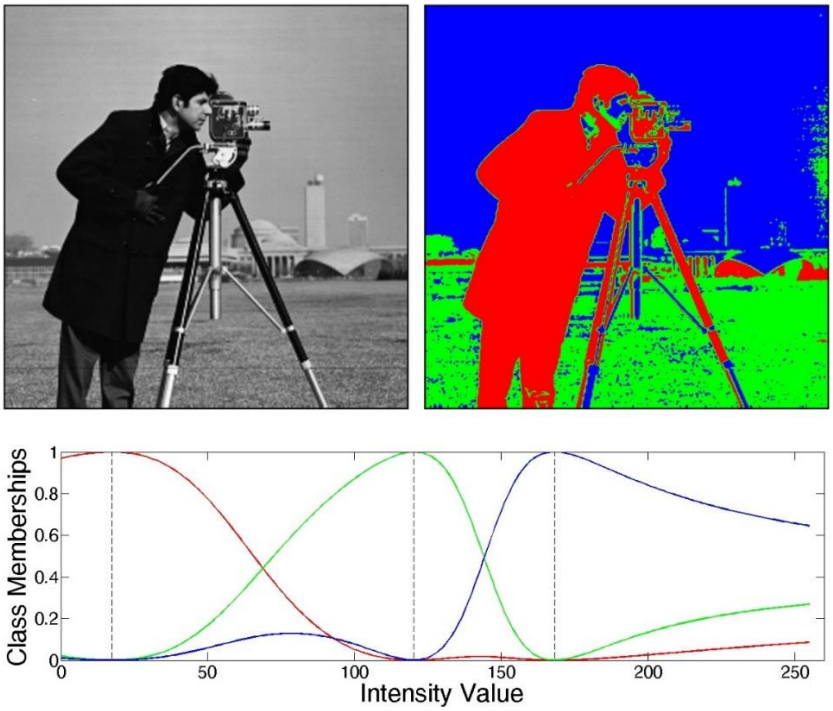

Figure I.10 Fast fuzzy c-means image segmentation.

I.5.4 Thresholding-based Segmentation

Image segmentation via thresholding is a simple but powerful process for segmenting images that have light objects on a dark background [32].

It divides pixels in an image by comparing the intensity of the pixel to a threshold value. It is useful when the required object is more intense than the background (unnecessary parts) [18].

The threshold value (T) can be considered as a constant However, it would only work if the image has very little noise (unnecessary information and data). It could be constant or dynamic according to the requirements.

Figure I.11 An example of threshold-based segmentation [31]

We can classify thresholding segmentation in the following categories:

Simple Thresholding

In this method, you replace image pixels with white or black. It is therefore a binary approach. Now, if the intensity of a pixel at a particular position is less than the threshold value, you will replace it with black. However, if it is above the threshold, you will replace it with white. It is simple thresholding and particularly suitable for beginners in image segmentation**.**

Otsu's Binarization

In simple thresholding, a constant threshold value was chosen, and used to perform image segmentation. How do you determine, though, that the value you chose was the correct value? Although the simplest method for this is to test different values and choose one, this is not the most effective.

By using Otsu binarization, we will take an image with a histogram having two peaks, one for the foreground and one for the background then choose the approximate value of the middle of those peaks as the threshold value. It will be calculated from the image's histogram if the image is bimodal.

This process is quite popular for scanning documents, recognizing patterns, and removing unnecessary colors from a file However, it has many limitations. You cannot use it for images that are not bimodal, (images whose histograms have multiple peaks).

Adaptive Thresholding

Having a constant threshold value may not be an appropriate approach to take with each image; some images have different background levels and conditions in various useable areas that affect their properties.

Therefore, instead of using one constant threshold value for performing segmentation on the entire image so we need to use an adaptive approach that can change the threshold for various image components by keeping the threshold value variable, different threshold values for different sections of an image will be retained. It works well with images that have different illumination conditions.

The idea behind this is to use an algorithm that segments the image into smaller sections and calculates the threshold value for each of them, by calculating the threshold value T. It can be either the average of the neighborhood area or the weighted sum of the neighborhood values.

I.5.5 Machine Learning-based Segmentation approach

With the development of machine learning methods, learning-based approaches have begun to increase their effectiveness in Object detection and segmentation methods tasks. Neutral networks made massive progress because a large amount of data is available, and the computing power is getting faster.

During the last decades, works in deep learning dealing with image segmentation have been significantly improved by using numerous artificial neural network architectures such as U-Net [33], and Fully Convolutional Neural Network [34]. These techniques based on the simulation of the learning process for decision with top accuracy for the various image segmentation task.

Artificial Neural Networks for Segmentation

It is the approach of segmenting the images via artificial neural networks, which use AI to analyze an image and identify its different components such as faces, objects, and text…; it is different from conventional segmentation algorithms.

An image is firstly designed into a Neural Network. Where every Neuron represents a pixel [35][36], the neural network was trained with a sample set to define the connection and weights between nodes. Then the new images were segmented with the trained neural network, it covered two important steps: feature extraction and image segmentation based on the neural network. Feature extraction is very critical as it determines input data of neural networks [37], extract features from images, such that they become appropriate for segmentation then became the input of the neural network.

Fully Convolutional Neural Network

Long et al. [34] proposed one of the first deep learning works for semantic image segmentation, using a fully convolutional network (FCN). Unlike the classical recognition [38][39] and segmentation methods, FCN can be seen as a CNN variant, which can extract the features of objects in the image.

By replacing all fully connected layers in the traditional CNN model with fully convolutional layers, the obtained model can achieve a significant improvement in segmentation accuracy compared with the CNN-based methods. With the help of deep learning architectures such as U-Net which is already based on FCN network most of the semantic segmentation issues could be solved, it can capture both the features of the context also the localization, which helps in predicting the image pixel by pixel, that achieves good performance and high-quality results on very different segmentation applications.

I.6 Applications of Segmentation

I.6.1 Autonomous driving

For complex driving scenarios, it is beneficial for the autonomous vehicle perception system to provide a more detailed understanding of its surrounding via sensors. To visualize the environment and recognize various objects, or other cars, free space on the roads, and also to detect traffic signs, all this happens through semantic segmentation, so that it can make decisions and navigate accordingly [40].

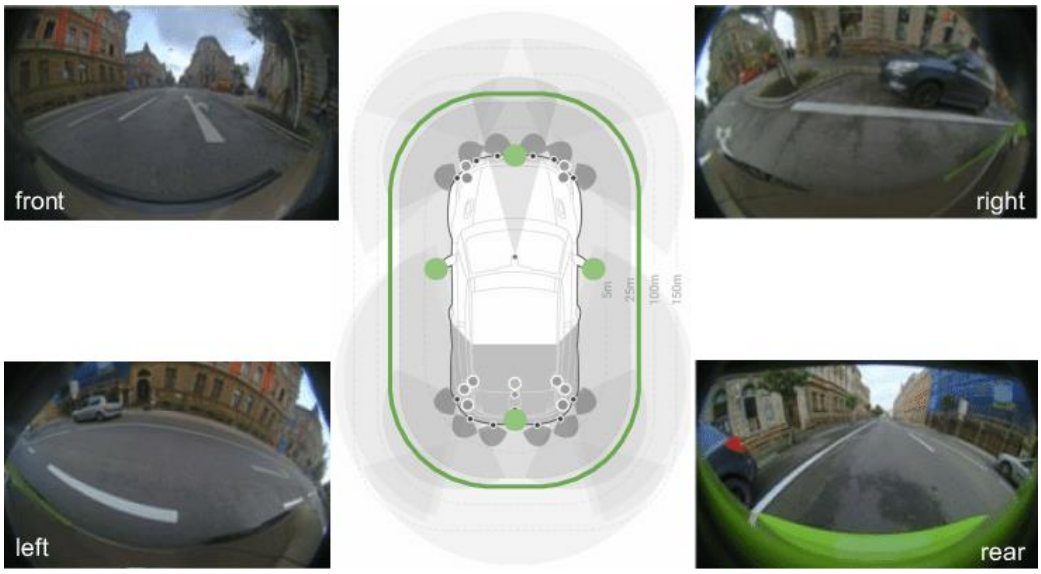

Figure I.12 Sample images from the surround-view camera network showing near field sensing and wide field of view. [41]

I.6.2 Facial Segmentation

Face labeling could be useful in a variety of scenarios. Huang et al. [42] found that high-level features might be predicted using simple learning algorithms, starting with face labeling and classifying them into skin, hair, eyes, nose, mouth, and background regions. In their vision, Face segmentation, provide important information for face recognition because of intermediate level features, and in many facial applications of computer vision, such as gender, estimation of age, skin color, ethnicity…

In all such applications, the precise knowledge - at the pixel level - of face segments is critical.

Finally, we can mention that variations in lighting conditions, facial expressions, face orientation and image resolution are among the factors that influence face segmentation dataset and model development. [41]

Figure I.13 Face Segmentation [43]

I.6.3 Medical image diagnosis

Based on what Dzung L & al [43] said, Magnetic Resonance Imaging (MRI), Computed Tomography (CT), digital mammography, and other imaging modalities, it is critical in the field of anatomy, diagnosis, and treatment planning.

The increasing number of these medical images needed computers to make it easier during processing and analysis. In particular, Computer algorithms for defining anatomical structures and other areas of interest are becoming more essential in helping and automating certain radiological procedures, known as image segmentation algorithms, and they contribute in:

1. The quantification of tissue volumes

2. Diagnosis.

3. Localization of pathology.

4. Study of anatomical structure.

5. Treatment planning.

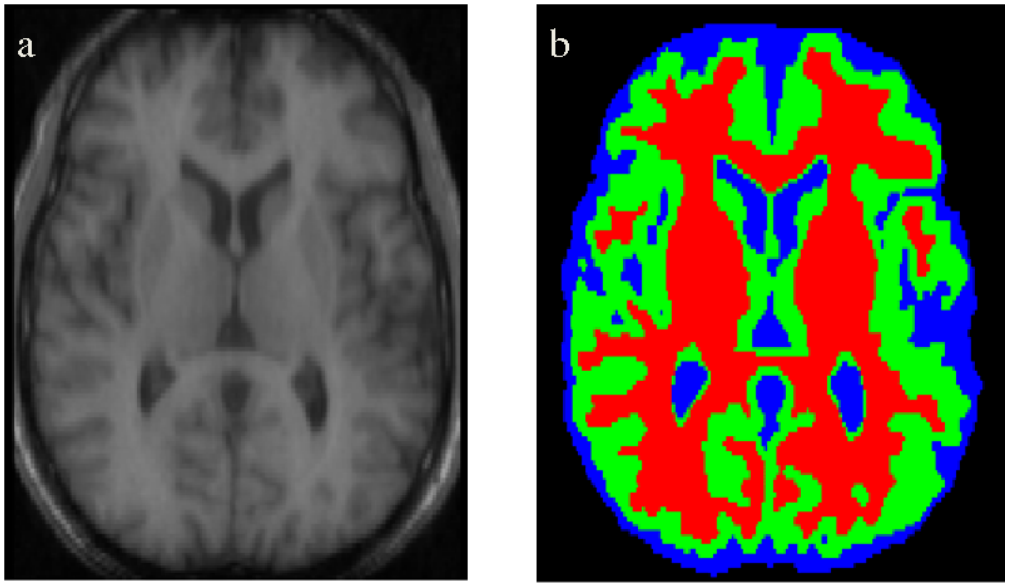

Figure I.14 Review of Medical image segmentation. [43]

I.6.4 Satellite (Aerial) Image Processing

Satellite imagery segmentation is one of the main tasks in remote sensing applications such as city planning, climate change research [44], or military purposes.
Color segmentation may also be used to track land cover changes over time. [45] The most notable goal of remote sensing data analysis is object detection, based on image segmentation. [46]

So basically the image segmentation process is to partition the image into a set of segments and map the individual regions to the corresponding real-world objects, like rivers, fields, roads, and others. [47]

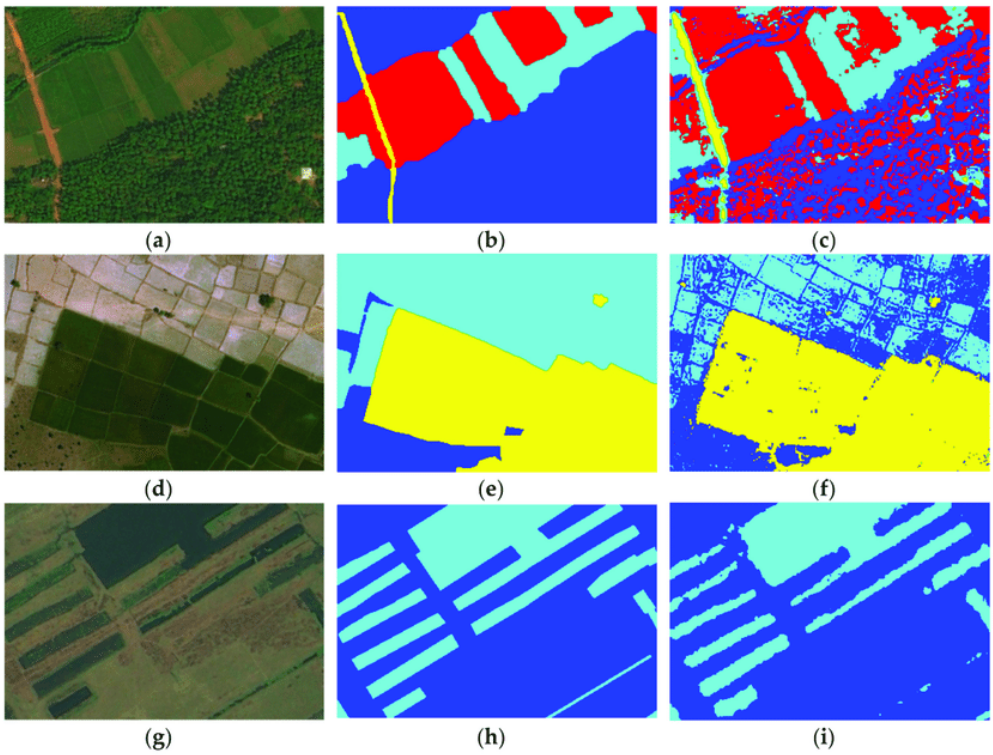

Figure I.15 Semantic segmentation of satellite/aerial images [47]

I.7 conclusion

In this chapter, we have defined image segmentation by developing briefly the different approaches used to perform segmentation; each approach uses different algorithmic, which is in full progress.

Various algorithms for image segmentation have been developed for specific use. We can mention that so far, there is no specific algorithm for all applications or all categories of images.

The main difference between conventional image segmentation and semantic segmentation is the utilization of semantic features in the process. Conventional methods for image segmentation such as thresholding, clustering, region growing, etc. use low-level features (i.e., edges) to locate object boundaries in images. Thus, in situations where the semantic information of an image is necessary for pixel-wise segmentation, these methods usually return a poor performance

Recently, deep learning (DL) networks have produced a new generation of image segmentation models such as U-Net, Fully Convolutional Neural Network, ParseNet 2015, and SegNet 2017 with remarkable performance and the highest accuracy rates compared to conventional methods.

In the next chapter, we will provide the concepts of Machine Learning and few of the corresponding algorithms, we will dive into deep learning methods including CNNs and fully convolutional pixel-labeling networks ending with a brief conclusion.

Chapter II Machine Learning, Traditional and Deep Learning

II.1 Introduction

At present, Machine Learning and Deep Learning are receiving considerable attention from specialists as well as non-specialists due to the significant breakthroughs they have made in the field of artificial intelligence.

In this chapter, we will discuss the concept of machine learning and some of its traditional algorithms. We will also cover deep learning, a brief background, and its approaches and important notions, particularly Convolutional Neural Networks, all the way to the most essential point in our manuscript: Fully Convolutional Networks, which are currently considered the ideal solution applied for Semantic Segmentation.

II.2 Machine Learning

II.2.1 Definitions

Many definitions were given to Machine learning (ML), we chose those of: Stephanie Chan:

"Machine Learning is a tool comprising a subset of artificial intelligence that enables the goals of artificial intelligence to be achieved.
ML has piqued attention for its broad range of uses in daily life from personalized online recommendations for videos and news to self-driving cars." [48]

And Jo-Hsuan Wu:

"ML uses existing data to train a computer to recognize a specific pattern or predict a specific outcome in a new data set" [49]

And finally, Tom Mitchell:

"*Machine learning (ML) seen as a part of Artificial Intelligence is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks "* [50]

II.2.2 Traditional Machine Learning

In classic Machine Learning, we notice that the models of "weak learners" and strong learners" are playing a major role in ensemble learning techniques. In what follows, we will detail these concepts more deeply

- **Weak learners**

"*A weak learner produces a classifier which is only slightly more accurate than random classification."* [51]

In the following section, a few weak learners will be briefly stated:

K-Nearest Neighbors

K-Nearest Neighbor [Fix and Hodges (1957)], is an extension of the Nearest Neighbor method, a very basic yet very efficient method for models classifications.

When an unknown pattern is to be classified, it chooses the class of the k nearest example among all the training samples as measured by a given distance. [52]

Decision Trees

Decision Tree is learning algorithm that mostly used for classification problems, with a pre-defined target variable, it splits samples into two or more sets based on their homogeneity.

It can be of two types: Categorical variable, and continuous variable decision tree. [53]

Decision trees define the final decision using a model graph (tree). The branches of this tree depend on the condition of the node above (YES or NO). The deeper we go

into the tree, the more conditions we combine. The figure below illustrates this procedure.

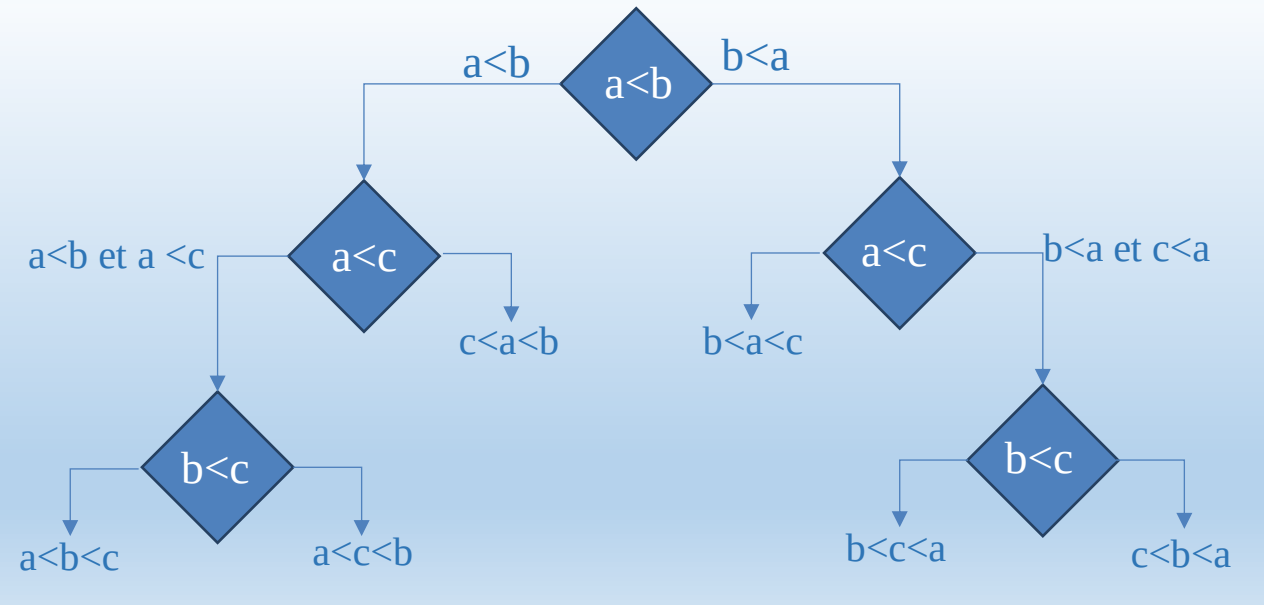

Figure II.1 Decision Trees illustration

CART

Classification and regression trees or CART for short, is one of the new statistical techniques of the computer age [Breiman, Friedman, Olshen, & Stone, 1984], CART is a decision-tree procedure thatis used widely on classification and regression predictive modelling problems.

CART progressively split individuals into smaller groups with increasing homogeneity in the dependent variable within each group. CART's mathematic results are easy to understand beside that it is fast to train and quite effective. [54]

C4.5

C4.5 [Quinlan (1993)] is an algorithm for decision trees construction, it results accurate trees, therefore, valuable classification tool. The major disadvantage of C4.5 is the loss in terms of time and memory. [55]

A main difference between CART and C4.5, is that trees produced by CART are binary recursive, while C4.5 allows two or more outcomes. [56]

❖ **Strong learners**

On the other hand, strong learners can be defined as:

"A class of concepts is learnable (or strongly learnable) if there exists a polynomial-time algorithm that achieves low error with high confidence for all concepts in the class." [57]

Some of these strong learners can be stated as follows:

Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm, mainly used for classification problems, One great advantage of this algorithm is that not only it does separate data into classes, but also find a separating hyper-plan.
Furthermore, SVM can handle data that are not linearly separable, "Soft margins" and the "kernel trick" are the two SVM updates for processing these data.
The SVM algorithm is set up in such a way that it searches the graph for points (or support vectors) that are closest to the dividing line. After the algorithm calculates the distance between these vectors and the dividing plane, that distance is named "the gap". The main goal of the algorithm is to maximize the neat distance. This gap should be as big as feasible for the optimum hyperplane. [58]

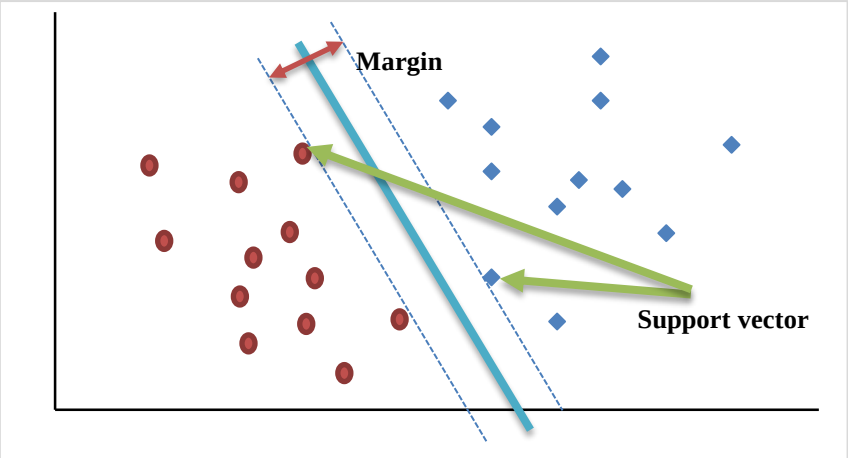

Figure II.2 SVM.

Random Forests

Random Forests is a supervised learning algorithm that, just as the name unveils, is an ensemble of several trees (i.e. Decision Tree algorithm). Generally, they are trained via bagging method.

To put it in simple terms: Random forests build several decision trees and merge them to get a more accurate and stable prediction. It is an efficient algorithm, for both classification and regression problems, to produce a predictive model. Its default hyper-parameters already return great results and the system is effect at avoiding overfitting.

The great disadvantage of Random Forests is that a high number of trees might make the computation process much slower and ineffective for real-time predictions. [60][61]

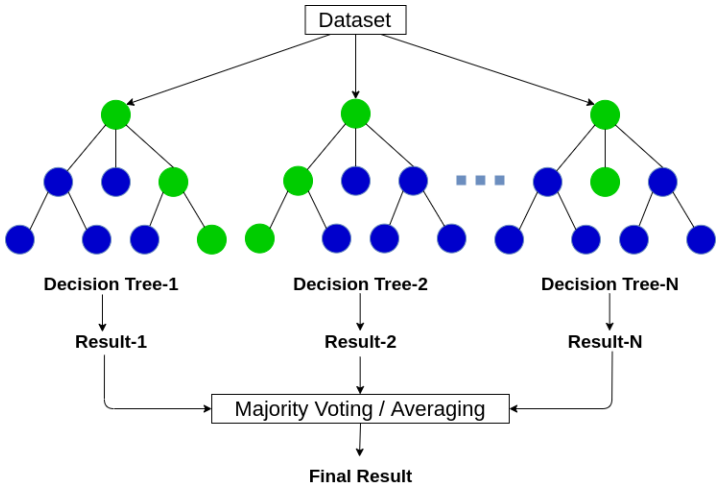

Figure II.3 Random Forest Architecture

Adaboost

AdaBoost, adaptive boosting for short, is a boosting algorithm that was mainly developed for binary classification. It is often used with weak learners such as decision trees. It has a solid theoretical foundation what led to fruitful contributions.

AdaBoost boost the weak classifier with a performance slightly better than random guessing into an accurate strong classifier, instead of directly designing a strong learning algorithm. [62][63]

After few iterations only, Adaboost achieves perfect prediction in samples, and this does not prevent generalization error to continue dropping. [62]

II.2.3 Limitations of Traditional Machine Learning

Although Machine Learning has reached unexpected limits in terms of keeping up with human abilities and behaviors, it has shown clear inadequacy in other critical tasks, such as:

Traditional ML algorithms still need human assistance to be capable of what they are designed to accomplish.

As for data, traditional Machine Learning algorithms are preferable with small data size only, which is inversely proportional to accurate results.

In addition, both SVM and Random Forest algorithms are noise-dependent models; moreover, their training time can be extremely expensive (i.e. it is directly commensurate with the number of samples). [64]

In traditional ML algorithms, human intervention is always needed due to a lack of domain understanding, as in the feature-engineering phase, where hand-crafted features are put on the table for the purpose of effective learning, this stage is manual and time-consuming as well. [65]

II.3 Deep Learning: A step forward

As a result to machine learning insufficiencies, Deep Learning (DL) has come with more efficient and inclusive solutions.

Deep Learning outperformed Traditional ML in the case of large data size, as Andrew Ng, one of the leaders of the Google Brain Project said "*The analogy to deep learning is that the rocket engine is the deep learning models and the fuel is the huge amounts of data we can feed to these algorithms.*"

Machine learning methods are much more useful with small databases where they give better results. For example, if we only have 100 images, decision trees, K-nearest neighbors and other ML models will be much more effective than using a deep neural network, However, it's big data age. [66]

DL eliminated the worries about features engineering, owing to the concept of "feature extraction from data automatically", via multi-layer non-linear transformations, with less human intervention.
Another major difference between DL and traditional ML is adaptability. Deep learning techniques can be adapted to different fields and applications.

For example, in computer vision, image classification networks are often used for the extraction of features for detection and segmentation of objects. The use of these pre-trained networks facilitates model learning and often results superior performance in a short time. Unlike ML algorithms, machine learning techniques differ from one domain to another and from one application to another. [66]

II.3.1 Definition

Deep Learning, a subset of Machine Learning (Fig.II.4), draws on information processing models found in the human brain. DL mainly does not require any human-designed guidelines to function; contrariwise, it uses a huge quantity of data to link the given input to custom labels. Numerous layers of algorithms make up part of how DL was designed; each provides a different interpretation of the supplied data.[67] [68]

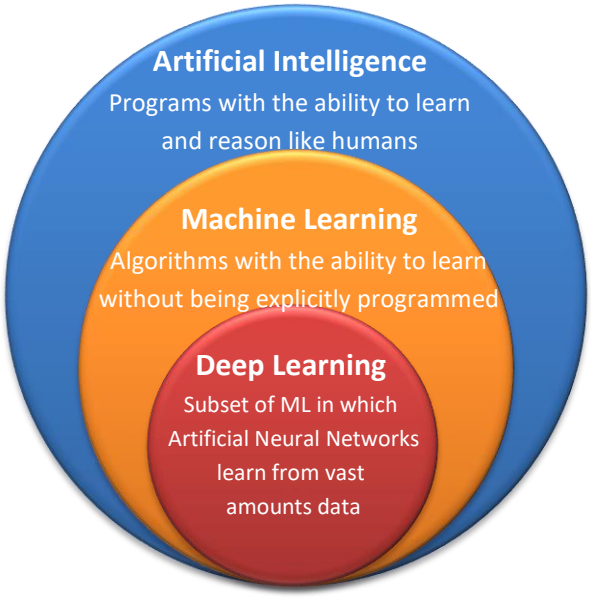

Figure II.4 Artificial Intelligence techniques

II.3.2 Background

People may believe that deep learning (DL) is a fresh discovery, given the current revolution it is causing. However, DL history dates back to the 1940s and particularly in 1943 when a human brain-based Neural Network computer model was developed by Walter Pitts and Warren McCulloch.

In 1960, Henry J. Kelley is credited with creating the fundamentals of a continuous backpropagation model. Stuart Dreyfus created a simpler version based solely on the chain rule in 1962.
The multilayer perceptron with a polynomial activation function was first presented in 1965 by Alexey Grigoryvich and Valentin Grigoryvich. [69][70]

In the early 70s, Seppo Linnainmaa implemented backpropagation in computer code; the research in backpropagation progressed significantly in recent years. [71]

The first step toward convolutional neural networks or CNN was when Kunihiko Fukushima introduced Neocognitron in the 80s. Furthermore.

Yann Le Cunn was influential in bringing CNN to where it is now, by using backpropagation to recognize handwritten digits. [72]

In 2014, Ian Goodfellow invented GANs, which opened new avenues for deep learning applications in fashion, art, and science, because of their ability to synthesize real data. [73]

II.3.3 Artificial Neural Networks:

The best description of the functioning of Artificial Neural Networks is simply, in order to find models or patterns; they give sense to the input data that has been fed to them.

Artificial Neural Networks are constituted of millions of neural (computational units) connected with high number of synapses (weighed connections), what allows to achieve real time solutions to complex problems, self-learning, resistance to errors.

One exciting thing about them is their ability to result practical outputs with given inputs that they never came across during the learning process. [74]

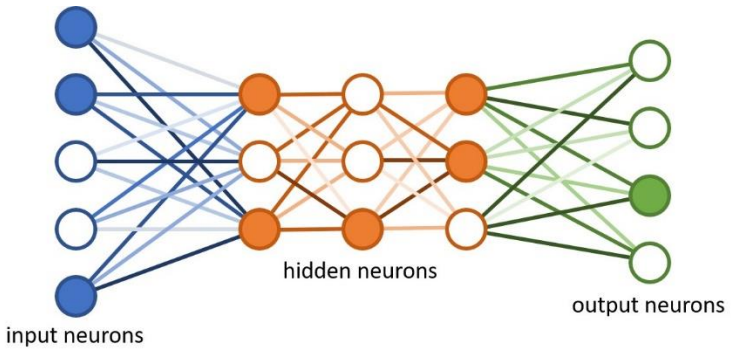

Figure II.5 Artificial Neural Networks

II.3.4 Deep Learning approaches

Similar to Machine Learning, Deep Learning approaches as well can be classified as Supervised, Semi-supervised and Unsupervised, beside Reinforcement Learning (RL)

which is usually discussed either as semi-supervised or unsupervised learning approach, more details in next sections. [75]

- **Supervised Learning**

The learning process in this technique is based on labeled datasets, several types fall under this approach, and we list some of the most important among them.

Convolutional Neural Network (CNN):

With the reputation as the most successful models in image classification, CNNs are a subcategory of neural networks. Their architecture is divided into two distinct parts: a convolutional part and a classification part. (To be discussed in section II.4)

Recurrent Neural Network (RNN):

Recurrent neural networks [76] are a supervised Learning model that -unlike CNNs-, allows sequential data in their both input and output, so it can for instance capture the sequential structure of a text, and that exactly why RNNs are powerful in language processing and modeling, and other different applications. [75]

In order to get better predictions, RNNs take is inputs and reuse the activations of both previous and later nodes in the sequence to affect the output. [77]

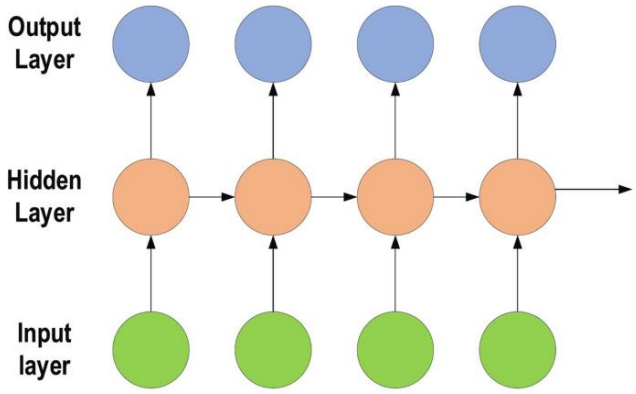

Figure II.6 Typical unfolded RNN diagram

This approach itself has other sub-models with slight variations:

Long Short-Term Memory (LSTM):

One of the principle issues with RNN is the vanishing gradient problem, and LSTM came as one of two solutions for this problem as a better RNN. It is also known for its efficiency with temporal information processing. [75].

Gated Recurrent Unit (GRU):

GRU [Cho, et al. 2014], principally came from LSTM, but considered a lighter version of RNN than LSTM regarding architecture, complexity and computation cost. [75]

In comparison, the GRU requires less network parameters, which allows the model to run quicker. LSTM, on the other hand, provides better performance.

- **Unsupervised Learning**

This type of learning techniques does not require labeled data, Auto Encoders (AE) and Generative Adversarial Networks are the most two common types included in this approach.

Auto encoder (AE)

An AE is a type of artificial neural network used for unsupervised learning. More precisely, to learn efficient coding of unlabeled data [78]. The main aim of AE is to learn a representation (encoding) of data, typically for dimensionality reduction, in order to ignore insignificant data by training the network.

This Auto encoder technique consists of two parts: the encoder and the decoder. Starting with the encoding phase that maps the input into the code in other words, the input samples are mapped usually to the lower dimensional features space with a

constructive feature representation. This process can be repeated until the desired feature dimensional space is reached.

While in the decoding phase, the current features are regenerated from lower dimensional features with inverse processing. [75]

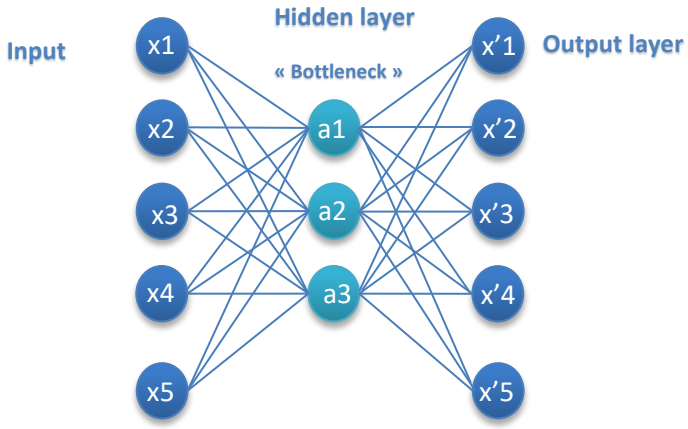

Figure II.7 Auto encoder architecture

GAN

Generative Adversarial Networks, or GAN for short [73], are unsupervised deep learning approach where two models are trained in a zero-sun game, the generator generates new examples, and the discriminator determines wither these example are good (Real: from the domain).or not (Fake: generated).[75]

GANs are used in wide range of domains as in image, natural language, and medical information processing etc. [79]

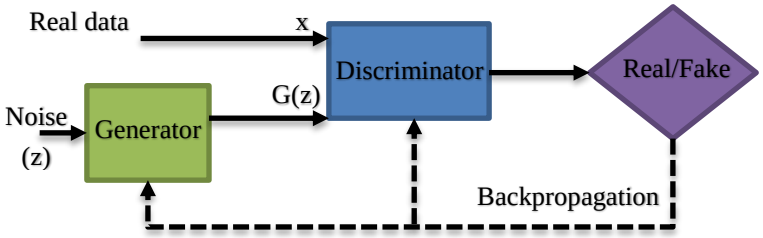

Figure II.8 Standard GAN architecture.

- **Reinforcement learning**

Commonly called semi-supervised learning, this approach required datasets that are partially labelled [75]. GANs and RNNs with LSTM and GRU are used also as semi-supervised learning approaches.

II.4 Overview of Convolutional Neural Networks

Convolutional networks were first introduced by Fukushima [80], he derived an architecture of the hierarchical neural network inspired by the research work of Hubel [81], Lecun [82].

Ciresan [83] used convolutional networks and performed best in the literature for multiple object recognition for image databases.

CNN networks focus primarily on the fact that the input will be image-based; this centralizes the architecture to be configured to best address the need to process a specific type of data.

The CNN algorithm takes an input image which is usually represented by a matrix; applying pooling and convolution operations will extract the features of the image that allow the computer to identify and differentiate an object from another.[84]

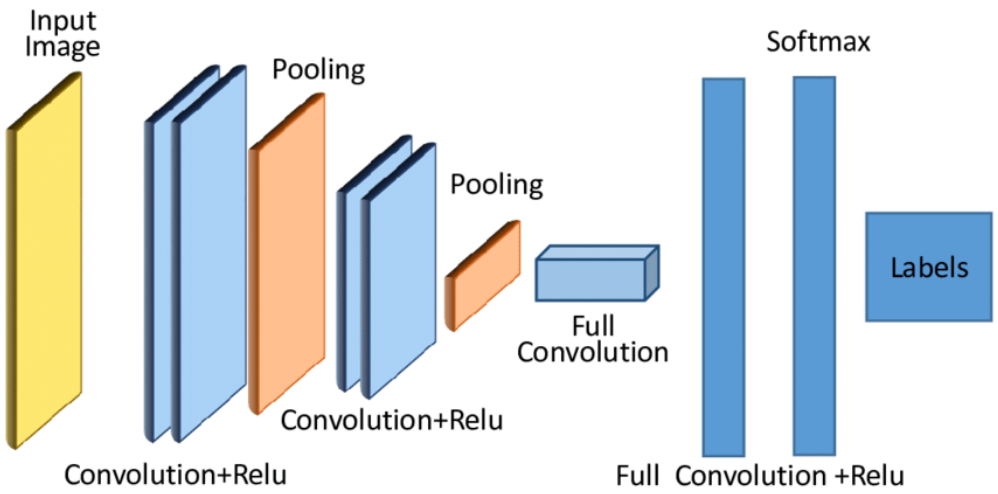

Figure II.9 Convolutional Neural Networks [59]

II.4.1 Covolutional Layer

The convolutional layer is the most important component in CNN design, it is made up of a set of kernels (convolutional filters).

The output feature map, with these filters, is generated by convolving the input image, expressed as N-dimensional metrics. [85]

The role of a convolution layer is to detect local features such as edges, lines, blobs of color, and other visual elements at different positions in the input feature maps by applying convolution operations on learnable kernels. [86] The more kernels that we give to a convolutional layer, the more features it can detect. [69]

II.4.2 Pooling Layer

Pooling layer is the next step after each convolutional layer; both the stride and the kernel are size-assigned before the pooling process is performed, identical to the convolutional operation. [87]

Down-sampling as the most significant idea of pooling, is performed in order to reduce the complexity for the further convolution layers. [88]

Two principle methods of pooling are commonly utilized, Max pooling, and the average pooling.

Max pooling

When using Max pooling, the output is produced by taking the maximum input in the kernel. To further understand how Max-Pooling works; consider the following scenario: we have a 4x4 matrix representing our initial input and a 2x2-size kernel that we will apply to it.

Max pooling will take the maximum for each of the regions scanned by the filter, resulting in a new output matrix where each element will correspond to the maximums of each region encountered. Figure II.10

Average Pooling

Comparing to Max pooling, the slight difference is that average pooling calculates the average value in the regions scanned by the kernel. For the use of creating down-sampled feature map. [89]

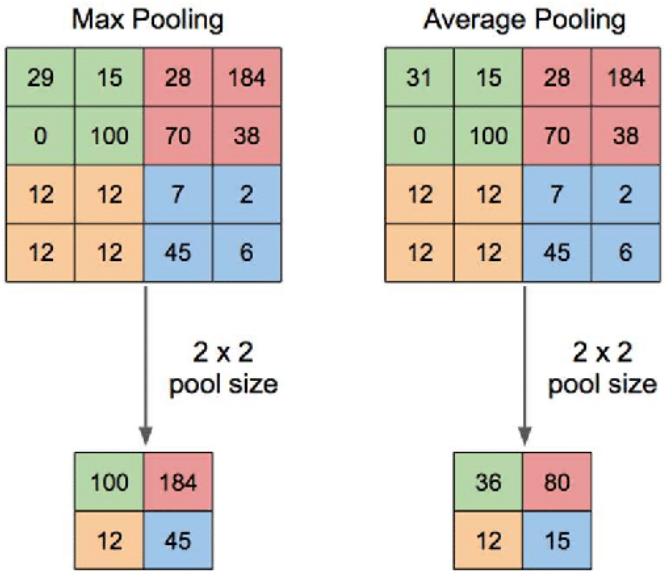

Figure II.10 Max pooling and Average pooling illustration [89]

II.4.3 Fully connected layer

After several layers of convolution and pooling, the high-level reasoning in the neural network is done through fully connected layers, exactly in the same way as a traditional feed-forward network, and are usually linked at the end of the network, in order to use the features learned by the previous layers in prediction.

Each Neuron in the fully connected layer is fully connected to every other neuron in the previous layer (The final Convolutional, activation, or Pooling layer).

Each feature in the final spatial layer is connected to each of the hidden states in the first fully connected layer [90]. Taking into consideration all the activations received, the neurons can determine which features could match more with which class. During this process, some of the spatial information were lost and to get over this, the fully connected layer is considered as its equivalent convolutional layer representation so they can be viewed as 1 × 1 convolution applied over the entire input with a full-connection mapping. [91]

II.4.4 Activation Functions

On a dynamic aspect of CNNs, relies the concept of activation function, it is a non-linear function that provides nonlinearity to networks, preventing the fusion of hidden layers in the neural network.

SIGMOID

It is the most widely used activation function as it is a non-linear function. Sigmoid function changes the values in the range 0 to 1. It can be defined as:

$$f(x) = \frac{1}{e^{-x}} \qquad \text{Eq. II.1}$$

Sigmoid function is continuously differentiable with an S-shaped curve, and it is asymmetric about zero, which will make the sign of all output values the same. [92]

TANH Function

Short for Hyperbolic Tangent function. It is comparable to the sigmoid function but is symmetrical at the origin. It can be defined as:

$$f(x) = 2sigmoid(2x) - 1 \qquad \text{Eq. II.2}$$

Tanh function is continuous and differentiable, and results values in the range -1 to 1. In most cases, Tanh function is better than sigmoid, as it has steeper gradients, the negative inputs will be listed as negative, where in sigmoid, they will be confused with near-zero values. [92]

ReLU FUNCTION

ReLU is a non-linear activation function that is frequently used in neural networks, it stands for Rectified Linear Unit, and it is commonly used and performs better than other functions.

The advantage of using ReLU is that all the neurons are not activated at the same time, using ReLU function is quite convenient due to the fact that all the neurons are not activated at the same time. This indicates that a neuron will be deactivated only when the output of linear transformation is zero [92]. It is defined as:

$$f(x) = \max(0, x) \qquad \text{Eq. II.3}$$

Leaky ReLU

Or Leaky Rectified Linear Unit, is another type of activation functions, it is both an extension and an attempt to solve the dying problem of ReLU and, with a range of [-∞, +∞] we have:

$$f(x)\begin{cases} \alpha x \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases} \qquad \text{Eq. II.4}$$

Where the parameter α is a small constant, therefore, for any negative value this function returns a really small value, as a result, the gradient of the left side of the graph comes out to be a non-zero value (Figure II.11), which lead to a better performance with higher resolution modeling [93][94].

ELU (Exponential Linear Unit)

This function is identical to ReLU for positive values, but for negatives it takes an e^x value [92], It is defined as:

$$f(x) = \begin{cases} x \; if \; x > 0 \\ e^x \; if \; x \leq 0 \end{cases} \qquad \text{Eq. II.5}$$

Softmax

Softmax function is used for multiclass classification problems, unlike sigmoid functions that are used for binary classification, in fact, it is considered as a combination of Sigmoid functions, with an output interval of (-∞;+∞), it can be expressed as:

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \; for \; j = 1, \dots, k \qquad \text{Eq. II.6}$$

The number of neurons in the network's output layer will match the number of classes in the target when we construct a network or model for multiple class classification. [92]

II.4.5 Loss Function

The loss function is the function that calculates the difference between the actual and the expected output of the algorithm; it can help the model to reduce the loss by updating the weights across the training samples.

The choice of the loss function is one of the most critical aspects of any deep learning approach.

Below is a brief overview of some of the frequently used loss functions:

Cross-entropy

The categorical cross-entropy loss function: It calculates the loss of an example by computing the following sum:

$$\text{Loss} = -\sum_{i=1}^{n} t_i * \log(p_i) \qquad \text{Eq. II.7}$$

(For n classes, where t_i is the truth label and p_i is softmax probability for the i^{th} class) [95]

Cross-entropy is used to minimize the loss through updating the weights during the training.

Exponential loss

Mainly used in AdaBoost algorithm [96], Exponential loss is defined as:

$$\frac{1}{2}\log\left(1 + e^{-Yf(x)}\right) \qquad \text{Eq. II.8}$$

Log-Loss

It is the binary cross-entropy, used for logistic regression [96] and defined as:

$$log\left(1 + e^{-Yf(x)}\right) \qquad \text{Eq. II.9}$$

II.4.6 Back-propagation

Backpropagation is an essential concept in learning with feedforward neural networks; it is about adjusting weights in a network based on the changes of the loss function, also the network is initialized with randomly chosen weights.[97]

In a neural network, there are two calculation steps, a feedforward step, and the back-propagation step.

II.4.7 Gradient descent

The gradient descent is an optimization algorithm that is used for the aim of minimizing the training error, in an iterative way; this algorithm updates the parameters of the network.
This algorithm has been effectively utilized to train ANNs in recent decades. [86]

II.4.8 Stochastic Gradient Descent (SGD)

The SGD approach is used for training Deep Neural Networks (DNN) [98].
In the gradient descent algorithm, all of the training samples are processed in one iteration for a single parameter update, whereas in SGD, only one sample or a subset of training samples is used in the iteration, in case of using a subset, it will be known as Mini-batch Stochastic gradient descent. [CS229 Lecture notes. Andrew Ng. Supervised learning.]

II.4.9 Learning rate (η)

The learning rate is an important component for training neural networks. The learning rate is a positive scalar determining the step size that accelerates the training process. It is an adjustable hyper-parameter, with a value that varies between 0.0 and 1.0.

However, selecting the value of the learning rate is sensitive. [98].

"When the learning rate is too large, gradient descent can inadvertently increase rather than decrease the training error. […] When the learning rate is too small, training is not only slower, but may become permanently stuck with a high training error."[99]

II.4.10 Optimization methods for DL

Below are some gradient descent optimization variants:

Momentum

Momentum has a significant impact on the speed of the Gradient Descent process [100], by adding to the initial expression the following vector: [74]

$$v_{t+1} = \mu v_t - \eta \Delta J(\theta_t),$$

$$\theta_{t+1} = v_{t+1} + \theta_t. \qquad \text{Eq. II.10}$$

ϑt + 1: The start of each iteration, **θt:** the gradient at the current location, **η**: Learning rate

Adagrad Optimizer

The objective of this method [101] is to calculate the adaptive learning rate through the training process, making it adjust automatically, depending on the "sparseness" of the parameters. Adagrad gradually lowers the learning rate but not in the same way for all parameters [100].

More formally, it is described with:

$$\forall i, (\theta_{t+1})_i = (\theta_t)_i - \alpha \frac{(\nabla j(\theta_t))_i}{\sqrt{\Sigma_{u=1}^{t} (\nabla j(\theta_u))^2}_i}, \alpha > 0. \qquad \text{Eq. II.11}$$

RMSprop

This algorithm [G. Hinton & al 2012], automatically adjusts the learning rate to each parameter, same as Adagrad. However, it divides gradients from recent iterations by a running average. [102]

Adam

Adaptive Moment Estimation (Adam) [103] is another method that computes adaptive learning rates for each parameter.

Based on the momentum and with a resemblance to RMSprop, it is liable on the magnitude of the gradient for calculating adaptive learning rate [74]
Its particularity is to calculate (mt; vt) "moments adaptive estimates".

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_1) {g_t}^2 \quad \text{Eq. II.12}$$

mt: the first moment of the gradient, **vt:** the second moment (uncentered variance).

Parameters **ß1** and **ß2** are used to perform running averages on the moment **mt** and **vt** respectively.

II.4.11 Overfitting

Overfitting occurs when a model learns all the details and achieve a good fit on the training data. It becomes capable of capturing non-useful information to accomplish its task, therefore, it becomes unable of generalizing the characteristics of the new data; and that has an effect on the performance of the model, especially when dealing with deep neural networks where the network is powerful on the training set.

A regularization method called ‘dropout’ is considered as one of the proposed solutions to overfitting, it drops randomly a number of neurons in a neural network during model training in each iteration.

By dropping a unit out, it means removing it from the network temporarily, with all its connections, this prevents units from overly co-adapting in order to avoid that the weights are less adjusted to the data than they should be, and this does not affect the model’s performance.

In Dropout regularization, neurons will not learn redundant details of inputs, which is important to make predictions due to useful knowledge that has been gained.

Dropout is not applied after training when making a prediction with the fit network.

When the network becomes less sensitive to the specific weights of neurons it will be capable of better generalization and it is less likely to over-fit the training data. [104]

II.4.12 Hyper-parameters tuning

Deep Learning and neural network is a learning model that contains a considerable number of hyper-parameters, that requires regular adjustment and configuration for more accurate results, this phase is extremely decisive along the learning process.

"Recently, it was shown that the state of the art on image classification benchmarks could be improved by configuring existing techniques rather than inventing new learning techniques" (Pinto et al. 2009, Coates et al. 2011, Bergstra et al. 2011, Snoek et al. 2012, Thornton et al. 2012). [105]

During the training process, there are several choices regarding the learning rate, a high learning rate value leads to an unstable convergence, while a low value slows down the convergence and in both cases; we will not reach the optimal performance. The most common method for achieving the best learning rate in deep learning is to start with a high value to speed up the gradient descent, and subsequently lower it to improve accuracy.

The use of mini-batches improves as well the convergence of the model and makes it more stable by reducing the number of passes on the learning basis. This also makes it possible to load partially the data in memory.

II.5 Fully Convolutional Networks

As mentioned in chapter I, FCN was proposed by Long et al. [106] for semantic segmentation as per-pixel image classification, that is to say, labeling each pixel with a class under the name of PixelWise classification. FCN uses a convolutional neural network to transform image pixels into pixel classes. In contrast to the CNNs that we already tackled for image classification or object detection, FCN includes only convolutional layers, which enable us to take an image of arbitrary size and produce a segmentation map of the same size, and the entire process appeared to be considerably faster.

The main objective was to create semantic segmentation networks by adapting classification networks (pre-trained networks) such as AlexNet, VGG16, and GoogLeNet into FCN to predict dense outputs from arbitrary-sized inputs [107].

This process is achieved through replacing all fully connected layers viewed as convolutions with kernels that cover their entire input region with the fully convolutional layers, via a 1×1 convolutional layer, which means less parameters for the model, mainly because fully connected and fully convolutional layers differ in terms of weights number. Also as mentioned, due to the structure that allows any input size. [107]

The first half of the network consists of down-sampling the spatial resolution of the image developing feature mappings.

With each convolution, a feature map shows the contribution of each layer to the final per-pixel classification of the image, as the first layer responds to low-level features, such as lines, colors, and edges, the middle layer responds to more difficult features such as textures and in the final layers, we obtain efficient differentiation in the objects of the class.

At this point, we can say that going deeper, deep features can be obtained but spatial location information are lost. To fix this issue and improve predictions, FCNs use deconvolutional layers by up-sampling lower resolution outputs and resizing them back to the original input image via “transposed convolution” and through “Skip connections” that recover the lost spatial information and allow information to flow by concatenating the outputs of disconnected layers. [107]

As a result, the model output is a high-resolution segmentation map that contains the predicted classes for the input pixel at the same spatial position.

II.5.1 Transposed Convolution (The deconvolutional layer)

The term deconvolution can be misleading, which implies that it is the inverse operation to convolution, and that is not true, the more relevant name is transposed convolution.

The deconvolution network produces object segmentation from the feature extracted from the convolution network.

It is a symmetric version of the convolution network that has multiple sets of unpooling, deconvolution, and activation layers that enlarge through the combination of unpooling and deconvolution operations. Contrary to convolution network, that reduces the size of activations. In the convolution network, pooling is designed to filter noisy activations in a lower layer, by retaining only robust activations in upper layers. During this operation spatial information is lost, which may be critical for precise localization that is required for semantic segmentation.

To solve this, we use unpooling layers in deconvolution network that work reversely to pooling, and to reconstruct the original size of activations.

According to the approach proposed in [108][109], it records the locations of maximum activations during pooling operation in switch variables, which are employed to place each activation back to its original pooled location, after that The deconvolution layers produce an activations map obtained by unpooling through convolution-like operations with multiple learned filters.

Convolutional layers connect multiple activations in a receptive field to a single activation, while deconvolutional layers associate a single activation to a field or window of multiple activations. The learned filters in deconvolutional layers correspond to standards to reconstruct the shape of an input object. Due to a hierarchical structure of deconvolutional layers being used to capture different levels of shape details.

II.5.2 Types of FCN

Encoder-Decoder Architecture

The so-called Encoder-Decoder architectures are composed of two parts, the encoder and decoder part, Encoder progressively reduces the spatial dimension with pooling layers, while the decoder gradually retrieves the object details and spatial dimension.

In the decoder part, each feature map receives directly the information from the feature map at the same level as the encoder part using skip connections. [110]

U-Net [110] and Seg-Net [111] are very well known examples, U-Net architectures have proven very beneficial for the different tasks of segmentation, such as medical images [110], satellite images [112].

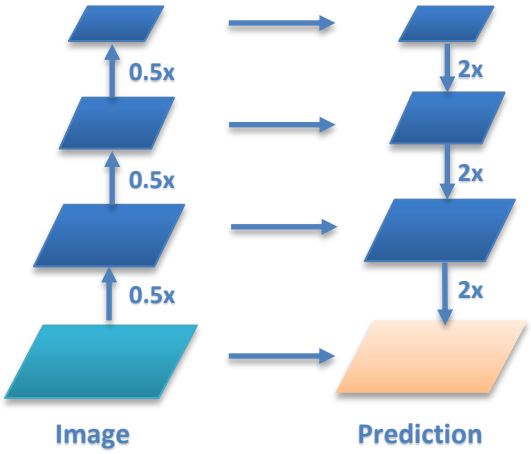

Figure II.11 Encoder-Decoder Architecture

Spatial Pyramid Pooling

Lazebnik et al. 2006, firstly proposed the idea of building a fixed-sized spatial pyramid, to prevent a BOW (Bag-of-Words) then after that; the approach was involved to CNNs by (He et al. 2015). So that Spatial Pyramid Pooling (SPP) removes the fixed-size constraint of the network, it allowed inputs of different sizes to be fed into CNNs which, create different-sized feature maps. [113][114]

The SPP layer is appended to the last convolutional layer that extracts the features and generates fixed-length outputs, which are then fed to the pixel-wise classifier. SPP-Net

allows the efficient training of images at different scales (or resolutions) by allowing different input sizes to a CNN.

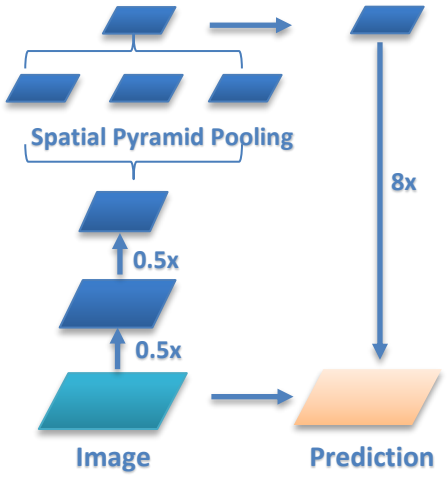

Figure II.12: Spatial Pyramid Pooling

Atrous Convolution

The idea of dilated convolutions is to replace the traditional convolutional layers with dilated convolution layers, which helps in increasing the receptive field. With adjacent convolutional filters, a receptive field can only grow linearly with layers; while with dilated convolution the effective receptive field would grow more quickly [115].

It is an efficient method for the detailed conservation of feature map resolutions, Instead of having deconvolution layers; it uses dilated convolutions to recover spatial resolution. The value of dilation specifies the sparsity while doing the convolution.

The downside of this technique is about its higher requirement for GPU storage and computation since the feature map resolutions do not get smaller within the feature hierarchy [116].

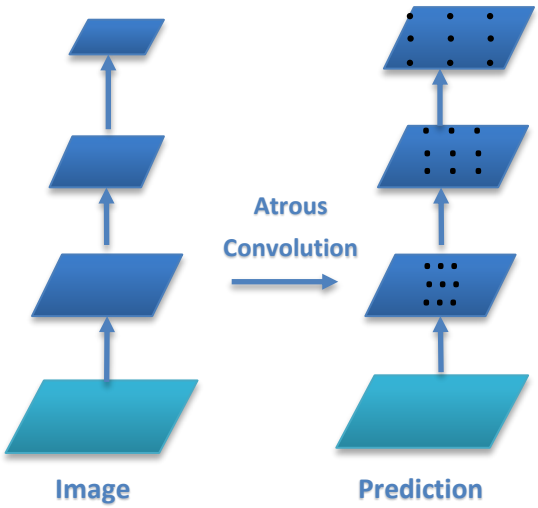

Figure II.13 Atrous convolution block

II.5.3 The encoder-decoder U-Net

U-Net is a convolutional neural network architecture developed for biomedical image segmentation designed by [U. Ronneberger & al 2015]. The network is based on the FCN architecture proposed by Long et al. It is able to locate and distinguish the borders of the elements composing a certain image by making the classification on each pixel. By looking at the architecture, we notice that the name U-net refers to the U-shape in which the layers are established. It consists of an encoder path followed by another symmetrical decoder**.**

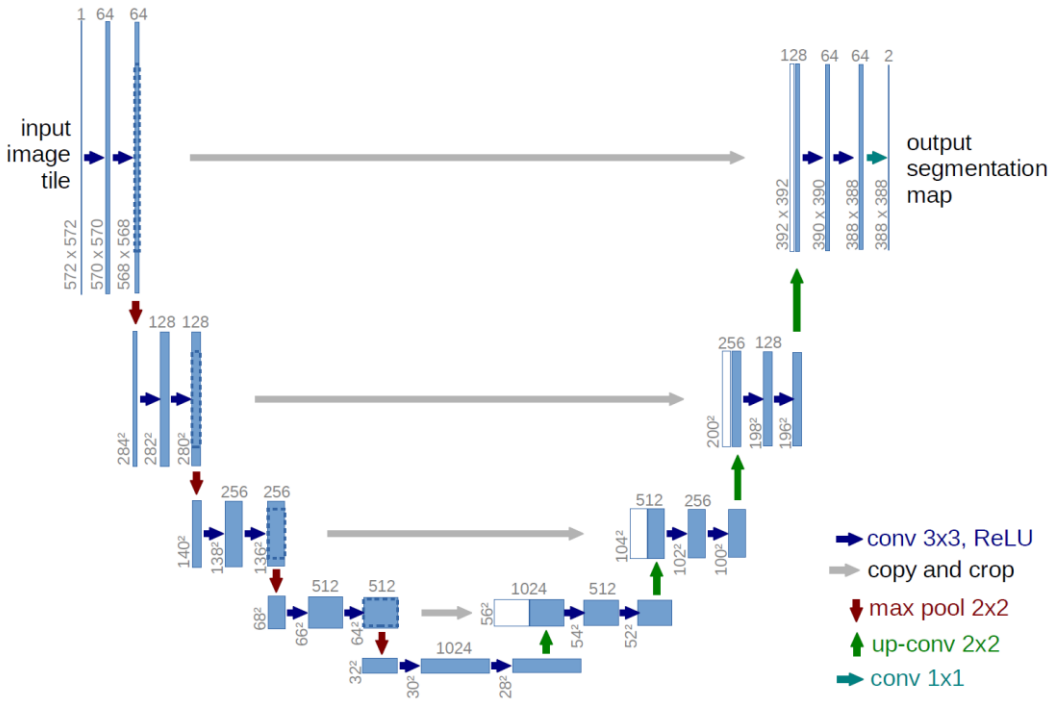

Figure II.14 U-Net Architecture [117]

The encoder path follows the typical architecture of a convolutional network. It consists of repeated convolutions layers, each one followed by ReLU and max-pooling operation which, reduce the size of an image, At each downsampling step we double the number of feature channels. [110]. Once we reach the decoder Path, The goal is to project semantically the features with a low resolution that have been learned by the encoder into the pixel space (high resolution) to get a dense classification. Every step in the decoder path consists of an upsampling of the feature map, followed by concatenation with the cropped feature map from the encoder path based on skip connection in order to learn better representations with the following convolutions.

II.6 Conclusion

In this chapter, we aimed at reviewing different Machine Learning and Deep Learning approaches, but mainly focused on how DL techniques overcome traditional ML, and then continued by providing an overview of the CNNs, its architecture and the

principle methods that represent the leap for an accurate performance such as backpropagation, activation function and some hyperparameters tuning. We also paid a particular attention to the most important unit in our research, which is Fully convolutional networks, theorically, these architectures have shown better performance concerning per-pixel image classification, owing to avoiding dense layers which leads to less parameters and less computational complications.

In practice, the following chapter, contains our attempt to adapt one of the previously seen architectures in FCN, U-NET, with VOC 2012 dataset, before and during the learning process, many regularizations and configurations are demanded regarding Batch Normalization, dropout, learning rate , adaptable activation functions and weights as well.

Chapter III Model design and Implementation

III.1 Introduction

Among the objectives of semantic segmentation is to build “fully convolutional” networks based on convolutional neural networks "CNN", which take input of arbitrary size and produce correspondingly sized output with efficient learning and accuracy.

In this chapter, we will start by giving an overview of the dataset, and the development tools needed to achieve this work. After that, we move on to explain the process of designing and building the three proposed models that are based on the U-Net architecture. Also, the different evaluation metrics that have been used to estimate the similarity of samples, at the end we will see the development process of the models to deal with the semantic segmentation task.

III.2 Used Dataset

The level of success for any Machine Learning application is determined by the quality of the data used for training. As for deep learning, data is even more important, the features are determined by the data itself.

Most studies concerning image segmentation were founded on 2D images; thus, many image segmentation datasets are available. Some of the most popular are Pascal VOC, MNIST, MS-COCO, ImageNet...

PASCAL VOC [118] is one of the most common datasets in computer vision, with image annotations that are available not only for semantic segmentation, but for also classification, detection, action classification, and person layout tasks. [107]

The PASCAL VOC semantic segmentation challenge image set includes 20 foreground object classes ordered alphabetically, and one background class. Which means there are 21 classes of object labels:

(1= aeroplane, 2= bicycle, 3= bird, 4= boat, 5= bottle, 6= bus, 7= car, 8= cat, 9= chair, 10=cow, 11= diningtable, 12=dog, 13= horse, 14= motorbike, 15= person, 16= potted plant, 17=sheep, 18=sofa, 19=train, 20=tv/monitor)

This dataset is divided into three subsets, training, validation, and a testing set.The main challenges have run each year since 2005, starting with a small database with only 4 classes and 1578 images, this challenge improved over the years until 2012 when finally reached a big database with over 11,530 images containing 27,450 objects annotations and 6,929 segmentations. [Everingham & al 2010].

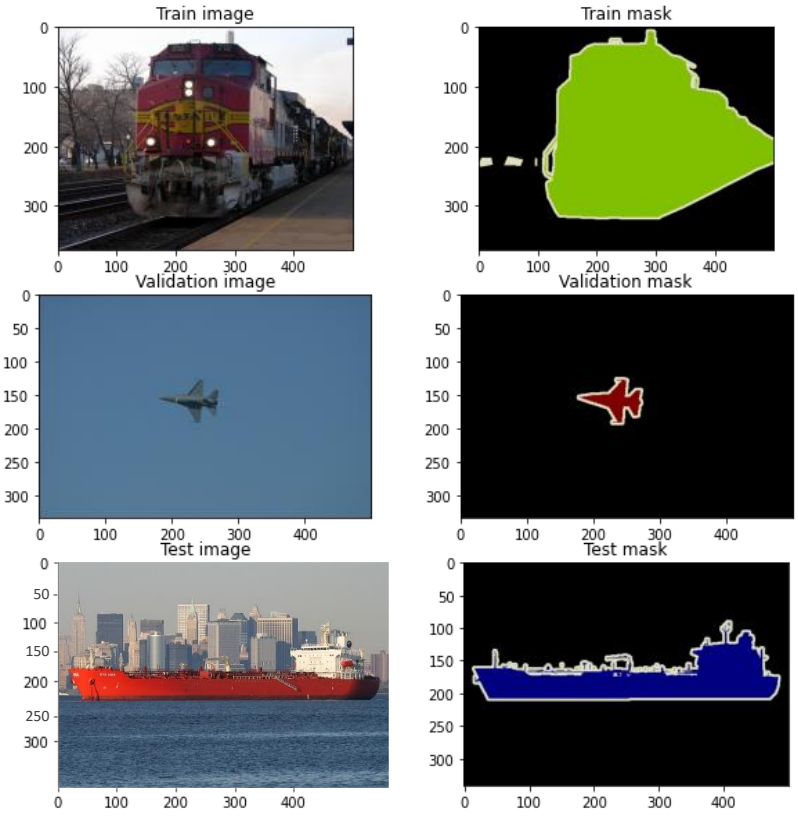

Figure III.1 Example of PASCAL VOC dataset

Regarding our project, we are working with The PASCAL VOC 2012 dataset, which used learning/validation data (2Go file), containing 2913 images in total for the implementation of the semantic segmentation task.

III.3 Development tools

The tools used to accomplish this project are mentioned below:

III.3.1 Software tools

Colaboratory

"Colab" for short, is a product from Google Research. Colab allows writing and executing python code through the browser, it is adequate to machine learning, data analysis, It will help to develop machine learning applications using Keras, TensorFlow and OpenCV libraries. [119]

TensorFlow

It is an open-source software library used to train and build both machine learning and deep learning models. It enables ML developers to build and deploy Machine Learning applications efficiently that exploit this technology. [120]

Keras

Same as Tensorflow, Keras is an open-source software library, which provides a Python interface for artificial neural networks. Keras acts as an interface for the Tensorflow library; it supports both convolutional and recurrent networks. [121]

Numpy

Numpy is a library for the Python programming language, dealing with multi-dimensional arrays and matrixes, and high-level mathematical functions.[122]

Matplotlib

It is a comprehensive library for the Python programming language and its numerical mathematics extension Numpy, it creates plots and interactive visualizations in Python. [123]

III.3.1 Hardware tools

CPU: Intel(R) Celeron(R) CPU N3350 @ 1.10GHz.

RAM: 12 Go.

OS: Windows 10 Professionnel 20H2.

III.4 Evaluation metric & Loss function

In our work, we used the Intersection-Over-Union (IoU) metric known as the Jaccard Index, considered generally the most used metric in semantic segmentation.

III.4.1 Jaccard Index

It measures the similarity between two finite sample sets A, and B by calculating the intersection of the predicted bounding box (A) and the ground truth (B) divided by their union, the larger the area of overlap the greater the IoU.

$$\text{Jac (A, B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \qquad \text{Eq. III.1}$$

For minimization purposes, it is recommended to use the Jaccard distance, it is a measure of how dissimilar two sets are where:

$$\text{Jaccard Loss (A, B)} = 1 - \text{Jac (A, B)} \qquad \text{Eq. III.2}$$

Where:

A is the predicted output of u-net.

B the ground truth or the image annotated by the expert.

III.4.2 Loss function

We chose the categorical cross-entropy loss function as it is used for multi-class classification tasks, which is similar to this case.

$$\text{Categorical Cross} - \text{Entropy (A, B)} = -\sum_{i=1}^{output\ size} \log A.B \quad \text{Eq. III.3}$$

Output size: is the number of scalar values in the model output.

In this situation, the pixels of some objects of interest are less than the pixels of the background as we already mentioned so for that reason we opted for a hybrid loss function, between categorical cross-entropy and that of Jaccard**.**

$$\text{Hybrid loss function (A, B)} = \text{Categorical Cross} - \text{Entropy (A, B)} + \text{Jaccard Loss (A, B)}$$

Eq. III.4

III.5 Proposed approaches

In addition to the standard U-Net [U. Ronneberger & al 2015], we have presented a model that groups both ‘U-Net and Google's "Inception" architecture’ features, by swapping the default U-Net network convolution layers for GoogLeNet's "Inception" layers in the encoder path.

The third proposed model consists of an Inception-based U-Net (Same as the second Model), with slight variations in terms of channels.

It is worth noting that our data set is randomly divided into two subsets:

The training set with a size equal to 2700 examples.

The validation set with a size equal to: 213 examples.

III.5.1 Standard U-Net

As a baseline model, we used the famous U-Net model, it took in images of (256, 256, 3) shape.

The contracting path (left side) follows the classic convolutional network architecture. It consists of a sequence of two 3x3 convolutions, each followed by an Exponential Linear Unit (ELU) and 2x2 max-pooling operation, with stride 2x2 for down-sampling.
On the right side, where the expanding path is constituted of an up-sampling of the feature map, then comes a 2x2 up-convolution, followed by 3x3 convolutions and a concatenation from the corresponding feature maps from the contracting path. For the output convolution, we used Softmax as an activation function.

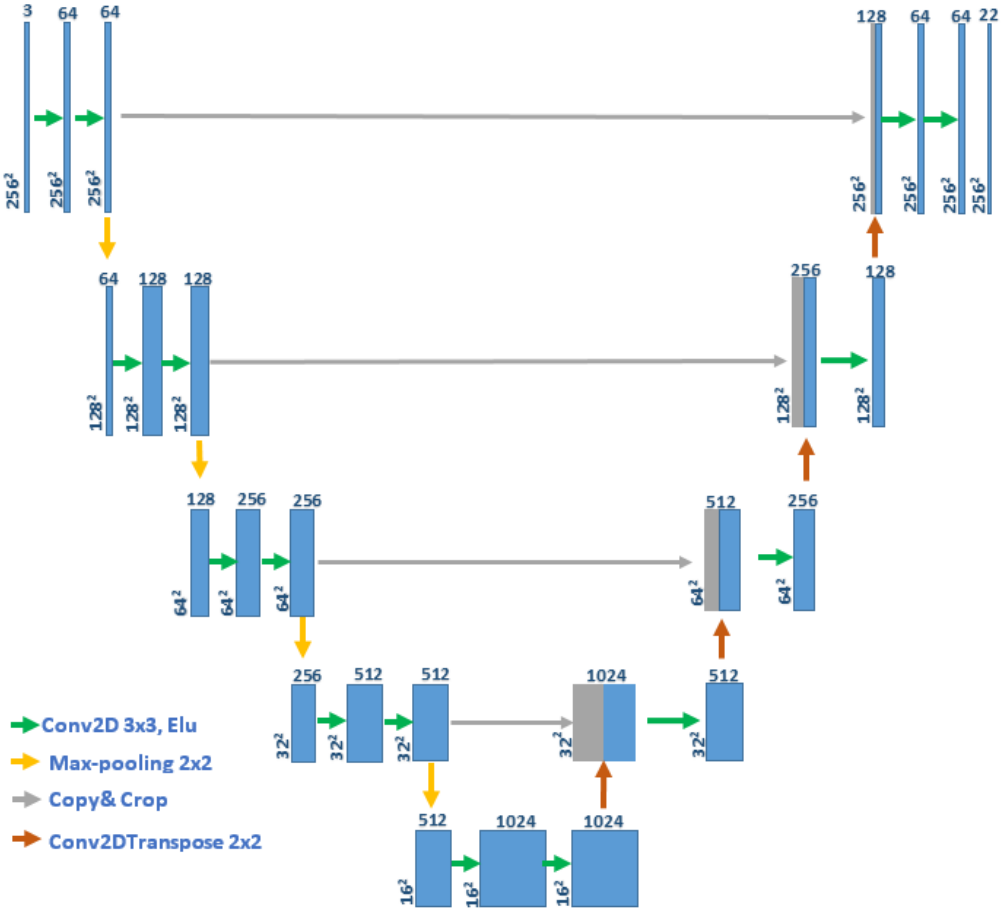

Figure III.2 Standard U-Net

III.5.2 Inception-based U-Net approach

Inception Modules

Inception Module of GoogleNet [C. Szegedy& al 2015]. This module allows cleverly using multiple convolution filter sizes, with some pooling, that are applied to the same input, instead of a single convolution. The outputs of each are then concatenated.

This module take advantages of feature extraction of different levels, For instance, it extracts general (5x5) and local (1x1) features at the same time, because we cannot know which size to use for a good filter beforehand.

What is interesting is that you can choose parameters in such a way that the total numbers of parameters of your model is very small, yet the model performs better than a single convolution.

In our model, we used Inception modules applying a concatenation of convolutions with different sizes (3x3, 5x5) followed by a stack of other two convolutions (3x3 + 5x5), the concatenated output is followed by another convolution (3x3).

Then we connected the resultant output to our U-Net model.

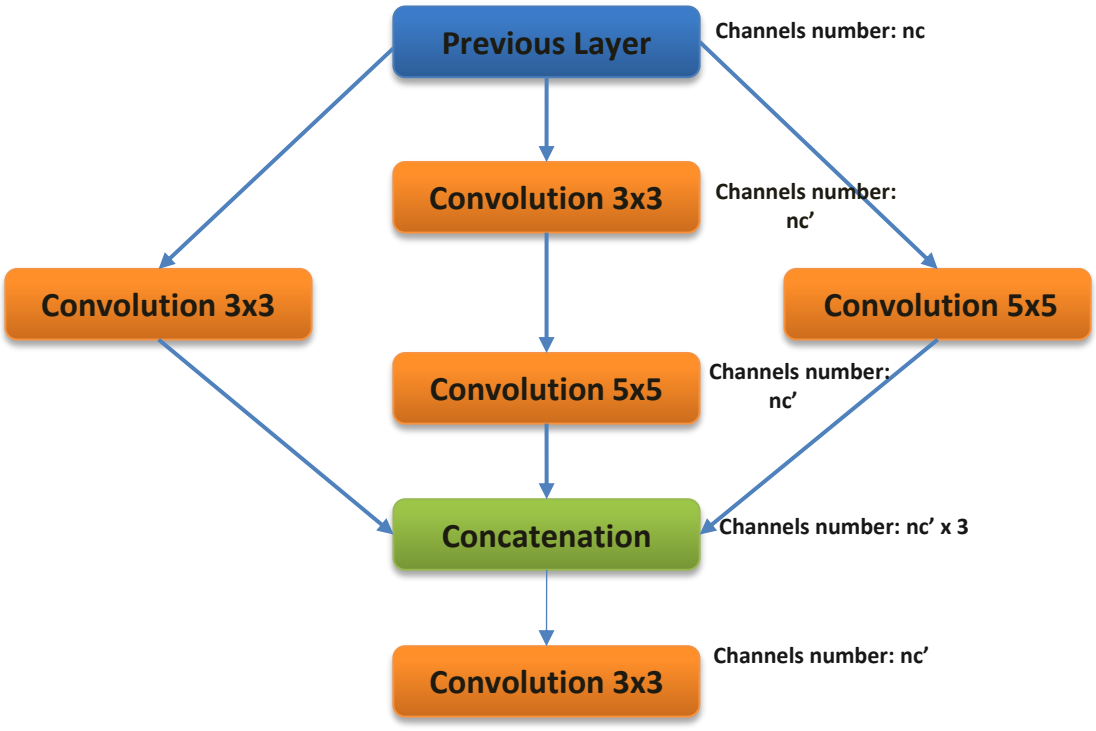

Figure III.3 Inception Modules.

Architecture layers

Integrated in a standard U-Net architecture, our proposed model had an input of size (256, 256, 3) resolution, and a number of channels that progressively changed from (32 to 512).

This approach comprises a series of convolution operations for each block, consisting of a contracting path with the down-sampling operation, and up-sampling operation in the expansive path in order to generate an output of the same size as the input.

The input images are normalized by Batch Normalization, and similarly to the first model, we used for each convolution the Exponential Linear Unit (ELU), except for the output convolution where we used Softmax as an activation function.

Each convolution uses an appropriate padding in order to keep the same dimensions, as well as a max pooling for each layer, the network here (as well as all proposed models) is trained with a hybrid loss function using categorical cross-entropy plus Jaccard loss and evaluated with the Jaccard score.

We used the He Normal initialization algorithm to initialize the models' parameters, and then we trained with one of the best stochastic gradient-based optimization algorithms, named Adam. (Section II.4.10)

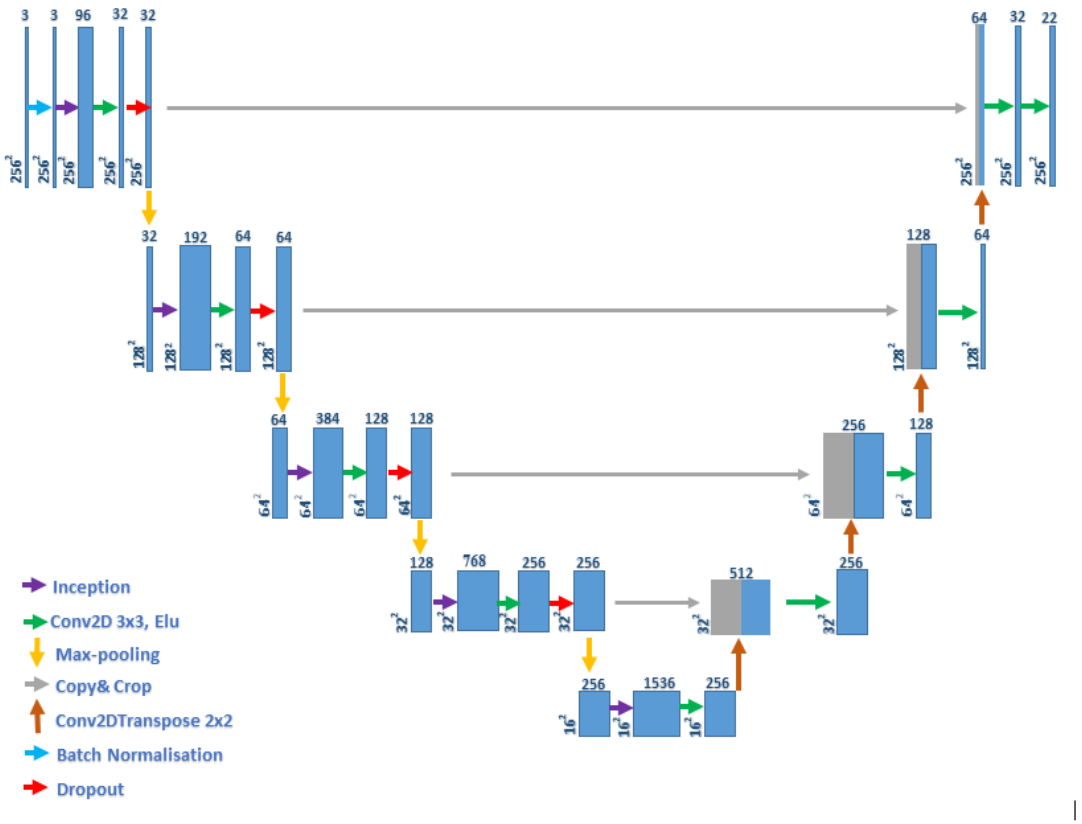

Figure III.4 Inception U-Net-based model architecture.

III.5.3 Inception-based U-Net (64)

This model as it is titled, is a version of the second model with a few changes concerning the number of channels in the training phase and the total number of trainable parameters, we chose a number of channels that changes increasingly from 64 to 1024.

(bloc1:64 channels, bloc2:128 channels, bloc3: 256 channels, bloc4: 510 channels, bloc5: 1024 channels).

With these variations, we obtained over 127 million trainable parameters.

III.6 Model development process

After we downloaded Pascal VOC 2012 Dataset which contains 2913 images with their associated annotations, we saved it in two different folders, one for the images and the second for the masks, then we created Numpy vectors containing the names of the images and their corresponding masks.

Next, we divided our dataset into subsets using the Sklearn library, where we took 213 examples for the validation set, and 2700 examples for the training set.

During the implementation phase, we faced issues that were memory demanding and did not allow us to upload the entire dataset, therefore, to tackle this problem we used a data generator, which basically load the data in batches instead of loading it all at once.

After the preprocessing of the data, we moved towards the conception of our model using Tensorflow and Keras library, then training the model, during this process, we were not able to use all the dataset.

The free version of Google Colab does not offer the means to accomplish this work (12 GB for RAM). However, we trained 700 examples by dividing it into slices of 100 examples with 8 epochs for each set and batch size equal to 8. Those examples needed to be saved in order not to start over again the training process. Each of the proposed models reached different epochs number as follows: Standard U-Net: 40 epochs, for the second model: 40 epochs and around 50 epochs for the third one.
In the training phase, our model took around 17 min for the two epochs, so, neither the memory issue was solved nor the time one.

III.7 Training

III.7.1 Hyper-parameters Tunning

Loss Function

For this hyper-parameter, we used a hybrid loss function combining both Categorical Cross Entropy (CCE) and Jaccard functions, we noticed better results than using only CCE.

Optimizer

For the optimizer, we used Adam optimizer (introduced in chapter II.4.10).

Batch size

We trained the model using two sizes of the batch, 4 and 8, the size 8 gave better results.

III.7.2 Results

- For the standard U-Net model we trained, we obtained the following results shown in the graphs.

We observe that the highest value for ExtendedJaccard metric in the validation phase is equal to 61.02% and it is obtained around 30 epochs.

For the fact that we loaded our dataset into slices (due to resources limitation), the validation graph of Jaccard index clearly shows zigzags, which indicates that the model is not performing as well as it should.

Regarding the loss function curve, we noticed that it's not readable because there was a big leap (30000->1859) in the error.

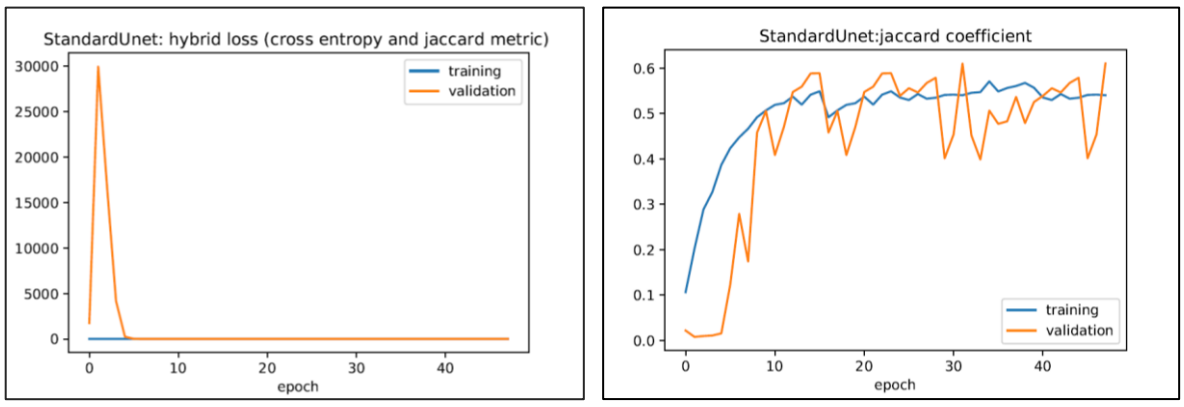

Figure III.5 Hybrid Loss and Jaccard Coefficient of the Standard U-Net

	Jaccard index	Error
Training	57.08 %	1.6735
Validation	61.02 %	1.8595

Table III.1 Jaccard Index and the Error for Standard U-Net

- As for the second model trained, which is Inception-based U-Net, the figure below shows the attained results:

The highest value for ExtendedJaccard metric in the validation phase is equal to 64.02% and it is obtained around 13 epochs.

Same for this model, Jaccard coefficient curve was not as good as expected, however, it contributed in better results.

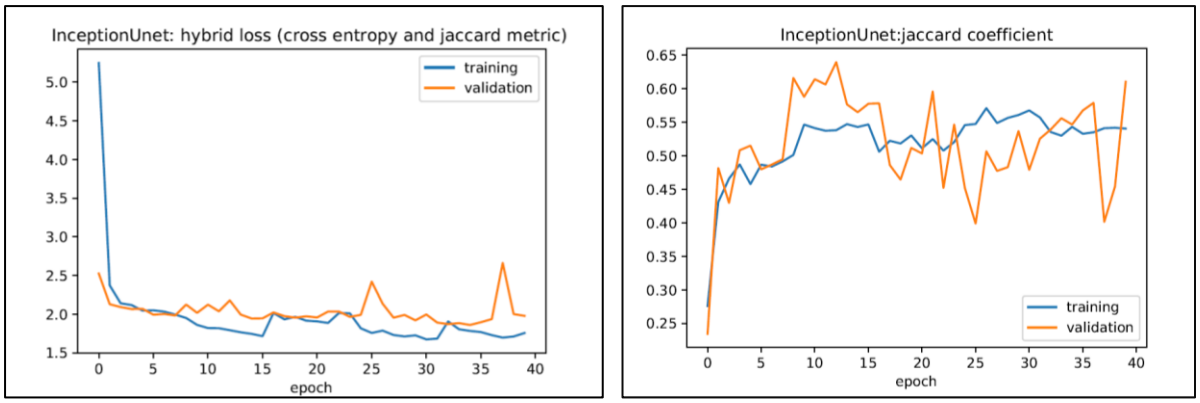

Figure III.6 Hybrid Loss and Jaccard Coefficient of the Inception-based U-Net

	Jaccard index	Error
Training	57.08 %	1.6735
Validation	64.02 %	1.8595

Table III.2 Jaccard Index and the Error for Inception- based U-Net second model

- In the last model, changing the number of channels did not improve the value of ExtendedJaccard metric in the validation phase, it reached a highest value that is equal to 62.18% after training for 15 epochs, as shown in the graph and the table below:

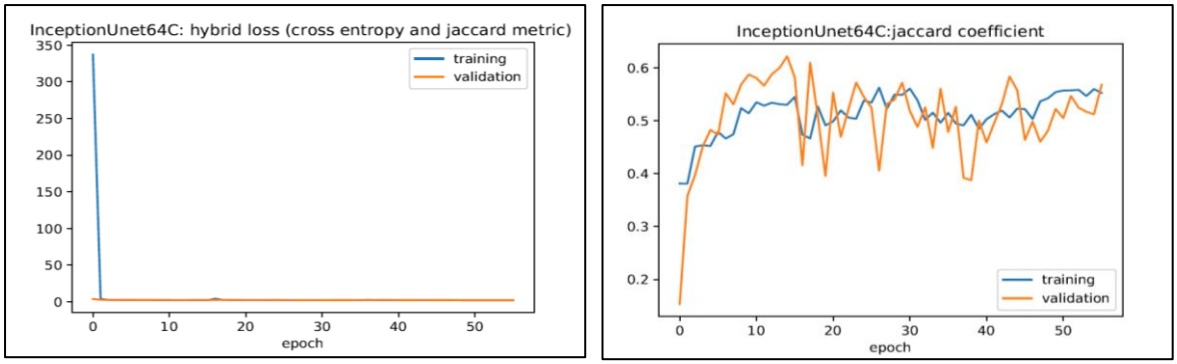

Figure III.7 Hybrid Loss and Jaccard Coefficient of the second Ineption-based U-Net model

	Jaccard index	Error
Training	57.08 %	1.6735
Validation	62.18 %	1.9586

Table III.3 Jaccard Index and the Error for the second Inception- based U-Net model

The next figure shows a few examples of semantic segmentation using our proposed model in respect to the expert-labeled image:

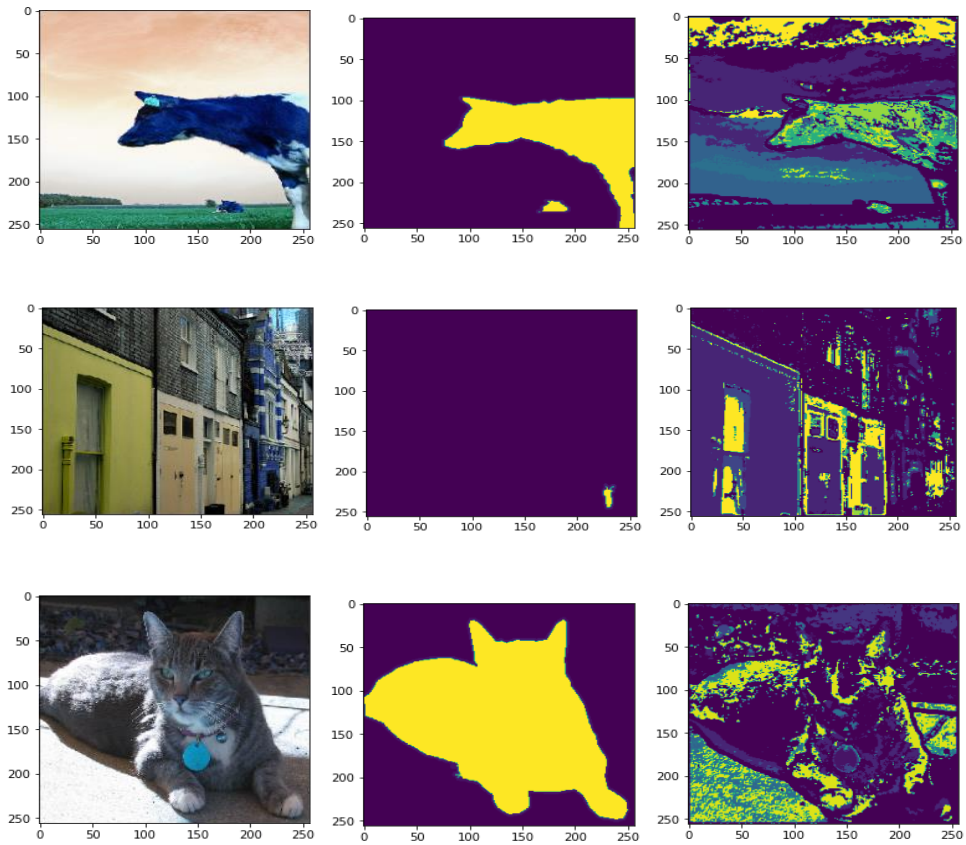

Figure III.8 Ground truth vs Prediction

III.8 Discussion

	Time	Trainable Parameters	epochs	Error	Jaccard coefficient
U-Net Standard	Around 42 min	31.044.886	40	1.8595	61.02 %
Inception-based U-Net 32	Around 42 min	31.930.876	40	1.8595	64.02 %
Inception-based U-Net 64	53 min	127 million	50	1.9586	62.18%

Table III.4 Comparison of the models

The table III.4 presents a comparison between the results obtained by the three proposed models in terms of the Jaccard coefficient, and the hybrid loss function. Also, it shows us the training time for each model, as well as the number of trainable parameters in each of them.

It can be said that all the proposed models took considerable training time, in relative to the number of parameters to train, and this is due to the use of a data generator, this has a strong impact on the results of our models, we could not have accurate results due to the fact that Google Colab doesn’t offer freely the means in terms of memory space and time to execute a model.

Using Inception's layers allowed our model to have more paths to follow, as well as a greater number of trainable parameters, which gave relatively good results.

The expansion in the number of channels per layer did not provide much, it is considered as overfitting, as it is the case in the third model with performance equal to 62.18% almost similar to the standard U-Net model (61.02%).

The best performance was obtained after training the second model "Inception-based U-Net" with a number of channels starting from 32 and 64% for accuracy.

To sum up, it can be argued that the architecture and the way the layers are connected have a significant impact on the outcomes.

III.9 Conclusion

In this last chapter, we put forward our contribution on semantic segmentation by showing our proposed models, which are a standard U-Net, inception-based U-Net.

We discussed the final results that were achieved after several attempts changing the way the layers are connected in the network, loss functions and other crucial hyper-parameters, as well as the dataset choice. These results could have been way improved using more sophisticated machines (GPU, Memory…).

General conclusion

Image segmentation is one of the most important operations in an image processing system because it lies at the intersection of image processing and analysis, which aims to group pixels together according to predefined criteria.

There is no universal solution to the segmentation problems, but rather a set of methods and algorithms, which can be combined together to solve specific problems.

In this context, we can distinguish a theoretical part in which we present the challenges and the aim of our subject, and an applicative part justified by the implementation of some operations that present the conception of a model devoted to the semantic segmentation.

In the first part, we discussed the fundamental notions of image segmentation, the various challenges that we faced, as well as the variety of existing segmentation methods such as: thresholding, clustering, region growing ... and others that are based on convolutional networks, concluding with some of the popular use cases in semantic segmentation.

In the next part, we started with quick review on different Machine Learning approaches, but mainly focused on deep learning and neural networks including CNN and FCN. Those approaches have shown better performance and high accuracy in semantic segmentation tasks. In addition, we discussed the several existing types of FCN where the U-Net architecture is considered one of the most remarkable solutions.

At the end, we presented an overview on the tests, the results and what has been achieved with regard to the primary objectives, by building three U-Net-based models inspired from FCN, and applied on PASCAL VOC 2012 images.

Perspective

As a perspective of this work, as we have had big issues in RAM and GPU, if we had the ability to train our model on a more powerful machine, higher performance and effectiveness would have been achieved.

Additionally, we can extend the presented model and use the concept of transfer learning where we reuse pre-trained models on huge dataset (ImageNet for instance), such as VGG-16, VGG-19, Inception V3, XCeption, ResNet-50 etc.

These pre-trained models allow the transfer of knowledge in order to perform faster and better process a new data. In fact, they can be used a decoder part of the UNET, or

they can be combined with traditional ML methods (such as SVM, Adaboost, Random Forest) to perform the pixel classification.

Furthermore, we can compare our model with existing encoder-decoder architectures such as Linknet, or Atrous based models.

Bibliography

[1] Boveiri, H. Reza. Vikramsingh R. Parihar. “Image Segmentation : A Guide to Image Mining”, *An ICSES Book published by ITIPPR*, Vol. 4, 2018.

[2] Chen, Long & Liu, Jiajie & Li, Han & Zhan, Wujing & Zhou, Baoding & Li, Qingquan. (2020). Dual context prior and refined prediction for semantic segmentation. Geo-spatial Information Science. 24. 1-1i3. 10.1080/10095020.2020.1785957.

[3] Rabinovich, Andrew et al. “Does image segmentation improve object categorization? “, University of California San Diego Technical Report cs2007-0908. 2007-a.

[4] Rabinovich, Andrew et al. “Objects in Context. IEEE International Conference of Computer Vision”, (ICCV), 2007-b, p.1-8.

[5] Shotton, Jamie et al. « TextonBoost for Image Understanding : Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context ». *International Journal of Computer Vision*, vol. 81, n^{o} 1, 2007, p. 2-23.

[6] Cyril Meurie. Segmentation d’images couleur par classification pixellaire et hierarchie de patitions. Thése de doctorat, Université de Caen/Basse-Normandie, octobre 2005.

[7] Cybéle Ciofolo. Segmentation de formes guidées par des modéles en neuro-imagerie. Intégration de la commande floue dans une méthode de segmentation par ensembles de niveau, Thése de Doctorat décembre 2005

[8] Yang, Allen Y. et al. « Unsupervised segmentation of natural images via lossy data compression ». *Computer Vision and Image Understanding*, vol. 110, n^{o} 2, 2008, p. 212-25.

Bibliography

[9] Zhuowen Tu et Song-Chun Zhu. « Image segmentation by data-driven markov chain monte carlo ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n° 5, 2002, p. 657-73.

[10] Pinheiro, Pedro O. Ronan, Collobert. "From image-level to pixel-level labeling with convolutional networks". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, p. 1713-1721.

[11] Papandreou, George et al. "Weakly and semi-supervised learning of a dcnn for semantic image segmentation". *In International Conference on Computer Vision (ICCV)*, 2015

[12] Guo, Dazhou et al. « Degraded Image Semantic Segmentation With Dense-Gram Networks ». *IEEE Transactions on Image Processing*, vol. 29, 2020, p. 782-95.

[13] Shiledarbaxi, Nikita. « Semantic vs Instance vs Panoptic : Which Image Segmentation Technique To Choose ». *Analytics India Magazine*, 12 avril 2021, analyticsindiamag.com/semantic-vs-instance-vs-panoptic-which-image-segmentation-technique-to-choose.

[14] Wilson, John. « Could you explain me how instance segmenation work ? » *AI pool*, 15 octobre 2019, ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works.

[15] P G, Greeshma. « Different Approaches for Semantic Segmentation ». *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020.

[16] Chakraborty, Debasish et al. *Image Segmentation Techniques*. Van Haren Publishing, 2012.

[18] Aljahdali, Sultan et Mohammad Junedul Haque. *Advanced Techniques for Image Segmentation : Image Processing*. LAP LAMBERT Academic Publishing, 2013.

[20] Yian-Leng Chang et Xiaobo Li. « Adaptive image region-growing ». *IEEE Transactions on Image Processing*, vol. 3, n° 6, 1994, p. 868-72.

Bibliography

[21] Gonzalez, Rafael C. et Richard E. Woods. *Digital Image Processing*. 3e éd., Pearson Prentice Hall, 2002.

[22] Najman, Laurent et Michel Couprie. « Watershed Algorithms and Contrast Preservation ». *Discrete Geometry for Computer Imagery*, 2003, p. 62-71.

[24] Charif, Houassine, « segmentation d'images par une approche biomimétique hybride. » Universite m'hamed bougara- boumerdes. 2012

[25] M. Melliani, segmentation d'image par cooperation regions-contours, magistère en informatique, Ecole national supérieur d'informatique, 2012

[27] Canny, John. « A Computational Approach to Edge Detection ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no 6, 1986, p. 679-98.

[28] Deriche, Rachid. « Using Canny's criteria to derive a recursively implemented optimal edge detector ». *International Journal of Computer Vision*, vol. 1, no 2, 1987, p. 167-87.

[29] Kass, Michael et al. « Snakes: Active contour models ». *International Journal of Computer Vision*, vol. 1, no 4, 1988, p. 321-31.

[30] Chen, Da et al. « Active contour for noisy image segmentation based on contourlet transform ». *Journal of Electronic Imaging*, vol. 21, no 1, 2012, p. 013009.

[32] Gonzalez, Rafael C. et Richard E. Woods. Digital Image Processing. 2e éd, Beijing: Publishing House of Electronics Industry, 2007

[33] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Bibliography

[34] Long, Jonathan et al. « Fully convolutional networks for semantic segmentation ». *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, p. 3431-40.

[35] W. X. Kang, Q. Q. Yang, R. R. Liang. "The Comparative Research on Image Segmentation Algorithms", IEEE Conference on ETCS, pp. 703-707, 2009.

[36] Pal, Nikhil R. et Sankar K. Pal. « A review on image segmentation techniques ». *Pattern Recognition*, vol. 26, n° 9, 1993, p. 1277-94

[37] C. Zhu, J. Ni, Y. Li, G. Gu,"General Tendencies in Segmentation of Medical Ultrasound Images", International Conference on ICICSE, 2009, pp. 113-117.

[38] Russell, Bryan C. et al. « LabelMe: A Database and Web-Based Tool for Image Annotation ». *International Journal of Computer Vision*, vol. 77, n° 1-3, 2007, p. 157-73.

[39] Ren, Shaoqing et al. « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ». Adv. Neural Inf. Process. Syst. 2015, pp. 91-99

[40] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand « Deep Semantic Segmentation for Automated Driving: Taxonomy, Roadmap and Challenges », 3 Aug 2017

[41] Khan, Khalil et al. « Multi-class semantic segmentation of faces ».*IEEE International Conference on Image Processing (ICIP)*, 2015.

[42] Gary, B Huang et al. "Towards unconstrained face recognition," in Computer Vision and Pattern Recognition Workshops. CVPRW'08. IEEE Computer Society Conference on. IEEE, 2008, pp. 1-8.

[43] Pham, Dzung L. et al. « Current Methods in Medical Image Segmentation ». *Annual Review of Biomedical Engineering*, vol. 2, n° 1, 2000, p. 315-337.

Bibliography

[44] Demir, Ilke et al. « DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images ». *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[45] Chitade, A.; Katiyar, S. Color Based Image Segmentation Using K-Means Clustering. International Journal of Engineering Science and Technology. 2010, 2, pp. 5319-5325 [14-3]

[46] Helber, Patrick et al. « EuroSAT : A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification ». *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, n° 7, 2019, p. 2217-26.

[47] Kazakeviciute-Januskeviciene, Giruta et al. « Assessment of the Segmentation of RGB Remote Sensing Images : A Subjective Approach ». *Remote Sensing*, vol. 12, n° 24, 2020, p. 4152.

[48] Chan, Stephanie et al. « Machine Learning in Dermatology: Current Applications, Opportunities, and Limitations ». *Dermatology and Therapy*, vol. 10, n° 3, 2020, p. 365-86.

[49] Wu, Jo-Hsuan et al. « Performance and limitation of machine learning algorithms for diabetic retinopathy screening: A meta-analysis (Preprint) ». *Journal of Medical Internet Research*, 2020.

[50] Mitchell, Tom. Machine Learning. 1re éd, *McGraw-Hill Education*, 1997.

[51] Rokach, Lior. Pattern classification using ensemble methods. *World Scientific*, Vol. 75, 2010, p.21.

[52] Rokach, Lior. Pattern Classification Using Ensemble. *World Scientific*. 2009.

[53] Sullivan, William. Machine Learning Beginners Guide Algorithms: Supervised & Unsupervised Learning, Decision Tree & Random Forest Introduction. *CreateSpace Independent Publishing Platform*, 2017.

Bibliography

[54] Ma, Xin. Using classification and regression trees: A practical primer. *IAP*, 2018.

[55] Quinlan J. Ross C4.5: Programs for machine learning. *Morgan Kaufmann Publishers*, San Mateo, 1993.

[56] Wu, Xindong et al. « Top 10 algorithms in data mining ». *Knowledge and Information Systems*, vol. 14, n° 1, 2007, p. 1-37.

[57] Schapire, Robert E. « The strength of weak learnability ». *Machine Learning*, vol. 5, n° 2, 1990, p. 197-227.

[58] Zocca, Valentino et al. Python Deep Learning. *Van Haren Publishing*, 2017.

[59] Afridi, Tariq et al. A Multimodal Memes Classification: A Survey and Open Research Issues 2020.

[60] Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed., O'Reilly, 2019.

[61] Singh, Aishwarya. "Ensemble Learning | Ensemble Techniques." Analytics Vidhya, www.analyticsvidhya.com, 18 June 2018.

[62] Wyner, Abraham J. Olson, Matthew. Bleich, Justin. Mease, David. "Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers". *Journal of Machine Learning Research*. Vol. 18 n°48, p. 1-33. Retrieved 17 March 2022.

[63] Cao Ying, Miao Qi-Guang, Liu Jia-Chen, Gao Lin. Advance and prospects of AdaBoost algorithm. *Acta Automatica Sinica*, Vol.39 n°6, 2013, p.745-758

[65] Nanni, Loris et al. « Handcrafted vs. non-handcrafted features for computer vision classification ». *Pattern Recognition*, vol. 71, 2017, p. 158-72

[66] Dahmane Khouloud. Analyse d'images par méthode de Deep Learning appliquée au contexte routier en conditions météorologiques dégradées. Vision par ordinateur et

Bibliography

reconnaissance de formes. Université Clermont Auvergne [2017-2020], 2020.

[67] LeCun, Yann et al. « Deep learning ». *Nature*, vol. 521, nº 7553, 2015, p. 436-44.

[68] Zhang, Ziwei et al. « Deep Learning on Graphs : A Survey ». *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, nº 1, 2022, p. 249-70.

[69] Heaton, Jeff. Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. 2016.

[70] KELLEY, HENRY J. « Gradient Theory of Optimal Flight Paths ». *ARS Journal*, vol. 30, nº 10, 1960, p. 947-54.

[71] Schmidhuber, Juergen. « Deep Learning ». *Scholarpedia*, vol. 10, nº 11, 2015, p. 32832.

[72] Ajit, Arohan et al. « A Review of Convolutional Neural Networks ». *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020.

[73] Goodfellow, Ian J et al. "Generative Adversarial Networks," Advances in Neural Information Processing Systems, vol. 3, June 2014.

[74] Nakamoto Pat. "Neural networks and deep learning: deep learning explained to your granny". CreateSpace Independent Publishing Platform. 2017

[75] Alom, Md Zahangir, et al. "The history began from alexnet: A comprehensive survey on deep learning approaches.". 2018.

[76] Rumelhart, David E. et al. « Learning representations by back-propagating errors ». *Nature*, vol. 323, nº 6088, 1986, p. 533-36.

[77] Pascanu, Razvan et al. "How to construct deep recurrent neural networks". In: Proceedings of the second international conference on learning representations (ICLR 2014); 2014.

Bibliography

[78] Kramer, Mark A. « Nonlinear principal component analysis using autoassociative neural networks ». *AIChE Journal*, vol. 37, nº 2, 1991, p. 233-43

[79] Gui, Jie et al. « A Review on Generative Adversarial Networks : Algorithms, Theory, and Applications ». *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[80] Fukushima, Kunihiko et Sei Miyake. « Neocognitron : A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition ». *Competition and Cooperation in Neural Nets*, 1982, p. 267-85.

[81] « Ferrier lecture - Functional architecture of macaque monkey visual cortex ». *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 198, nº 1130, 1977, p. 1-59.

[82] Lecun, Y. et al. « Gradient-based learning applied to document recognition ». *Proceedings of the IEEE*, vol. 86, nº 11, 1998, p. 2278-324.

[83] CIRESAN, Dan C. et al. « Flexible, high-performance convolutional neural networks for image Classification». 2011

[84] Heaton, Jeff. Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. 2015.

[85] Yamashita, Rikiya et al. « Convolutional neural networks : an overview and application in radiology ». *Insights into Imaging*, vol. 9, nº 4, 2018, p. 611-29.

[86] Zhou, Kevin et al. *Deep Learning for Medical Image Analysis (The MICCAI Society Book Series)*. 1re éd., *Academic Press*, 2017.

[87] Alzubaidi, Laith et al. « Review of deep learning : concepts, CNN architectures, challenges, applications, future directions ». *Journal of Big Data*, vol. 8, nº 1, 2021.

[88] Albawi, Saad et al. « Understanding of a convolutional neural network ». *2017 International Conference on Engineering and Technology (ICET)*, 2017.

Bibliography

[89] Yani, Muhamad et al. « Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail ». *Journal of Physics : Conference Series*, vol. 1201, nº 1, 2019, p. 012052.

[90] Aggarwal, Charu. *Neural Networks and Deep Learning: A Textbook*. 1st ed. 2018, Springer, 2018.

[91] Shivaprakash, Muruganandham. Master's Thesis. " Semantic Segmentation of Satellite Images using Deep Learning". 16thAugust, 2016

[92] Sharma, Siddharth et al. « ACTIVATION FUNCTIONS IN NEURAL NETWORKS ». *International Journal of Engineering Applied Sciences and Technology*, vol. 04, nº 12, 2020, p. 310-16.

[93] Maas, Andrew L, Hannun, Awni Y, and Ng, Andrew Y. Rectifier nonlinearities improve neural network acoustic models. *In Proc. ICML*, volume 30, 2013.

[94] Xu, Bing. et al. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853, 2015.

[96] Hastie, Trevor et al. « The Elements of Statistical Learning: data mining, inference, and prediction. 2nd ed ». *Springer Series in Statistics*, 2009.

[97] Goller, C. et A. Kuchler. « Learning task-dependent distributed representations by backpropagation through structure ». *Proceedings of International Conference on Neural Networks (ICNN'96)*, 1996..

[98] Bottou, Léon. "Stochastic gradient descent tricks." Neural networks: Tricks of the trade. Springer Berlin Heidelberg, pp. 421-436, 2012.

[99] Goodfellow, Ian, et al. Deep Learning. Illustrated Edition. *The MIT Press* .2016, pp 429.

[100] Gillot, Pierre et al. « Algorithmes de Descente de Gradient Stochastique avec le filtrage des paramètres pour l'entraînement des réseaux à convolution profonds» .2018

Bibliography

[101] Duchi, John. Hazan, Elad. Singer, Yoram. “Adaptive subgradient methods for online learning and stochastic optimization”. *Journal of Machine Learning Research* . Vol. 12. N°61. p. 2121-2159, July 2011.

[102] Hinton, Geoffrey. Srivastava,Nitish. Swersky, Kevin. Neural networks for machine learning: overview of mini-batch gradient descent. 2012

[103] Kingma, Diederik P. Ba, Jimmy Lei. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 2015. p. 1-3

[104] Srivastava, Nitish et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.2014. pp. 1929-1958.

[105] Bidoit, Nicole. An empirical approach to machine learning: algorithm selection, hyperparameter optimization, and automatic principle design. Ph.D. thesis proposal in computer science 2014-2017

[106] Shelhamer, Evan et al. « Fully Convolutional Networks for Semantic Segmentation ». *in Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 39, n^{o} 4, 2015, pp. 3431- 3440.

[107] Ulku, Irem. Akagündüz, Erdem. “A SURVEY ON DEEP LEARNING-BASED ARCHITECTURES FOR SEMANTIC SEGMENTATION ON 2D IMAGES” March 17, 2022

[108] Zeiler, Matthew D. et Rob Fergus. « Visualizing and Understanding Convolutional Networks ». *Computer Vision – ECCV 2014*, 2014, p. 818-33.

[109] Zeiler, Matthew D et al.” Adaptive deconvolutional networks for mid and high level feature learning” .*In ICCV*, 2011.

[110] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, MICCAI, Cham, 2015, pp.

Bibliography

234–241. Springer International Publishing.

[111] Badrinarayanan, Vijay et al. « SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, nº 12, 2017, p. 2481-95.

[112] Ulku, Irem et al. « Comparison of single channel indices for U-Net based segmentation of vegetation in satellite images ». *Twelfth International Conference on Machine Vision (ICMV 2019)*, édité par Wolfgang Osten et Dmitry P. Nikolaev, 2020.

[113] Lazebnik, S. et al. « Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories ». *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* - Volume 2 (CVPR'06), 2006

[114] He, Kaiming et al. « Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition ». *Computer Vision – ECCV 2014*, 2014, p. 346-61.

[115] Chen, Liang-Chieh et al. « DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, nº 4, 2018, p. 834-848.

[116] He, Kaiming et al. "Deep residual learning for image recognition", In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. pp. 770-778.

[117] Chan, Stephanie et al. « Machine Learning in Dermatology : Current Applications, Opportunities, and Limitations ». *Dermatology and Therapy*, vol. 10, nº 3, 2020, p. 365-86.

Bibliography

Webography

[17] Bandyopadhyay, Hmrishav. « An Introduction to Image Segmentation: Deep Learning vs. Traditional [+Examples] ». V7, 26 mai 2022, www.v7labs.com/blog/image-segmentation-guide.

[19] Splitting and merging. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/medium/segment/split.htm, 1996

[23] Canny edge detector demos. http://robotics.eecs.berkeley.edu/~sastry/ee20/cademo.html, 1996.

[31] Wikipedia contributors. « Thresholding (Image Processing) ». Wikipedia, 14 juin 2022, en.wikipedia.org/wiki/Thresholding (image_processing).

[64] Raschka, Sebastian. « When Does Deep Learning Work Better Than SVMs or Random Forests® ? » KDnuggets, www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html. Consulté le 18 juin 2022.

[95] Koech, Kiprono Elijah. « Cross-Entropy Loss Function - Towards Data Science ». *Medium*, 16 décembre 2021, towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e.

[118] « The PASCAL Visual Object Classes Homepage ».

host.robots.ox.ac.uk/pascal/VOC/index.html. Consulté le 22 juin 2022.

[119] "Google Colab." *Google Colab*, research.google.com, https://research.google.com/colaboratory/faq.html?hl=fr. Accessed 25 June 2022.

[120] "TensorFlow." *TensorFlow*, www.tensorflow.org, https://www.tensorflow.org/. Accessed 25 June 2022.

Bibliography

[121] Team, Keras. “Keras: The Python Deep Learning API.” *Keras*: The Python Deep Learning API, keras.io, https://keras.io/. Accessed 25 June 2022.

[122] “NumPy.” *NumPy*, numpy.org, https://numpy.org/. Accessed 25 June 2022.

[123] “Matplotlib — Visualization with Python.” *Matplotlib* — Visualization with Python, matplotlib.org, 2 May 2022, https://matplotlib.org/.

ملخص

تعد التجزئة الدلالية جزءا هاما في مجال رؤية الكمبيوتر، حيث يتم تسمية الفئة التي ينتمي إليها كل بكسل على حِدة في الصورة، الطرق التقليدية قدمت أداءآت ضعيفة خصوصا مع مجموعة البيانات كبيرة الحجم، لذلك اعتمدنا في عملنا هذا على احدى التقنيات الحالية المتمثلة في الشبكات الالتفافية بالكامل، وبشكل أخص U-Net، هذا الأخير جعل من الممكن التوفيق بين ماهية الأشياء التي يمكن إيجادها في الصورة المجزئة وبين موقعها في الصورة، خاصة بعد إدراج تغييرات في بنيته بإضافة طبقات (Inception على سبيل المثال)، أو تغيير من حيث عدد القنوات.

كلمات مفتاحية: التجزئة الدلالية، شبكات التفافية كاملة، U-Net، تحدي فئات الكائن المرئي باسكال، التعلم العميق.

Abstract

Semantic segmentation is an important part of the computer vision field, where the class to which each pixel in an image belongs, is labeled automatically, traditional methods have provided poor performance especially with the large-sized datasets, therefore, we have relied in our work on one of the current techniques, represented in Fully convolutional neural network, specifically, U-Net architecture. This latter made it possible to balance between "what" objects could be found in the image and their "localization", especially after incorporating changes in its architecture by adding layers (Inception modules layers for example), or changes in terms of channels number.

Key Words: Semantic Segmentation, Fully Convolutional Networks, U-Net, Pascal VOC, Deep Learning.

Résumé

La segmentation sémantique est une partie importante du champ de vision par ordinateur, où la classe à laquelle chaque pixel d'une image appartient, est étiquetée automatiquement, les méthodes traditionnelles ont fourni des performances faible, surtout avec les grands ensembles de données, par conséquent, nous nous sommes appuyés dans notre travail sur l'une des techniques actuelles, représentée dans Réseau de neurones entièrement convolutionnel, en particulier, l'architecture U-Net. Ce dernier a permis d'équilibrer entre les objets on pouvait trouver dans l'image et leur « localisation », notamment après avoir intégré des changements dans son architecture en ajoutant des couches (Inception modules par exemple), ou un changement en termes de numéro de canaux.

Mots clés: segmentation sémantique, Réseaux entièrement convolutifs, U-Net, Pascal VOC, L'apprentissage profond.