PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS

N°:……

DOMAIN: SCIENCE AND TECHNOLOGY

FIELD: TELECOMMUNICATIONS

OPTION: TELECOMMUNICATIONS SYSTEMS

Thesis presented for the
Academic Master's degree

By: **ABDELMOULA Zouhir.**
MADJIDI Mohamed Adib.

Entitled

Multi-interest U-Net for Medical Image Segmentation

Defended in front of the jury composed of:

Dr.ZERDOUMI Zohra	University of M'SILA	President
Pr. LALAOUI Lahouaoui	University of M'SILA	Supervisor
Dr.BENAHCENE Madani	University of M'SILA	Examiner

Academic year: 2023/2024

Acknowledgements

First of all, we thank ALLAH the Almighty for giving us health, courage and a great will to complete this work.

We would like to express our deepest gratitude to our supervisor Pr. LALAOUI Lahouaoui for his invaluable guidance, support and encouragement throughout this research effort.

We would like to express our sympathy to all the staff and students of the Department of Electronics at the University of M'Sila who have greatly contributed to our development.

We would also like to thank the president and the members of the jury for having done us the great honour of agreeing to evaluate this thesis.

We express our sincere gratitude to our families for their endless love, understanding and encouragement. Their unwavering support has been the cornerstone of our perseverance and success.

Madjidi Mohamed Adib and Abdelmoula Zouhir

Dedication

This study is entirely dedicated to our beloved parents, who have been our source of inspiration, who continually provide us with their moral, spiritual, emotional and financial support. To our brothers, sisters, parents, friends and classmates.

Madjidi Mohamed Adib and Abdelmoula Zouhir

ABSTRACT

Medical image segmentation plays a central role in various clinical applications, aiding in diagnosis, treatment planning, and disease monitoring. Among the leading architectures, U-net, U-net++, and U-net3+ have emerged as top choices, each offering unique advancements in capturing contextual information and improving segmentation accuracy. This thesis presents a comparative analysis of these three models, focusing on their performance and computational efficiency in a medical imaging dataset. Through experimentation and evaluation, information on the strengths and weaknesses of each architecture is provided, allowing for a better understanding of their applicability in real-world cases

Keywords: Medical image segmentation,U-net,U-net++,U-net3+,Comparative analysis Performance evaluation

RÉSUMÉ

La segmentation des images médicales joue un rôle central dans diverses applications cliniques, facilitant le diagnostic, la planification des traitements et la surveillance des maladies. Parmi les principales architectures, U-net, U-net++ et U-net3+ sont apparus comme des choix de premier ordre, chacune offrant des avancées uniques dans la capture d'informations contextuelles et l'amélioration de la précision de la segmentation. Cette thèse présente une analyse comparative de ces trois modèles, en se concentrant sur leurs performances et leur efficacité de calcul dans un jeu de données d'imagerie médicale. Grâce à l'expérimentation et à l'évaluation, des informations sur les forces et les faiblesses de chaque architecture sont fournies, ce qui permet de mieux comprendre leur applicabilité dans des cas réels.

Mots clés:Segmentation d'images médicales, Unet,Unet++,Unet3+, Analyse comparative, Évaluation de la performance.

ملخص

تلعب تجزئة الصور الطبية دورا مركزيا في التطبيقات السريرية المختلفة ، مما يساعد في التشخيص وتخطيط العلاج ومراقبة الأمراض. من بين البنى الرائدة ، برزت U-net و ++ U-net و + U-net3 كأفضل الخيارات ، حيث يقدم كل منها تطورات فريدة في التقاط المعلومات السياقية وتحسين دقة التجزئة. تقدم هذه الأطروحة تحليلا مقارنا لهذه النماذج الثلاثة ، مع التركيز على أدائها وكفاءتها الحسابية في مجموعة بيانات التصوير الطبي. من خلال التجريب والتقييم ، يتم توفير معلومات حول نقاط القوة والضعف في كل بنية ، مما يسمح بفهم أفضل لإمكانية تطبيقها في حالات العالم الحقيقي.

الكلمات المفتاحية: تجزئة الصور الطبية ، U-net، ++ U-net، + U-net3، التحليل المقارن ،تقييم الأداء

List of Figures

List of Tables

Contents

General Introduction

Interpretation of medical images such as CT and MRI requires extensive training and skills because the segmentation of organs and lesions needs to be performed layer by layer. Manual segmentation means a heavy workload to the doctors, which can introduce bias if it involves the subjective opinions of doctors. To analyze complicated images, it usually requires doctors to make a joint diagnosis, which is time consuming. Furthermore, automatic segmentation is a challenging task, and it is still an unsolved problem for most medical applications due to the wide variety connected with image modalities, encoding parameters, and organic variability. Oncology is a global problem with a significant impact on the health and lives of people around the world. Cancer occupies a special place as one of the most frequently diagnosed oncological diseases. Early diagnosis plays a crucial role in preventing the spread of cancer cells through the lymphatic pathways and damage to critical organs. Detecting malignant neoplasms in their early stages allows healthcare providers to take timely action to limit the spread of cancer and preserve the integrity of vital body structures. Methods include invasive biopsy and non-invasive techniques such as ultrasound. However, the complexity of interpreting these images is attributed to their low resolution and the interference of speckle noise. In response, computer-aided diagnosis (CAD) systems employing deep learning (DL) have been developed to streamline medical image analysis and offer precise decision support for radiologists. For automated cancer segmentation using deep learning, it is necessary to select models and algorithms that give the best results. This allows changes to be made to the medical images and allows the tumor boundaries to be accurately defined. The development of an accurate and user-friendly computer-aided design system for accurate tumor identification (malignant or benign) using machine learning has the potential to improve the efficiency of doctors, improve diagnosis and contain cancer progression. Hopefully, with the help of such systems, it will be possible to reduce the progression of such a terrible disease. Nowadays, the application of deep learning technology in medical imaging has attracted extensive attention. How to automatically recognize and segment the lesions in medical images has become one of the issues that concern lots of researchers. Ronneberger et al. proposed U-Net at the MICCAI conference in 2015 to tackle this problem, which was a breakthrough of deep learning in segmentation of medical imaging. U-Net is a Fully Convolutional Network (FCN) applied to biomedical image segmentation, which is composed of the encoder, the bottleneck module, and the decoder. The widely used U-Net meets the requirements of medical image

segmentation for its U-shaped structure combined with context information, fast training speed, and a small amount of data used. Since U-Net was proposed, its encoder-decoder-hop network structure has inspired a large amount of segmentation means in medical imaging. Such deep learning technologies as attention mechanism, dense module, feature enhancement, evaluation function improvement, and other basic U-Net structures have been introduced into medical image segmentation and become widely adopted. These variations of U-Net-related deep learning networks are designed to optimize results by improving the accuracy and computing efficiency of medical image segmentation through changing network structures, adding new modules.

The aim of this work is to efficiently use U-Net-related deep learning network techniques in the segmentation of medical images for biomedical applications.

This document is organized as follows: The thesis comprises three main chapters. The first chapter, we provide a general description of segmentation methods. In the second chapter, several models of Deep Learning are explained and essentially the convolutional neural network CNN. The last chapter, we ended by presenting the results and testing of the application of the different U-net models on Medical Images. Finally, a general conclusion summarizes the various works carried out in this works.

Chapter I
Image segmentation

I.1 Introduction:

Image segmentation is a fundamental task in computer vision that involves partitioning an image into multiple segments or regions. The goal is to simplify the representation of an image to make it easier to analyze and understand. In segmentation, pixels with similar attributes such as color, intensity, or texture are grouped together to form meaningful regions. This technique is widely used in various applications such as object detection, medical imaging, autonomous driving, and satellite image analysis. It plays a crucial role in tasks like identifying objects, separating foreground from background, and extracting meaningful information from images. Mathematically complete segmentation of an image **I** is a finite set of regions R1,. . , Rn

$$\bigcup_{i=1}^{n} R_i = I \quad \text{(I.1)}$$

$$\forall i \neq j, R_i \cap R_j = \varnothing \quad \text{(I.2)}$$

I.2 Concept of image:

I.2.1 Definition of the digital image:

An image is defined as a two-dimensional function,F(x,y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is called the intensity of that image at that point. When x,y, and amplitude values of F are finite, we call it a digital image. In other words, an image can be defined by a two-dimensional array specifically arranged in rows and columns. Digital Image is composed of a finite number of elements, each of which elements have a particular value at a particular location. These elements are referred to as picture elements, image elements, and pixels. A Pixel is most widely used to denote the elements of a Digital Image.

I.2.2 Types of an image:

BINARY IMAGE– The binary image as its name suggests, contain only two pixel elements i.e. 0 & 1, where 0 refers to black and 1 refers to white. This image is also known as Monochrome.

BLACK AND WHITE IMAGE– The image which consists of only black and white color is called BLACK AND WHITE IMAGE.

8 bit COLOR FORMAT– It is the most famous image format. It has 256 different shades of colors in it and commonly known as Grayscale Image. In this format, 0 stands for Black, and 255 stands for white, and 127 stands for gray.

16 bit COLOR FORMAT– It is a color image format. It has 65,536 different colors in it. It is also known as High Color Format. In this format the distribution of color is not as same as Grayscale image. [1]

I.2.3 Histogram of an image:

An image histogram is a graphical representation of the distribution of pixel intensities in an image. It plots the frequency of occurrence of each intensity value (usually ranging from 0 to 255 for grayscale images) along the horizontal axis, against the number of pixels with that intensity value along the vertical axis.

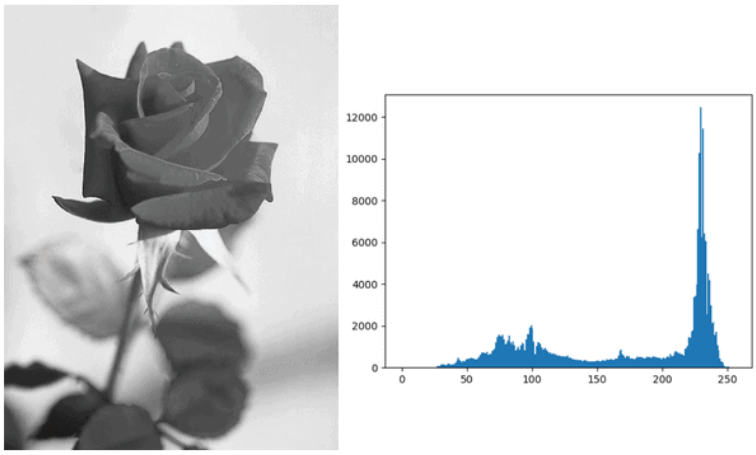


Figure I.1: An image with its histogram [2]

I.3 Traditional Image Segmentation Techniques:

I.3.1 Manual Segmentation:

Manual segmentation is usually performed by an expert: often a radiologist or a specialized clinician. This is usually done in a slice-by-slice manner, but is also possible in 3D, with the expert either encircling the ROI or annotating the voxels of interest. While an advantage of this segmentation method is that we can utilize expert knowledge, drawbacks of this method are that it is very time consuming and prone to intra and interobserver variability, as can be seen in Fig. I.2. Thus, although manual segmentation is usually regarded as the golden standard, it often has a large variability, which can result in a large difference in the extracted radiomics values [3].

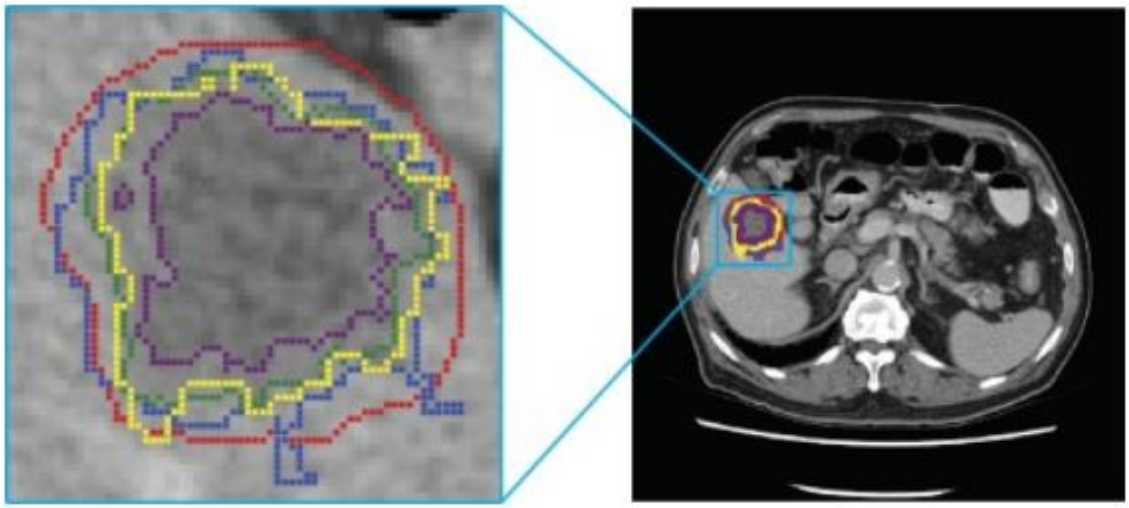


Figure I.2. Manual segmentations of multiple observers of a colorectal liver metastasis on an axial slice of a CT scan.

I.3.2 Threshold Method:[4]

It is an example of pixel based segmentation algorithm and one of the oldest methods where segmentation is achieved by collecting all pixels having intensity between two thresholding values into the same class. For this segmentation method, pixel amplitude is the most important feature that can be applied in the segmentation process to segment an object from the background. If the desired object is white and the background is black, or vice versa, it is an easy task to set an appropriate threshold to segment this object from its background. According to how bright or how dark the pixels are, they are labeled as belonging to the object or background. Threshold is selected to differentiate between the dark and the bright levels. The grey-level histogram of an image is studied to determine the image dark and bright regions. The histogram is supposed to have two peaks and the threshold is selected as the point at the minimum in between. To segment more than one object, a multimodal histogram with multiple thresholds could be used. If the mean grey-level in the image varies throughout the image, where the threshold suitable for one region may be unsuitable for the other region, the image can be split into smaller regions in which the mean greylevel is nearly constant and threshold each region as before.

Threshold techniques can be divided into two classes: the traditional or the global thresholding and the other is the adaptive thresholding. In the global thresholding, only one single threshold value is used for the entire image. In the adaptive threshold, the threshold value is determined for each pixel to locate it to the object or to the background.

I.3.2.1 Basic Global Thresholding;

For a global threshold T, a point (x, y) for which is f(x, y)>T is classified as object point otherwise is called background point. In grayscale images (binary converted), the values greater than T will be one (white) otherwise zero (black).

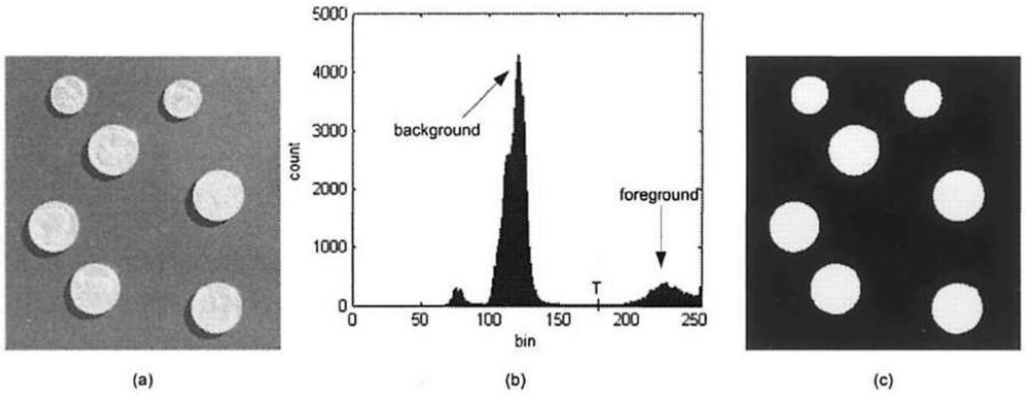


Figure I.3 Global Thresholding [5]

This produces two groups of pixels, R1 which contains pixels with gray level values >T and R2 which contains pixels with gray level values < T. Then the mean (average gray level) values mean1 and mean2 for pixels in R1 and R2 are computed, and finally compute the threshold: T= 0.5(mean1+mean2) (I.3)

Otsu's method is a famous example of global thresholding. This method works well when the image has uniform lighting conditions and good contrast between foreground and background.

I.3.2.2 Multimodal Thresholding:

The process of determining more than one threshold value is called multithresholding. Multimodal thresholding sets a point (x,y) to one object class if T1< f(x,y) <T2 , to the other object class if f(x,y) >T2 and to the background if **f(x,y) < T1.**

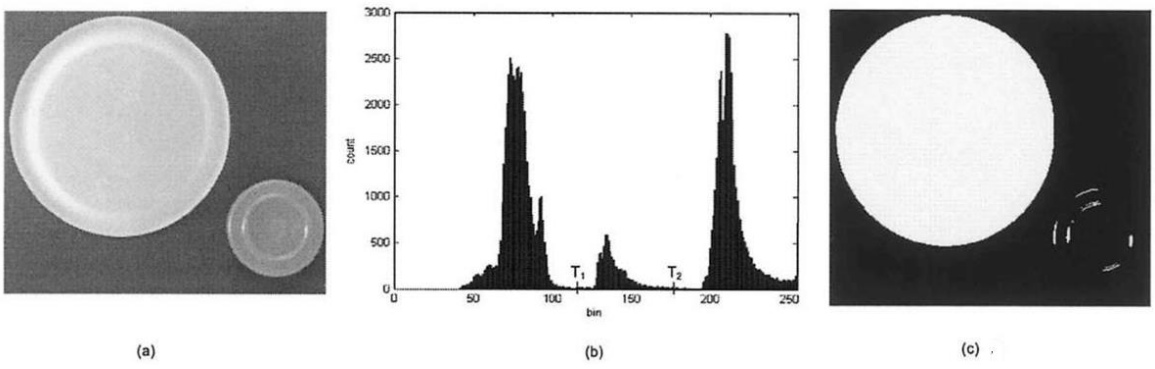


Figure I.4 Multimodal Thresholding [4]

I.3.2.3 Adaptive Thresholding:

Adaptive thresholding is a technique used in image processing to separate objects from the background in images with varying lighting conditions. In standard thresholding, a single fixed threshold value is applied to the entire image, resulting in either foreground or background pixels. However, this approach might not be suitable for images with non-uniform lighting.

Adaptive thresholding, on the other hand, calculates the threshold for each pixel based on its local neighborhood. This means that different parts of the image can have different threshold values, which can better account for variations in lighting.

There are several methods for adaptive thresholding, including:

1. **Mean Adaptive Thresholding**: The threshold for each pixel is calculated based on the mean intensity of its local neighborhood.
2. **Gaussian Adaptive Thresholding**: Similar to mean adaptive thresholding, but the threshold is calculated based on the weighted mean using a Gaussian window.
3. **Median Adaptive Thresholding**: The threshold for each pixel is calculated based on the median intensity of its local neighborhood.
4. **Adaptive Thresholding using Integral Image**: This method uses integral images to efficiently compute the sum of intensities within local neighborhoods, making it faster than traditional methods.

Adaptive thresholding is particularly useful in scenarios where the lighting conditions vary across the image or when the contrast between foreground and background is not consistent. It's commonly used in applications such as document binarization, object detection, and image segmentation.

I.3.3 Edge-Based Segmentation:

Edge-based segmentation is a method used in image processing to identify boundaries or edges of objects within an image. The basic idea is to detect abrupt changes in intensity or color, which typically correspond to object boundaries. Once these edges are detected, they can be used to segment the image into different regions or objects**.**

I.3.3.1 Edge-based segmentation process:

The classic edge detector process follows these steps:

1. **Edge Detection**: The first step is to detect edges in the image. This is typically done using edge detection algorithms such as the Sobel operator, the Prewitt operator, the Canny edge detector, or the Laplacian of Gaussian (LoG). These algorithms highlight regions in the image where intensity changes rapidly, which often correspond to object boundaries.
2. **Edge Enhancement (Optional)**: In some cases, edge enhancement techniques may be applied to enhance the detected edges and make them more prominent. This can involve operations such as edge sharpening or filtering.
3. **Edge Linking**: Detected edge pixels may not form continuous lines due to noise or other factors. Edge linking algorithms are used to connect adjacent edge pixels into continuous curves or contours, which represent the boundaries of objects in the image.
4. **Segmentation**: Once the edges are detected and linked, they can be used to segment the image into different regions or objects. This segmentation can be based on various criteria, such as the enclosed area within the contours or the properties of the regions defined by the contours.

I.3.3.2 Edge Models:

Edge models are classified according to their intensity profiles. A step edge involves a transition between two intensity level levels occurring ideally over the distant of 1 pixel. Figure I.5(a) shows a section of a vertical step and a horizontal intensity profile through the edge. Step edges occur, for example, in images generated by a computer for use in areas such as solid modelling and animation. These clean, ideal edges can occur over the distance of 1 pixel, provided that no additional processing (such as smoothing) is used to make them look real. Digital step edges are used frequently as edge models in algorithm development. In practice, digital images have edges that are blurred and noisy, determined by limitations in the focusing mechanism and noise level determined by the electronic components of imaging system. In such situations, edges are more closely modelled as having an intensity ramp profile such as edge in Figure I.5(b). The slope of the ramp is inversely proportional to the degree of blurring in the edge. It can be seen that in this model, there is no thin (1

pixel thick) path. Instead, an edge point which is now any point contained in the ramp, and an edge segment would then be a set of such points that are connected. A third model of an edge is called roof edge, having the characteristics illustrated in Figure I.5(c). Roof edges are models of lines through a region, with the base (width) of a roof edge being determined by the thickness and sharpness of the line [6]. The figures below show the various edge models.

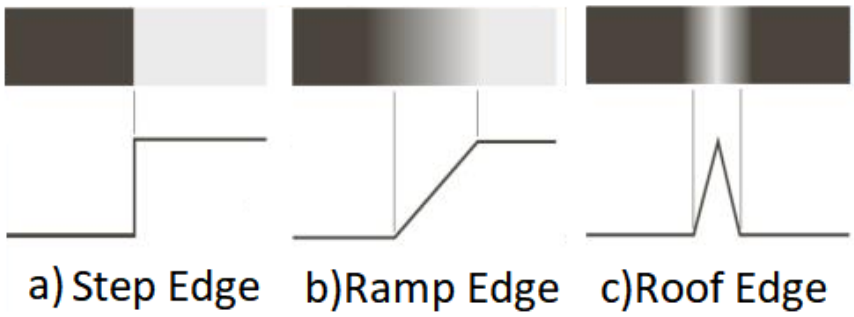


Figure I.5 Edge Models with intensity profile

I.3.3.3 Edge Detection Techniques:

Edge detection is a fundamental tool for image segmentation. Edge detection methods transform original images into edge images benefits from the changes of grey tones in the image. In image processing especially in computer vision, the edge detection treats the localization of important variations of a gray level image and the detection of the physical and geometrical properties of objects of the scene. It is a fundamental process detects and outlines of an object and boundaries among objects and the background in the image. Edge detection is the most familiar approach for detecting significant discontinuities in intensity values. Edges are local changes in the image intensity. Edges typically occur on the boundary between two regions. The main features can be extracted from the edges of an image. Edge detection has major feature for image analysis. These features are used by advanced computer vision algorithms. Edge detection is used for object detection which serves various applications like medical image processing, biometrics etc. Edge detection is an active area of research as it facilitates higher level image analysis. There are three different types of discontinuities in the grey level like point, line and edges. Spatial masks can be used to detect all the three types of discontinuities in an image. There are many edge detection techniques in the literature for image segmentation. The most commonly used discontinuity based edge detection techniques are reviewed in this section. Those

techniques are Roberts edge detection, Sobel Edge Detection, Prewitt edge detection, , LoG edge detection and Canny Edge Detection.[7]

I.3.3.3.1 Roberts Edge Detection

The Roberts edge detection is introduced by Lawrence Roberts (1965). It performs a simple, quick to compute, 2-D spatial gradient measurement on an image. This method emphasizes regions of high spatial frequency which often correspond to edges. The input to the operator is a grayscale image the same as to the output is the most common usage for this technique. Pixel values in every point in the output represent the estimated complete magnitude of the spatial gradient of the input image at that point.

-1	0
0	+1

G_x

0	-1
+1	0

G_y

I.3.3.3.2. Sobel Edge Detection :

The Sobel edge detection method was introduced by Sobel in 1970 (Rafael C. Gonzalez (2004)). The Sobel technique of edge detection for image segmentation finds edges using Sobel approximation derivative . It performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that corresponds to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image . The operator consist of a pair of 3x3 convolution mask, one mask is simply the other rotated by 90°. This is very similar to Robert Cross operator.

-1	-2	-1
0	0	0
+1	+2	+1

G_x

-1	0	-1
-2	0	+2
-1	0	+1

G_y

I.3.3.3.3 Prewitt Edge Detection:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images..

I.3.3.3.4 LoG Edge Detection:

The Laplacian of Gaussian (LoG) was proposed by Marr(1982). The LoG of an image f(x,y) is a second order derivative defined as,

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{(I.4)}$$

It has two effects, it smoothes the image and it computes the Laplacian, whch yields a doubleedge image. Locating edges then consists of finding the zero crossings between the double edges. The digital implementation of the Laplacian function is usually made through the mask below,

0	-1	0
-1	4	-1
0	-1	0

G_x

-1	-1	-1
-1	8	-1
-1	-1	-1

G_y

I.3.3.3.5 Canny Edge Detection:

In industry, the Canny edge detection technique is one of the standard edge detection techniques. It was first created by John Canny for his Master's thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the image afterwards it applying the tendency to find the edges and the serious value for threshold. The algorithmic steps are as follows:

• Convolve image f(r, c) with a Gaussian function to get smooth image f^(r, c).

$$f\hat{}(r, c)=f(r,c)*G(r,c,6) \quad \text{(I.5)}$$

• Apply first difference gradient operator to compute edge strength then edge magnitude and direction are obtain as before.

• Apply non-maximal or critical suppression to the gradient magnitude.

• Apply threshold to the non-maximal suppression image. Unlike Roberts and Sobel, the Canny operation is not very susceptible to noise. If the Canny detector worked well it would be superior.

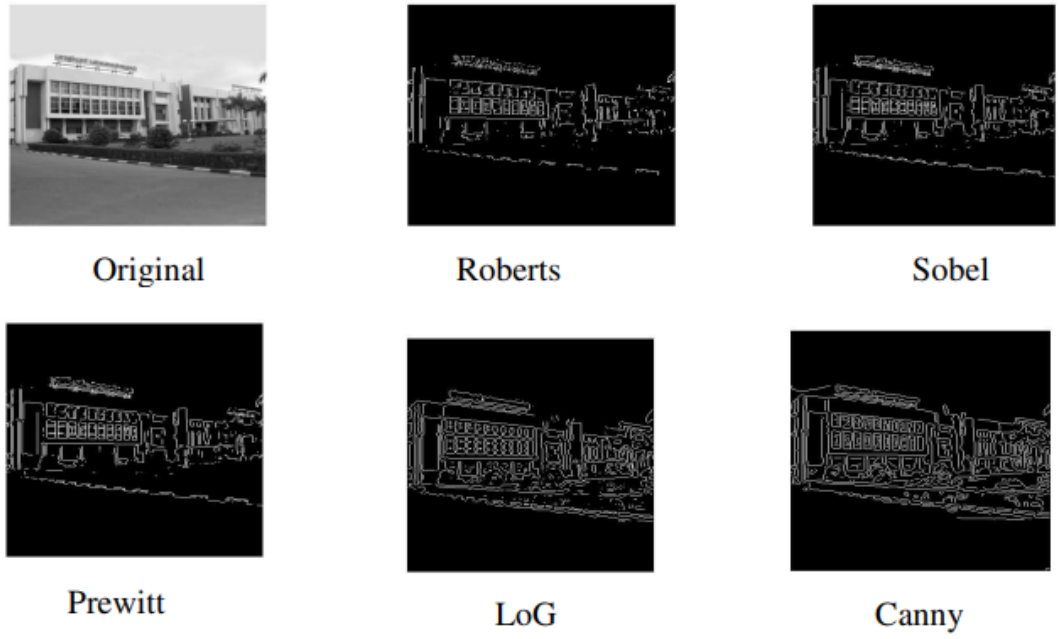


Figure I.6 Original Image with the result of various edge detection techniques

I.3.4 Region-Based Segmentation:

This process involves dividing the image into smaller segments that have a certain set of rules. This technique employs an algorithm that divides the image into several components with common pixel characteristics. The process looks out for chunks of segments within the image. Small segments can include similar pixes from neighboring pixels and subsequently grow in size. The algorithm can pick up the gray level from surrounding pixels. [8]

Figure I.7 Region-Based Segmentation

I.3.4.1 Region Growing Segmentation:

Region growing is a procedure that groups pixels or sub regions in to layer regions based on predefined criteria. The basic approach is to start with a set of seed points and from there grow regions by appending to each seed these neighboring pixels that have properties similar to the seed. The objective of segmentation is to partition an image into regions. We approached this problem by finding boundaries between regions based on discontinuities in gray levels, whereas segmentation was accomplished via thresholds based on the distribution of pixel properties, such as gray-level values or color

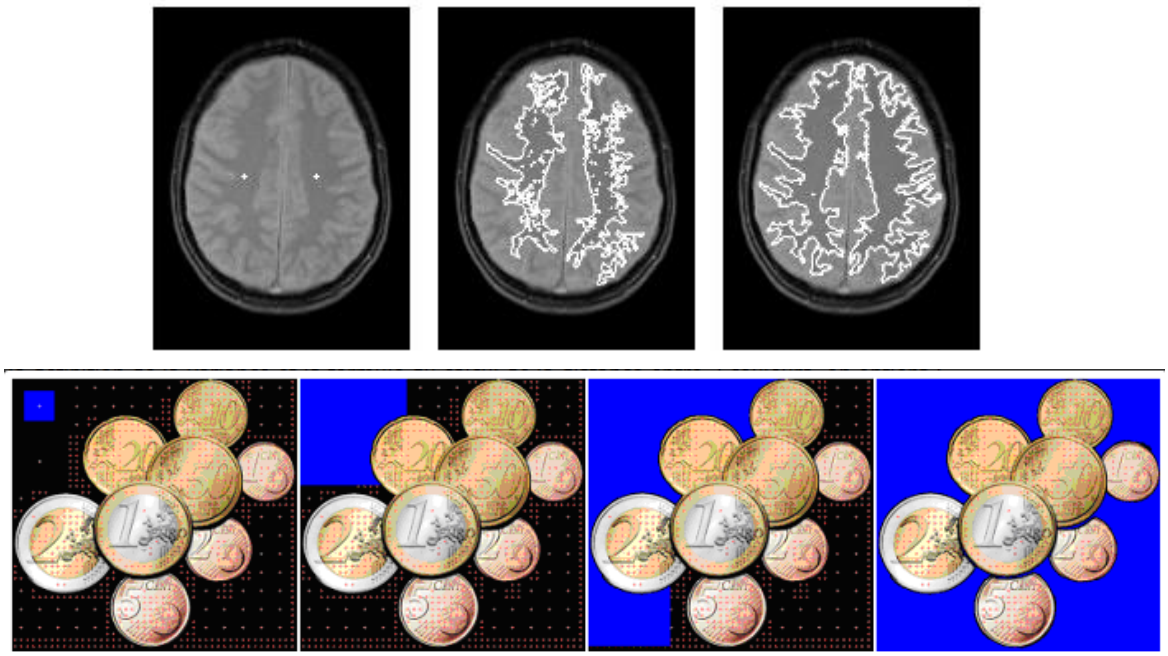


Figure I.8 Examples of Region Growing Segmentation

Basic Formulation:

The growing continues until a stopping rule takes effect. The region growing has several practical problems: how to select the seed points growing rule (or uniform criterion) stopping rule. Let R represent the entire image region. We may view segmentation as a process that partitions R into n subregions, R1, R 2..., Rn, such that The predicate P(Ri) is used to check the condition. In any region, if P(Ri)= true, then image is subdivided into various subimages. If P(Ri) = false, then divide the image into quadrants. If P(Ri) = false, then further divide the quadrants into sub quadrants

Region Growing:

As its name implies, region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria. The basic approach is to start with a set of "seed" points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (such as specific ranges of gray level or color). When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagerydepends heavily on the use of color. This problem would be significantly more difficult, or even impossible, to handle without the inherent information available in color images. When the images are monochrome, region analysis must be carried out with a set of descriptors based on gray levels and spatial properties (such as moments or texture). Basically, growing a region should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as gray level, texture, and color, are local in nature and do not take into account the "history" of region growth. Additional criteria that increase the power of a region growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the gray level of a candidate and the average gray level of the grown region), and the shape of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available.[9]

I.3.4.2 Region Merging Segmentation:

Region merging segmentation is a technique used in image processing and computer vision to partition an image into regions or segments based on certain criteria such as similarity in color, texture, or intensity. The basic idea behind region merging segmentation is to iteratively merge neighboring pixels or regions that satisfy predefined similarity conditions until no further merging is possible.

Here's a simplified outline of the region merging segmentation process:

1. **Initial Segmentation**: Start with an initial segmentation where each pixel or group of pixels constitutes a separate region.
2. **Merge Criterion**: Define a criterion for merging adjacent regions. This criterion can be based on various factors such as color similarity, intensity, texture, or a combination of these.
3. **Merge Process**: Iteratively examine neighboring regions and merge them if they satisfy the merge criterion. This process continues until no further merging is possible or until a stopping condition is met (e.g., when the regions become too large or when the number of regions reaches a certain threshold).
4. **Stopping Condition**: Define a stopping condition to halt the merging process. This can be based on parameters such as the maximum allowable region size, a threshold on region similarity, or a predefined number of iterations.
5. **Output**: The final output of the region merging segmentation is a partition of the image into homogeneous regions, where each region consists of pixels that are similar according to the defined criteria.

Region merging segmentation algorithms vary in their specific implementation details, such as the choice of merge criterion, the method for representing regions, and the stopping conditions. These algorithms can be further optimized for specific applications or types of images to achieve better segmentation results.

I.3.4.3 Region Splitting Segmentation:

The basic idea of region splitting is to break the image into a set of disjoint regions which are coherent within themselves. The algorithm is recursive and terminates when all regions are homogeneous. An example of this is Olhander's algorithm, which uses the peaks of the grayscale histogram as a dividing criterion. A disadvantage of splitting methods is that they are based on global statistics; A small region next to a large one will therefore rarely be returned.

The commonly used method is to make a blocky dichotomy of the image. To do this, we start by defining a block the size of the image, and then we look at the contents of this block. If the block is homogeneous (contains only similar pixels) then the decomposition is stopped. Otherwise, we divide the block into 4 sub-blocks and examine the contents of

each sub-block. The geometry of the slicing has a direct influence on the segmentation result. For example, the quad-tree method shows square regions. There are other types of sharing (triangle, pyramid). The choice of the type of sharing is made according to the forms you want to segment. The goal of these methods is to automatically divide and share an image into a set of homogeneous regions according to the chosen criterion (the same as before).

- Initially take the image as a whole to be the area of interest.
- Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.
- If **TRUE** then the area of interest corresponds to a region in the image.
- If **FALSE** split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.
- This process continues until no further splitting occurs. In the worst case this happens when the areas are just one pixel in size.
- This is a ***divide and conquer*** or ***top down*** method.

If only a splitting schedule is used then the final segmentation would probably contain many neighboring regions that have identical or similar properties.

Thus, a ***merging*** process is used after each split which compares adjacent regions and merges them if necessary. Algorithms of this nature are called ***split and merge*** algorithms.[10]

I.3.4.4 Region Splitting and Merging Segmentation:

Region Splitting and Merging (RSM) segmentation is a hybrid approach that combines the processes of region splitting and region merging to achieve image segmentation. It aims to overcome the limitations of each individual method by leveraging their strengths.

RSM segmentation combines the advantages of both region splitting and region merging techniques, allowing for more flexible and adaptive segmentation that can handle a wide range of image characteristics and complexities. It is often used in scenarios where neither splitting nor merging alone can provide satisfactory segmentation results.

A. Splitting Top-down approach:

In the context of image segmentation, a top-down approach to splitting refers to a method where the segmentation process starts with the entire image considered as a single region, and then proceeds to recursively split this region into smaller and more homogeneous sub-regions until certain criteria are met. This approach is often used in hierarchical segmentation methods and is particularly suitable for situations where the goal is to partition the image into regions of varying sizes and complexities.

Recursive splitting:

1 Partition the region, for example from its histogram.

2 For each resulting region, if possible (and necessary) go back to 1.

B. Quadtree:

A quadtree is a tree data structure commonly used in computer science and image processing for spatial partitioning of two-dimensional space. It recursively divides a space into four equal quadrants until a certain condition is met, typically a minimum size threshold or a maximum depth limit. Quadtree structures are particularly useful in various applications, including image compression, collision detection, and spatial indexing.

Here's how a quadtree typically works:

1. **Initial Region**: Start with the entire two-dimensional space to be partitioned, often represented as a square or rectangle.
2. **Partitioning**: Divide the space into four equal quadrants, typically by splitting it into two equal halves horizontally and two equal halves vertically. Each resulting quadrant becomes a child node of the root node in the quadtree.
3. **Recursion**: Repeat the partitioning process recursively for each quadrant until a termination condition is met. This condition is often based on a minimum size threshold for the quadrants or a maximum depth limit for the quadtree.
4. **Leaf Nodes**: Once the termination condition is reached, the quadrants that cannot be subdivided further become leaf nodes of the quadtree. These leaf nodes represent the smallest regions or units of the space.

5. **Tree Structure**: The resulting quadtree forms a hierarchical structure, with each node representing a quadrant of the space and containing references to its four child nodes (if applicable). Leaf nodes contain information about the corresponding region of the space, such as pixel values in the case of image processing applications.

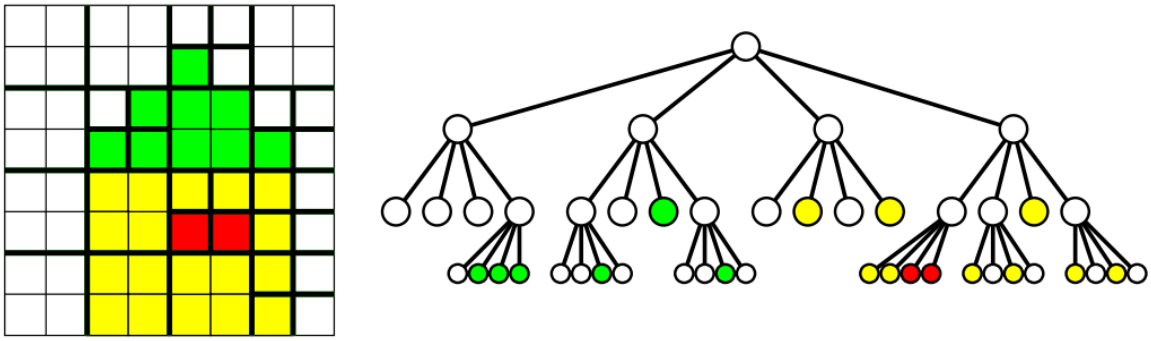


Figure I. 9 Quadtree

C. Merging Bottom-up approach:

In image segmentation, a bottom-up approach to merging involves starting with atomic elements (e.g., individual pixels or small homogeneous regions) and iteratively merging adjacent elements or regions based on certain similarity criteria until larger and more coherent regions are formed. This approach is also known as agglomerative clustering or hierarchical clustering.

D. Data structure for merging : RAG

Region Adjacency Graph : non oriented graph where the nodes correspond to the image regions. There is an edge between 2 nodes if and only if the 2 corresponding regions are adjacent. We need a geometric description of the regions. [11]

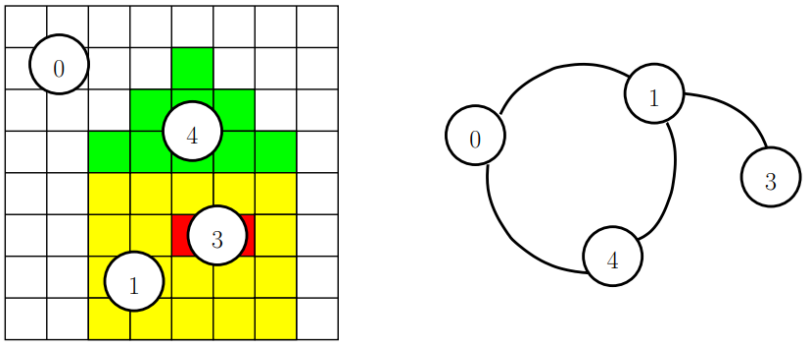


Figure I.10 Region Adjacency Graph

Figure I.11 Example of Splitting and merging

I.3.5 Watershed Segmentation:

Watershed algorithm which is a mathematics morphological method for image segmentation based on region processing, has many advantages. The result of watershed algorithm is global segmentation, border closure and high accuracy. It can achieve one-pixel wide, connected, closed and exact location of outline. The basic concept of watershed is based on visualizing a gray level image into its topographic representation, which includes three basic notions: minima, catchment basins and watershed lines. In the image of Fig.I.12 a), if we imagine the bright areas have "high" altitudes and dark areas have "low" altitudes, then it might look like the topographic surface illustrated by Fig. I.12 b). In this surface, it is natural to consider three types of points: (1) points belonging to the different minima; (2) points at which water would fall with certainty to a single minimum; and (3) points at which water would be equally likely to fall to more than one minimum. The first type of points forms different minima of the topographic surface. The second type points which construct a gradient interior region is called catchment basin. The third type of points form crest lines dividing different catchment basins, which is termed by watershed lines.

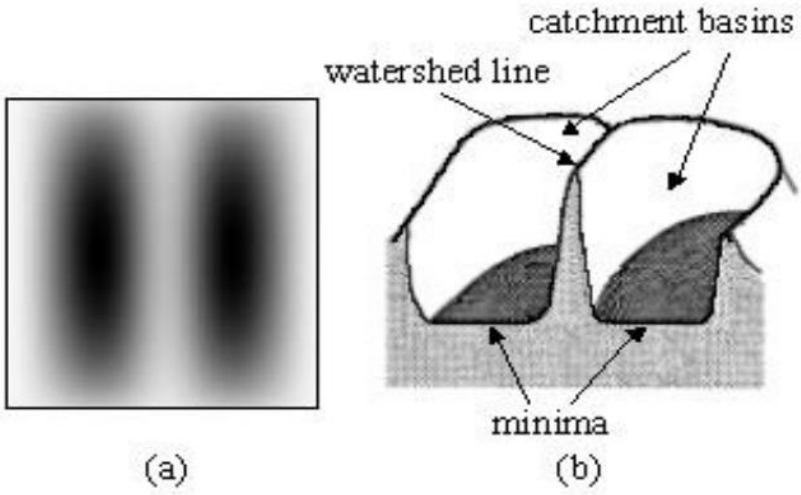


Figure I.12 Watershed Segmentation. a) A gray level image b)Topographic surface

Watershed transform:

The algorithm introduced by Luc Vincent and Pierre Soille is based on the concept of "immersion". Each local minima of a gray-scale image I which can be regarded as a surface has a hole and the surface is immersed out into water. Then, starting from the minima of lowest intensity value, the water will progressively fill up different catchment basins of image (surface) I. Conceptually, the algorithm then builds a dam to avoid a situation that the water coming from two or more different local minima would be merged. At the end of this immersion process, each local minimum is totally enclosed by dams corresponding to watersheds of image (surface) Figure I.13 shows this procedure graphically [12].

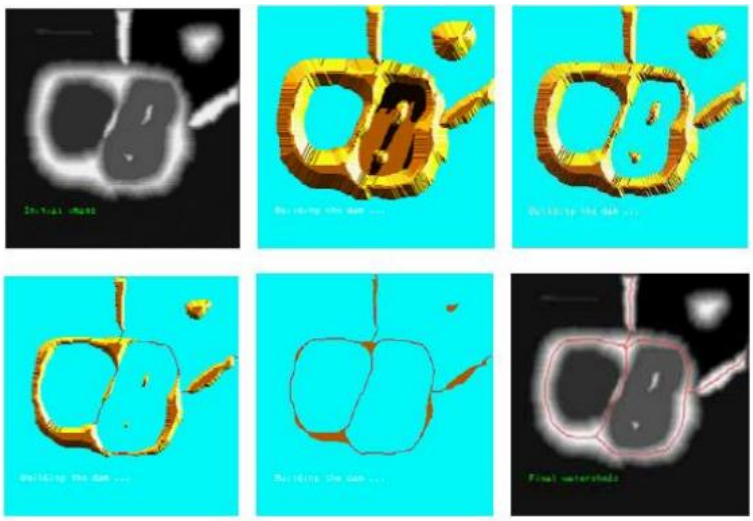


Figure I.13 Flooding process in watershed transforms

I.4 Advanced Image Segmentation Approaches:

I.4.1 Clustering-Based Segmentation:

Thresholding apply well on a grayscale image, for which it is easy to define a threshold from the modes of the histogram. However, this approach cannot be applied on a color or multiband image because there is no histogram. Each pixel in a B-band image can be represented by a point in a B-dimensional space. By doing so, pixels with similar colors form groups in the space, as illustrated in Fig. I.14.

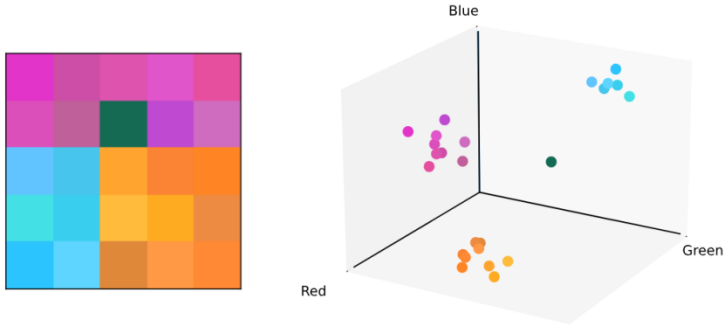


Figure. I.14 Showing the pixels of a color image in a B-dimensional space.

Clustering methods consists in defining groups of pixels. Therefore, all the pixels in the same group define a class in the segmented image. A classical clustering method for image segmentation is the k-means method .

The k-means algorithm [Steinhaus 1957, MacQueen 1967] is an iterative method that affects every point in the space $\mathbb{R}^B$ to a group (called cluster). The number **K** of groups is chosen by the user. In the sequel, the centroid defines the center of a group. Its coordinates are the mean of the coordinates of the points in the group.[13]

Algorithm: K-means:

1. Initialize (randomly) the **K** centroids
2. While the centroids vary:
    1. STEP A For each point:
        1. Calculate the distances from the point to all centroids
        2. Assign the point to the nearest group
    2. STEP B Calculate the centroid of each group

Fig. I.15 illustrate this algorithm, in the simple case of an image with two bands (hence the two-dimensional space) segmented into K=2 classes.

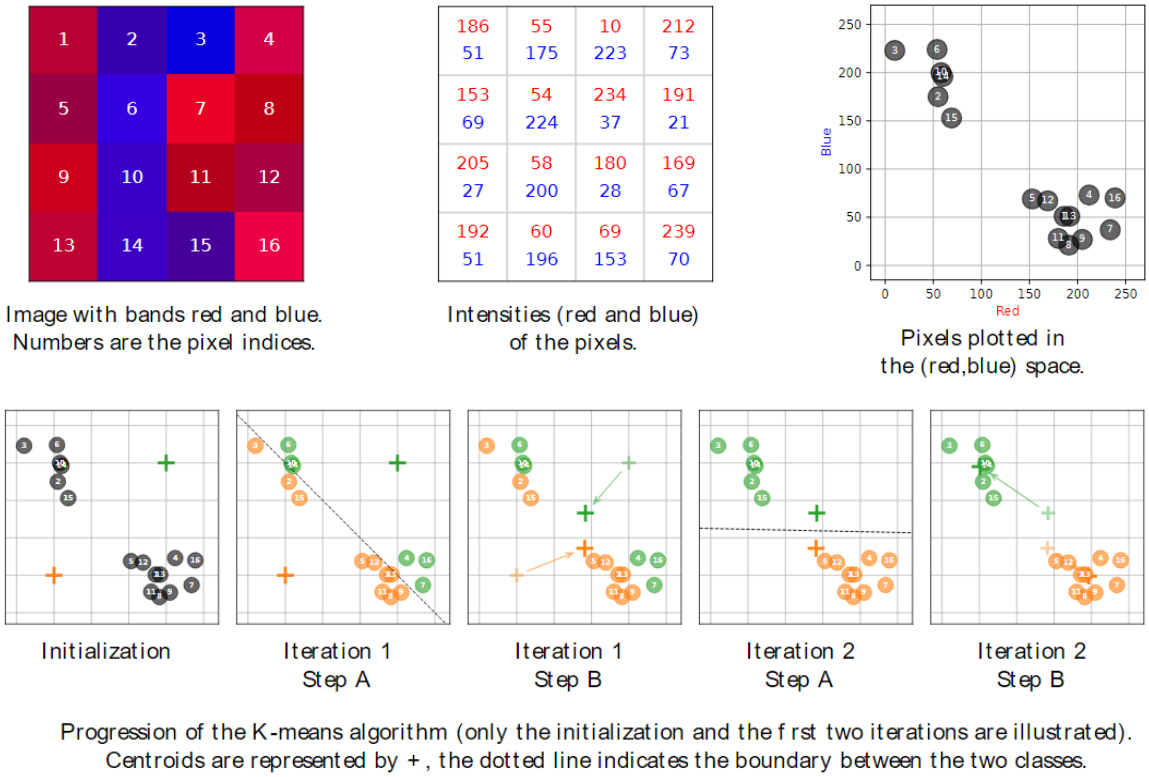


Figure I.15 Illustration of the k-means algorithm

Figure.I.16 gives the result of the k-means algorithm on an image**.**

Figure I.16 Segmentation with the k-means algorithm on the left image (center: K=2 classes, right: K=4 classes).

I.4.2 Graph-Based Segmentation:

GBS involves the application of a graph theory to construct a representation of an image in the form of a graph. In this approach, each image pixel is represented as a node, while the edges connecting the nodes represent the degree of similarity between the corresponding pixels. The main objective of GBS is to divide an image into separate regions, each one representing a segment in the image. Moreover, GBS uses graph partitioning algorithms aiming to reduce the cost of separating segments in the image by minimizing the total weight of the edges that need to be cut.[14]

I.4.2.1 Graph-Based Segmentation process:

GBS algorithms follow a set of standard steps for segmenting images. Hence, this section outlines the steps involved in it. In the first step, we construct a graph to represent the image, where each pixel is a node. So, we must define the weights of the edges between the nodes. It bases on the similarity or dissimilarity between the pixels, like the difference in color or intensity values. Then, we partition the graph into disjoint regions employing a graph partitioning algorithm. This algorithm's goal is to minimize the weights of the edges between the segments. Finally, we refine the segments by merging or splitting them based on several criteria, such as size, shape, or texture. The figure next shows the steps of the GBS and how an image is transformed into a segmented image:

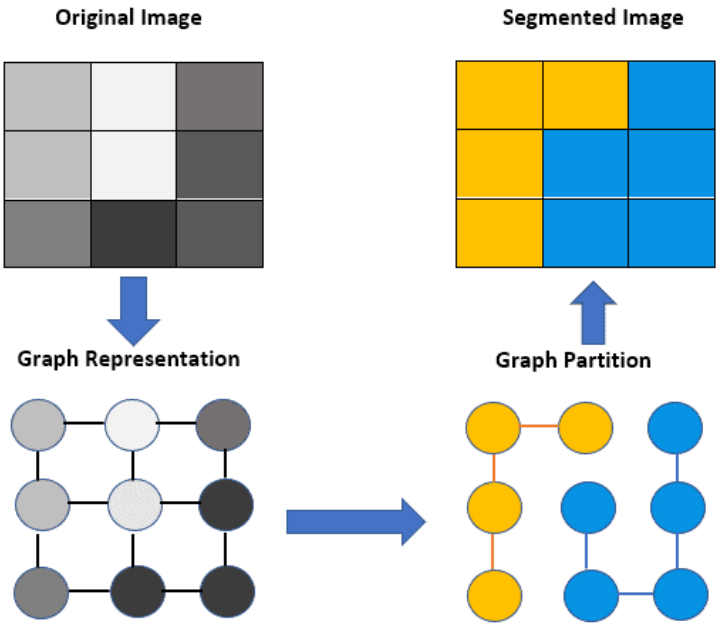


Figure I.17 Graph-Based Segmentation

I.4.2.2 Graph Partitioning Algorithms:

This section will introduce three different graph partitioning algorithms: spectral clustering, normalized cut, and minimum cut. It's relevant to highlight that graph partitioning has a critical role in GBS. It ensures that the resulting segments are meaningful and accurate.

A. Spectral Clustering

Spectral clustering is a popular algorithm for GBS. It's based on the eigenvectors of the Laplacian matrix, which captures the connectivity of the graph. Spectral clustering works by first computing the Laplacian matrix of the graph. Thus, it finds the eigenvectors corresponding to the smallest eigenvalues. Finally, it applies k-means clustering to the eigenvectors to obtain the segments.

B. Normalized Cut

The normalized cut is another algorithm used for graph partitioning. It bases on the notion of minimizing the ratio of the cut cost to the sum of the weights of the edges within each segment. The normalized cut algorithm recursively divides the graph into smaller sub-graphs. It occurs until the sub-graphs meet predefined stopping criteria.

C. Minimum Cut

Minimum cut is a classic algorithm for performing the graph partitioning task. It works by finding the minimum cut in a graph, which means the cut with the lowest weight. The minimum-cut algorithm partitions the graph into two segments by removing the edges with the smallest weights until it's disconnected.

I.5 Conclusion:

In conclusion, the segmentation of medical images represents a critical step in medical image analysis, aiding in the extraction of meaningful information for diagnosis, treatment planning, and research purposes. Throughout this chapter, we have explored various segmentation techniques.

Traditional segmentation methods, such as thresholding, region-based segmentation, and edge detection, have laid the foundation for medical image segmentation and are still utilized in certain scenarios due to their simplicity and computational efficiency. However, these methods often struggle with complex image structures, noise, and variability in medical images, limiting their applicability in clinical settings.

The segmentation of medical images is a multifaceted task that bridges the gap between

image acquisition and clinical decision-making. By leveraging a combination of traditional techniques and cutting-edge deep learning methods, researchers and clinicians can unlock new insights from medical images, ultimately enhancing patient care and advancing medical science.

Chapter II
Convolutional neural network

II .1 Introduction:

Convolutional Neural Networks (CNNs) have emerged as a groundbreaking technology in the realms of machine learning and computer vision, revolutionizing the way computers perceive and interpret visual information. CNNs are a specialized class of neural networks designed to process and analyze structured grid-like data, such as images.

Neural networks, the foundation of CNNs, are computational models inspired by the human brain's interconnected neurons. These networks consist of layers of interconnected nodes, or neurons, each contributing to the overall task of learning patterns and relationships within the input data. Neural networks have found applications in diverse fields, ranging from natural language processing and speech recognition to image and video analysis.

The key innovation of CNNs lies in their use of convolutional layers, which involve the application of convolutional operations to input data. These operations enable the networks to capture local patterns and relationships in the visual data, preserving spatial hierarchies and reducing the number of parameters compared to fully connected networks. This design is particularly effective for tasks where the spatial arrangement of features is crucial, such as image recognition.

II .2 Artificial Neural Networks (ANN):[15]

II .2.1 Human neural network:

Artificial neural networks are inspired by the human neural network. When a neuron fires, usually when triggered by a stimulus, signals are sent down the axon of that neuron to another neuron’s dendrites through a synapse. After that, the new neuron might fire and cause another neuron to fire, repeating the process in the whole system. (Figure II .1)

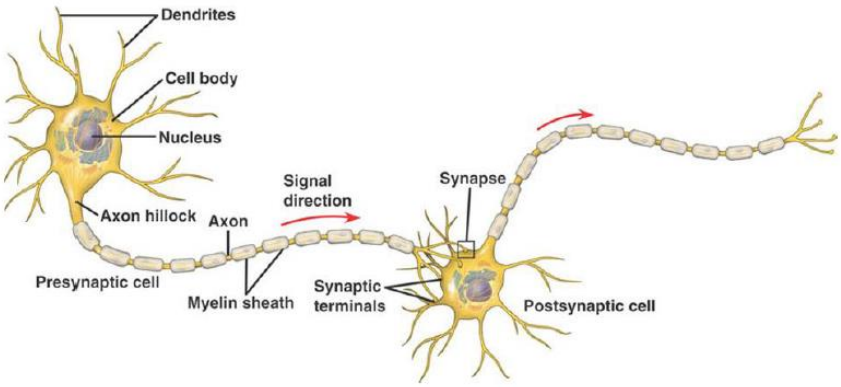


Figure II .1 A diagram of the neuron highlighting the chain structure between the axon and dendrite.

II .2.2 Artificial neural networks:

An artificial neural network (ANN) is a set of layers of neurons . There is an input layer, where the network takes all the information needed. Between the input layer and the output layer are hidden layers. Each hidden layer is used to detect a different set of features**.** The output layer is where the network makes predictions**.** After enough learning, a network can make classifications automatically without human help**.**

II .2.3 Weights, biases:

Weights and structure of a neuron:

The connections between the units in a neural network are weighted, meaning that the weight indicates how much influence the input from a previous unit has on the output of the next unit (Figure II .2).

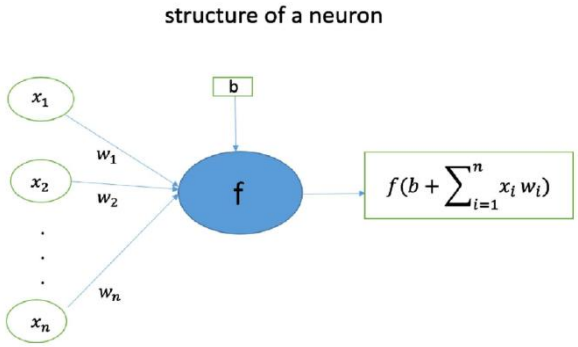


Figure II .2 A diagram to show the work of a neuron: input x, weights w, bias b, activation function f.

Biases:

A bias (*b*) is an extra input to a neuron and it is the number 1 multiplied by a weight**.**

II .2.4 Activation functions:[16]

Sigmoid Function

A saturated activation layer, which is a different form of a logistic function where the input is a real number and the output is a number in the range of [0,1], can be defined by

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

$$f(x) \in (0,1)$$

Tanh Activation Function

The hyperbolic tangent function is a saturated activation layer commonly used when a negative gradient is essential. It outputs a number in the range of [-1, +1]. The following formula defines

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad \text{(II .2)}$$

$$f(x) \in (-1,1)$$

ReLU Activation Function

The rectified linear activation layer (Nair & Hinton, 2010) is considered one of CNN's most important activation layers. It is a non-saturated activation function that is mainly used to remove any negative values. It is advantageous in CNN because it eliminates any negative gradients when the threshold is at zero.

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

LeakyReLU Activation Function

A leaky rectified linear activation layer (Maas et al., 2013) is a non-saturated activation function that allows some negative gradients to pass. It is used to reduce the effect of the negative gradients by factor α. $f(x) = \begin{cases} x, x > 0 \\ \alpha x, x < 0 \end{cases}$ (II .4)

Softmax Activation Function

Softmax is an activation layer usually at the end of a network, and it produces a discrete probability distribution vector.

$$P(y = j|X) = \frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T w_k}} \qquad \text{(II .5)}$$

where X is the input vector and wi is the predicted probability of $y=j$.

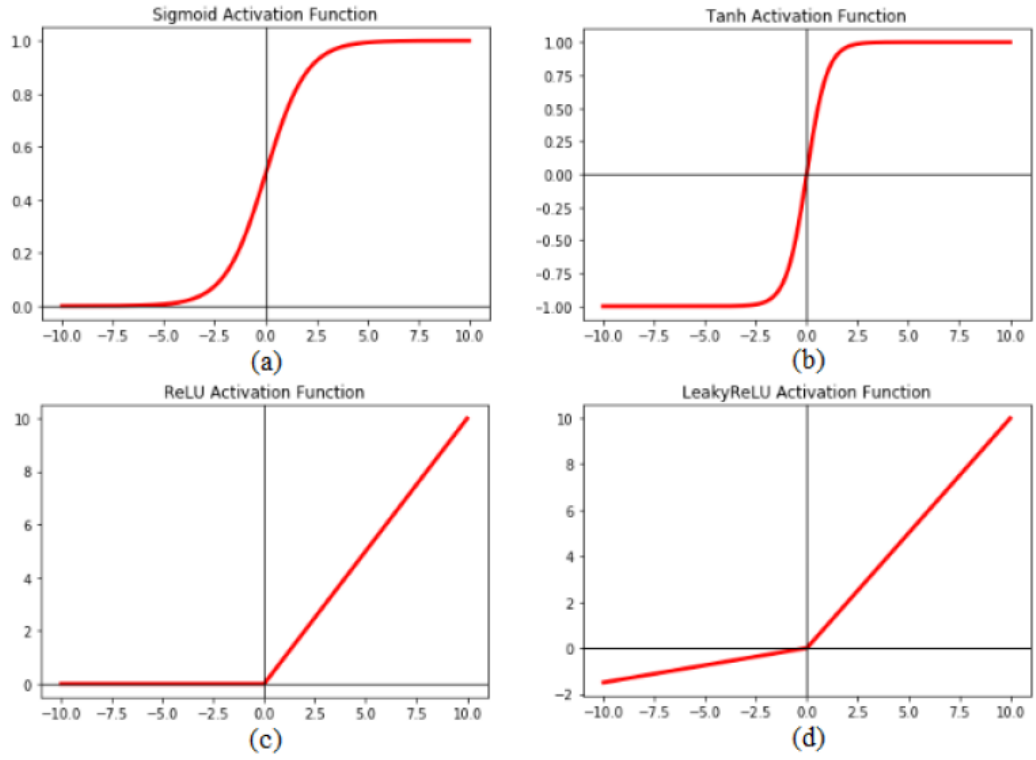


Figure II .3. Plot of different activation functions: (a) Sigmoid activation function; (b) Tanh activation function; (c) ReLU activation function; and (d) LeakyReLU activation function. The x-axis represents the input x and the y-axis represents the output function.

II .3 Deep Convolutional Neural Networks (DCNNs):

The strength of DCNNs is in their layering. A DCNN uses a three-dimensional neural network to process the red, green, and blue elements of the image at the same time. This considerably reduces the number of artificial neurons required to process an image, compared to traditional feed forward neural networks.

Deep convolutional neural networks receive images as an input and use them to train a classifier. The network employs a special mathematical operation called a “convolution” instead of matrix multiplication.

The architecture of a convolutional network typically consists of four types of layers: convolution, pooling, activation, and fully connected [17].

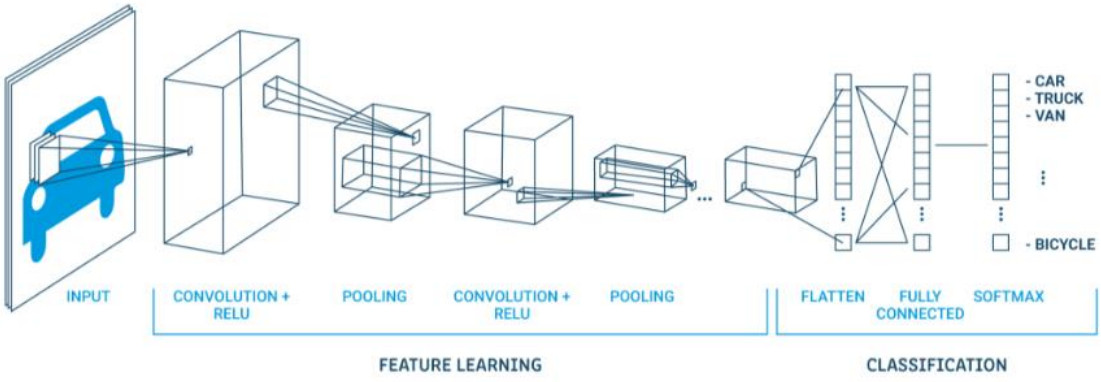


Figure II .4 A typical CNN

II .3.1 Convolutional Layers:

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let’s assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.[18]

For the convolution layers, a *convolution operation* is defined, in which a filter is used to map the activations from one layer to the next. A convolution operation uses a 3-dimensional filter of weights with the same depth as the current layer but with a smaller spatial extent. The dot product between all the weights in the filter and any choice of spatial

region (of the same size as the filter) in a layer defines the value of the hidden state in the next layer (after applying an activation function like ReLU). The operation between the filter and the spatial regions in a layer is performed at every possible position in order to define the next layer (in which the activations retain their spatial relationships from the previous layer).[19]

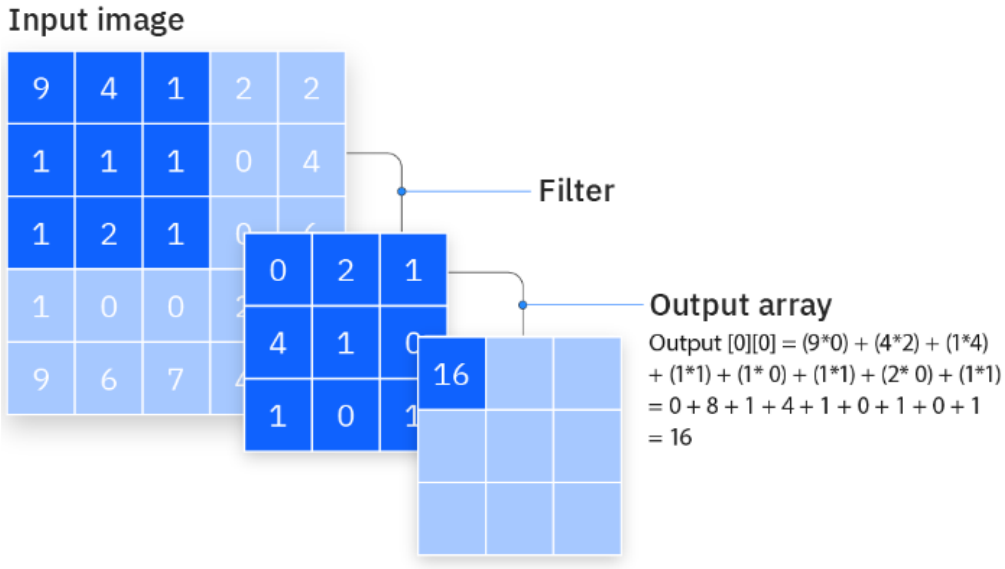


Figure II .5 Convolution operation

II .3.2 Activation Layers:[20]

Just as the final step in evaluating a neuron in a general ANN is to apply an activation function, convolution layers are typically followed by an activation layer. In this work, we will be using “rectified linear unit” (Aggarwal, 2018) activation, “ReLU” for short.

II .3.3 Pooling Layers:

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model. The pooling layer operates over each activation map in the input, and scales its dimensionality using the “MAX” function. In most CNNs, these come in the form of max-pooling layers with kernels of a dimensionality of 2 × 2 applied with a stride of 2 along the spatial dimensions of the input. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size. Due to the destructive nature of the pooling layer, there are only two generally observed methods of max-pooling. Usually, the stride and filters of the pooling layers are both set to 2 × 2, which will allow the layer to extend through the entirety of the spatial dimensionality of the input. Furthermore, overlapping pooling may be utilized, where the stride is set to 2 with a kernel size set to 3. Due to the destructive nature of pooling, having a kernel size above 3 will usually greatly decrease the performance of the model.[21]

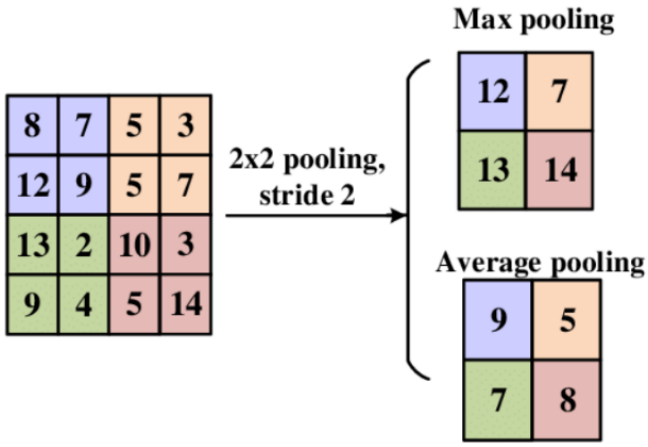


Figure II .6 Pooling operation [22]

II .3.4. Flattening Layers:[16]

The output of the pooling layer is flattened to a $1D$ vector because the subsequent dense layers can only receive $1D$ vectors. A flattening layer can be seen in Figure II .7. The dimensionality of the resulting vector is given by:

$DimFlat=Dimimg*Dimimg*numcolor$

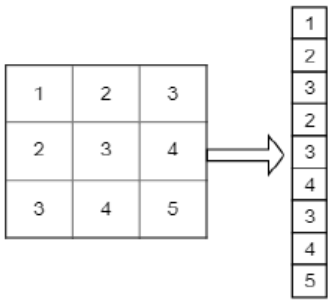


Figure II .7. Flattening 2D feature maps to 1D vector**.**

II .3.5 Fully connected Layer

The fully connected layer appears at the end of the CNN. The output of convolution/pooling is flattened into a single vector of values, with each representing a probability that a certain feature belongs to a label. In this layer, each neuron is connected to every neuron in the superior layer, such that the distributed features extracted and learned in the previous layers can be mapped into the tag space. In other words, all the features previously obtained are consolidated into a unified output, which reduces the burden of subsequent classification tasks. To improve the nonlinear expression, the fully connected layer is usually activated by an activation function. Figure II .7 presents the structure of a relatively simple, fully connected layer.[23]

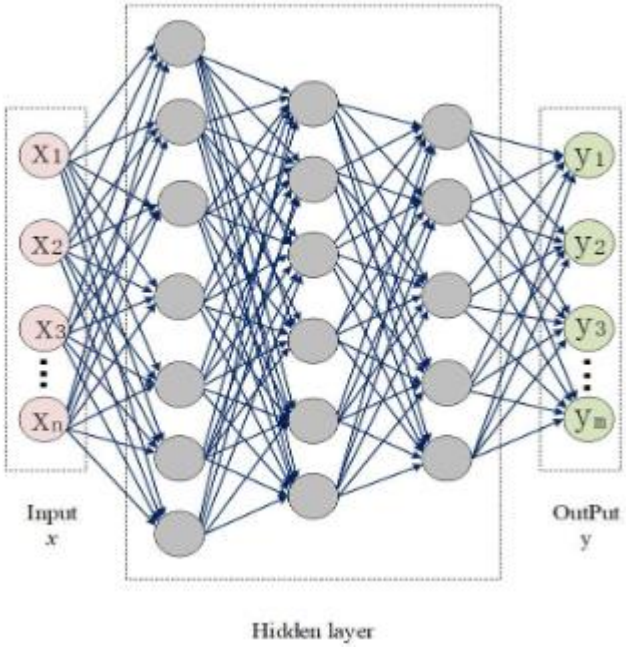


Figure II .8 The structure of the fully connected layer.

II .3.6. Dropout Layer:

A dropout layer is a regularization layer that was first introduced by (Srivastava et al., 2014). It can be applied to any layer in the network. During network training, some neurons are disabled with a predefined dropout-rate probability *P*. It can be thought of as bagging for neural networks. [16]

II .3.7. Regularization Layers:

Complex models with large weights usually have low generalizability since they can learn noise instead of learning the true model patterns (Chollet, 2017a). Under the assumption that models with small weights have better generalizability than those with large weights, regularization functions are commonly used to limit overfitting. Regularization works by adding a penalty term to the loss function to avoid large weights being used by the model (James et al., 2014). The main idea of regularization is to eliminate the weights that do not contribute to the model accuracy by shrinking them to zero. Three types of regularization have been introduced in the literature: L1, L2, and elastic nets. The main differences between these regularizations lie in the penalty terms. [16]

II .3.8. Batch Normalization Layers :

Batch normalization can speed up the network's training and increase its robustness against overfitting (Ioffe & Szegedy, 2015). It reduces the network covariance shift (Santurkar et al., 2018). Additionally, batch normalization adds noise to each layer to increase its robustness. It works by normalizing each layer's inputs by subtracting the batch mean and dividing by the batch standard deviation.

II .4 Training CNNs:

The process of adjusting the value of the weights is defined as the "training" of the neural network.Firstly, the CNN initiates with the random weights. During the training of CNN, the neural network is being fed with a large dataset of images being labelled with their corresponding class labels. The CNN network processes each image with its values being assigned randomly and then make comparisons with the class label of the input image.
If the output does not match the class label which mostly happen initially at the beginning of the training process and therefore makes a respective small adjustment to the weights of its CNN neurons so that output correctly matches the class label image.[24]

II .5 Evaluation Metrics:[16]

Accuracy:

This metric quantifies how accurate the classifier is. It is calculated as the number of correctly classified data points divided by the total number of data points. The formula is shown in Equation

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \quad \text{(II .6)}$$

where TP stands for true positive, TN stands for true negative, FP stands for false positive, and FN stands for false negative. In the context of this study, images without fractures belong to the negative class, whereas images with a bone fracture belong to the positive class.

Kappa:

This is an evaluation metric that is usually used to consider the probability of selecting by chance, especially in cases of unbalanced datasets, and it was introduced by Cohen (1960). The upper limit of the Kappa metric is 1, which means that the classifier classified everything correctly. At the same time, the lower bound can go below zero, which indicates the classifier is just classifying by luck. The Kappa formula is presented in Equation

$$Kappa = \frac{Agreement_{Observed} - Agreement_{Expected}}{1 - Agreement_{Expected}} \quad \text{(II .7)}$$

II .6 Transfer learning:

Transfer learning is a deep learning technique that is used to rapidly and accurately train a CNN in which its weights are not initialized from scratch. Instead, they are imported from another CNN that was trained on a larger dataset. The most popular set of weights used for transfer learning is from the ImageNet dataset (Deng et al., 2009). Several CNN architectures have been trained on the ImageNet dataset and have achieved high accuracy. These weights can be used to classify another completely different dataset instead of randomly initializing the weights from scratch. There are four strategies in transfer learning. The first strategy is to remove the original fully connected layers that act as classifiers, freeze the entire network weights, use the CNN pre-trained layers as feature extraction, and then add a classifier layer such as a fully connected layer or another machine learning classifier, like a support vector machine. The second strategy is to remove the original fully connected layers, fine-tune the entire network weights by using a minimal learning rate (LR), and add a new classifier layer that suits the new task. The third strategy is to remove the fully connected layers, fine-tune only the top layers while keeping the bottom layers frozen, and then add a new classifier layer that suits the new task. Many researchers have suggested that the bottom layers only detect generic features such as edges and circles, while the top layers detect more dataset-specific features.

The fourth strategy is to use a state-of-the-art architecture and start training it from scratch by using only the architecture proven to work on different challenging datasets. [16]

II .7 Popular CNN Model Architectures:[25]

LeNet:

LeNet 5, built by Yann Lecun in 1998. This network takes a 32 x 32 image as input, which goes to the convolution layers (**C1**) and then to the subsampling layer (**S2**). Today, the subsampling layer is replaced by a pooling layer. Then, there is another sequence of convolution layers (**C3**) followed by a pooling (that is, subsampling) layer (**S4**). Finally, there are three fully connected layers, including the **OUTPUT** layer at the end. This network was used for zip code recognition in post offices. Since then, every year various CNN architectures were introduced with the help of this competition:

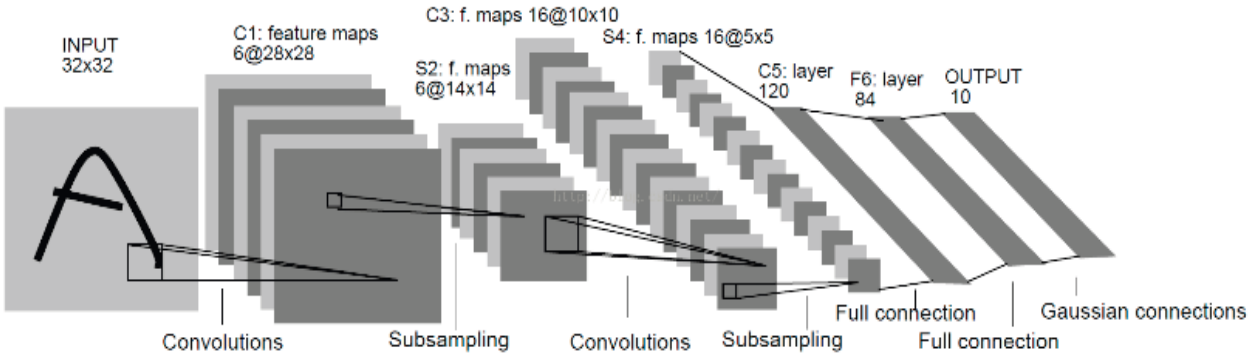


Figure II .9 LeNet 5 -CNN architecture fromYann Lecun's article in 1998

Therefore, we can conclude the following points:

The input to this network is a grayscale 32 x 32 image

The architecture implemented is a CONV layer, followed by POOL and a fully connected layer

CONV filters are 5 x 5, applied at a stride of 1

AlexNet:

The first breakthrough in the architecture of CNN came in the year 2012. This award-winning

CNN architecture is called **AlexNet**. It was developed at the University of Toronto by Alex Krizhevsky and his professor, Jeffry Hinton

In the first run, a ReLU activation function and a dropout of 0.5 were used in this network to fight overfitting. As we can see in the following image, there is a normalization layer used in the architecture, but this is not used in practice anymore as it used heavy data augmentation. AlexNet is still used today even though there are more accurate networks available, because of its relative simple structure and small depth. It is widely used in computer vision:

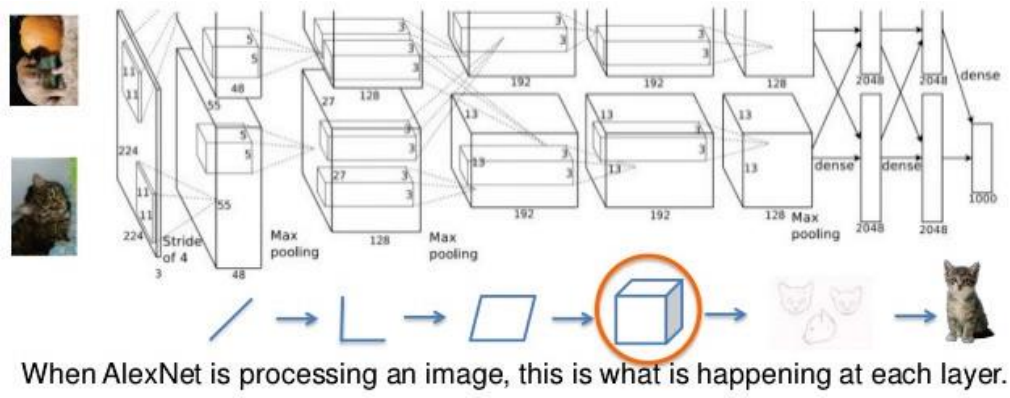


Figure II .10 An illustration of AlexNet architecture

VGGNet:

The runner-up in the 2014 ImageNet challenge was VGGNet from the visual geometric group at Oxford University. This convolutional neural network is a simple and elegant architecture with a 7.3% error rate. It has two versions: VGG16 and VGG19. VGG16 is a 16-layer neural network, not counting the max pooling layer and the softmax layer. Hence, it is known as VGG16. VGG19 consists of 19 layers. A pre-trained model is available in Keras for both Theano and TensorFlow backends

The key design consideration here is depth. Increases in the depth of the network were achieved by adding more convolution layers, and it was done due to the small 3 x 3 convolution filters in all the layers. The default input size of an image for this model is 224 x 224 x 3. The image is passed through a stack of convolution layers with a stride of 1 pixel and padding of 1. It uses 3 x 3 convolution throughout the network. Max pooling is done over a 2 x 2-pixel window with a stride of 2, then another stack of convolution layers followed by three fully connected layers. The first two fully connected layers have 4,096 neurons each, and the third fully connected layers are responsible for classification with 1,000 neurons. The final layer is a softmax layer. VGG16 uses a much smaller 3 x 3 convolution window, compared to AlexNet's much larger 11 x 11 convolution window. All hidden layers are built with the ReLU activation function. The architecture looks like this:

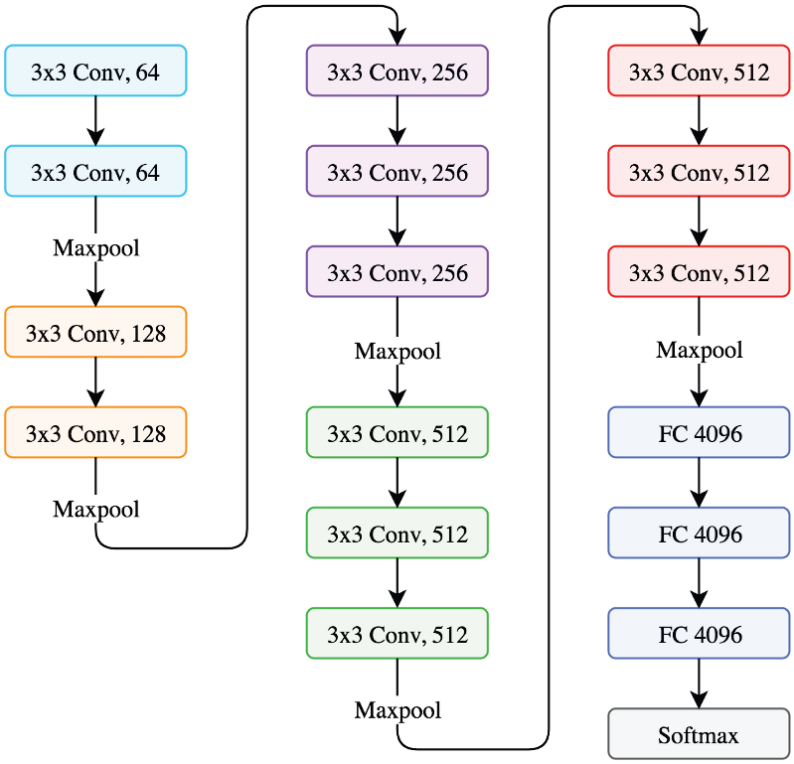


Figure II .11 VGG16 network architecture

GoogleNet:

In 2014, ILSVRC, Google published its own network known as **GoogleNet**. Its performance is a little better than VGGNet; GoogleNet's performance is 6.7% compared to VGGNet's performance of 7.3%. The main attractive feature of GoogleNet is that it runs very fast due to the introduction of a new concept called **inception module**, thus reducing the number of parameters to only 5 million; that's 12 times less than AlexNet. It has lower memory use and lower power use too.

It has 22 layers, so it is a very deep network. Adding more layers increases the number of parameters and it is likely that the network overfits. There will be more computation, because a linear increase in filters results in a quadratic increase in computation. So, the designers use the inception module and GAP. The fully connected layer at the end of the network is replaced with a GAP layer because fully connected layers are generally prone to overfitting. GAP has no parameters to learn or optimize.

Instead of choosing a particular filter size as in the previous architectures, the GoogleNet designers applied all the three filters of sizes 1 x 1, 3 x 3, and 5 x 5 on the same patch, with a 3 x 3 max pooling and concatenation into a single output vector. The use of 1 x 1 convolutions decreases the dimensions wherever the computation is increased by the expensive 3 x 3 and 5 x 5 convolutions. 1 x 1 convolutions with the ReLU activation function are used before the expensive 3x3 and 5 x 5 convolutions.

In GoogleNet, inception modules are stacked one over the other. This stacking allows us to modify each module without affecting the later layers. For example, you can increase or decrease the width of any layer:

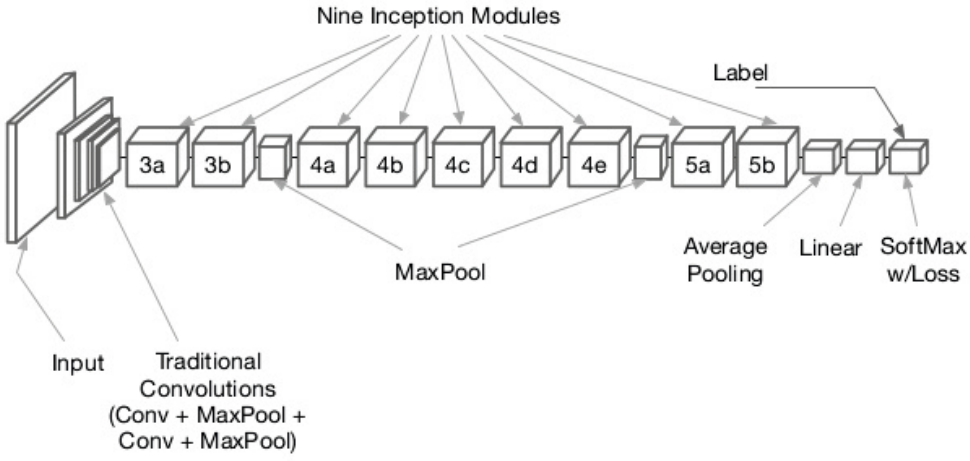


Figure II .12 GoogleNet architecture

ResNet:

After a certain depth, adding additional layers to feed-forward convNets results in a higher training error and higher validation error. When adding layers, performance increases only up to a certain depth, and then it rapidly decreases. In the **ResNet** (**Residual Network**) paper, the authors argued that this underfitting is unlikely due to the vanishing gradient problem, because this happens even when using the batch normalization technique. Therefore, they have added a new concept called **residual block**. The ResNet team added connections that can skip layers:

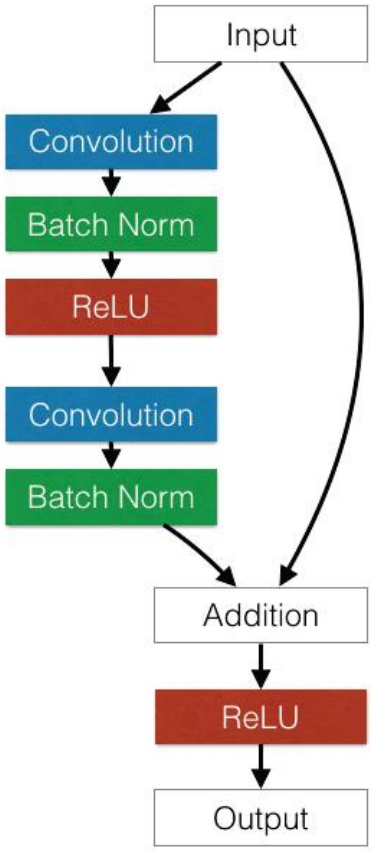


Figure II .13 Residual block

In the 2015 ImageNet ILSVRC competition, the winner was ResNet from Microsoft, with an error rate of 3.57%. ResNet is a kind of VGG in the sense that the same structure is repeated again and again to make the network deeper. Unlike VGGNet, it has different depth variations, such as 34, 50, 101, and 152 layers. It has a whopping 152 layers compared to AlexNet 8, VGGNet's 19 layers, and GoogleNet's 22 layers. The ResNet architecture is a stack of residual blocks. The main idea is to skip layers by adding connections to the

neural network. Every residual block has 3 x 3 convolution layers. After the last conv layer, a GAP layer is added. There is only one fully connected layer to classify 1,000 classes. It has different depth varieties, such as 34, 50, 101, or 152 layers for the ImageNet dataset. For a deeper network, say more than 50 layers, it uses the **bottleneck** features concept to improve efficiency. No dropout is used in this network.

DenseNet:

DenseNet, short for Densely Connected Convolutional Networks, is a neural network architecture introduced by Gao Huang, Zhuang Liu, and Kilian Q. Weinberger in their paper titled "Densely Connected Convolutional Networks" (2016). It's a significant innovation in the field of computer vision and has found applications in tasks like image classification, object detection, and segmentation.

The core idea behind DenseNet is to establish dense connections between layers, allowing each layer to receive direct input from all its preceding layers. This approach is in contrast to traditional feedforward architectures, where each layer only connects to the one before it. Dense connections facilitate feature reuse, promote gradient flow, and significantly reduce the number of parameters, making DenseNet both accurate and computationally efficient.[26]

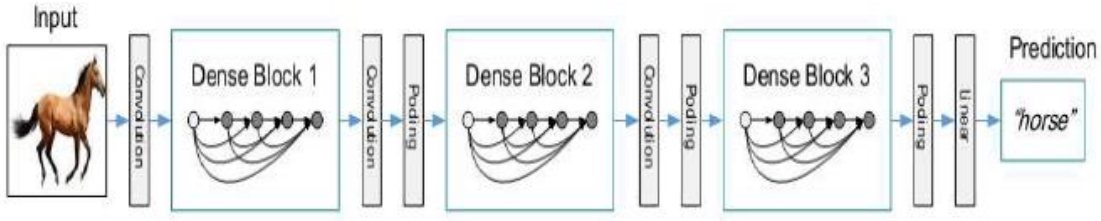


Figure II .14 DenseNet with three dense blocks

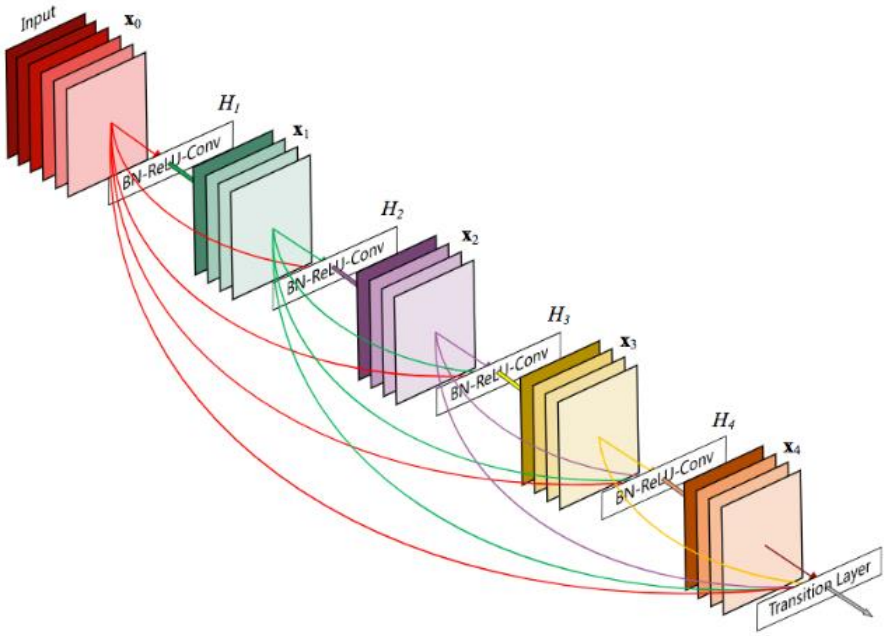


Figure II .15 A dense block

II .8 CNN based segmentation:

II .8.1 Sliding-window approach:

In this approach, the segmentation can be performed by dividing the image into patches (square windows) and passing them into a CNN classifier to obtain a label for the central pixel of each patch. However, this method has some drawbacks. First, it is very slow since it can only predict one pixel label by once forward computation. Secondly, there is a trade-off between computational cost and the use of context. Large patches require more computational time but allow the network to use more contextual information, whereas small patches can only use small context, but the computation cost is low.

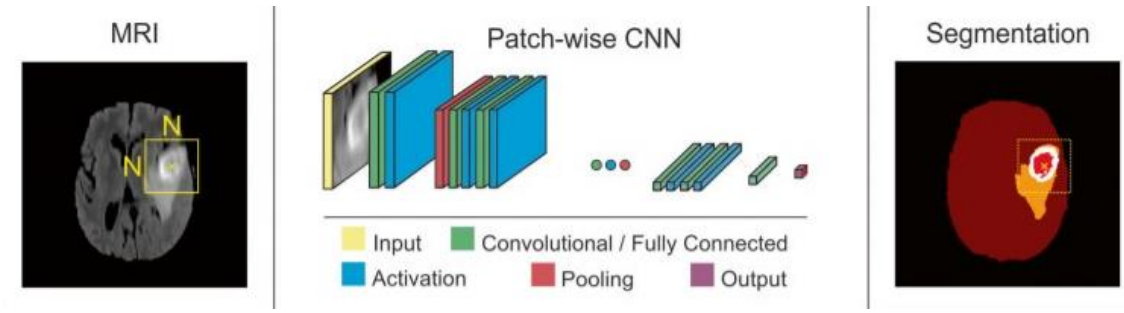


Figure II .16 Illustration of Sliding-window approach [27]

II .8.2 Fully convolutional networks (FCNs):

In 2015, Jonathan Long, Evan Shelhamer and Trevor Darrell [28] introduced the first FCN to image segmentation task. They adapted the classification CNNs pre-trained on ImageNet (AlexNet, VGG-16, and GoogleNet) into FCNs by converting the fully connected layers to convolutional layers. The resulting fully convolutional network (FCN) can take input of any size and produce a probability map for each pixel with a single pass that is much more efficient than output for a single pixel predicted by sliding-window approach. However, because of consecutive pooling layers or striding convolutions, the resolution of the output is far lower than the input. To recover the original input resolution, upsampling layers such as unpooling, max-unpooling, and up convolution (transpose convolution) can be used [29] (see Fig. II .17).

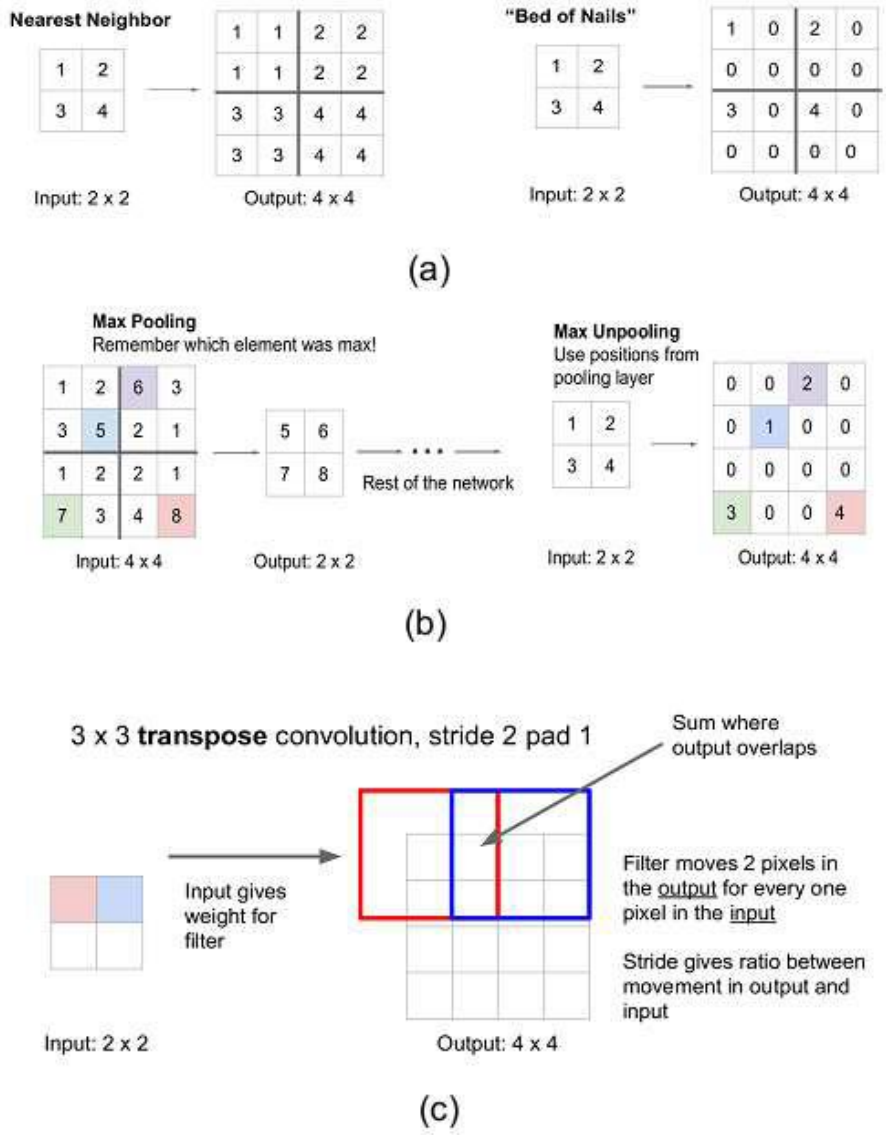


Figure II .17. Upsampling techniques. (a) unpooling, (b) max-unpooling, (c) transpose convolution

In FCN [28], the authors upsampled the last convolutional layer to the size of the input image (FCN32). To enhance the segmentation, the upsampled feature maps were summed with the corresponding feature maps skipped from the encoder in one (FCN16) or two (FCN8) levels (Fig II .19).

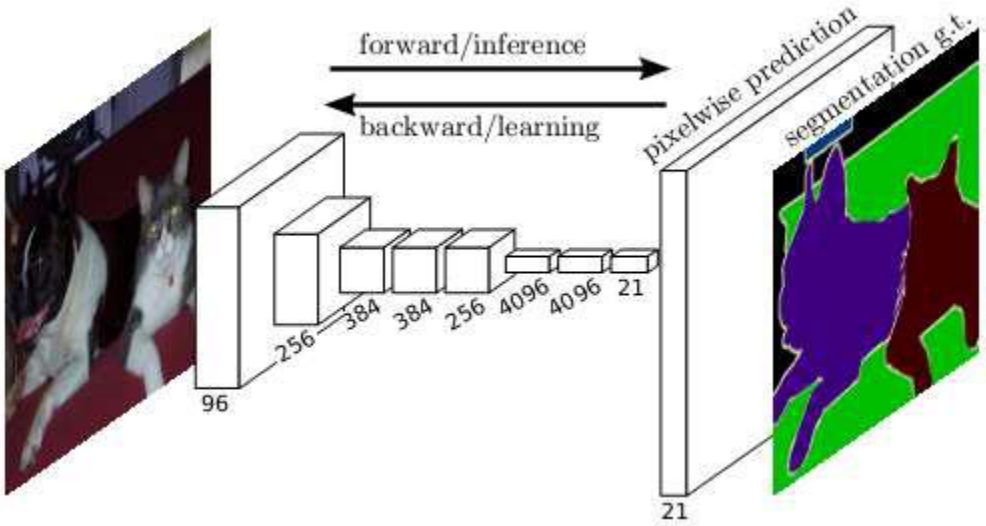


Figure II .18 FCN

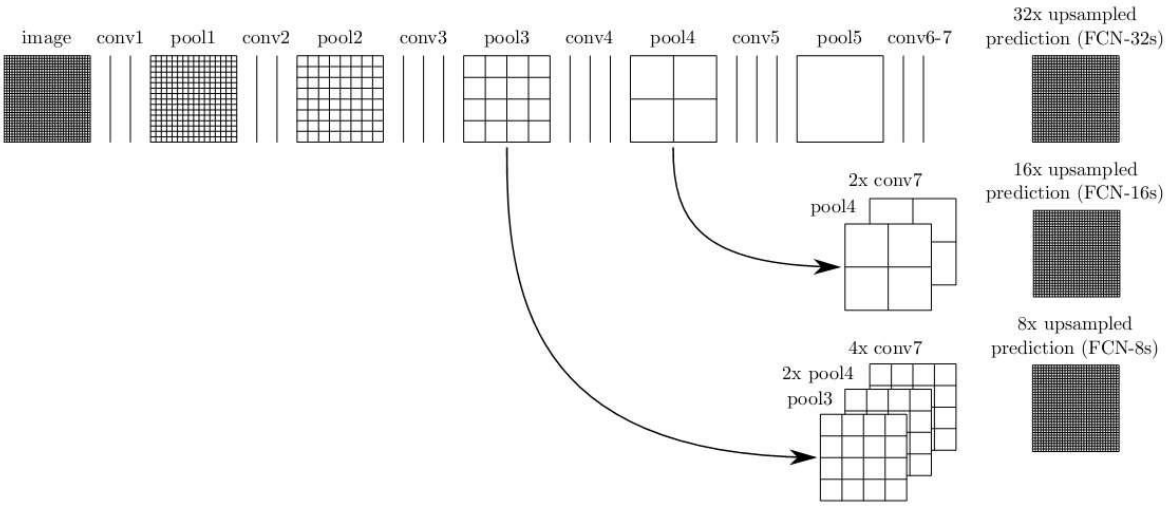


Figure II .19 Skip connection via addition

II .8.3 Encoder-decoder models:

DeconvNet:

The deconvolution network (DeconvNet) is composed of deconvolution and unpooling layers. For the conventional FCN, the output is obtained by high ratio (32×, 16× and 8×) upsampling, which might induce rough segmentation output (label map). In this DeconvNet, the output label map is obtained by gradual deconvolution and unpooling..[30]

The following is the overall architecture of DeconvNet:

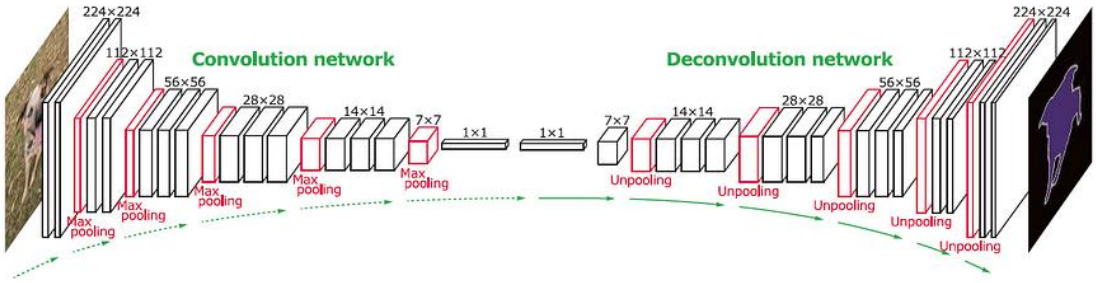


Figure II .20 DeconvNet Architecture [31]

As we can see, it uses VGGNet as backbone. The first part is a convolution network which is as usual like FCN, with conv and pooling layers. The second part is the deconvolution network which is a novel part in this paper.

SegNet:

SegNet is a semantic segmentation model. This core trainable segmentation architecture consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network. The role of the decoder network is to map the low-resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature maps. Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling.[32]

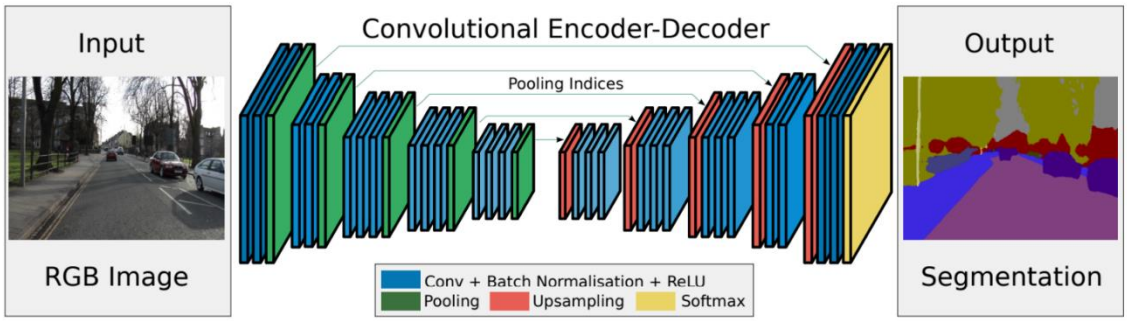


Figure II .21 An illustration of SegNet architecture [33]

II .9 Conclusion:

This chapter focused on the basic concepts of Convolutional Neural Networks, explaining the layers required to build one and detailing how best to structure the network in most image analysis tasks. We presented some popular CNN based models for semantic image segmentation.

Convolutional Neural Networks differ to other forms of Artificial Neural Network in that instead of focusing on the entirety of the problem domain, knowledge about the specific type of input is exploited. This in turn allows for a much simpler network architecture to be set up.

Chapter III Experimental Results

III.1 Introduction:

In this chapter we apply the architecture of three convolutional neural networks (CNNs) models for ultrasound images of breast cancer. The first is the U-net, the second is the Unet++ and the third is the Unet 3+. We're making a comparison of this different methods.

III.2 About Dataset:

Breast cancer is one of the most common causes of death among women worldwide. Early detection helps in reducing the number of early deaths. The data reviews the medical images of breast cancer using ultrasound scan. Breast Ultrasound Dataset is categorized into three classes: normal, benign, and malignant images. Breast ultrasound images can produce great results in classification, detection, and segmentation of breast cancer when combined with machine learning.The data collected at baseline include breast ultrasound images among women in ages between 25 and 75 years old. This data was collected in 2018. The number of patients is 600 female patients. The dataset consists of 780 images with an average image size of 500*500 pixels. The images are in PNG format. The ground truth images are presented with original images. The images are categorized into three classes, which are normal, benign, and malignant

The models are trained on a publicly available dataset sourced from Kaggle.[34]

Data of normal class:

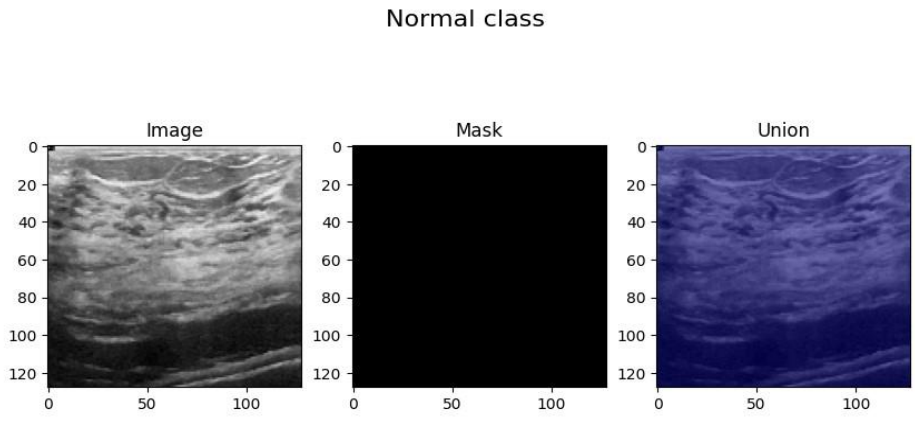


Figure III.1 Data of normal class

Data of benign class:

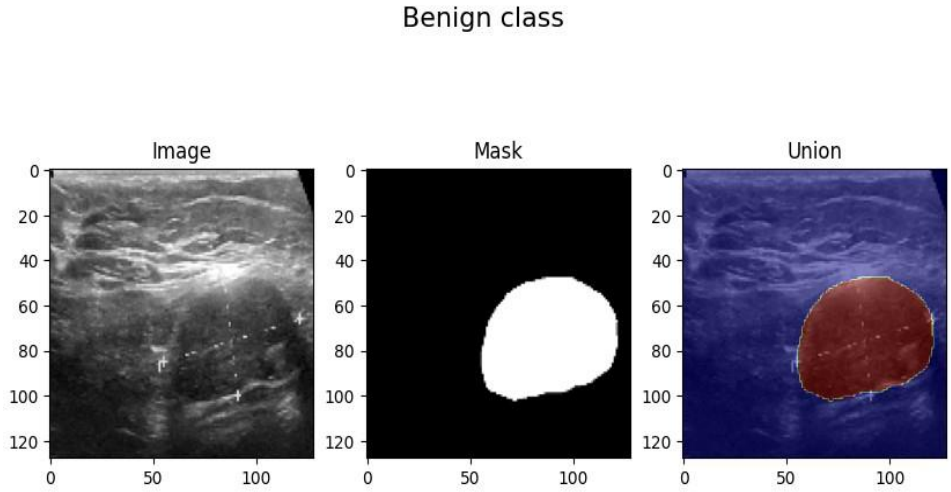


Figure III.2 Data of benign class

Data of Malignant class:

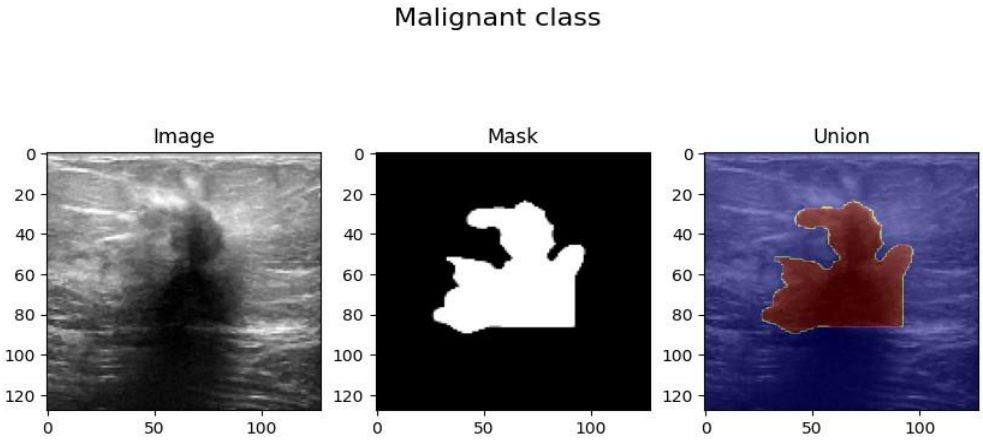


Figure III.3 Data of Malignant class

Average view of masks each class:

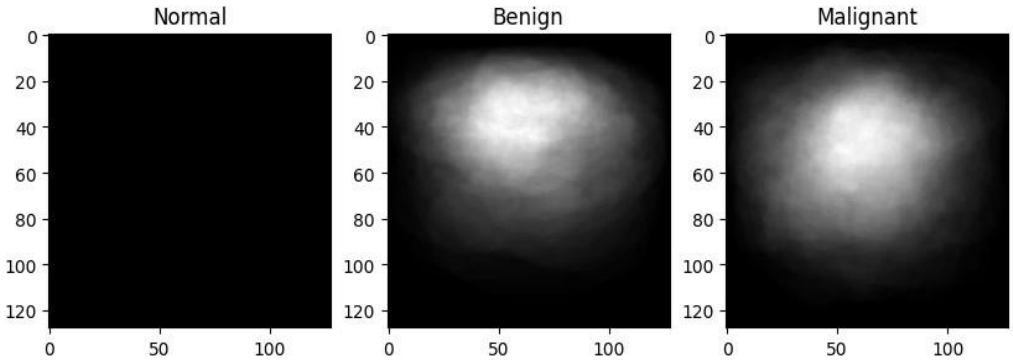


Figure III.4 Average view of masks each class

III.3 Model evaluation metrics:

Accuracy: Measures the overall correctness of the prediction: the percentage of correctly classified samples (true positive and true negative) from the total data:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{(III.1)}$$

Precision: metric assesses the accuracy of positive forecasts by measuring the proportion of TP among all forecasts as positive:

$$\text{Precission} = 100 \times \frac{TP}{TP + FP} \quad \text{(III.2)}$$

Recall: this metric measures the percentage of correctly classified samples, including both True Positives (*TP*) and True Negatives (*TN*):

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{(III.3)}$$

Dice index: that metric measures the similarity in image segmentation. It evaluates the consistency between two sets A (segmented regions) and B (benchmark regions):

$$\text{Dice}(A, B) = 2\frac{|A| \cap |B|}{|A| + |B|} \quad \text{(III.4)}$$

III.4 Experiments:

The experiments were carried out at Google Colab Pro using Python programming. Network models were implemented with Keras and TensorFlow.

III.4.1 U-Net:

U-Net is an architecture for semantic segmentation. It consists of a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers.[35]

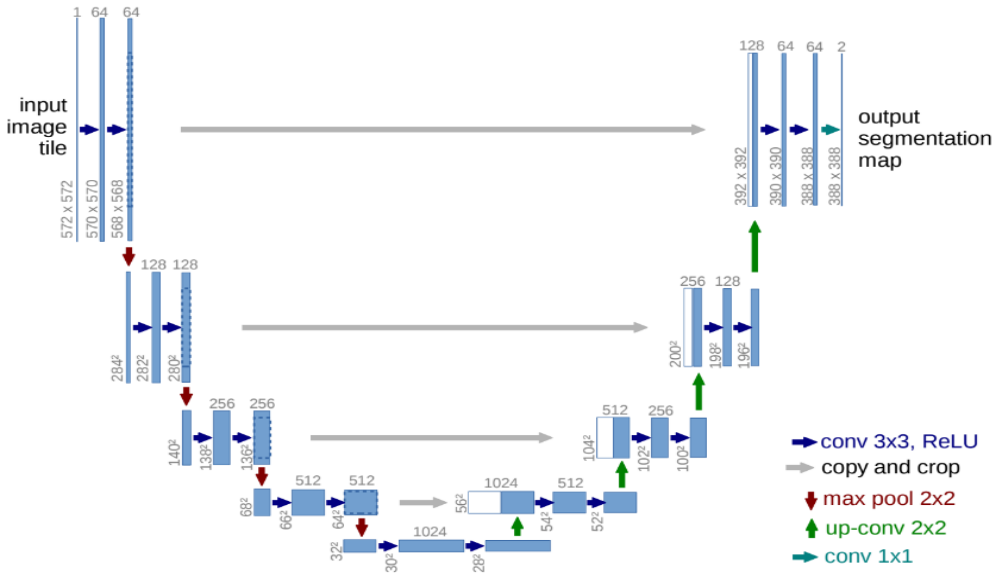


Figure III.5 U-Net architecture [36]

Results:

After completing training and testing on a dataset of original ultrasound images with U-Net model, we obtain these results:

The Figure III.7 shows how the loss and accuracy values change during training and validation at different epochs for U-Net model.

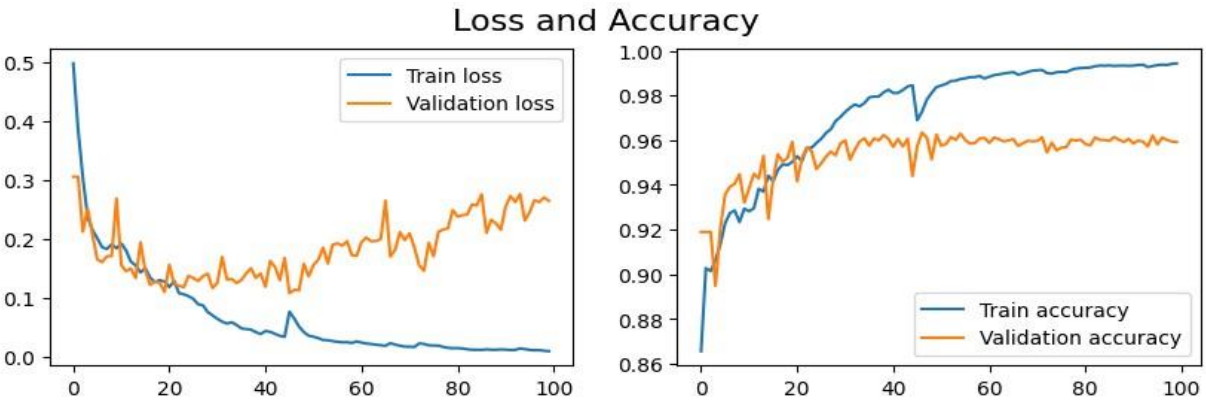


Figure III.6 The training and validation loss as well as the training and validation accuracy over epochs (U-Net)

The Figure III.7 shows the images, their corresponding masks, and the model's predictions for a randomly selected subset of test data

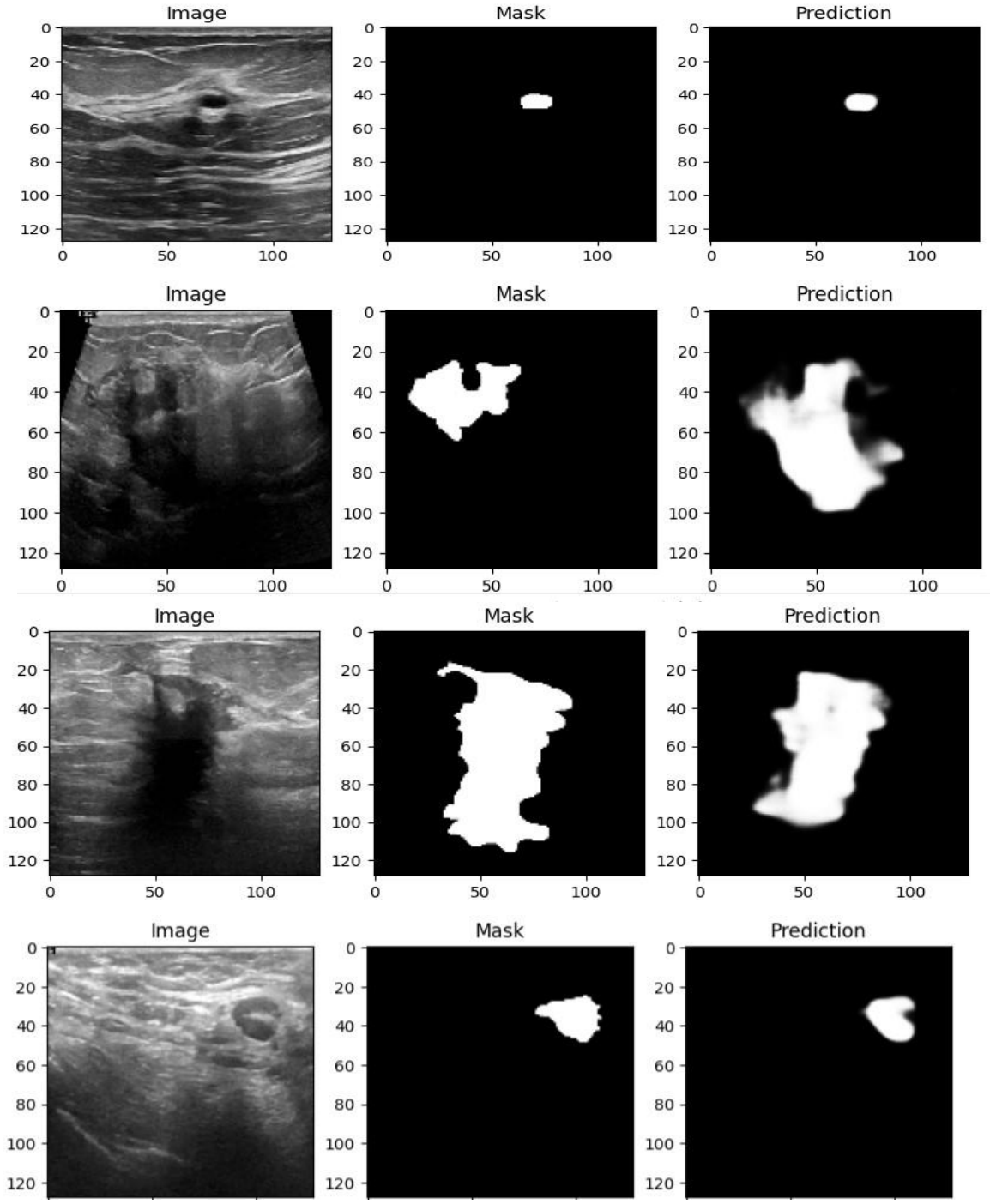


Figure III.7 Cancer prediction results of U-Net model on the original dataset

	Accuracy	Precision	Recall	Dice coef
U-Net	0.9330	0.9793	0.9578	0.9406

Table III.1 The average values of evaluation metrics for U-Net

III.4.2 U-Net++:

UNet++ is an architecture for semantic segmentation based on the U-Net. Through the use of densely connected nested decoder sub-networks, it enhances extracted feature processing and was reported by its authors to outperform the U-Net in Electron Microscopy (EM), Cell, Nuclei, Brain Tumor, Liver and Lung Nodule medical image segmentation tasks.[37]

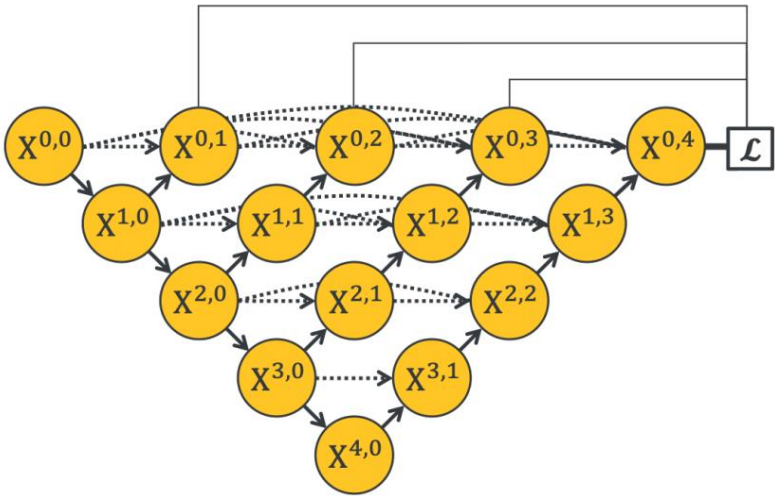


Figure III.8 UNet++ architecture

Results:

After completing training and testing on a dataset of original ultrasound images with U-Net++ model, we obtain these results:

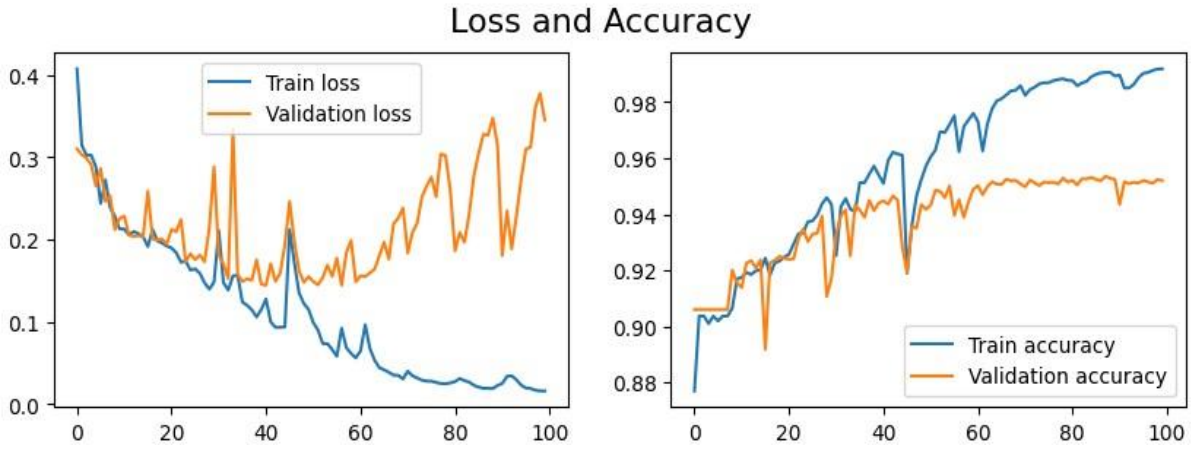


Figure III.9 The training and validation loss as well as the training and validation accuracy over epochs (U-Net++)

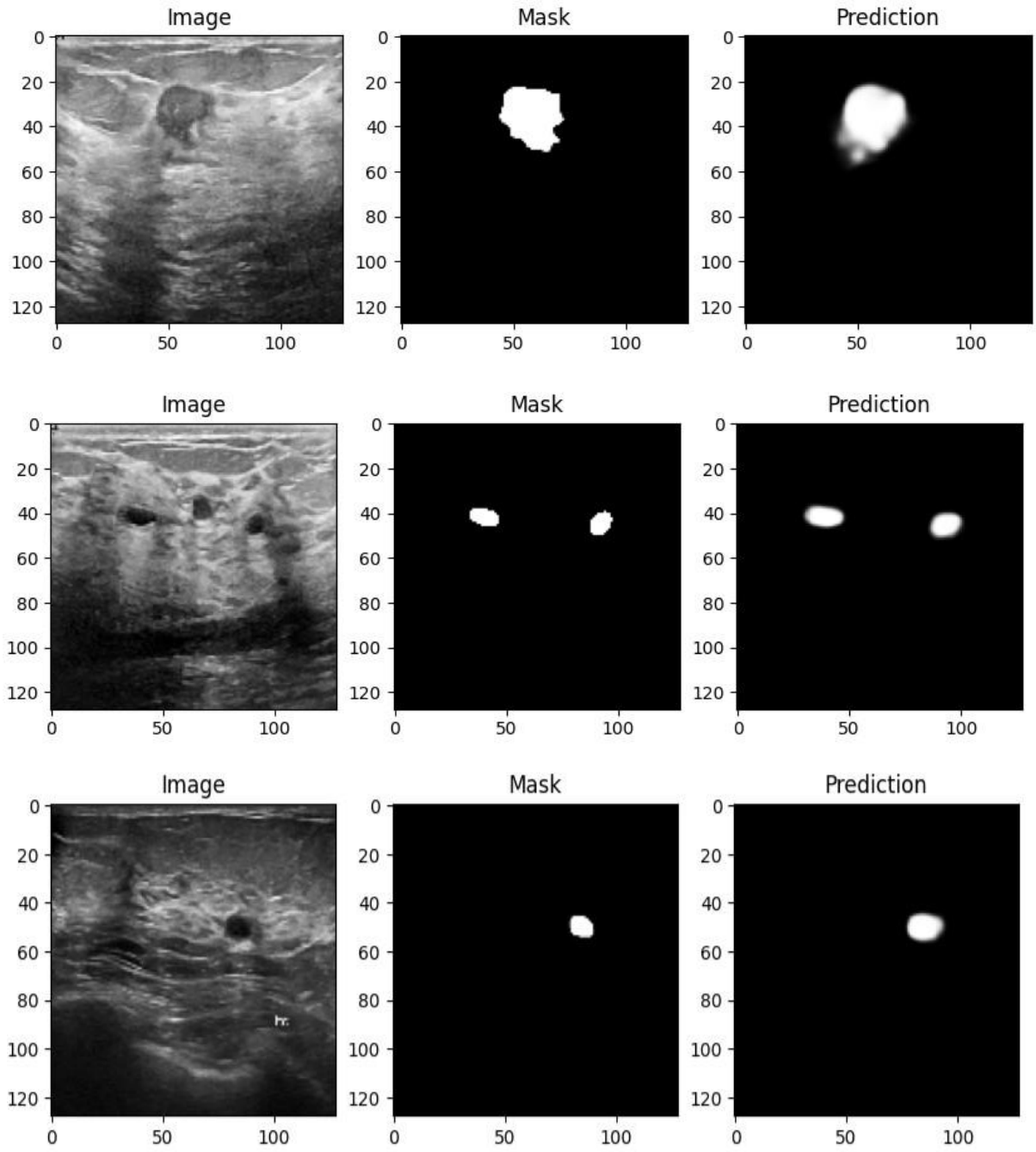


Figure III.10 The images, their corresponding masks, and the model's predictions for a randomly selected subset of test data

	Accuracy	Precision	Recall	Dice coef
U-Net++	0.9845	0.9417	0.8903	0.8849

Table III.2 The average values of evaluation metrics for U-Net++

III.4.3 U-Net3+:

In 2020, H. Huang et al. proposed the next generation architecture of U-Net family, UNet3+ (*Fig. III.11)*, which improved on the UNet++ model by,

1) adopting its deep supervision technique and modifying it to take in **full-scale semantic information.**

2) changing the **design of the dense connections** which accounts for both low and high levels of details from feature maps more effectively for segmentation.

Previous studies have shown that feature maps at different scales (aka. different levels) exploit different types of information. For example: feature maps at lower levels capture spatial information better, like the boundaries of organs, while at higher levels positional information like the relative positions of the organs would be exploited more.[38]

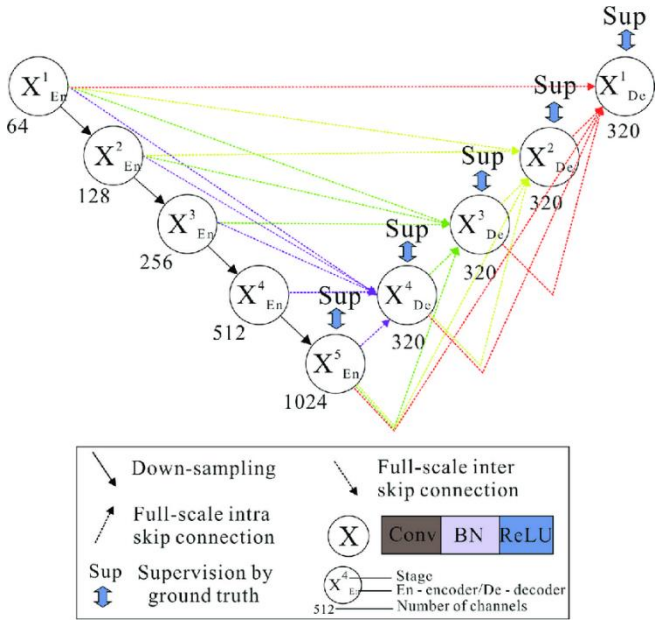


Figure III.11 UNet 3+ architecture [39]

Results:

After completing training and testing on a dataset of original ultrasound images with UNet**3+** model, we obtain these results:

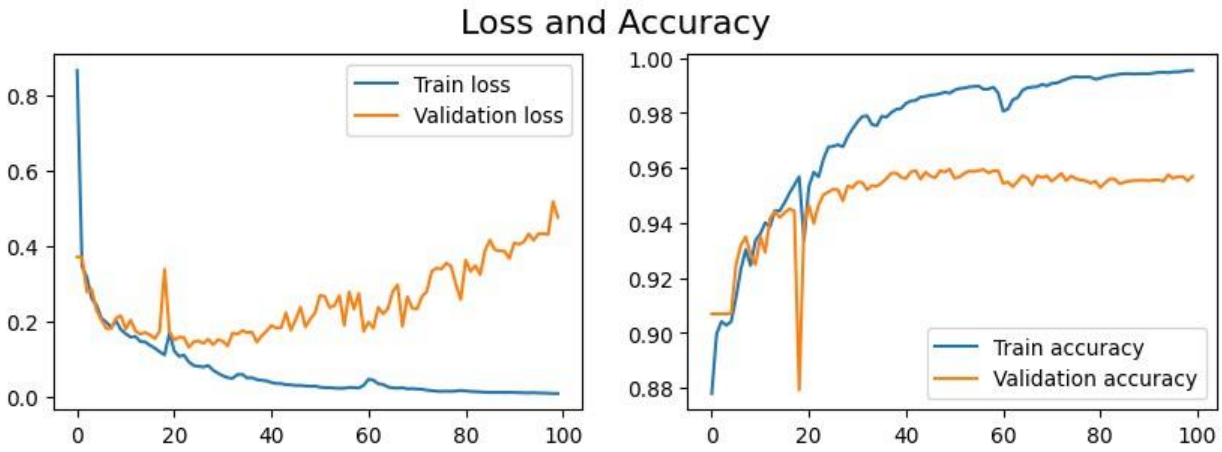
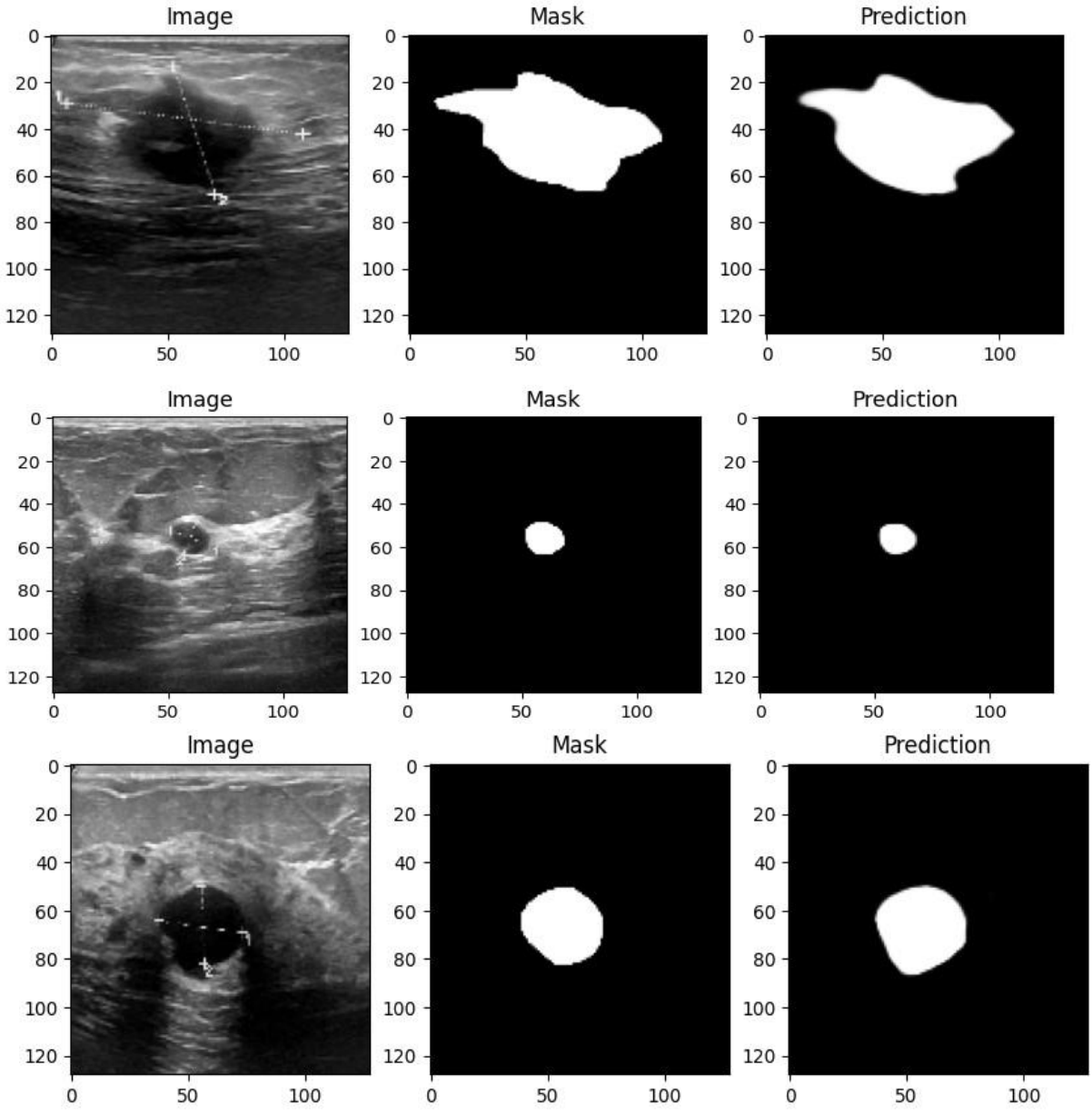


Figure III.12 The training and validation loss as well as the training and validation accuracy over epochs (U-Net**3+**)



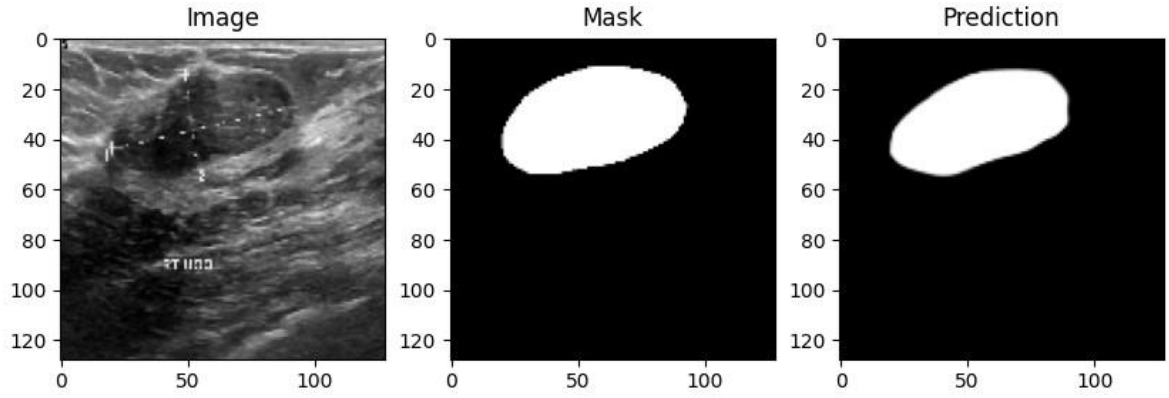


Figure III.13 The images, their corresponding masks, and the model's predictions for a randomly selected subset of test data

	Accuracy	Precision	Recall	Dice coef
U-Net**3+**	0.9841	0.9331	0.8848	0.8855

Table III.3 The average values of evaluation metrics for U-Net3+

III.5 Discussion:

Table III.4 summarizes the results of different methods on this dataset. The models, trained for a total of 300 epochs, showcase noteworthy results. On Accuracy, the average score of UNet is 0.9930, the average score of UNet ++ is 0.9845, and the average score of UNet 3+ is 0.9841. On Precision, the score of UNet is 0.9793, which is higher than UNet++ and UNet 3+. On DC, the UNet 3+ score is 0.8855, which is lower than UNet and UNet++.

	Accuracy	Precision	Recall	Dice coef
U-Net	0.9930	0.9793	0.9578	0.9406
U-Net++	0.9845	0.9417	0.8903	0.8849
U-Net**3+**	0.9841	0.9331	0.8848	0.8855

Table III.4. Comparison of different models for breast cancer segmentation.

Additionally, the training progress of these models is depicted through learning curves. The diagrams in Figures III.6, III.9 and III.12 illustrate the progression of accuracy and loss values throughout the training and validation phases for different models. This visualization provides insight into the models' ability to delineate object boundaries and the accuracy of their segmentation performance.

Figure III.7, III.10 and III.13 shows a random selection of Breast Ultrasound (BUS) images from the test set to simplify understanding of the results by comparing mask prediction from models to the actual mask. As is evident from the figures, U-Net3+ technique offers distinct advantages over previous models since it is more sensitive to regions of varying sizes and attentive to nuances.

III.6 Conclusion:

In this chapter, we performed three experiments on three image segmentation models (Unet, Unet++ and Unet3+). The results obtained show that the Unet3+ model performs better than the other two models. Unet3+ gives best predictions from original images, in fact, prediction masks are very similar to ground truth masks.

General Conclusion

In conclusion, this thesis has explored the critical aspects of medical image segmentation and the efficacy of different convolutional neural network architectures in this domain, ultimately leading to significant findings that contribute to the state-of-the-art in medical image analysis.

Chapter one provided a comprehensive overview of image segmentation techniques, emphasizing their importance in medical image analysis. The chapter highlighted the limitations of conventional approaches in dealing with the complexities of medical images, including variability in anatomical structures, presence of noise, and diverse imaging conditions. This set the stage for the need for more sophisticated methods, paving the way for the introduction of deep learning-based techniques. Chapter two delved into the intricacies of convolutional neural networks (CNNs), highlighting their significance and various architectures commonly used for medical image segmentation tasks. We elucidated the architecture of CNNs, explaining how layers of convolution, pooling, and non-linear activation functions enable the extraction of hierarchical features from images. The chapter also covered various CNN architectures, emphasizing their applicability to medical image segmentation. In Chapter Three, three leading CNN architectures, namely U-net, U-net++, and U-net3+, were implemented and evaluated using Google Colab. Through meticulous evaluation against predefined criteria, it has been established that U-net3+ outperforms U-net++ and U-net in terms of segmentation accuracy and robustness. This conclusion is drawn on the basis of experimentation and comparison of models on a dataset and evaluation measures.

The results of this thesis provide information on the performance of state-of-the-art CNN architectures. In addition, the superiority of U-net3+ over U-net++ and U-net underscores the importance of exploring and developing new architectural improvements to further improve the accuracy and efficiency of medical image segmentation systems.

The models studied offer the medical community a more precise and reliable analysis to segment cancer on medical images. The results can be implemented in clinical practice to help radiologists in the early diagnosis of cancer. By addressing remaining challenges and continuing to innovate, the potential of deep learning to transform medical image analysis and improve patient care can be fully realized.

References:

[1] https://www.geeksforgeeks.org/digital-image-processing-basics/

[2] https://www.baeldung.com/cs/image-histograms

[3] Martijn P.A. Starmans, Wiro J. Niessen, in "*Handbook of Medical Image Computing and Computer Assisted Intervention*", 2020.

[4] Amira A. Mahmoud1 , El-Sayed M. El-Rabaie2 , Taha E. Taha3 , Adel Elfishawy4 , Osama Zahran5 , Fathi E. Abd El-Samie." *Medical Image Segmentation Techniques, a Literature Review, and Some Novel Trends*". Nov. 2017.

[5] https://what-when-how.com/embedded-image-processing-on-the-tms320c6000-dsp/segmentation-image-processing/

[6] Fari Muhammad Abubakar." *Study Of Image Segmentation By Using Edge Detection Techniques* ". Vol. 1 Issue 9, November- 2012.

[7] Muthukrishnan.R1 and M.Radha." *EDGE DETECTION TECHNIQUES FOR IMAGE SEGMENTATION* ". Vol 3, No 6, Dec 2011.

[8]https://www.tutorialspoint.com/region-and-edge-based-segmentation#:~:text=Region%20Based%20Segmentation,of%20segments%20within%20the%20image.

[9] https://www.rcet.org.in/uploads/academics/rohini_48305134656.pdf

[10] https://users.cs.cf.ac.uk/dave/Vision_lecture/node34.html

[11] https://dept-info.labri.fr/~vialard/Image3D/cours/slides-seg.pdf

[12] Amandeep Kaur, Aayushi." *Image Segmentation using Watershed Transform* " . International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4, Issue-1, March 2014

[13] https://vincmazet.github.io/bip/segmentation/clustering.html

[14] https://www.baeldung.com/cs/graph-based-segmentation

[15] Hung Dao."*Image Classification Using Convolutional Neural Networks*". Spring 2020

[16] Ibrahem Hamdy, Abdelhamid Kandel."*Deep Learning Techniques for Medical Image Classification*". A thesis submitted in partial fulfillment of the requirements for the degree of Doctor in Information Management. May2021

[17] https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks#:~:text=Deep%20convolutional%20neural%20networks%20(CNN,the%20visual%20cortex%20of%20animals.

[18] https://www.ibm.com/topics/convolutional-neural-networks

[19] Charu C. Aggarwal. "*Neural Networks and Deep Learning*." Springer Nature 2018

[20] Darnowsky, Philip. 2022. "*Image Classification with Evolved Convolutional Neural Networks*". Master's thesis, Harvard University Division of Continuing Education.

[21] Keiron O'Shea, Ryan Nash. *"An Introduction to Convolutional Neural Networks*" arXiv:1511.08458

[22] Huo Yingge, Imran Ali and Kang-Yoon Lee." *Deep Neural Networks on Chip - A Survey*" February 2020

[23] Jinsong Zhu and Jinbo Song."*An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges*". February 2020.

[24] https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/

[25] Mohit Sewak, Md. Rezaul Karim, Pradeep Pujari " *Practical Convolutional Neural Networks*" Packt Publishing First published: February 2018

[26] https://mindfulmodeler.hashnode.dev/exploring-densenet-a-concise-overview

[27] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson, "*Deep learning for brain MRI segmentation: state of the art and future directions*," Journal of digital imaging, vol. 30, pp. 449-459, 2017.

[28] Jonathan Long, Evan Shelhamer and Trevor Darrell " *Fully Convolutional Networks for Semantic Segmentation* " appear in CVPR (2015) arxiv:1411.4038

[29] Li, F.-F., J. Johnson, and S. Yeung, Lecture 11: *detection and segmentation*, 2017, Stanford University http://cs231 n.stanford.edu/slides/2017/cs231n 2017 lecture11.pdf

[30] https://towardsdatascience.com/review-deconvnet-unpooling-layer-semantic-segmentation-55cf8a6e380e

[31] Hyeonwoo Noh, Seunghoon Hong and Bohyung Han." *Learning Deconvolution Network for Semantic Segmentation* " published in **2015 ICCV**

[32] https://paperswithcode.com/method/segnet

[33] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla." *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* " arXiv:1511. 00561v3.Oct 2016

[34] W. Al-Dhabyani, M. Gomaa, H. Khaled, A. Fahmy: "*Dataset of Breast Ultrasound* Images,Data in Brief", Vol. 28, February 2020, p. 104863

[35] https://paperswithcode.com/method/u-net

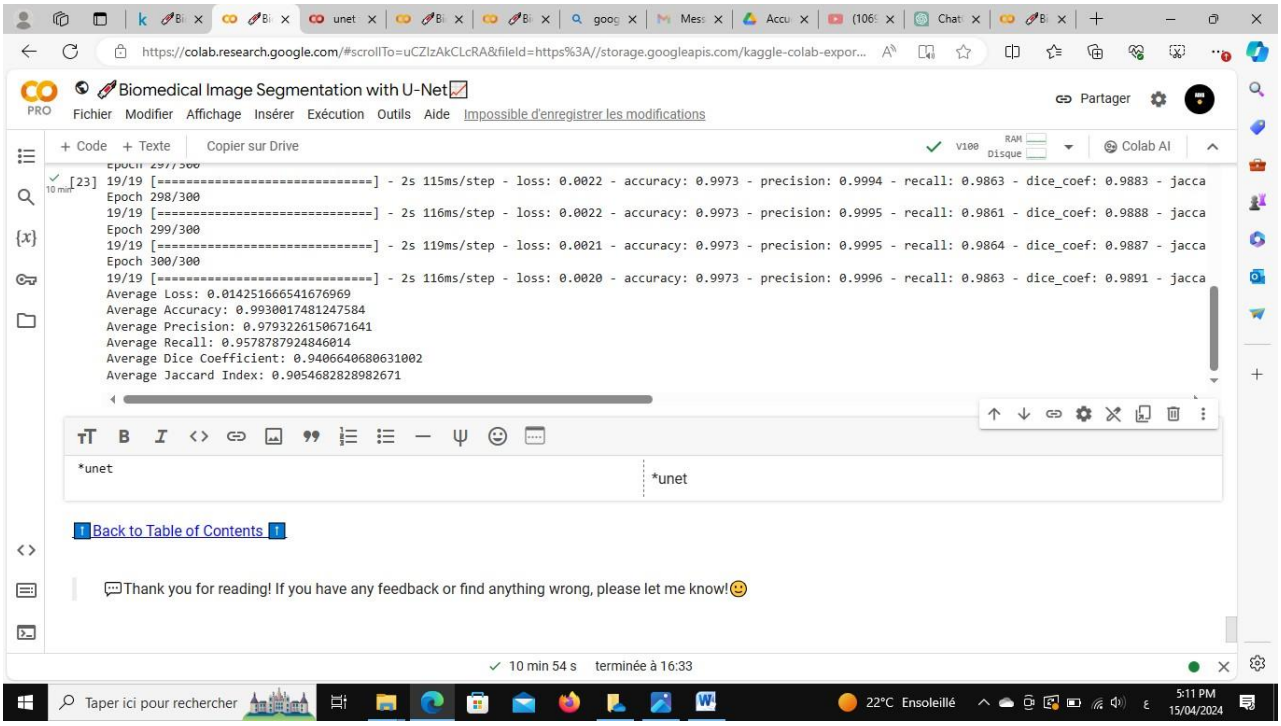
[36] Olaf Ronneberger, Philipp Fischer, Thomas Brox." *U-Net: Convolutional Networks for Biomedical Image Segmentation*" arXiv:1505. 04597v1.May 2015

[37] https://paperswithcode.com/method/unet

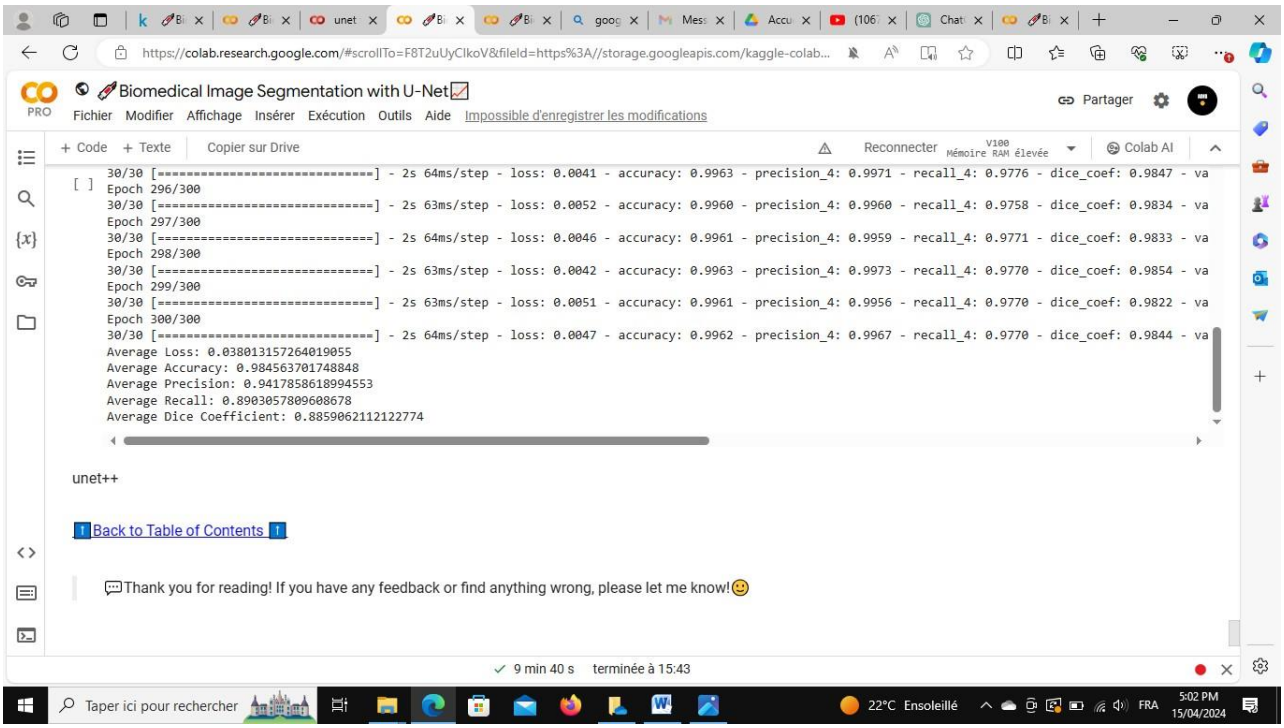
[38] https://medium.com/@mlquest0/unet-3-fully-explained-next-generation-unet-2a8e204e4cf9

[39]https://www.researchgate.net/figure/The-UNet3-network-architecture-for-semantic-segmentation_fig1_358954556

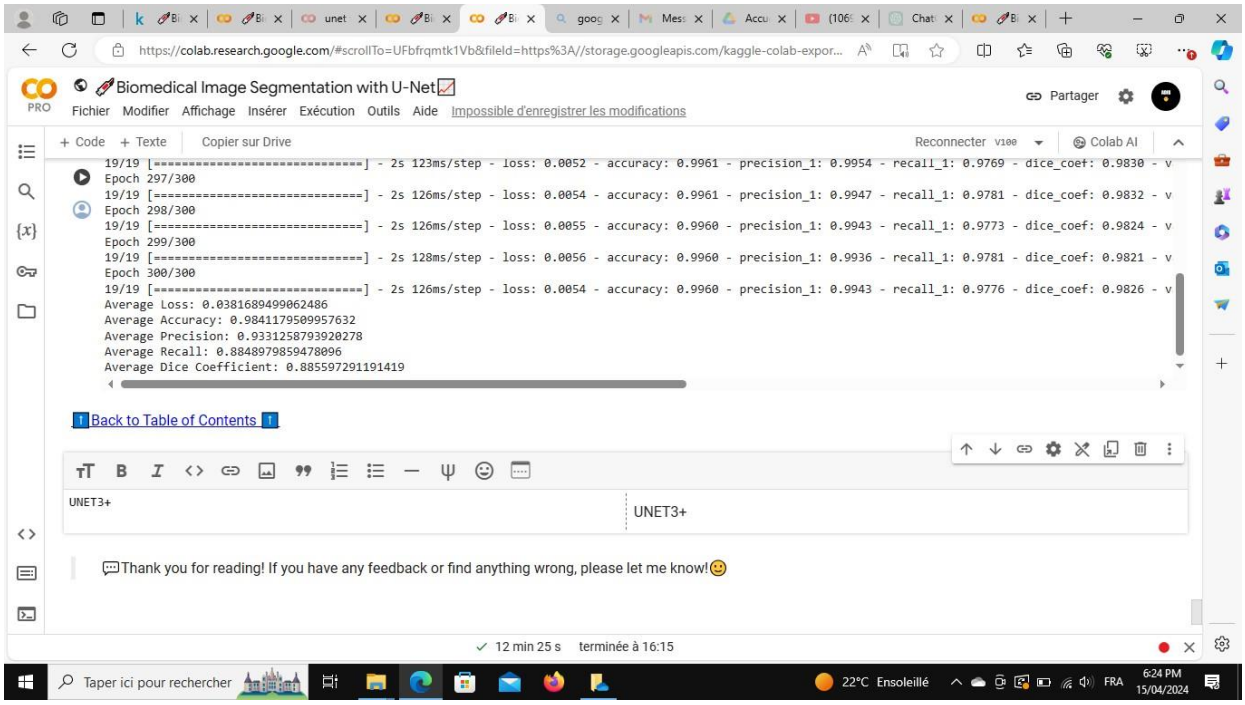
Annexes:



Annex A Experimental results with U-Net model on Google Colab Pro

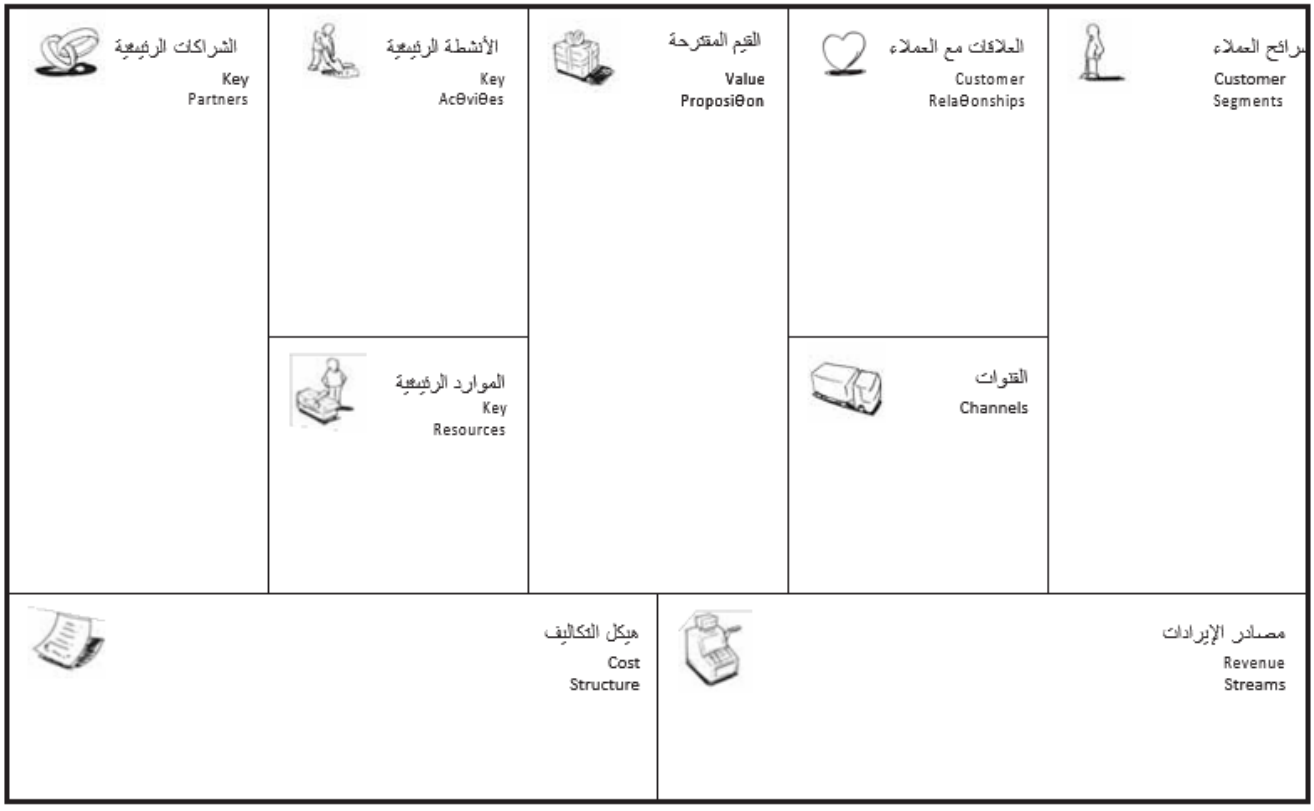


Annex B Experimental results with U-Net++ model on Google Colab Pro



Annex C Experimental results with U-Net3+ model on Google Colab Pro

مخطط نموذج العمل التجاري



الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليـــــم العالـــــي والبحث العلمي
Ministère de l'Enseignement supérieur et de la Recherche scientifique
جامعة محمد بوضياف – المسيلة
Université Mohamed Boudiaf - M'sila
مركز دعم التكنولوجيا والابتكار
Centre d'Appui à la Technologie et à l'Innovation (CATI)

Fiche de renseignements (Brevet)

في إطار متابعة ملفات براءات الاختراع المودعة على مستوى مركز دعم التكنولوجيا والابتكار بجامعة المسيلة، ولتسهيل عملية التواصل مع الطلبة والأساتذة المخترعين، نطلب من حضراتكم ملأ المعلومات التالية:

Dans le cadre du suivi des dossiers brevets déposés au Centre d'Appui à la Technologie et à l'Innovation (CATI) de l'Université de M'sila, et afin de faciliter la communication avec les inventeurs, nous vous demandons de bien renseigner les informations suivantes:

1- Titre de l'invention :	1- عنوان الاختراع:

2- Type de l'invention :	Appareil, dispositif ou machine	2- نوع الاختراع:

3- Renseignements sur l'inventeur principal : (À qui adresser toute correspondance)		3- معلومات حول المخترع الرئيسي: (لمن يتم توجيه جميع المراسلات)	
اللقب Nom de famille		الاسم Prénom	
القسم أو الكلية Dept. /Fac.		الصفة Qualité	Enseignant / Chercheur
الهاتف N° Tél.		الإيميل E-mail	

4- Les autres inventeurs						4- المخترعون الأخرون:
N°	اللقب والاسم Nom et Prénom	الصفة Qualité	التخصص Spécialité	الجامعة او المؤسسة Université/Enterprise	الهاتف N° de Tél.	البريد الالكتروني E-mail
2		Doctorant				
3		Enseignant				
4		Etudiant				
5		Fonctionnaire				
6		Incubé				
7		Ingénieur				
8		Partenaire				
9		Enseignant				
10		Etudiant				

(كل أعضاء الفريق متساوون في ملكية براءة الاختراع ما لم يتفق على خلاف ذلك مسبقا)
(Tous les inventeurs ont une propriété égale du brevet, sauf accord contraire préalable)

5- Dans quel cadre avez-vous conduit ce travail ?					5- في أي سياق قمتم بهذا العمل؟
مشروع بحث Projet de recherche ○	عمل مستقل Travail indépendant ○	مشروع تخرج Projet fin étude ○	ضمن القرار 1275 Arrêté 1275 ●	عمل محتضن Travail incubé ○	

ملاحظة: يرجى إرسال هذه الاستمارة مملوءة مع الملف النهائي لبراءة الاختراع **حصريا** الى إيميل المركز:
Remarque : Merci d'envoyer ce formulaire rempli avec le dossier de brevet, **exclusivement**, via l'e-mail du centre :
cati@univ-msila.dz

M'sila, le 17/08/2023

مركز دعم التكنولوجيا والابتكار، حاضنة أعمال جامعة المسيلة، القطب الجامعي الشمالي، جامعة محمد بوضياف بالمسيلة
رقم الهاتف: 035133849
إيميل: cati@univ-msila.dz