الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة الشهيد الشيخ العربي التبسي – تبسة

Echahid Cheikh Larbi Tebessi University– Tebessa

Faculty of Science and Technology

Departement of Electronic and Telecommunications

MASTER THESIS

Presented for obtaining the **Academic Master's degree**

Field: Telecommunications

specialty : Network and Telecommunications

Presented by: ABIDAT Mohammed

THEME

Medical image denoising. An Auto Encoders based approach

Presented and evaluated, on 27/06/2024 , Committee members:

Mr.	Karim FERROUDJI	MCB	President
Mr.	Lotfi HOUAM	MCB	Supervisor
Mrs.	Hanane DJELLAB	MCA	Examiner

Academic Year: 2023/2024

Dedication

Dedicated to my parents.

Acknowledgment

I would like to thank my advisor, Professor HOUAM Lotfi for his continuous support and guidance.
This thesis work would not have been successful without his feedback and teachings.

Table of Contents

Dedication ..... **i**

Acknowledgment ..... **ii**

Table of Contents ..... **iii**

List of Tables ..... **x**

List of Figures ..... **xi**

Chapter 1 Introduction **1**

1.1 Background and motivation 1

1.2 Problem statement 2

1.3 Objectives of the master thesis 2

Chapter 2 An Overview on Medical imaging modalities **4**

2.1 Introduction 4

2.2 Image definition 4

2.3 Types of Images 5

2.3.1 Classification of Images on the basis of Attributes 5

2.3.1.1 Vector graphics 5

2.3.1.2 Raster images 6

2.3.2 Classification of images on the basis of color 6

2.3.2.1 Gray-scale images 6

2.3.2.2 True color (or full color) 7

2.3.3 Categorizing Images Based on Dimensions 7

2.4 Characteristics of a Digital Image 8

2.4.1 Dimension 8

2.4.2 Resolution 8

2.4.3 Noise 8

2.4.4 Histogram 8
2.4.5 Luminance 9
2.4.6 Contrast 9
2.5 Applications 9
2.6 Medical imaging 10
2.7 Overview of medical imaging modalities 10
2.7.1 X-ray imaging 10
2.7.2 Computed Tomography (CT) Imaging 12
2.7.3 Magnetic Resonance Imaging (MRI) 13
2.7.4 Ultrasound imaging 14
2.7.5 Nuclear medicine imaging 16
2.7.6 Electrical Impedance Tomography (EIT) 16
2.7.7 Cardiovascular Imaging 17
2.8 Comparison between different medical imaging medical 18
2.9 conclusion 19
Chapter 3 Medical image denoising techniques and artificial neural networks 21
3.1 Introduction 21
3.2 image denoising 22
3.2.1 Image denoising problem statement 22
3.2.2 Noise sources 22
3.2.3 Noise models 23
3.2.3.1 Additive Noise Model 23
3.2.3.2 Multiplicative Noise Model 23
3.2.4 Types of noises 23
3.2.4.1 Gaussian Noise 23

3.2.4.2 Salt and Pepper Noise 24
3.2.4.3 Poison Noise 24
3.2.4.5 Impulse Noise 25
3.2.4.6 Speckle Noise 26
3.2.5 Medical Image Denoising Techniques 26
3.2.5.1 Adaptive Filter 26
3.2.5.2 Median Filter 27
3.2.5.3 FIR Filter (finite impulse response) 27
3.2.5.4 Linear Filter 27
3.2.5.5 Non-Local Means Filter 27
3.2.5.6 Wavelet transform 28
3.2.5.7 Curvelet transform 29
3.2.5.8 Convolutional Networks (CNNs) 29
3.3 Artificial Neural Networks 30
3.3.1 Biological neuron 30
3.3.2 Artificial neural networks 30
3.3.3 Perceptron in ANN 31
3.3.3.1 Inputs and outputs 33
3.3.3.2 Weights 33
3.3.3.3 Summation function 34
3.3.3.4 Activation function 34
3.3.4 The differences between biological and artificial neural networks 34
3.3.4.1 Size 34
3.3.4.2 Signal transport and processing 35
3.3.4.3 Processing speed 35

3.3.4.4 Topology 35
3.3.4.5 Speed 35
3.3.4.6 Fault-tolerance 35
3.3.4.7 Power consumption 35
3.3.4.8 Learning 36
3.3.4.9 Field of application 36
3.3.4.10 Training algorithm 36
3.3.5 Types of Artificial Neural Networks 36
3.3.5.1 Feed Forward Neural Network 36
3.3.5.2 Feedback Neural Network 37
3.3.6 ANN Learning Techniques 38
3.3.6.1 Supervised Learning 38
3.3.6.2 Unsupervised Learning 38
3.3.6.3 Reinforcement Learning 38
3.3.7 Artificial neural network applications 38
3.3.7.1 Speech recognition 38
3.3.7.2 Handwritten characters recognition 38
3.3.7.3 Signature Classification 38
3.3.7.4 Medical: 39
3.3.8 What is Convolutional Neural Network 39
3.3.9 Autoencoders 43
3.3.9.1 What Are Autoencoders? 43
3.3.9.2 Types of Autoencoder 44
3.3.9.3 Applications of Autoencoders? 47
3.4 Related studies 50

3.5 Conclusion 50

Chapter 4 Methodology 52

4.1 Introduction 52

4.2 Convolutional Autoencoder architecture 53

4.2.1 Input Layer 54

4.2.2 Encoder layers 54

4.2.2.1 Convolutional Layer 54

4.2.2.2. Pooling Layer 55

4.2.3 Bottleneck Layer 55

4.2.4 Decoder Layers 56

4.2.4.1. Up-sampling Layer 56

4.2.4.2. Convolutional Layer 56

4.2.5 Output Layer 56

4.3 Autoencoder hyperparameters 58

4.3.1 Activation function 58

4.3.2 Loss function 59

4.3.3 Optimization technique 59

4.3.4 Callback 60

4.3.5 Batch size 60

4.3.6 Number of epochs 60

4.4 Training procedure 60

4.4.1 How autoencoder train 60

4.4.2 How convolutional autoencoder train 62

4.5 Conclusion 63

Chapter 5 Experiments and results 64

5.1 Introduction 64
5.2 Types of Datasets 64
5.2.1 mini-MIAS database 65
5.2.2 Panoramic Dental X-rays database 65
5.3 Noise Types 66
5.4 Types of Losses 67
5.5 Optimizer Types 68
5.6 preprocessing 68
5.6.1 Loading and Resizing Images 68
5.6.2 Splitting Dataset 68
5.6.3 Normalizing Images 68
5.6.4 Adding Noise 68
5.6.5 Preparing Noisy Images for Training 69
5.6.6 median and gaussian filter 69
5.6.6.1 Median Filter: 69
5.6.6.2 Gaussian Filter 69
5.7 Evaluation metrics 70
5.7.1 PSNR (Peak Signal-to-Noise Ratio) 70
5.7.2 SSIM (Structural Similarity Index) 71
5.8 Tools 72
5.8.1 Software Tools 72
5.8.2 Hardware Tools 72
5.9 Fine tuning 72
5.9.1 Batch Size Tuning 73
5.9.2 Epoch Tuning 73

5.9.3 Image Size Tuning 73
5.10 Fine tuning results 74
5.10.1 mini-MIAS database results 74
5.10.1.1 Batch Size Tuning 74
5.10.1.2 Epoch Tuning 74
5.10.1.3 image size Tuning 75
5.10.2 Panoramic Dental X-rays database results 76
5.10.2.1 Batch Size Tuning 76
5.10.2.2 Epoch Tuning 76
5.10.2.3 image size Tuning 77
5.11 Empirical evaluation 77
5.12 Result and discussion 83
5.13 Conclusion and future work 84
Chapter 6 Conclusion 86
References 88

List of Tables

Table 2- 1 Comparative analysis [2]. 19

Table 5- 1 For each type of noise in the, there are both low and high levels of perturbation. 69

Table 5- 2 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels). 74

Table 5- 3 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels). 74

Table 5- 4 SSIM and PSNR results with different image sizes (fixing batch size in 10 and number of epochs in 300). 75

Table 5- 5 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels). 76

Table 5- 6 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels). 76

Table 5- 7 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 200). 77

Table 5- 8 comparing mean SSIM and PSNR scores using denoising CDAE without and with filtering for mini_MIAS database. 79

Table 5- 9 comparing mean SSIM and PSNR scores using denoising CDAE without and with filtering for Panoramic Dental X-rays database. 81

List of Figures

Figure 2- 1 (a) Sample of binary digital image, and (b) Matrix of the Image [8]. 5

Figure 2- 2 vector graphics [9]. 6

Figure 2- 3 raster graphics [9]. 6

Figure 2- 4 (a)True color image, and (b) it's red, green and blue components respectively. ...7

Figure 2- 5 Image histogram and palette [6]. 9

Figure 2- 6 Conventional X-ray Radiography. 11

Figure 2- 7 X ray image of a human hand[11]. 11

Figure 2- 8 X-ray CT Image acquisition using a circular sensor array [10]. 12

Figure 2- 9 CT scan of fluid collection at the gastro-esophageal junction[12]. 13

Figure 2- 10 Block diagram of MRI device. 14

Figure 2- 11 MRI image of a human brain[13]. 14

Figure 2- 12 Block diagram of Ultrasound imaging device. 15

Figure 2- 13 Ultrasound image of the pancreas[14] 15

Figure 2- 14 whole body scan for thyroid cancer evaluation[15]. 16

Figure 2- 15 human thorax[16]. 17

Figure 2- 16 axial image of the heart[17]. 18

Figure 3- 1 Schematic image of a biological neuron [24]. 30

Figure 3- 2 Schematic diagram of a neural network [25]. 31

Figure 3- 3 A mathematical model of perceptron in a neural network [26]. 32

Figure 3- 4 Schematic diagram of a shallow neural network [1]. 33

Figure 3- 5 Schematic diagram of a deep learning neural network [1]. 33

Figure 3- 6 Plot of most commonly used activation function [26]. 34

Figure 3- 7 Feed Forward Neural Network. 37

Figure 3- 8 Feedback Neural Network (Recurrent NN).....37

Figure 3- 9 A generic CNN pipeline with 6 types of layers: 1 input, 2 convolutional, 2 pooling, 1 flattening, 2 fully connected, and 1 output [28]. 39

Figure 3- 10 Image and a filter [29].....40

Figure 3- 11 Applying The Filter To The Image [29].....40

Figure 3- 12 A max pooling in action. You can think of each colored region as a position of the 2x2 filter [30].....41

Figure 3- 13 how flattening layer function [28]. 42

Figure 3- 14 The architecture of Fully Connected Layers [31].....42

Figure 3- 15 Demonstrates The Basic Architecture Of An Autoencoder [32].....44

Figure 3- 16 Architecture of fully connected autoencoders [32].....45

Figure 3- 17 Architecture of Convolutional Autoencoder for Image Segmentation [32].....45

Figure 3- 18 A denoising autoencoder processes a noisy image, generating a clean image on the output side [32].....46

Figure 3- 19 The topology of Sparse Autoencoder [32]. 46

Figure 3- 20 Architecture of variational autoencoder [32].....47

Figure 3- 21 Sequence-to-Sequence Autoencoder [33]. 47

Figure 3- 22 Face completion by filling in the missing pixels [32].....48

Figure 3- 23 An illustration of the fully convolutional SegNet architecture [34]. 49

Figure 4- 1 Block diagram of the proposed model architecture. 52

Figure 4- 2 Architecture of the proposed CDAE.....53

Figure 4- 3 Convolution of a 5×5 image and a 3×3 kernel with stride =1. Observe how a feature map is generated step by step [28].....54

Figure 4- 4 Convolution of zero-padded image with a kernel. Observe how the dimensions of the input remain preserved as compared to no zero-padding [28]. 55

Figure 4- 5 Demonstration of the max pooling operation [28]. 55

Figure 4- 6 Architecture of the CDAE used [3]. 57

Figure 4- 7 Sigmoid activation function [31]. 58

Figure 4- 8 ReLU activation function [31] 59

Figure 4- 9 Autoencoder training architecture [37]. 62

Figure 5- 1 Random samples of medical images taken from mini-MIAS dataset [38]. 65

Figure 5- 2 Random samples of medical images taken from the Panoramic Dental X-rays database [39]. 66

Figure 5- 3 Effect of different type of noise on the original image, taken from the mini-MIAS dataset (first row show the minimal level of noise second row show a higher noise level). ..66

Figure 5- 4 Effect of different type of noise on the original image, taken from the Panoramic Dental X-rays dataset (first row show the minimal level of noise second row show a higher noise level). 67

Figure 5- 5 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels). 74

Figure 5- 6 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels). 75

Figure 5- 7 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 300). 75

Figure 5- 8 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels). 76

Figure 5- 9 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels). 76

Figure 5- 10 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 200). 77

Figure 5- 11 denoising performance of CDAE on the mini-MIAS database with and without filtering in the preprocessing. The top row displays the real images. The second and fifth rows show the noisier versions with minimal and higher noise levels, respectively. The third and sixth rows present the denoising results of CDAE without filtering. The fourth and eighth rows show the results of CDAE with filtering in the preprocessing. 78

Figure 5- 12 denoising performance of CNNDAE on the Panoramic Dental X-rays database with and without filtering in the preprocessing. The top row displays the real images. The second and fifth rows show the noisier versions with minimal and higher noise levels, respectively. The third and sixth rows present the denoising results of CNNDAE without filtering. The fourth and eighth rows show the results of CNNDAE with filtering in the preprocessing. 80

Figure 5- 13 loss and validation loss from 300 epochs using a batch size of 10 and 200*200 pixels image size with salt and pepper noise (density=0.3, proportion=0.5).....82

Figure 5- 14 loss and validation loss from 200 epochs using a batch size of 10 and 200×200 pixels image size with Gaussian noise (mean=0, variance=0.08).....82

Figure 5- 15 loss and validation loss from 300 epochs using a batch size of 10 and 200*200 pixels image size with Speckle noise (mean=0, variance=0.04) with a log transformation in the preprocessing.....83

Acronyms

AE: Autoencoder

ANN: Artificial Neural Network

CBCT: Cone Beam Computed Tomography

CNN: Convolutional Neural Network

CT: Computed Tomography

DL: Deep Learning

DNN: Deep Neural Network

DSA: Digital Subtraction Angiography

EIT: Electrical Impedance Tomography

GAN: Generative Adversarial Network

ML: Machine Learning

MRI: Magnetic Resonance Imaging

MRS: Magnetic Resonance Spectroscopy

PET: Positron Emission Tomography

PSNR: Peak Signal-to-Noise Ratio

SNR: Signal-to-Noise Ratio

SPECT: Single Photon Emission Computed Tomography

SSIM: Structural Similarity Index

US: Ultrasound

Chapter 1 Introduction

1.1 Background and motivation

Artificial intelligence has had a profound impact on the medical industry, revolutionizing the process of medical diagnosis. The utilization of this technology has significantly enhanced the diagnostic procedure, resulting in improved efficacy, speed, and reliability. Its main objective is not to supplant doctors, but rather to facilitates their work [1]. One of the services provided is medical imaging, which encompasses the methods and technology employed to generate visual depictions of the inside structures of the body.

These visual representations, commonly referred to as pictures, can be employed for the purposes of diagnosing, monitoring, or treating a range of medical disorders. The user's text is simply "M".Medical imaging facilitates the analysis and depiction of different anatomical components, including bones, muscles, organs, blood vessels, and other interior structures [2]. Medical imaging techniques such as X-rays, magnetic resonance imaging (MRI), computer tomography (CT), ultrasound, and others are used.All of these medical photos are prone to noise. The reasons for this variation range from the utilization of diverse image collecting techniques to efforts aimed at reducing patients' radiation exposure.

As the level of radiation lowers, the level of noise increases. Noisy images often need to be denoised in order to facilitate accurate image analysis, whether performed by humans or machines [3]. Image denoising is a procedure that eliminates noise from an image, resulting in a crisp and distinct image. Medical imaging machines primarily utilize it to mitigate the noise in the resultant image. Prior to the final printing process, there are various approaches that can be employed to prevent any distortions in the image.

Autoencoders are a prominent type of software utilized for the purpose of removing noise from photographs prior to their final printing [4]. An autoencoder is a neural network that encodes the input into a compact and meaningful representation, and then decodes it to rebuild the input as accurately as possible to the original [5]. In order to handle image processing, it is necessary to employ a certain kind of autoencoder known as convolutional

autoencoders. These autoencoders make use of the complete potential of convolutional neural networks to effectively utilize the structure of images [3].

1.2 Problem statement

The task of removing noise from medical images is fraught with multiple challenges, including intricate noise patterns originating from several sources, such as electrical system noise, patient motion, and differences in scanning techniques. The noise might exhibit Gaussian, Poisson, or more intricate forms such as speckle noise. One of the primary challenges is the high dimensionality of images, especially 3D images like as CT and MRI scans. These images have a huge number of dimensions, which makes the denoising process computationally demanding. Efficient algorithms capable of handling big datasets are necessary to address this issue.

It is essential to maintain the intricate intricacies and structural information. Preserving anatomical features and borders is crucial when minimizing noise. Any error has the potential to result in the loss of crucial diagnostic data and various additional complications.

CNN-based denoising autoencoders are capable of efficiently tackling these difficulties by utilizing big datasets to acquire knowledge about intricate noise patterns that are specific to different medical imaging modalities and originate from diverse sources of noise. In addition, they have the capability to address the issue of image structure loss by extracting features at many scales. This allows them to retain intricate details and structural information while reducing noise.

1.3 Objectives of the master thesis

The main aim of this master thesis is to tackle the problem of reducing noise in medical images by employing a convolutional denoising autoencoder (CAE). Despite extensive study in the field of image denoising, the task of extracting significant information from noisy images while preserving crucial details remains a formidable challenge. Efficiently eliminating noise from medical images is crucial as accurate diagnosis and treatment necessitate high-quality images.

The aim of our project is to develop a dependable procedure for eliminating noise from medical images while preserving essential diagnostic patterns.

In order to accomplish this, we utilize the capabilities of the CAE model, which is a specialized convolutional neural network specifically intended to detect and analyze noise patterns from the feature maps at each layer. The CAE model can effectively differentiate between noise and important information by understanding the context of the images, thus ensuring that only the noise is removed. The objective of this strategy is to improve the clarity and quality of medical images, hence enabling more accurate diagnoses and improved overall healthcare results.

Chapter 2 An Overview on Medical imaging modalities

2.1 Introduction

Medical imaging is an essential tool in the diagnosis and treatment of various diseases. It refers to the non-invasive techniques and approaches used to create visual representations of the internal organs and tissues of the human body. We can utilize these visual representations, also known as images, to detect and diagnose a variety of diseases, guide disease treatment strategies, and monitor treatment effectiveness. Specifically, it examines and visualizes various body parts such as bones, muscles, organs, blood vessels, and other internal structures.

Diagnostic and therapeutic imaging are the two main categories into which medical imaging falls.

Diagnostic imaging uses modalities like X-ray radiography, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound (US), and nuclear medicine to detect and diagnose diseases and assess their severity.

Modalities such as fluoroscopy, angiography, and interventional radiology are part of therapeutic imaging, which guides procedures like surgery or radiation therapy. Technology advancements have led to the development of a wide range of medical imaging modalities that offer detailed visual information about the internal structures and functions of the body. This chapter provides an overview of the most prominent imaging modalities, starting with the basics of diagnostic x-ray imaging, the historical starting point of medical imaging, and then discussing various imaging modalities [2].

2.2 Image definition

The digital image itself is really a data structure within the computer, containing a number or code for each pixel or picture element in the image. This code determines the color of the pixel. Consider each pixel as a discrete sample of a continuous real image. From a photographer's point of view, it is a photograph (i.e., a projection of the real world), and from a computer engineer's point of view, an image may be a two-dimensional (2D) signal.

Therefore, an image is a two-dimensional function f(x,y), where for each position (x,y) in the projection plane, the values of the function f(x,y) represent the amplitude or light intensity of the image [6].

We define a digital image as a 2D discrete signal that varies over the spatial coordinates x and y, represented mathematically as f(x,y). It is also an n×n array of elements, and each element represents the sampled intensity [7].

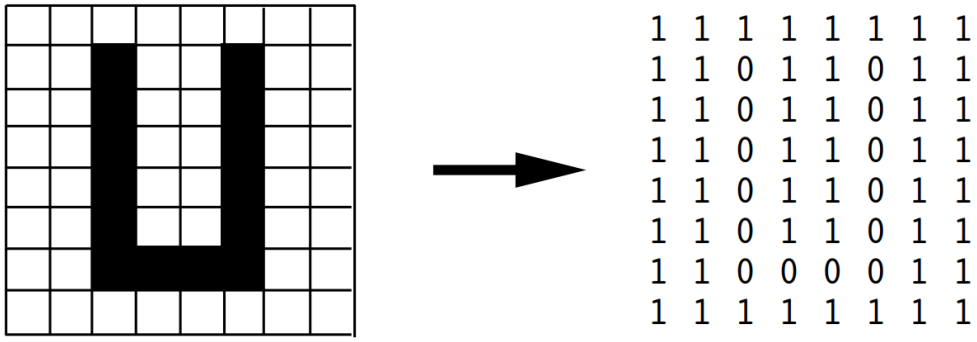

Figure 2- 1 (a) Sample of binary digital image, and (b) Matrix of the Image [8].

2.3 Types of Images

Various criteria such as attributes, color, dimension, and data types can perform image classification. We classify images as raster and vector based on attribute criteria, while we can classify them as binary, grayscale, true color, or pseudo color based on color. Furthermore, we distinguish between 2D and 3D images based on their dimensions, and we classify the images into signed integer, unsigned integer, float, logical, and double types based on their data types. So there is no single accepted way of classifying images. Here is a detailed description of the image classification process:

2.3.1 Classification of Images on the basis of Attributes

Any image's attributes determine whether it is a raster or vector graphic image.

2.3.1.1 Vector graphics

Uses graphic primitives, such as points, lines, circles, and ellipses, to depict an image. Hence, the notion of resolution is practically not present in graphics [7].

Figure 2- 2 vector graphics [9].

2.3.1.2 Raster images

Are pixel-based, meaning that their quality depends on the quantity of pixels. Therefore, operations such as enlarging or blowing up a raster image frequently result in a quality reduction [7].

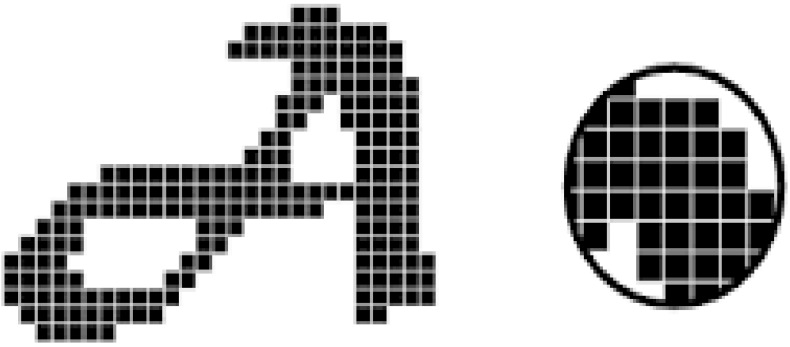

Figure 2- 3 raster graphics [9].

2.3.2 Classification of images on the basis of color

The images fall into the following categories based on color:

2.3.2.1 Gray-scale images

The term gray-scale refers to the range of shades between white and black or vice versa; such images have many shades of gray, and eight bits (2^8 = 256) are enough to represent the gray-scale because the human visual system cannot differentiate more than 32 different gray levels, and the additional bits are necessary to cover noise margins. Most

medical images, such as X-rays, CT images, MRIs, and ultrasound images, are often gray-scale images [7].

2.3.2.2 True color (or full color)

The primary colors red, green, and blue combine to create the color of an image. We represent each color component like a grayscale image using eight bits. Most true-color images use 24 bits to represent all the colors. Hence, a true color image can be considered a three-band image. The number of colors that are possible is 256^3 (i.e., 256×256×256 = 1,67,77,216 colors) [7].

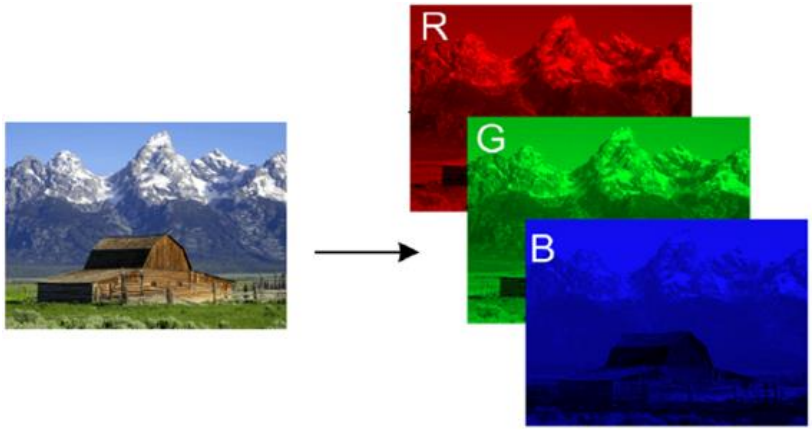

Figure 2- 4 (a)True color image, and (b) it's red, green and blue components respectively.

2.3.3 Categorizing Images Based on Dimensions

Dimensions can also classify images. Normally, digital images are a 2D rectangular array of pixels. When we consider another dimension, such as depth or any other characteristic that may be required, we generate a higher-order stack of images similar to 3D images.

A volume image, where pixels are known as voxels, serves as a good example of a 3D image. The term '3D image' refers to the three-dimensional (x, y, and depth) dimension of the target in the imaging system, which could be a scene or an object. In medical imaging, some of the frequently encountered 3D images are CT images, MRIs, and microscopy images. Basically, we store these as 2D image slices taken across the body or the skull. Range images (often used in remote sensing applications) are also 3D images, because they incorporate depth information [7].

2.4 Characteristics of a Digital Image

An image is a structured set of information characterized by the following parameters:

2.4.1 Dimension

This is the size of the image. It takes the form of a matrix, whose elements are numerical values representing light intensities (pixels). The number of rows in this matrix multiplied by the number of columns gives us the total number of pixels in an image [6].

2.4.2 Resolution

In image production, a monitor or printer achieves clarity or fineness of detail. The number of pixels per unit of measurement (inch or centimeter) on computer monitors expresses resolution. We also use the word resolution to indicate the total number of pixels that a monitor can display horizontally or vertically; the higher the number, the higher the resolution [6].

2.4.3 Noise

The illumination of the sensor's optical and electronic devices causes noise in an image, which is defined as a sudden variation in a pixel's intensity in relation to its neighbors [6].

2.4.4 Histogram

The grayscale, or color histogram, of an image is a function that gives the frequency of appearance of each grayscale (color) in the image. In the case of an image that is too light or too dark, it provides a wealth of information on the distribution of gray levels (color) and indicates the bounds between which the majority of gray levels (color) fall.

By introducing a few modifications, one can enhance the quality of an image and extract useful information from it. We often modify the corresponding histogram to reduce quantization error, compare two images obtained under different lighting conditions, or measure certain properties of an image [6].

.

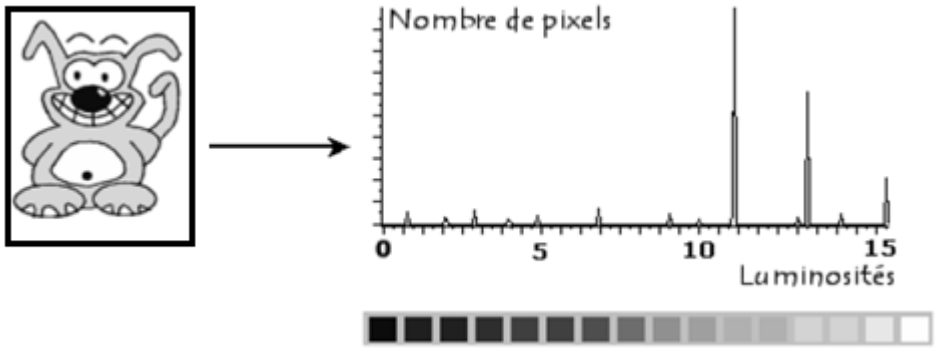

Figure 2- 5 Image histogram and palette [6].

2.4.5 Luminance

This is the degree of brightness of the image points. It is also defined as the quotient of a surface's luminous intensity divided by its apparent area. A distant observer substitutes the word luminance for the word brilliance, referring to the brightness of an object. Good luminance is characterized by:

- Luminous (bright) images,

- Good contrast: avoid images where the contrast range tends towards white or black; these images result in loss of detail in dark or bright areas.

- Absence of noise [6].

2.4.6 Contrast

It's the marked opposition between two regions of an image, more precisely between the dark and light regions of the image. Contrast is defined as the luminance of two image areas [6]. If L_1 and L_2 are the respective luminance levels of two adjacent image areas, A_1 and A_2, contrast C is defined by the ratio:

$$C = \frac{L1-L2}{L1+L2} \qquad (2.1)$$

2.5 Applications

The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas, to know that:

1. Image enhancement/restoration

2. Medical visualization

3. Artistic effect

4. Industrial inspection

5. Law enforcement

6. Human computer interfaces

7. Remote sensing via satellites and other space crafts

8. Image transmission and storage for business applications

9. Radar, SONAR, Acoustic image processing

10. Robotics [10]

2.6 Medical imaging

Medical imaging refers to the techniques and technologies used to create visual representations of the body's interior. We can use these visual representations, also known as images, to diagnose, monitor, or treat various medical conditions. Medical imaging enables the examination and visualization of various body parts, such as bones, muscles, organs, blood vessels, and other internal structures.

We can divide medical imaging into two main categories: diagnostic imaging and therapeutic imaging. Diagnostic imaging employs modalities like X-ray, CT, MRI, ultrasound, and nuclear medicine to diagnose and assess the severity of a condition. Modalities such as fluoroscopy, angiography, and interventional radiology are part of therapeutic imaging, which guides procedures like surgery or radiation therapy.

Some common modalities include X-ray, CT, MRI, ultrasound, and nuclear medicine; each has its own indications and limitations. These modalities are powerful diagnostic tools that can reveal the internal structure of the body and its functions [2].

2.7 Overview of medical imaging modalities

2.7.1 X-ray imaging

X-ray imaging, also referred to as radiography, is a diagnostic imaging technique in medicine that produces high-resolution images of internal anatomical structures such as bones. The basic principles of X-ray imaging are as follows:

X-rays are a form of electromagnetic radiation that can pass through many solid objects, including the human body. A beam of X-rays directs itself towards the area of the body under examination in this type of imaging. A special detector, such as an X-ray film or digital detector, detects the X-rays as they pass through the body. Because bones are denser than other tissues, they absorb more X-rays and appear white on the final image, while softer tissues such as muscles and organs absorb fewer X-rays and appear darker. The machine must use the appropriate amount of radiation and correctly position the body part in relation to the X-ray beam to produce a clear image.

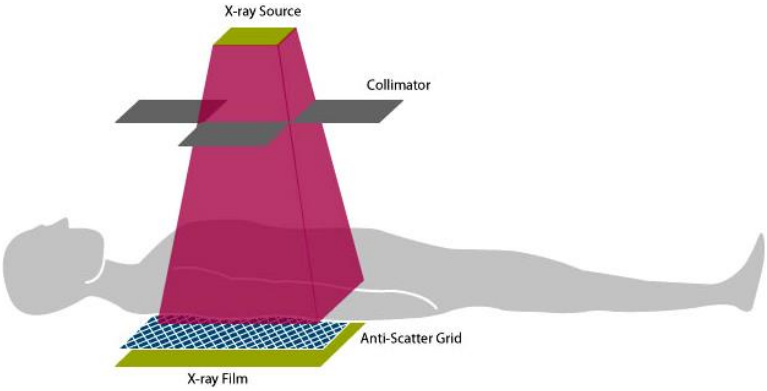

Figure 2- 6 Conventional X-ray Radiography.

A radiologist, a medical doctor who specializes in interpreting medical images, takes different views to examine the same area from different angles, resulting in a more complete picture. A radiologist, a medical doctor specializing in medical image interpretation, then interprets the images to identify any abnormalities or issues, such as broken bones, tumors, or other conditions, and provides a diagnosis [2].

Figure 2- 7 X ray image of a human hand[11].

2.7.2 Computed Tomography (CT) Imaging

CT imaging, commonly called CAT scanning due to its use of X-rays to make cross-sectional pictures of the body, is a common medical imaging procedure. The basic principles of CT imaging are as follows:

CT scans use X-rays to create detailed images of internal structures by directing the X-rays at the body from different angles and measuring the intensity of the X-rays that pass through the body with detectors. CT scans use a special type of X-ray detector known as a multi-slice detector, which can acquire multiple images simultaneously from different angles, enabling the creation of detailed cross-sectional images of the body. To improve image quality, CT scans also use techniques such as spatial filtering, which removes noise and improves contrast, and multi-energy imaging, which uses different energy levels of X-ray beams to capture different information and increase image contrast.

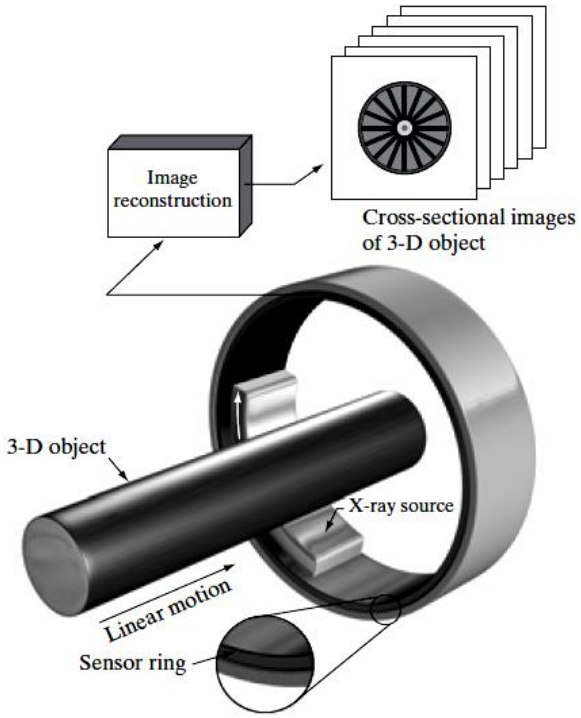

Figure 2- 8 X-ray CT Image acquisition using a circular sensor array [10].

Additionally, CT scans can employ a technique known as dose modulation, which modifies the radiation dose based on the size, shape, and composition of the scanned body part, thereby mitigating the risk of side effects. A computer then processes the images from CT scans and displays them in various formats, including cross-sectional slices, 3D images,

and even virtual reality images. Radiologic technologists perform CT scans, while radiologists, medical doctors specializing in medical image interpretation, interpret the images [2].

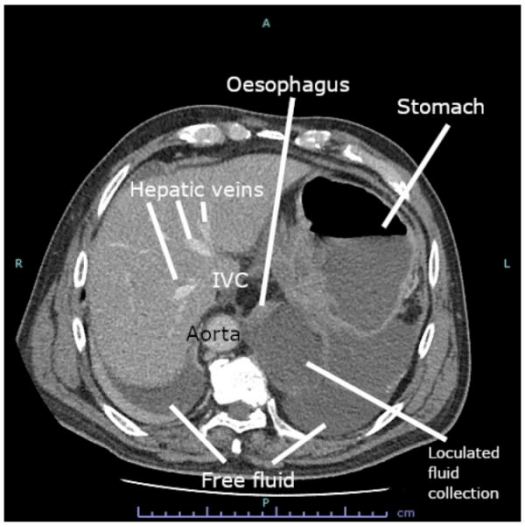

Figure 2- 9 CT scan of fluid collection at the gastro-esophageal junction[12].

2.7.3 Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) and Magnetic Resonance Microscopy (MRM) are medical imaging techniques that use a magnetic field and radio waves to produce detailed images of internal structures. The basic principles of both MRI and MRM are as follows.

Magnetic Resonance Imaging (MRI) and Magnetic Resonance Angiography (MRA) are advanced imaging modalities that use a powerful magnetic field to align the nuclei of hydrogen atoms in the body. This generates a small magnetic moment, enabling the creation of detailed images of the body's internal structures.

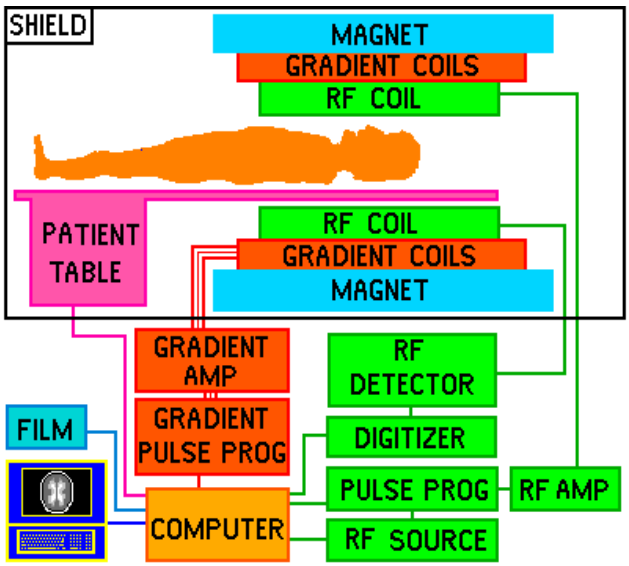

Figure 2- 10 Block diagram of MRI device.

The process begins by using radiofrequency (RF) pulses to change the alignment of the hydrogen nuclei, which causes them to emit a weak radio signal. A detector picks up these signals, and a computer processes them to create detailed images of the body's internal structures [2].

Figure 2- 11 MRI image of a human brain[13].

2.7.4 Ultrasound imaging

Ultrasound imaging is a medical imaging technique that uses high-frequency sound waves to produce detailed images of the internal structures of the body. The basic principles of ultrasound imaging are:

Ultrasound imaging is a medical imaging technique that uses a transducer, a device that emits high-frequency sound waves and detects the echoes of these waves as they bounce back from internal structures. The sound waves are sent into the body and as they hit a boundary between different types of tissue, some of the sound waves are reflected back to the transducer.

These echoes are picked up by the transducer and converted into electrical signals, which are then processed by a computer to create detailed images of the internal structures of the body. These images can be used to evaluate organs, blood vessels, and fetuses during pregnancy, among other things [2].

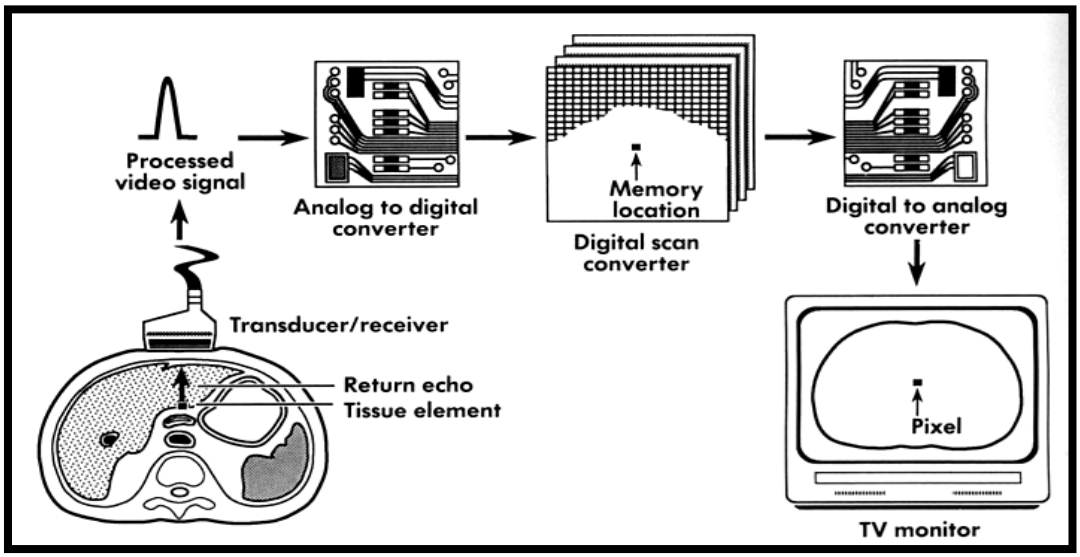

Figure 2- 12 Block diagram of Ultrasound imaging device.

Figure 2- 13 Ultrasound image of the pancreas[14]

2.7.5 Nuclear medicine imaging

Nuclear imaging is a kind of medical imaging that creates high-resolution pictures of the body's inner workings by using minute quantities of radioactive material, called radiotracers. The basic principles of nuclear imaging are:

Nuclear imaging is a medical imaging technique that uses radiotracers that are introduced into the body, either by injection, inhalation, or ingestion, depending on the type of exam. The radiotracers emit gamma rays, which are detected by a special camera, called a gamma camera. This camera creates an image of the distribution of the radiotracer in the body. The gamma camera detects the gamma rays and converts them into an image, which is then processed by a computer and can be viewed in different ways, such as 2D images, 3D images, or functional images. The images obtained from nuclear imaging provide functional information about the body, such as blood flow, metabolism, or chemical activity. This information can be used to evaluate certain disorders such as cancer, inflammation, or heart function [2].

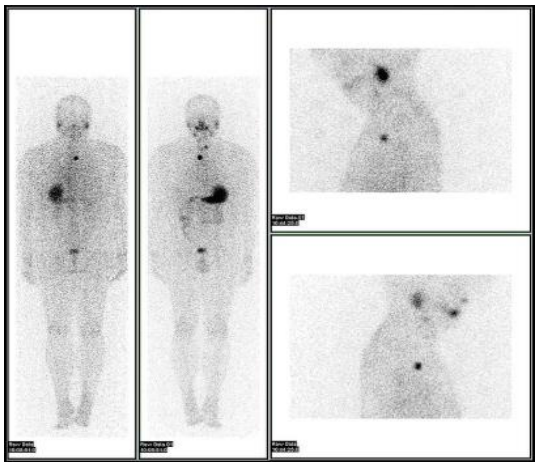

Figure 2- 14 whole body scan for thyroid cancer evaluation[15].

2.7.6 Electrical Impedance Tomography (EIT)

Electrical Impedance Tomography (EIT) is a medical imaging technique that uses electrical currents to produce images of the interior of the body. The EIT's basic principles are:

EIT (Electrical Impedance Tomography) is a medical imaging technique that uses a small number of electrodes placed on the surface of the body. The electrodes apply small

electrical currents to the body and measure the resulting voltage changes. When the electrical currents pass through the body, they encounter different types of tissue with different electrical properties, such as conductivity and permittivity.

These properties influence the distribution of the electrical current, enabling the creation of images depicting the internal structure of the body. Electrodes measure electrical activity in various parts of the body, and the computer uses this information to build representations of organs, including the lungs, heart, and brain.

EIT is a non-invasive, radiation-free imaging method with potential applications in lung, brain, and breast imaging [2].

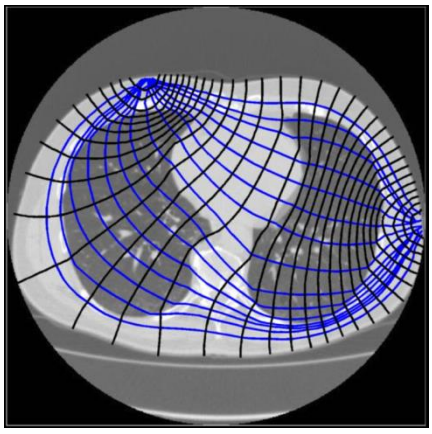

Figure 2- 15 human thorax[16].

2.7.7 Cardiovascular Imaging

It is a subspecialty of medical imaging that uses various modalities to visualize the structure and function of the heart and blood vessels. It includes techniques such as Echocardiography, Cardiac Computed Tomography (CCT), and Magnetic Resonance Imaging (MRI). These techniques help in the diagnosis of heart diseases, such as coronary artery disease, valvular heart disease, and congenital heart disease. They also play an important role in guiding interventional procedures such as angioplasty and stent placement [2].

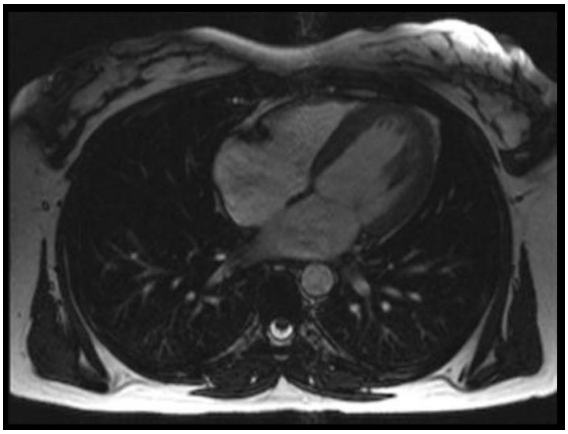

Figure 2- 16 axial image of the heart[17].

2.8 Comparison between different medical imaging medical

This table provides a detailed overview of different medical imaging modalities, highlighting their working principles, common applications, advantages, and limitations.

Modality	Working Principle	Applications	Advantages	Limitations
X-ray	Using ionizing radiation to produce images of the internal structure of a body	Detecting broken bones, monitoring treatment of conditions such as pneumonia, monitoring the healing of fractures	Inexpensive, widely available, quick results	Low resolution images, ionizing radiation exposure
CT Scan	X-ray technology combined with computer processing to produce detailed images	Detecting cancers, identifying blood clots, assessing organ damage, diagnosing spinal problems	High resolution images, non-invasive	Ionizing radiation exposure, high cost
MRI	Combination of powerful magnetic fields and radio waves allows for the creation of high-	Detecting tumors, brain and spinal cord injuries, joint problems, and monitoring the progression of conditions such as	Non-ionizing radiation, detailed images	Long examination time, high cost, not suitable for patients with metal

	resolution photographs	multiple sclerosis		implants
Ultrasound Imaging	Using high frequency sound waves to produce images	Monitoring the growth and development of a fetus, evaluating organs and tissues, detecting tumors and cysts	Non-invasive, no ionizing radiation exposure	Operator dependent, limited view of deep structures
Nuclear Imaging	Using radioactive tracers to produce images	Detecting diseases and conditions such as cancer, heart disease, and neurological conditions	High specificity For certain conditions, non-invasive	Limited view of the structure, exposure to ionizing radiation
Electrical Impedance Tomography	Using electrical currents to produce images	Monitoring changes in tissue, measuring organ function	Non-invasive, portable	Limited spatial resolution, operator dependent
Cardiovascular Imaging	Various techniques to produce images of the heart and blood vessels	Diagnosing heart disease, monitoring treatment	High resolution images, non-invasive	High cost, operator dependent

Table 2- 1 Comparative analysis [2].

2.9 conclusion

In conclusion, medical imaging modalities are a crucial tool in modern healthcare and play a vital role in the diagnosis, management, and treatment of various diseases. There are several different modalities available, each with its own advantages and limitations. X-ray, computed tomography (CT), magnetic resonance imaging (MRI), nuclear imaging, ultrasound, electrical impedance tomography (EIT), and emerging technologies for in vivo imaging are the common medical imaging modalities.

In addition to these modalities, there are several advanced techniques, such as contrast-enhanced MRI, MR approaches for osteoarthritis, cardiovascular imaging, and medical imaging data mining and search, that can provide additional information and improve diagnosis accuracy.

Chapter 3 Medical image denoising techniques and artificial neural networks

3.1 Introduction

The quality of digital images is determined by a variety of factors. The image tends to have a good quality when it has minimal disturbing factors such as salt and pepper noise, Gaussian noise, mixed noise, etc. Any disturbance from the outside causes a deficiency in the image signal, which we refer to as noise. Noise is one of the most important factors in the degradation of image quality. The acquisition procedure for digital images converts optical signals into electrical signals. Furthermore, these signals undergo conversion into digital signals, which are then presented in digital images based on the noise present. Images can contain a variety of noise types. The most popular noise, or, in other words, the disturbing factors, such as salt and pepper noise (impulse noise), has black and white pixels on the images.

Image denoising is to remove noise from a noisy image so as to restore the true image. However, due to the high-frequency characteristics of noise, edges, and textures, this process can be challenging.

To distinguish them during the denoising process, the denoised images may unavoidably lose some details. Overall, recovering meaningful information from noisy images is an important problem nowadays in the process of noise removal to obtain high-quality images. In fact, researchers have studied image denoising for a long time, making it a classic problem. However, it remains a challenging and open task [18].

There are two sections in this chapter. The first section covers the problem of image denoising, along with noise sources, models, and denoising methods. The second section discusses artificial neural networks, including CNNs and autoencoders, and their application in image denoising.

3.2 image denoising

3.2.1 Image denoising problem statement

We can mathematically model the problem of image denoising as follows:

$$y = x + n \quad (3.1)$$

In practical applications, various methods such as median absolute deviation, block-based estimation, and principal component analysis (PCA)-based methods can estimate additive white Gaussian noise (AWGN) with standard deviation. The purpose of noise reduction is to decrease the noise in natural images while minimizing the loss of original features and improving the signal-to-noise ratio (SNR). The major challenges for image denoising are as follows:

- Flat areas should be smooth,
- Edges should be protected from blurring,
- Textures should be preserved, and
- New artifacts should not be generated.

Because solving the clean image x in Eq. (3.1) is an ill-posed problem, we cannot get a unique solution from the image model with noise. To obtain a trustworthy estimation

Over the past several years, the field of image processing has extensively studied image denoising. We will introduce spatial domain methods and transform domain methods in more detail in the upcoming sections.

Owing to solve the clean image x from the Eq. (3.1) is an ill-posed problem, we cannot get the unique solution from the image model with noise. To obtain a good estimation of image x, image denoising has been well-studied in the field of image processing over the past several years. Generally, image denoising methods can be roughly classified as: spatial domain methods, transform domain methods, which are introduced in more detail in the next couple of sections [19].

3.2.2 Noise sources

Image acquisition and transmission can introduce noise into the image. There may be a variety of reasons why noise appears in the image. The number of pixels corrupted in the

image determines the noise quantification. The primary sources of noise in digital images are as follows:

- Environmental circumstances may influence the imaging sensor.
- Low light and sensor temperature may cause noise in the image.
- Dust particles present on the scanner may cause noise in the digital image.
- There is interference in the transmission channel [20].

3.2.3 Noise models

Noise in the image can take two forms: additive form and multiplicative form.

3.2.3.1 Additive Noise Model

The addition of an additive noise signal to the original signal results in a corrupted, noisy signal, which adheres to the following model:

$$w(x,y) = s(x,y) + n(x,y) \quad (3.2)$$

Gaussian noise is a kind of additive noise that uniformly distributes itself over the signal. This kind of noise has a gaussian distribution [20].

3.2.3.2 Multiplicative Noise Model

This model involves multiplying the noise signal by the original signal. The multiplicative noise model adheres to the following rules:

$$w(x,y) = s(x,y) \times n(x,y) \quad (3.3)$$

Where s(x, y) represents the original image intensity and n(x, y) denotes the noise introduced to compose the corrupted signal w(x, y) at (x, y) pixel position [20].

3.2.4 Types of noises

The noise's pattern and probabilistic characteristics distinguish it. There is an abundance of different types of noise. We focus solely on the most notable types, which include Gaussian noise, salt and pepper noise, poison noise, impulse noise, and speckle noise.

3.2.4.1 Gaussian Noise

Gaussian noise, also known as the Gaussian distribution, is statistical noise with a probability density function (PDF) that is equal to the normal distribution. In other words, the

values that the noise can take on are Gaussian-distributed. A special case is white Gaussian noise, in which the values at any given times are equally distributed and statistically independent (un co-related). In applications, Gaussian noise is most frequently used as additive white noise to acquire additive white Gaussian noise. Some scholars have referred to gaussian noise as additive noise [20]. The signals' haphazard oscillation produces Gaussian noise. We can calculate the probability density function P for a Gaussian random variable (z) as follows [18]:

$$P_G(Z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(Z-\mu)^2}{2\sigma^2}} \qquad (3.4)$$

3.2.4.2 Salt and Pepper Noise

Salt and pepper noise is a kind of noise normally observed on images. It manifests as randomly occurring white and black pixels. An efficient noise elimination technique for this kind of noise involves the use of a median filter, morphological filter, or contra-harmonic mean filter. Salt and pepper noise sneaks into images in situations where rapid transients occur, such as faulty switching [20].

Salt and pepper noise values may be either minimum (0) or maximum (255). For pepper noise, the typical intensity value is close to 0; for salt noise, it is close to 255. Furthermore, the unaffected pixels remain unchanged [18, 21].

$$n(x,y) = \begin{cases} 0, & \text{Pepper noise} \\ 255, & \text{Salt noise} \end{cases} \qquad (3.5)$$

3.2.4.3 Poison Noise

The nonlinear responses of the image detectors and recorders induce poisson noise. This kind of noise is image-data-reliant. This term arises because detection and recording processes involve arbitrary electron emission with a Poisson distribution and a mean response value [20].

If we assume that the noise has a unity variance, the image-dependent term has a standard deviation because the mean and variance of a Poisson distribution are identical [22].

$$P(k) = \frac{e^{-\lambda}\lambda^k}{k!} \qquad (3.6)$$

P(k) is the probability of having *k* pixels affected by noise in a window of a certain dimension. It also represents the average number of affected pixels in a window of the same size, also known as the Poisson distribution variance [22].

3.2.4.5 Impulse Noise

Impulse noise is a class of acoustic noise that comprises unwanted, approximately instantaneous (impulse-like) sharp sounds (like clicks and pops). Electromagnetic interference, scratches on recording disks, and ill-synchronization in digital recording and communication are the main causes of this type of noise. Impulse noise corruption is most familiar in digital images. Impulse noise consistently operates independently of the image pixels, distributing itself randomly throughout the image.

Hence, unlike Gaussian noise, for an impulse noise-corrupted image, all the image pixels are not noisy; a number of image pixels will be noisy, and the rest of the pixels will be noise-free. There are two types of impulse noise: salt and pepper noise and random-valued impulse noise. In the salt and pepper type of noise, the noisy pixels take either a salt value (gray level = 225) or a pepper value (gray level = 0), and they appear as black and white spots on the images. If p is the total noise density, then salt noise and pepper noise will have a noise density of p/2. Eq. (3.7) provides a mathematical representation of this.

$$\mathrm{Yij} = \begin{cases} 0 \text{ or } 255 \text{ with probability p} \\ \mathrm{x_{ij}} \text{ with probability } 1 - \mathrm{p} \end{cases} \qquad (3.7)$$

where Y_{ij} indicates the noisy image pixel, represents the total noise density of impulse noise, and represents the uncorrupted image pixel. At times, the salt noise and pepper noise may have different noise densities, and thus the total noise density will be $p = p_1 + p_2$. Random-valued impulse noise can take any gray-level value between 0 and 225. In this scenario, noise disperses randomly throughout the entire image, and the likelihood of encountering any gray level value as noise remains constant [20]. We can mathematically represent random-valued impulse noise as in equation (3.8).

$$Y_{ij} = \begin{cases} n_{ij} \; with \; probability \; p \\ x_{ij} \; with \; probability \; 1 - p \end{cases} \qquad (3.8)$$

3.2.4.6 Speckle Noise

Since random values multiplied by pixel values can model the noise, speckle noise is also known as multiplicative noise. It causes major problems in some radar applications. We can calculate its probability function using the following equation [18]:

$$\boldsymbol{F(g) = \frac{g^{\alpha-1}\, e^{\frac{-g}{a}}}{\alpha-1!\, a^{\alpha}}} \qquad (3.9)$$

3.2.5 Medical Image Denoising Techniques

Denoising medical images is a significant step in improving the performance of various medical imaging applications. It is one of the crucial preprocessing steps for medical image analysis, such as image segmentation and registration, tissue and organ delineation, computer-aided diagnosis (CAD), deblurring, and edge and image quality enhancement. For further medical imaging tasks like tissue segmentation, renal cyst volume measurement, skull stripping, and brain voxel classification, we must solve two problems in image quality assessment: noise corruption and noise reduction.

Common medical image denoising techniques include Gaussian filters, mean filters, median filters, Wiener filters, non-local means filters, guided filters, convolutional neural networks, wavelet transforms and variations, and denoising auto-encoding neural networks are some examples of common medical image denoising techniques.

In this section, we examine the primary groups of the previously mentioned techniques.

3.2.5.1 Adaptive Filter

It is commonly used to enhance or restore data by removing noise without significantly blurring the structure in the image.

$$f(x,y) = g(x,y) - (\sigma_{\eta}^2/\sigma_L^2)[g(x,y) - m_L] \qquad (3.10)$$

- *g(x,y) : pixel value at position (x,y)*
- σ_{η}^2 *: variance of overall noise*
- σ_L^2: *Local variance of local region*
- m_L : *Local Mean* **[18]**

3.2.5.2 Median Filter

Median filter is nonlinear filtering technique. It is widely used to remove salt and pepper noise and preserves the edges. The median filter calculates by first sorting all the pixel values from window into numerical order, and then replacing the pixel being considered with the middle (median) pixel value [18].

3.2.5.3 FIR Filter (finite impulse response)

FIR filter has symmetrical impulse response in the spatial domain (region of support centered about the origin) which offer a significant reduction in number of multiplications necessary for filter realization. Advantages of FIR filter is it's always stable [18].

3.2.5.4 Linear Filter

Linear filter in which the value of an output pixel is linear combination of the values of the pixels in the input pixel neighborhood. Disadvantages of this filter that noise become increase in it [18].

3.2.5.5 Non-Local Means Filter

NLM algorithm can be classified into 4 types, which boost up the SNR value, in this way to obtain the best edge preservation[23]. The algorithm can be given below:

Algorithm: Non-Local Means

Input 1: Image with random value impulsive noise

Output 1: NLM (Denoised Image)

For each pixel I, where i $\boldsymbol{\epsilon}$ $[\mathbf{1}, \boldsymbol{N}]$,

Do

For each pixel in $\boldsymbol{N_k}$, where $\boldsymbol{N_k}$Is the square patch around the center pixel $\boldsymbol{k}$,

Do

Evaluate, normalization constant $\boldsymbol{Z(i)} \sum_j e^{\frac{||v(N1°-v(N1)||^Z}{h^Z}}$

Where j refers to the $\boldsymbol{N_k}$ patches

Calculate, weights matrix $\boldsymbol{W(I,j)} \frac{1}{z(i)} e^{\frac{||v(N1°-v(N1)||^Z}{h^Z}}$

Done

Denoise pixel i ;$\boldsymbol{NL[v](i)} \sum_{j=1} w(i,j)v(j)$

Done

3.2.5.6 Wavelet transform

A time-frequency illustration of an analyzed signal is provided by a powerful tool named as wavelet transform that has the main advantage of its ability to obtain the information like time, location, and frequency of an image simultaneously, while the Fourier transform provides only the frequency information of the signal. Mathematically, a wavelet can be explained as a real-valued function ψ(t) which satisfy the conditions:

$$\int_{-\infty}^{+\infty} \Psi(t)dt = 0 \text{ and } \int_{-\infty}^{+\infty} |\Psi(t)^2|dt = 1 \qquad (3.11)$$

The probability density function of the wavelet transform is given by

$$P(g) = \frac{1}{\sqrt{s}}\, p(\frac{i-j}{s}) \qquad (3.12)$$

where$\frac{1}{\sqrt{s}}$, is an energy normalization [23].

3.2.5.7 Curvelet transform

Is a new type of multi-scale transform that relies upon wavelet. Curvelet transform is a type of run of typical multi-scale geometric investigation technique, which is produced based on the wavelet transform. It has the better peculiarity characteristics and it could keep more edge data of images.

By using denoising technique we obtain the better output from the original image. The restored image has less noise while comparing other images. Suppose *f(m,n)* an image is corrupted by additive noise, as given by:

$$g(m, n) = f(m, n) + \eta(m, n) \qquad (3.13)$$

where η(m,n) are autonomous similarly distributed Gaussian variable with zero mean and variance σ. Image denoising algorithms differ from simple threshold to difficult representation based methods. However simple threshold techniques can eliminate majority of the noise [23].

Algorithm: Curvelet

1.Apply the Forward Curve-let transform to the noisy image.

2.Threshold the Curve-let coefficients to remove some unimportant curve-let coefficients by using a threshold function in the curve-let domain.

3.Inverse Curve-let transform of the threshold coefficients to reconstruct a function.

3.2.5.8 Convolutional Networks (CNNs)

Convolutional Neural Networks (see § 3.3.8) in image denoising are advanced deep learning models designed to automatically reduce noise from images. They leverage multiple layers of convolutional filters to learn and identify patterns that differentiate noise from actual image content. These networks are trained on large datasets of noisy and clean image pairs, enabling them to effectively remove various types of noise while preserving important image details and features.

CNN-based denoising methods often outperform traditional denoising techniques, offering superior performance in maintaining the clarity and quality of the denoised images [23].

3.3 Artificial Neural Networks

3.3.1 Biological neuron

A neuron has a roughly spherical cell body called a soma. An extension on the cell body known as the axon, or nerve fibers, transmits the signals generated in the soma to other neurons. Another kind of extension around the cell body, like a bushy tree, is the dendrites, which are responsible for receiving the incoming signals generated by other neurons.

At the very end, the axon enlarges and forms terminal buttons. Special structures called synapses, which are the junctions transmitting signals from one neuron to another, place terminal buttons.

The signaling process is partly electrical (from the beginning of the dendrites until the end of the axon) and partly chemical (between every neuron and another, certain chemicals called neurotransmitters in the synapse are responsible for transmitting the signal)[24].

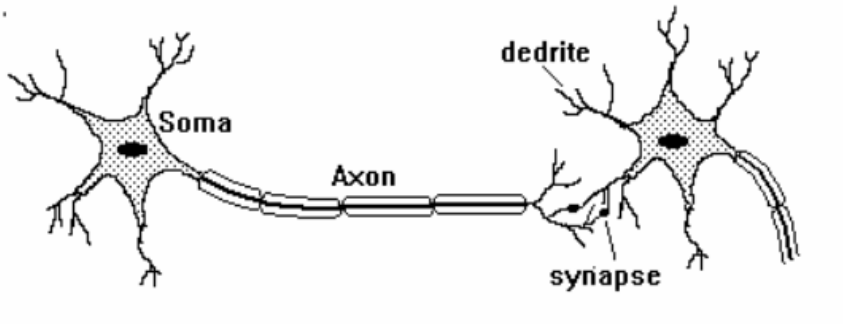

Figure 3- 1 Schematic image of a biological neuron [24].

3.3.2 Artificial neural networks

An artificial neural network (ANN) is a mathematical model that tries to simulate the structure and functionalities of biological neural networks. The basic building block of every artificial neural network is an artificial neuron, that is, a simple mathematical model (function). Such a model has three simple sets of rules: multiplication, summation, and

activation. At the entrance of an artificial neuron, the inputs are weighted, which means that every input value is multiplied by its individual weight.

In the middle section of an artificial neuron is a sum function that sums all weighted inputs and biases. At the exit of an artificial neuron, the sum of previously weighted inputs and bias is passing through an activation function that is also called a transfer function [25].

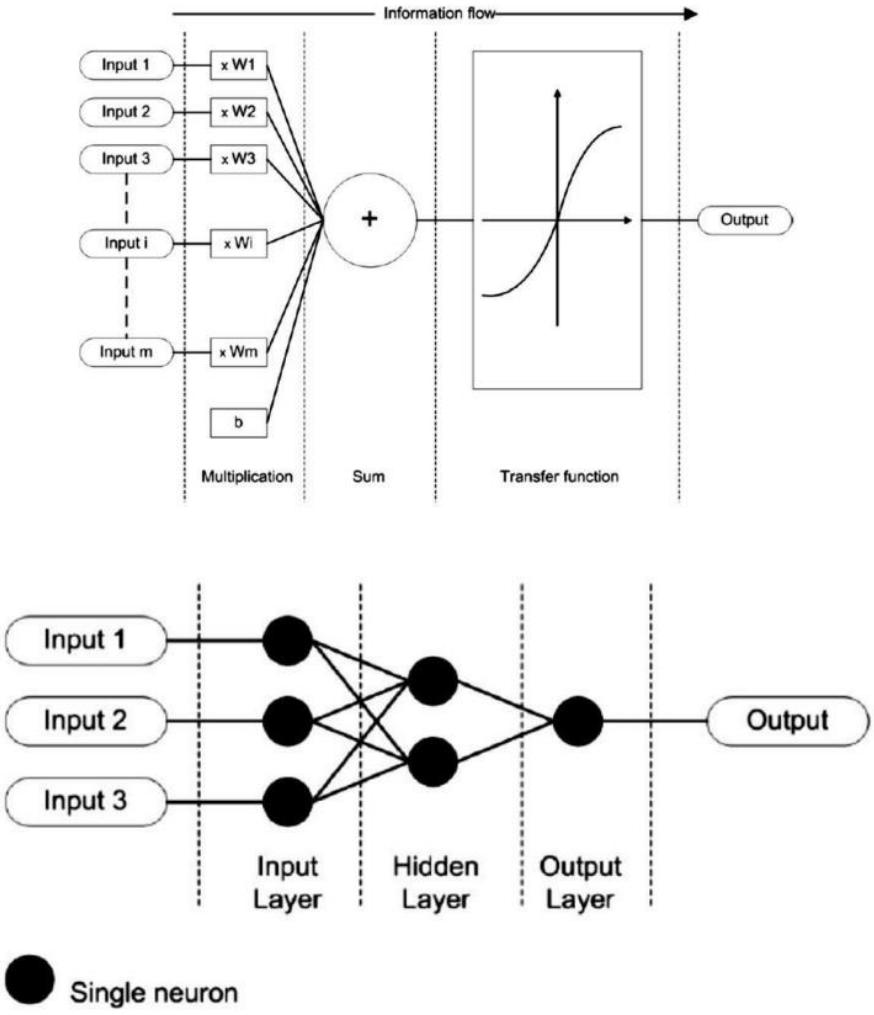

Figure 3- 2 Schematic diagram of a neural network [25].

3.3.3 Perceptron in ANN

A perceptron is a machine learning algorithm specifically designed for the purpose of binary categorization. A simplified version of a biological neuron, created to imitate the cognitive processes of the brain.

The perceptron is a computational model that receives many inputs, assigns separate weights to each input, adds them together, and then applies an activation function to get an output. Subsequently, you may utilize this output to categorize data into one of two distinct classifications.

The perceptron was initially introduced by Frank Rosenblatt in 1958. While the perceptron is regarded less complex in comparison to modern machine learning models, it served as the basis for the development of deep learning. The understanding of how it functions can be beneficial in comprehending more sophisticated models, making it a crucial asset in the toolkit of any machine learning engineer [24].

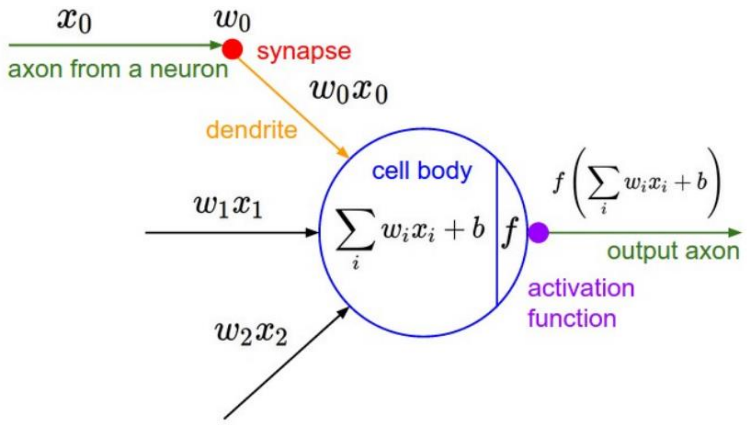

Figure 3- 3 A mathematical model of perceptron in a neural network [26]**.**

Basically, neural networks can be classified into two categories:

- Shallow neural network (in which there is only one single hidden layer between the input and output layer)[1]

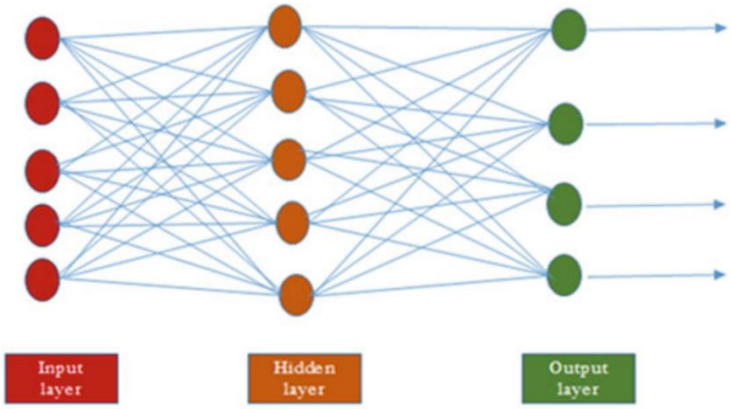

Figure 3- 4 Schematic diagram of a shallow neural network [1].

- Deep neural network (in which there are multiple hidden layers between the input and output layer and it is the most widely used network)[1]

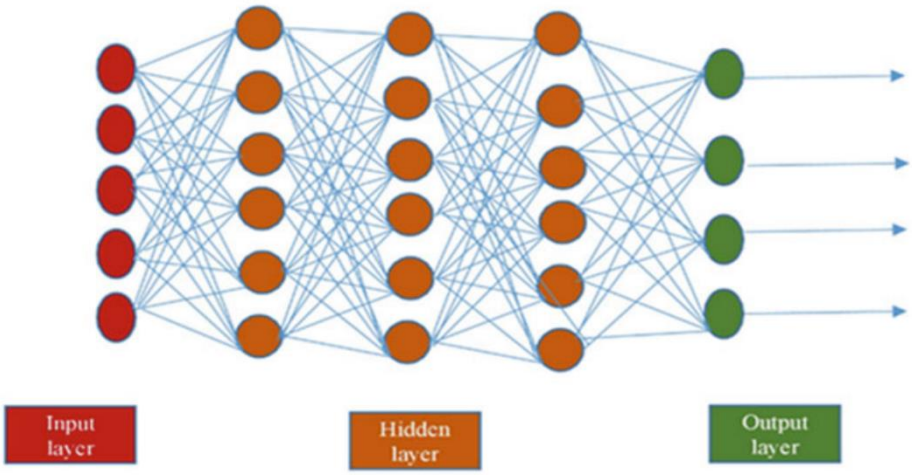

Figure 3- 5 Schematic diagram of a deep learning neural network [1].

3.3.3.1 Inputs and outputs

the network inputs represent the variables of the study and its outputs represent the solution to the problem for which the network was designed [27].

3.3.3.2 Weights

it is the basic element in the neural network, it expresses the relative strength of the variables included in the network, or the strength of the connection between the layers of the network [27].

3.3.3.3 Summation function

this function calculates the weight of all the inputs and sums them [27].

$$net = \sum_{i=1}^{n} x_i w_i \qquad (3.12)$$

3.3.3.4 Activation function

An activation function is a mathematical equation that determines the output of each element (perceptron or neuron) in the neural network. It may be defined as the extra force or effort applied over the input to obtain an exact output [27].

Most commonly used activation functions:

Sigmoid $\sigma(z) = \frac{1}{1+\exp(-z)}$ (3.13)

Tanh $tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$ (3.14)

ReLU (Rectified Linear Unit) $ReLU(z) = \max(0, z)$ (3.15)

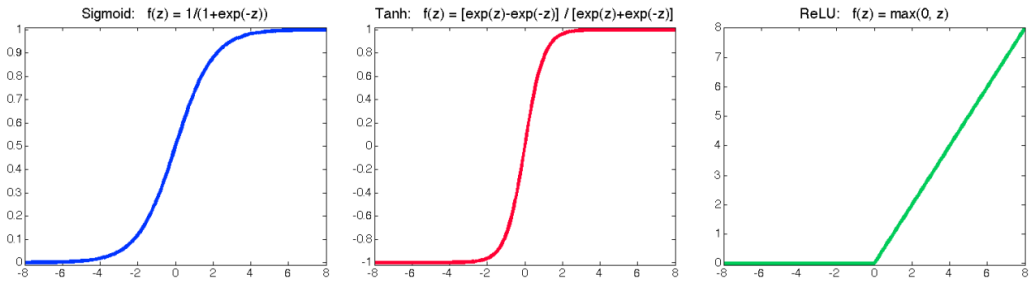

Figure 3- 6 Plot of most commonly used activation function [26].

3.3.4 The differences between biological and artificial neural networks

3.3.4.1 Size

Our brain contains about 86 billion neurons and more than 100 synapses (connections). The number of "neurons" in artificial networks is much less than that.

3.3.4.2 Signal transport and processing

The human brain works asynchronously, ANNs work synchronously.

3.3.4.3 Processing speed

Single biological neurons are slow, while standard neurons in ANNs are fast.

3.3.4.4 Topology

Biological neural networks have complicated topologies, while ANNs are often in a tree structure.

3.3.4.5 Speed

Certain biological neurons can fire around 200 times a second on average. Signals travel at different speeds depending on the type of nerve impulse, ranging from 0.61 m/s up to 119 m/s. Signal travel speeds also vary from person to person depending on their sex, age, height, temperature, medical condition, lack of sleep, etc.

The continuous, floating point number values of synaptic weights carry over information in artificial neurons. There are no refractory periods for artificial neural networks (periods while it is impossible to send another action potential due to the sodium channels being locked shut), and artificial neurons do not experience “fatigue”: they are functions that can be calculated as many times and as fast as the computer architecture would allow.

3.3.4.6 Fault-tolerance

Biological neuron networks, due to their topology, are also fault-tolerant. Artificial neural networks are not modeled for fault tolerance or self-regeneration (similarly to fatigue, these ideas are not applicable to matrix operations), though recovery is possible by saving the current state (weight values) of the model and continuing the training from that saved state.

3.3.4.7 Power consumption

The brain consumes about 20% of all the human body’s energy; despite its large size, an adult brain operates on about 20 watts (barely enough to dimly light a bulb), making it extremely efficient. Considering that humans can continue to function for a considerable amount of time with just a small amount of vitamin-rich lemon juice and beef tallow, this efficiency is truly remarkable.

For a benchmark, a single Nvidia GeForce Titan X GPU runs on 250 watts alone and requires a power supply. Our machines are way less efficient than biological systems. When used, computers also generate a lot of heat, with consumer GPUs operating safely between 50 and 80 °C rather than 36.5 and 37.5 °C.

3.3.4.8 Learning

We still do not understand how brains learn or how redundant connections store and recall information. Learning builds upon previously stored information in the brain. Repetition and sleep deepen our knowledge, and once we master tasks that once required focus, we can execute them automatically.

On the other hand, artificial neural networks have a predefined model that does not allow for the addition or removal of additional neurons or connections. Only the connections' weights (and biases representing thresholds) can be changed during training.

The networks begin with random weight values and gradually try to reach a point where further weight changes will no longer improve performance. Biological networks usually don't stop or start learning. ANNs have different fitting (train) and prediction (evaluate) phases.

3.3.4.9 Field of application

ANNs are specialized. They can perform one task. They might be perfect at playing chess, but they fail at playing go (or vice versa). Biological neural networks can learn completely new tasks.

3.3.4.10 Training algorithm

ANNs often use Gradient Descent algorithm for learning. Human brains use something different (but we don't know what?) [25].

3.3.5 Types of Artificial Neural Networks

There are two important types of ANNs:

3.3.5.1 Feed Forward Neural Network

In feedforward ANNs, the information flow is only in one direction. That is, data flows from the input layer to the hidden layer, and then to the output layer. There are no

feedback loops. We commonly use these neural networks for supervised learning tasks such as classification and image recognition. When the data is not in consecutive order, we use them. Feedforward networks are comparable to convolutional neural networks (CNNs).

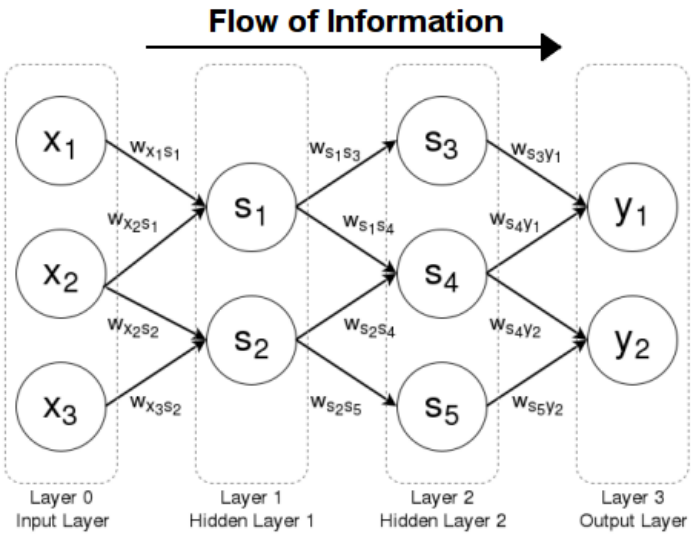

Figure 3- 7 Feed Forward Neural Network.

3.3.5.2 Feedback Neural Network

The feedback loops are an element of the feedback ANNs. Recurrent neural networks, among others, primarily serve memory retention purposes. Situations where the data is sequential or time-dependent best suit these networks. The feedback loops define recurrent neural networks (RNNs).

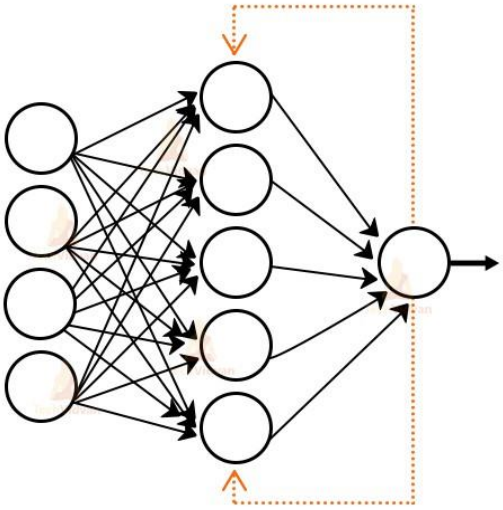

Figure 3- 8 Feedback Neural Network (Recurrent NN).

3.3.6 ANN Learning Techniques

3.3.6.1 Supervised Learning

In this learning method, the user trains the model with labeled data. It signifies that the user has already tagged some data with the appropriate responses. We refer to learning that occurs in the presence of a supervisor as supervised learning.

3.3.6.2 Unsupervised Learning

For this learning, the model does not require supervision. Typically, it handles unlabeled data. The user gives permission for the model to categorize the data on its own. It organizes the data based on similarities and patterns without requiring any prior data training.

3.3.6.3 Reinforcement Learning

The output value is unknown in this case, but the network provides feedback on whether it is correct or incorrect. It's referred to as "Semi-Supervised Learning".

3.3.7 Artificial neural network applications

The following are some important applications of artificial neural networks:

3.3.7.1 Speech recognition

heavily relies on artificial neural networks (ANNs). Earlier speech recognition models used statistical models, such as hidden Markov models. With the introduction of deep learning, several forms of neural networks have become the only way to acquire a precise classification.

3.3.7.2 Handwritten characters recognition

Neural networks have trained to recognize handwritten characters, which can take the form of letters or digits.

3.3.7.3 Signature Classification

When developing these authentication systems, we use artificial neural networks to recognize signatures and categorize them according to the person's class. Furthermore, neural networks can determine whether or not a signature is genuine.

3.3.7.4 Medical:

It provides detailed results by detecting cancer cells and analyzing MRI images.

3.3.8 What is Convolutional Neural Network

A convolutional neural network (CNN), also known as ConvNet, is a specialized type of deep learning algorithm mainly designed for tasks that necessitate object recognition, including image classification, detection, and segmentation. Various practical scenarios, including autonomous vehicles and security camera systems, employ CNNs.

Six layers generally make up a CNN pipeline (or architecture): the input layer, the convolutional layer(s), the pooling layer(s), the flattening layer, the fully connected layer(s), and the output layer [28].

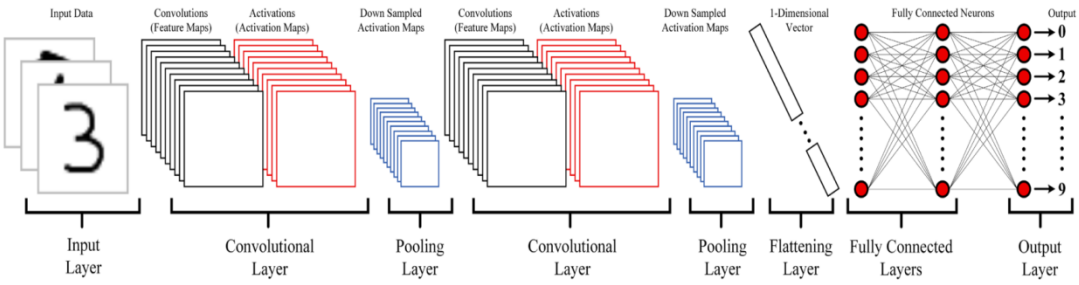

Figure 3- 9 A generic CNN pipeline with 6 types of layers: 1 input, 2 convolutional, 2 pooling, 1 flattening, 2 fully connected, and 1 output [28].

- Input Layer

The input layer is the first layer in CNNs, concerned with receiving the raw input data, e.g., images, videos, audios, texts, etc. It represents images as tensors (multi-dimensional matrices), with each element representing the value of a pixel's color intensity. In CNNs, the input layer is not learnable, and there are no neurons present.

As per the network architecture, the input layer does batch normalization (which makes sure that the data's scale and distribution are consistent to speed up training), pre-processing (which includes things like cropping, resizing, and data augmentation to make sure that the input sizes are consistent and to increase the data's variability), and forward propagation (which sends the data to the next part of the network to be processed) [28].

- **Convolutional layer**

Convolution, from a simplistic point of view, is the application of a mathematical filter to an image. From a more technical point of view, it is a matter of dragging a matrix over an image, and for each pixel, use the sum of the multiplication of that pixel by the value of the matrix. This technique allows us to find parts of the image that might be of interest to us. Take the figure below **M** as an example of an image and **F** as an example of a filter [29].

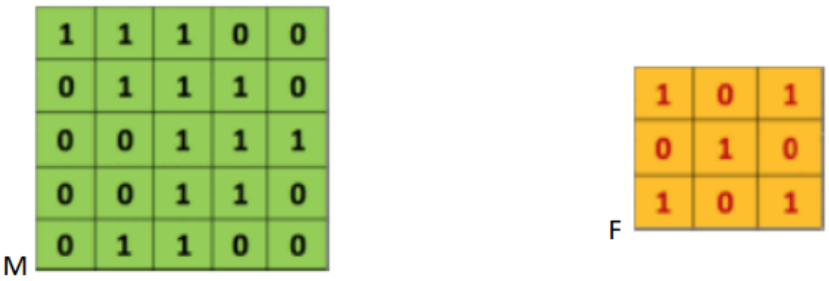

Figure 3- 10 Image and a filter [29].

Apply the filter to the image: in the M image matrix, we can see that each value of the pixels of the tile image (the orange boxes) is multiplied by each corresponding filter value (1x1, 1x0, 1x1....). Then add all these values to get a single value '4' that will be part of a new convoluted image [29].

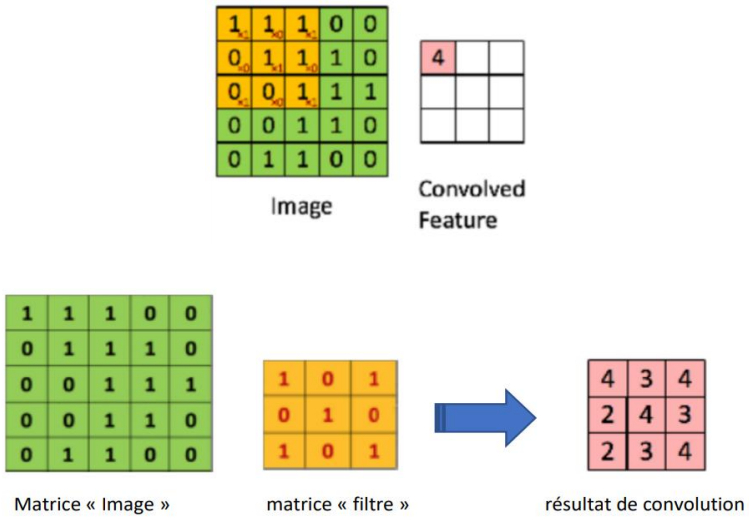

Figure 3- 11 Applying The Filter To The Image [29].

- Poolling

Is another operation similar to convolutions performed on an image? Sometimes, applying a convolution to an image doesn't reduce it enough, so we can further utilize pooling operations. Pooling is simpler than convolution because we do not have a kernel full of elements. Instead, we have a filter that we slide across the image, and for every position of this filter, we simply take the minimum, maximum, mean, or median of the set of pixels inside the image that fall underneath this filter. It is most typical to use a max pool with a 2x2 filter and a 2x2 stride, such that there is no overlap between filters. Max Pool has appeared to work better than the average pool [30].

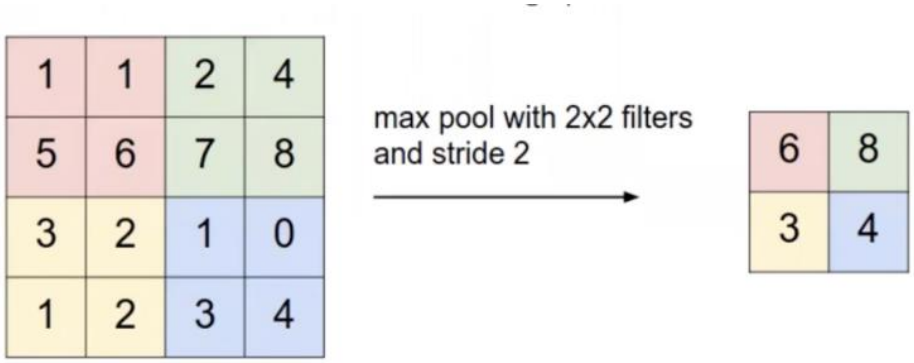

Figure 3- 12 A max pooling in action. You can think of each colored region as a position of the 2x2 filter [30].

- Flattening Layer

The role of a flattening layer is to "flatten" the multi-dimensional feature maps or activation maps received from the previous convolutional or pooling layers. The flattening layer is located before the fully connected layers. The flattening layer transforms the multi-dimensional data into a 1-dimensional vector (refer to Figure 3.13) and then transfers it to the fully connected layer, specifically designed to receive 1-dimensional data as input [28].

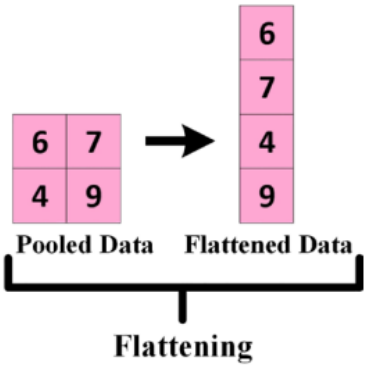

Figure 3- 13 how flattening layer function [28].

- Fully Connected (FC) Layer

Typically, the final segment (or layers) of any CNN architecture comprises fully connected layers, with each neuron within a layer establishing a connection with its antecedent neuron. The CNN architecture uses the last layer of fully connected layers as its output layer. Fully-connected layers are a type of feed-forward artificial neural network (ANN), and they follow the principles of traditional multi-layer perceptron neural networks (MLP). The FC layers receive input in the form of a set of metrics (feature maps) from the final convolutional or pooling layer, flatten these metrics into a vector, and feed this vector into the FC layer to produce the final CNN output [31], as illustrated in Figure 3.14.

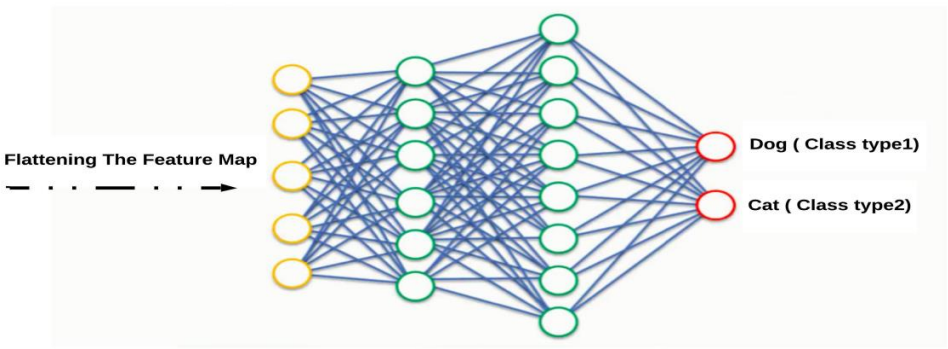

Figure 3- 14 The architecture of Fully Connected Layers [31].

- Output layer

Is the final layer in a CNN, placed after the FC layer (see Figure 3.14). It generates the model's final output. It is an essential component of CNNs, and its function is to make decisions based on the network's extracted and learned features. The preceding FC layer fully connects all neurons in the output layer. The number of neurons in the output layer of CNNs is equivalent to the number of classes for classification tasks and the number of regression outputs for regression tasks. Each neuron represents a class, and the network predicts the corresponding class based on the neuron's highest activation [28].

3.3.9 Autoencoders

3.3.9.1 What Are Autoencoders?

An autoencoder (AE) is an artificial neural network used for unsupervised learning tasks (i.e., no class labels or labeled data) such as dimensionality reduction, feature extraction, and data compression. They seek to:

- Accept an input set of data (i.e., the input)
- Internally compress the input data into a latent space representation (i.e., a single vector that compresses and quantifies the input)
- Reconstruct the input data from this latent representation (i.e., the output)

An autoencoder consists of the following two primary components:

- **Encoder:** The encoder compresses input data into a lower-dimensional representation known as the latent space or code. This latent space, often called embedding, aims to retain as much information as possible, allowing the decoder to reconstruct the data with high precision. If we denote our input data as x and the encoder as E, then the output latent space representation, s would be.

$$s = E(x) \qquad (3.16)$$

- **Decoder:** The decoder reconstructs the original input data by accepting the latent space representation s. If we denote the decoder function as D and the output of the detector as o, then we can represent the decoder as o=D(s).

$$o = D(s) \qquad (3.17)$$

Both encoder and decoder are typically composed of one or more layers, which can be fully connected, convolutional, or recurrent, depending on the input data's nature and the autoencoder's architecture [32].

By using our mathematical notation, the entire training process of the autoencoder can be written as shown in the following figure:

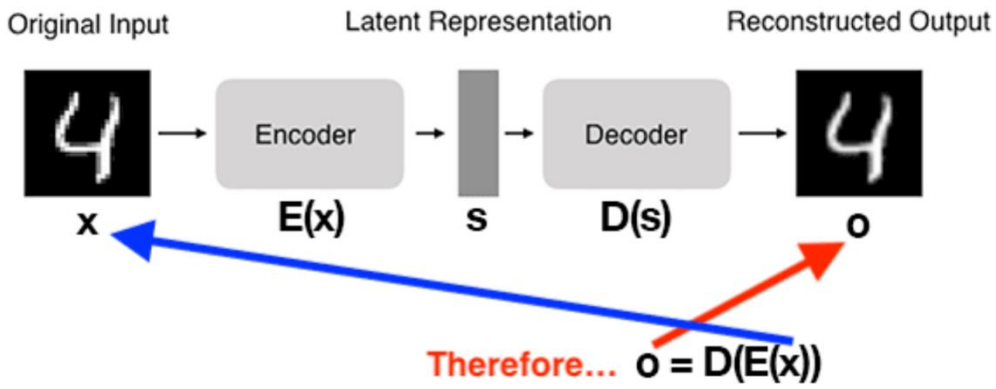

Figure 3- 15 Demonstrates The Basic Architecture Of An Autoencoder [32].

3.3.9.2 Types of Autoencoder

- Vanilla Autoencoder

Figure 3.16 shows the simplest form of an autoencoder, consisting of one or more fully connected layers for both the encoder and decoder. It works well for simple data but may struggle with complex patterns [32].

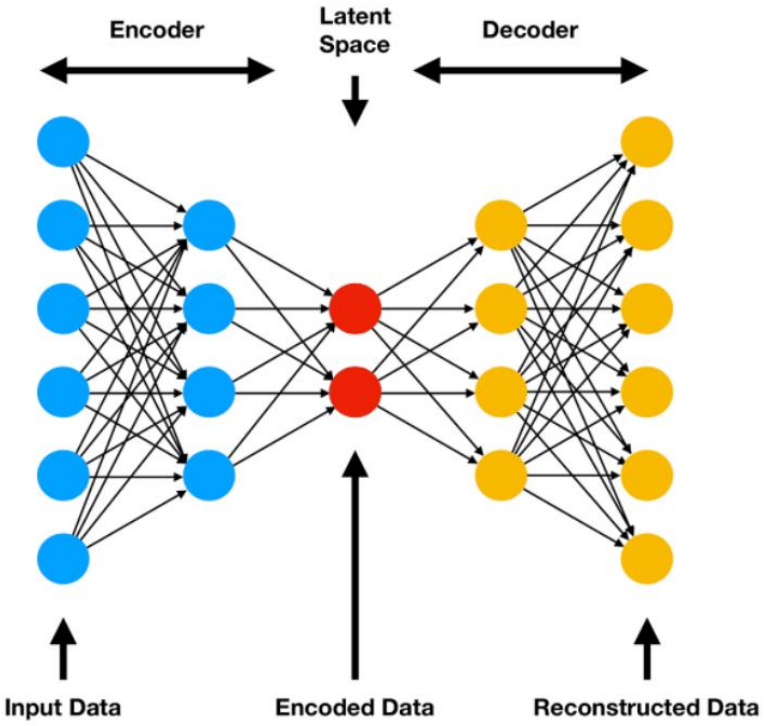

Figure 3- 16 Architecture of fully connected autoencoders [32].

- Convolutional Autoencoder (CAE)

Utilizes convolutional layers in both the encoder and decoder, making it suitable for handling image data. By exploiting the spatial information in images, CAEs can capture complex patterns and structures more effectively than vanilla autoencoders and accomplish tasks such as image segmentation[32], as shown in Figure 3.17.

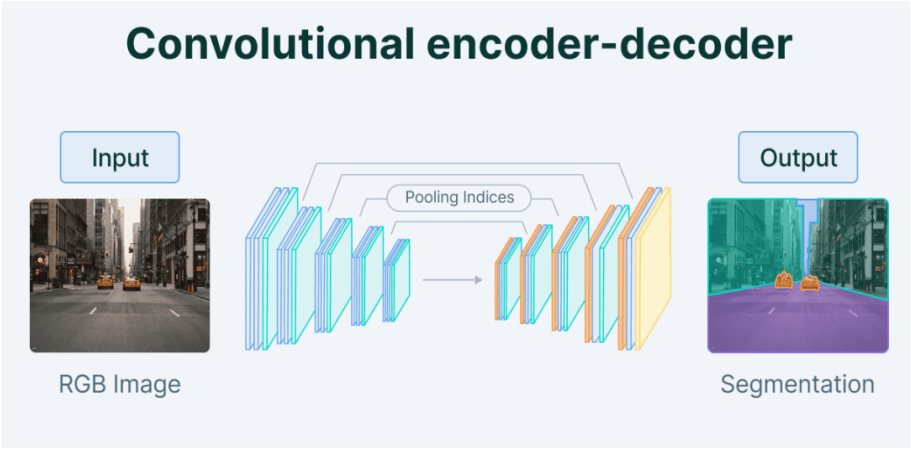

Figure 3- 17 Architecture of Convolutional Autoencoder for Image Segmentation [32].

- Denoising Autoencoder

This autoencoder is designed to remove noise from corrupted input data, as shown in Figure 3.18. During training, the input data is intentionally corrupted by adding noise, while the target remains the original, uncorrupted data. The autoencoder learns to reconstruct the clean data from the noisy input, making it useful for image denoising and data preprocessing tasks [32].

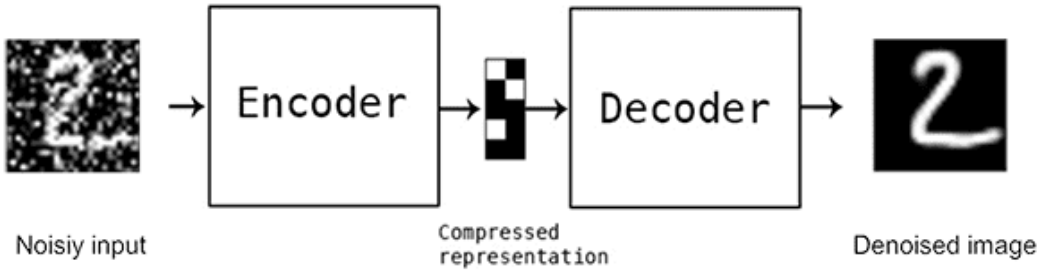

Figure 3- 18 A denoising autoencoder processes a noisy image, generating a clean image on the output side [32].

- Sparse Autoencoder

This type of autoencoder enforces sparsity in the latent space representation by adding a sparsity constraint to the loss function (as shown in Figure 3.19). This restriction makes the autoencoder represent the input data with a small group of active neurons in the latent space. This makes feature extraction more efficient and reliable [32].

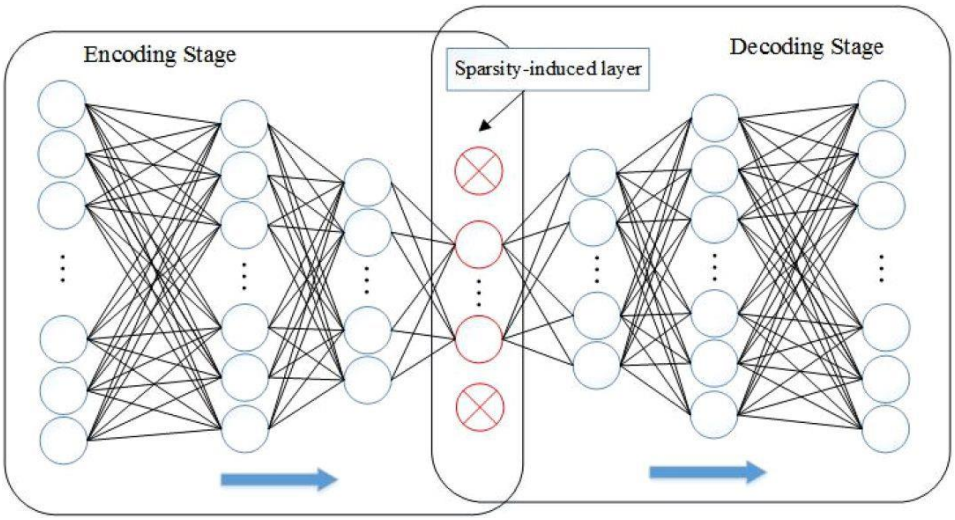

Figure 3- 19 The topology of Sparse Autoencoder [32].

- Variational Autoencoder (VAE)

Figure 3.20 shows a generative model that introduces a probabilistic layer in the latent space, allowing for sampling and generation of new data. VAEs can generate new samples from the learned latent distribution, making them ideal for image generation and style transfer tasks [32].

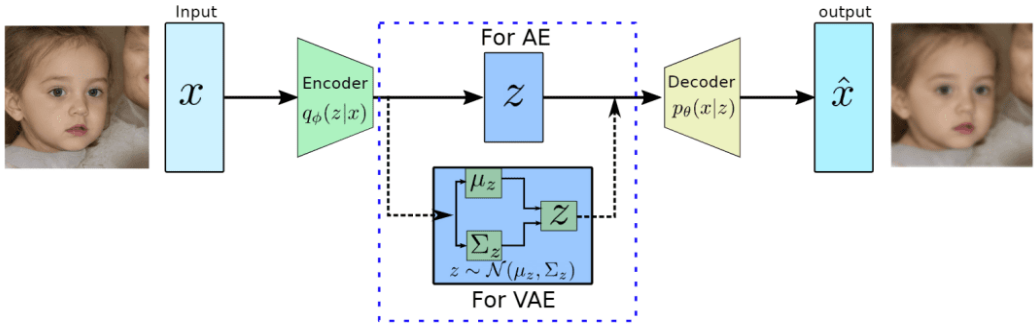

Figure 3- 20 Architecture of variational autoencoder [32].

- Sequence-to-Sequence Autoencoder (SSAE)

Also known as a Recurrent Autoencoder, this type of autoencoder utilizes recurrent neural network (RNN) layers (e.g., long short-term memory (LSTM) or gated recurrent unit (GRU)) in both the encoder and decoder shown in Figure 3.21. This architecture is well-suited for handling sequential data (e.g., time series or natural language processing tasks) [33].

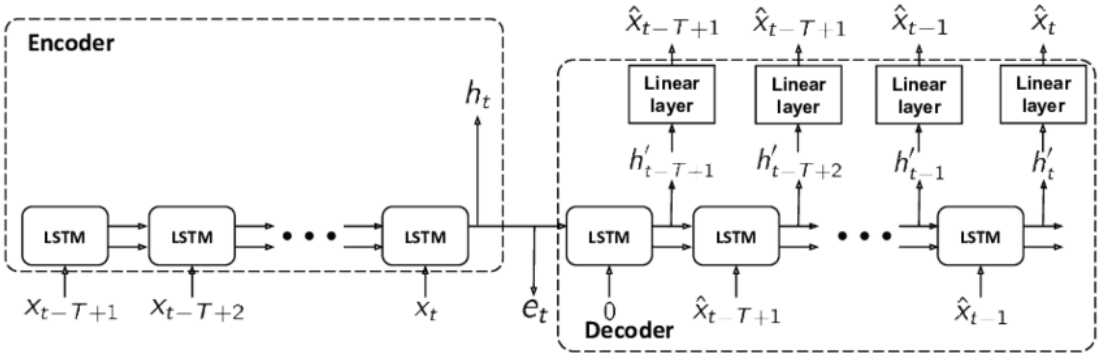

Figure 3- 21 Sequence-to-Sequence Autoencoder [33].

3.3.9.3 Applications of Autoencoders?

Autoencoders have a wide range of applications across various domains, including:

- Dimensionality Reduction

Autoencoders can reduce the dimensionality of input data by learning a compact and efficient representation in the latent space. This can be helpful for visualization, data compression, and speeding up other machine learning algorithms.

- Feature Learning

Autoencoders can learn meaningful features from input data, which can be used for downstream machine learning tasks like classification, clustering, or regression.

- Anomaly Detection

By training an autoencoder on normal data instances, it can learn to reconstruct those instances with low error. When presented with an anomalous data point, the autoencoder will likely have a higher reconstruction error, which can be used to identify outliers or anomalies.

- Denoising Images

Autoencoders can be trained to reconstruct clean input data from noisy versions. The denoising autoencoder learns to remove the noise and produce a clean version of the input data.

- Image Inpainting

As shown in Figure 3.22 , autoencoders can fill in missing or corrupted parts of an image by learning the underlying structure and patterns in the data [32].

Figure 3- 22 Face completion by filling in the missing pixels [32].

- Generative Modeling

Variational autoencoders (VAEs) and other generative variants can generate new, realistic data samples by learning the data distribution during training. This can be useful for data augmentation or creative applications.

- Recommender Systems

Autoencoders can be used to learn latent representations of users and items in a recommender system, which can then predict user preferences and make personalized recommendations.

- Sequence-to-Sequence Learning

Autoencoders can be used for sequence-to-sequence tasks, such as machine translation or text summarization, by adapting their architecture to handle sequential input and output data [32].

- Image Segmentation

Autoencoders are commonly utilized in semantic segmentation as well. A notable example is SegNet (Figure 3.23), a model designed for pixel-wise, multi-class segmentation on urban road scene datasets. This model was created by researchers from the University of Cambridge's Computer Vision Group [34].

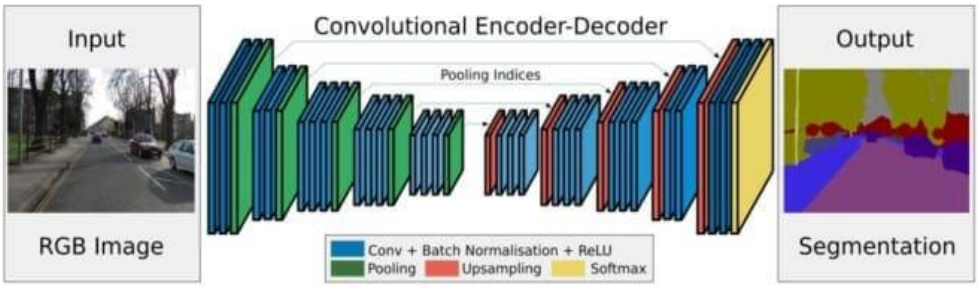

Figure 3- 23 An illustration of the fully convolutional SegNet architecture [34].

These are just a few examples of the many possible applications of autoencoders. Their versatility and adaptability make them an important tool in the machine learning toolbox.

3.4 Related studies

In [3], *Gondara et al*. have presented empirical evidence that stacked denoising autoencoders built using convolutional layers work well for small sample sizes, which are typical of medical image databases. This contradicts the commonly held belief that deep architecture-based models require very large training datasets for optimal performance. These methods have demonstrated the ability to recover signals even in the presence of extremely high noise levels, a threshold beyond which most other denoising methods would fail.

Xie et al. have presented a novel approach to image denoising and blind inpainting that combines sparse coding and pre-trained deep neural networks with denoising auto-encoders [35]. It has proposed a new training scheme for DA that makes it possible to denoise and inpaint images within a unified framework. In the tests, their method worked about as well as the standard linear sparse coding algorithm at getting rid of white Gaussian noise that was added on top of other noise. Moreover, their non-linear approach successfully tackles the much harder problem of blind inpainting complex patterns, which, to the best of their knowledge, has not been addressed before. It has also been demonstrated that the proposed training scheme is capable of improving DA's performance in unsupervised feature learning tasks.

Qi et al. have proposed a contrastive-center loss for deep neural networks [36]. The contrastive-center loss takes into account both how compact the classes are within themselves and how separate the classes are from each other. It does this by penalizing the differences in the distances of training samples from their class centers and the sum of the distances of training samples from their non-class centers. The contrastive-center loss has a very clear intuition and geometric interpretation, which is more appealing. The experimental results on several benchmark datasets prove the effectiveness of the proposed contrastive-center loss.

3.5 Conclusion

In summary, Chapter 2 focuses on medical image denoising, a crucial process for improving image quality by removing noise such as salt and pepper, Gaussian, and other types. Noise, caused by various factors during image acquisition and transmission, degrades image quality and complicates the extraction of meaningful information. The chapter

discusses various noise models, including additive and multiplicative noise, as well as Gaussian, salt and pepper, Poisson, impulse, and speckle noise.

The chapter also explores several denoising techniques, including adaptive filters, median filters, FIR filters, and linear filters. It delves into advanced methods like wavelet and curvelet transforms and highlights the role of convolutional neural networks (CNNs) and autoencoders in denoising. With their specialized architecture, CNNs excel in image recognition tasks, while autoencoders, including variations like denoising autoencoders and variational autoencoders, serve unsupervised learning tasks like dimensionality reduction and data compression. Overall, the chapter underscores the ongoing challenges and advancements in medical image denoising, aiming to achieve high-quality, noise-free images that preserve essential details and structures.

Chapter 4 Methodology

4.1 Introduction

Our primary goal is to eliminate noise from images. Understanding the image's context is necessary to ensure the elimination of noise only. We extract the context of an image by passing it through the CAE (convolutional autoencoder) model, a convolutional neural network whose primary task is to capture the noise patterns in an image from the feature map at each layer. This model is used to identify noise artifacts in the image.

In this part, we delve into the methodology employed for developing and training a Convolutional Neural Network-based Denoising Autoencoder (CDAE). The aim of this model is to effectively reduce noise in grayscale images while preserving essential features.

The methodology encompasses the detailed architecture of the autoencoder, including its input, encoder, bottleneck, decoder, and output layers. Furthermore, we discuss the activation functions, loss function, optimization techniques, and training procedures integral to the model's development. This comprehensive approach ensures a robust framework for image denoising, optimizing the model's performance through meticulous design and training strategies.

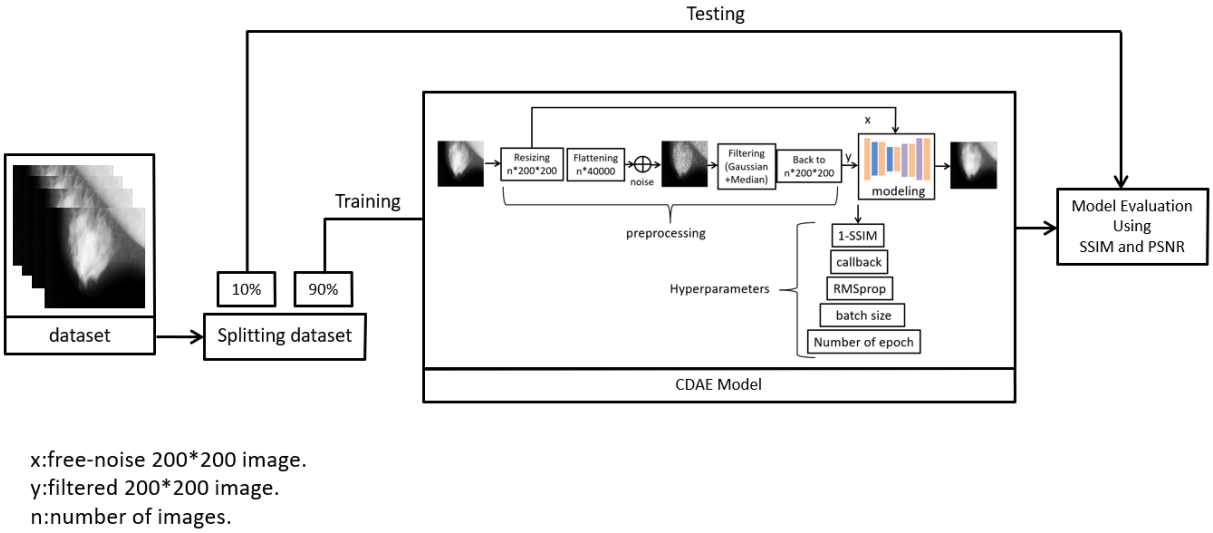

Figure 4- 1 Block diagram of the proposed model architecture.

4.2 Convolutional Autoencoder architecture

The basic architecture of a convolutional autoencoder (CAE) is shown in Figure 4.2. Convolutional autoencoders utilize the classic autoencoder design, with convolutional encoding and decoding layers. Convolutional autoencoders are more suitable for image processing compared to conventional autoencoders because they fully employ the capabilities of convolutional neural networks to exploit the structure of images.

Convolutional autoencoders utilize weight sharing across all input locations to maintain local spatiality. The i^{th} feature map is represented as follows:

$$h^i = s(x * W^i + b^i) \tag{4.1}$$

In the above scenario, bias is being transmitted over the entire map. The symbol "*" represents a convolution operation in 2D, and "s" specifies an activation function. A single bias is assigned to each latent map, and the reconstruction is derived accordingly:

$$y = s(\sum_{i \in H} h^i * \tilde{W}^i + c) \tag{4.2}$$

The variable "c" represents the bias per input channel, "H" represents a collection of latent feature mappings, and "$\tilde{W}$" represents a flip operation across both weight dimensions.

Back propagation is employed to calculate the gradient of the error function in relation to the parameters [3].

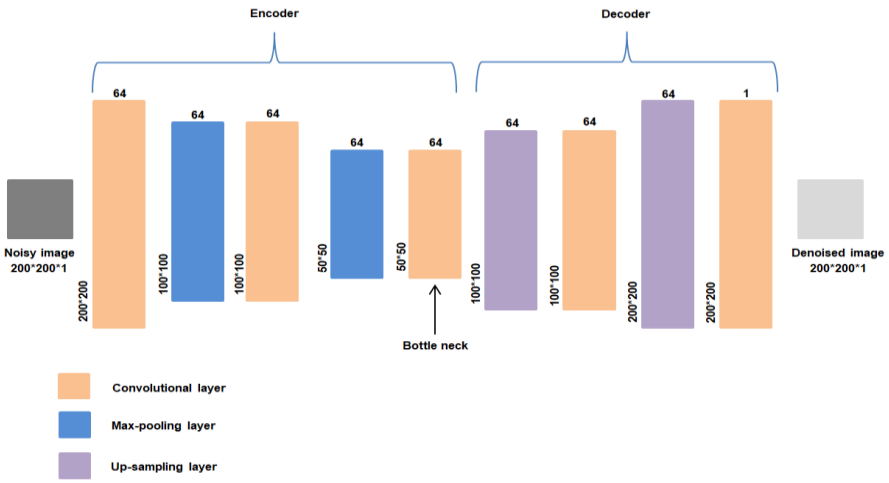

Figure 4- 2 Architecture of the proposed CDAE.

4.2.1 Input Layer

In the input layer images are in specified shape (200,200,1), which means that the images are grayscale (1 channel) and 200×200 pixels.

4.2.2 Encoder layers

4.2.2.1 Convolutional Layer

Convolutional Layer is the fundamental layer of CNNs and it has a set of learnable "filters" (also called kernels). These filters are used to perform convolutions with images to extract features. Convolution is a linear operation in which one matrix (filter) is moved over another (image) [28].

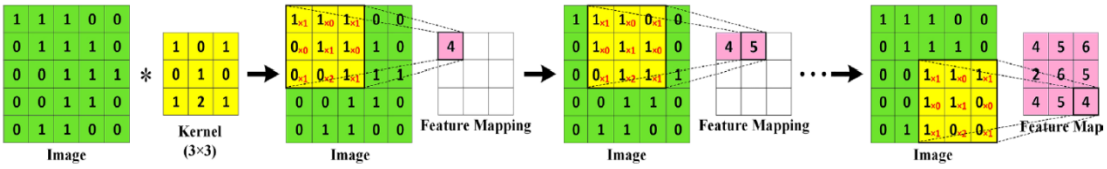

Figure 4- 3 Convolution of a 5×5 image and a 3×3 kernel with stride =1. Observe how a feature map is generated step by step [28].

This layer has a 64 filter, each with a size of 3×3pixels, the activation function is ReLU (rectifier linear unit).

"padding='same'": The process of adding pixels of "zero" values at the borders of the input images or the feature maps (before performing convolutions) is called "zero-padding". Padding size refers to the width appended at each edge. Larger padding sizes result in spatially larger feature maps, somewhat increasing the computational cost [28].

Advantages of zero-padding are preservation of spatial dimensions and capturing features at the edges (avoiding information loss).

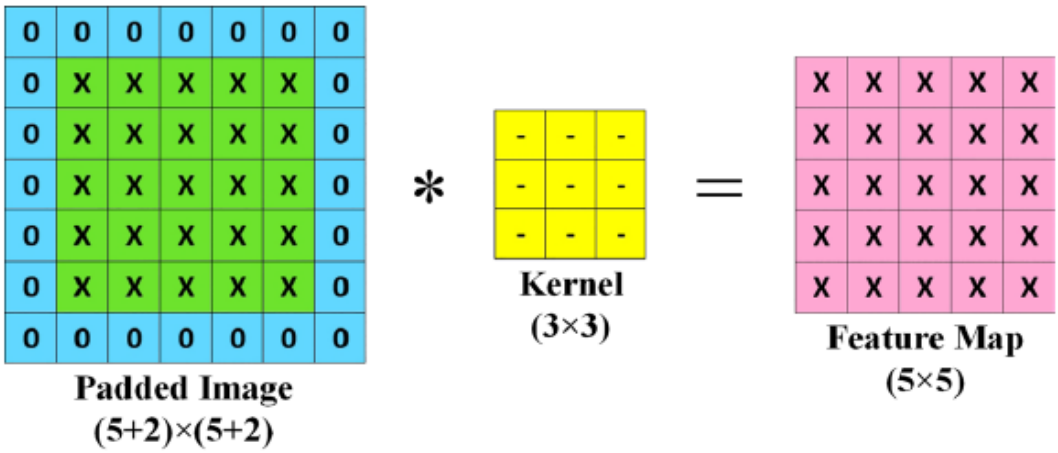

Figure 4- 4 Convolution of zero-padded image with a kernel. Observe how the dimensions of the input remain preserved as compared to no zero-padding [28].

4.2.2.2. Pooling Layer

Pooling Layer is used to down-sample (sub-sample) the data inside CNNs and it typically comes after the convolutional layer(s). A network can have multiple pooling layers.

The main purpose of a pooling layer is to reduce the computational load of the network. It decreases spatial dimensions of the feature maps by extracting important information from them and discarding the rest [28].

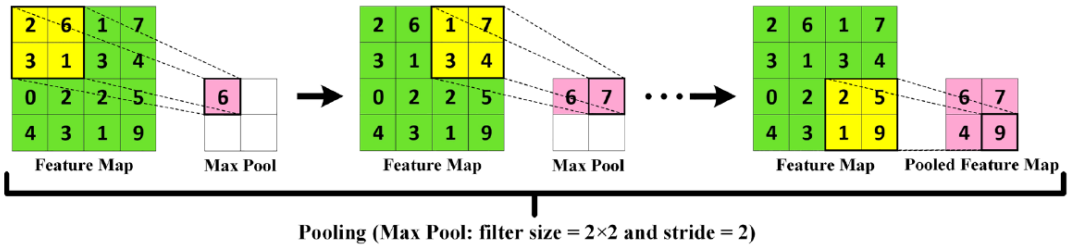

Figure 4- 5 Demonstration of the max pooling operation [28].

This layer performs with a pool size of 2×2 pixels, which reduces the spatial dimension of the feature map by taking the maximum value in each 2×2pixels.

4.2.3 Bottleneck Layer

This layer contain the most compressed representation of the input data. The encoding is generated in the lower-dimensional hidden layer. The bottleneck layer contains a reduced number of nodes, and this number also determines the dimension of the input's encoding.

4.2.4 Decoder Layers

4.2.4.1. Up-sampling Layer

This layer performs an up-sampling with a scale of 2×2 pixels, which increases the spatial dimensions of the feature map by replicating each pixel value.

4.2.4.2. Convolutional Layer

This layer has a 64 filter, each with a size of 3×3pixels, associate with an activation function ReLU.

"padding='same'", the padding ensure that the spatial dimension of this output is the same as the input, and the convolutional layers in the decoder are for reconstructing the feature map with a higher spatial dimension.

4.2.5 Output Layer

The output layer is a convolutional layer with 64 filters of 3×3 pixels size, and "padding='same'", and a sigmoid activation function.

The convolutional denoising autoencoder (CDAE) utilized a straightforward architecture, as depicted in Figure 4.6.

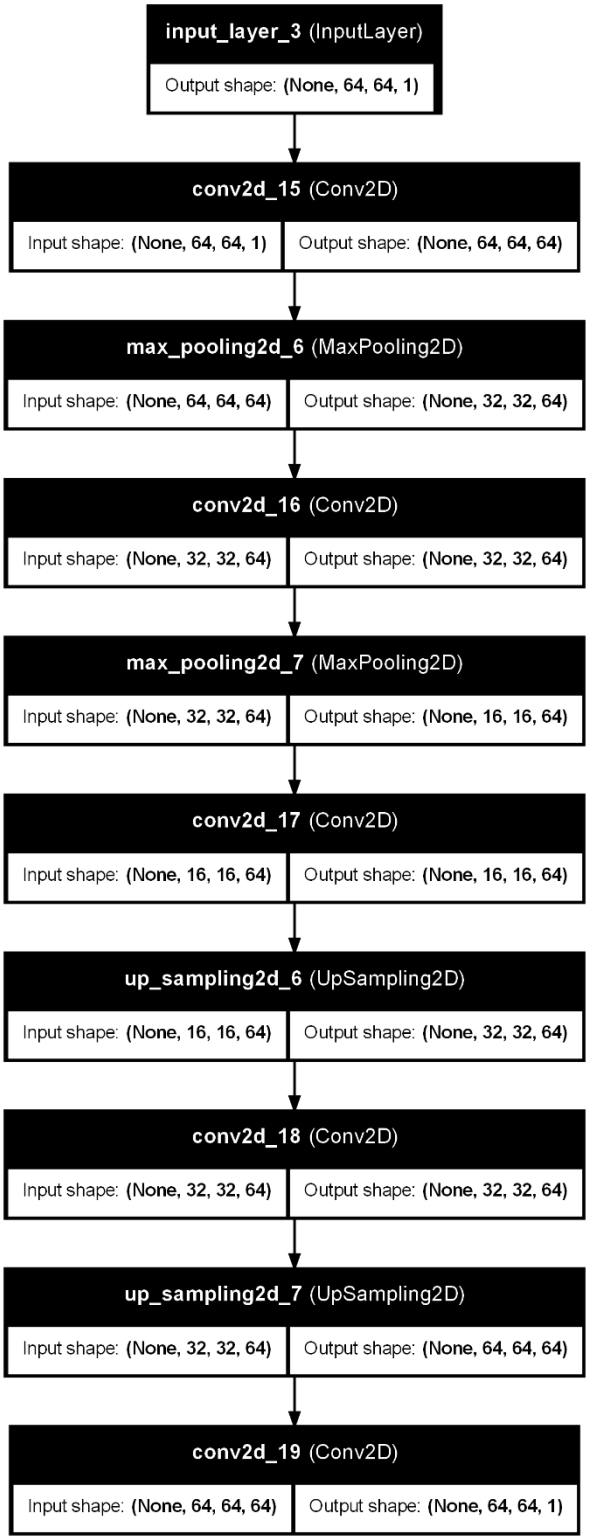

Figure 4- 6 Architecture of the CDAE used [3].

4.3 Autoencoder hyperparameters

4.3.1 Activation function

The main task of any activation function in any neural network-based model is to map the input to the output.

-Sigmoid: The sigmoid activation function takes real numbers as its input and bind the output in the range of [0,1]. data shall be mapped to values between 0 and 1 to validate with the matrix multiplication and neural network calculations [31].

$$f(x)sigm = \frac{1}{1 + e^{-x}} \tag{4.3}$$

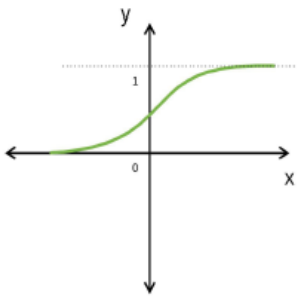

Figure 4- 7 Sigmoid activation function [31].

-ReLU: The Rectifier Linear Unit (ReLU) is the most commonly used activation function in Convolutional Neural Networks. It is used to convert all the input values to positive numbers. The advantage of ReLU is that it requires very minimal computation load compared to others [31]. The mathematical representation of ReLU is given by:

$$f(x)ReLU = \max(0, x) \tag{4.4}$$

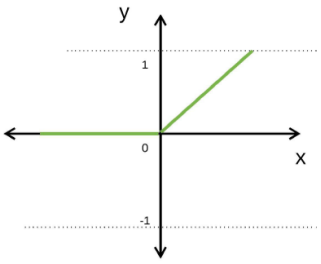

Figure 4- 8 ReLU activation function [31]

4.3.2 Loss function

By employing a loss function, we compute the prediction error produced by the CNN model on the training data in the output layer of the CDAE. The prediction error measures the discrepancy between the network's forecast and the real output, and the CNN model improves this error while it learns.

The loss function utilizes two parameters to compute the discrepancy: the initial parameter represents the predicted output of the CNN model (referred to as the prediction), while the second parameter represents the true output, also known as the ground truth.

we used a custom loss function as:

$$L = 1 - \text{SSIM} \tag{4.5}$$

which computes the dissimilarities between the original and the denoised image, with a maximum value of 1. the less loss value gets, means better denoising performance [3].

4.3.3 Optimization technique

An optimizer is a crucial element that fine-tunes a neural network’s parameters during training. Its primary role is to minimize the model’s error or loss function, enhancing performance. Various optimization algorithms, known as optimizers, employ distinct strategies to converge towards optimal parameter values for improved predictions efficiently. In our case, we used RMS prop optimizer for training process.

4.3.4 Callback

It allows the training process to automatically stop when the model performance is not improving, by monitoring the validation loss between the original free-noise image and the denoised image. The training will stop if this value of the validation loss does not improve for 5 consecutive epochs.

4.3.5 Batch size

The batch size refers to the number of images that are inputted into the model simultaneously. “Batch = 10”, which means after 10 samples the model will update the weights to optimize model performance and accuracy.

4.3.6 Number of epochs

The number of epochs is an hyperparameter that determines the number of iterations the learning algorithm will traverse over the complete training dataset. An epoch refers to the point at which every sample in the training dataset has been used to update the internal model parameters.

“epoch=100”, means the number of times the entire dataset passed through the model during the training process.

4.4 Training procedure

The goal of training an autoencoder is to minimize the difference between the input and the reconstructed output, and we measure this using a loss function.

4.4.1 How autoencoder train

Training an autoencoder is unsupervised in the sense that no labeled data is needed. The training process is still based on the optimization of a cost function, and can be summarized in the following steps [3]:

Step 1: Forward Pass

Input: An autoencoder first takes an input x

Encoding: The encoder maps(encode) x into a latent representation y using deterministic mapping, such as:

$$y = f(x; \theta e) = s(Wex + be) \quad (4.6)$$

where:

"*s*": is an activation function.

"W_e, b_e": encoder parameters (weight, bias).

Decoding: y is then mapped back(decode) into a reconstruction z, which is of same shape as x using similar mapping.

$$z = g(z; \theta d) = s(Wdz + bd) \quad (4.7)$$

where:

"*s*": is an activation function.

"W_{dz}, b_d": decoder parameters (weight, bias).

Step 2: Compute Loss

The difference between the original input x and the reconstructed input z is measured using a loss function.in our case we used "1_ssim" as a custom loss function.

$$L = 1 - SSIM \quad (4.8)$$

where: $$SSIM = [l(x,z)]^{\alpha}[c(x,z)]^{\beta}[s(x,z)]^{\gamma}$$

Step 3: Backpropagation

The loss is then backpropagated through the network to compute the gradients of the loss with respect to all the parameters (weights and biases).

Step 4: Update Parameters

Using an optimization algorithm in our case we used RMS propagation, the parameters of the encoder and decoder are updated to minimize the loss.

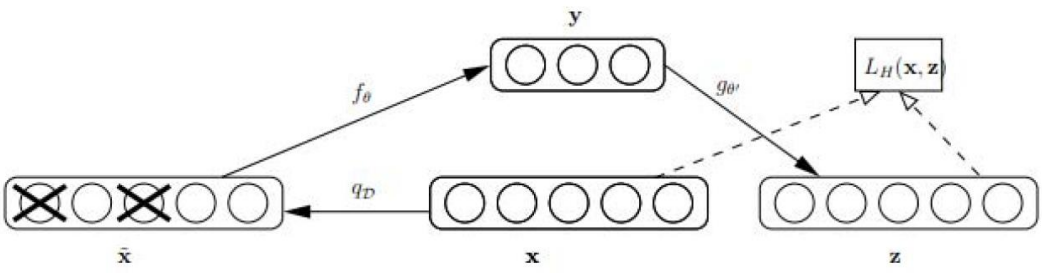

Figure 4- 9 Autoencoder training architecture [37].

4.4.2 How convolutional autoencoder train

The difference of training between a basic AE and a CAE, is in the forward pass and the parameters [3].

Step 1: Forward Pass

Input: An autoencoder first takes an input image x of shape (H, W, C), where H is height, W is width, and C is the number of channels.

Encoding: The input x is passed through a series of convolutional layers, to encode x into a latent representation $y = h^{(L)}$. Each convolutional layer performs:

$$y = f(x; \theta e) = h^{(l)} = s(We^{(l)} * h^{(l-1)} + be^{(l)}) \quad (4.9)$$

where:

"s": is an activation function.

"W_e, b_e": encoder parameters for convolutional layers (kernel coefficients, bias).

"l": is the number of the layer. And $h^{(0)} = x$.

Decoding: The latent representation y is passed through a series of deconvolutional layers, to decode it into a reconstruction $z = h^{(M)}$. Each convolutional layer performs:

$$z = z(y; \theta d) = h^{(m)} = s(Wd^{(m)} * h^{(m-1)} + bd^{(m)}) \quad (4.10)$$

where:

"s": is an activation function.

"W_d, b_d": encoder parameters for deconvolutional layers (kernel coefficients, bias).

"m": is the number of the layer. And $h^{(0)} = y$.

Step 2: Compute Loss

The difference between the original input *x* and the reconstructed input z is measured using a loss function. In our case we used "1_ssim" as a custom loss function, it's the same as the previous case (AE).

Step 3: Back propagation

The loss is then back propagated through the network to compute the gradients of the loss with respect to all the parameters (weights and biases).

Step 4: Update Parameters

Using an optimization algorithm in our case we used RMS prop, the parameters of the encoder and decoder are updated to minimize the loss.

4.5 Conclusion

In summary, this chapter provided a thorough examination of the methodology used to create and train a CNN-based denoising autoencoder. By detailing the architecture and its components, from the input layer to the output layer, we established a clear blueprint for the model.

The use of specific activation functions, a custom loss function, and the RMSprop optimizer contributed to the model's effectiveness in noise reduction. Additionally, the implementation of early stopping callbacks and a structured training procedure ensured optimal performance. This methodological framework lays a solid foundation for achieving superior denoising capabilities, ultimately enhancing image quality and clarity.

Chapter 5 Experiments and results

5.1 Introduction

Medical imaging techniques like MRI, CT, and ultrasound are prone to noise due to various factors, including efforts to reduce patient radiation exposure. As radiation decreases, noise increases, making effective denoising crucial for accurate image analysis.

Image denoising is a long-standing problem in computer vision with many traditional approaches, such as partial differential equations (PDEs), wavelet transforms, and non-local means. These methods aim to reconstruct the original image from its noisy version, often assuming a specific noise process.

Deep learning advancements, particularly convolutional neural networks (CNNs), have shown promise in denoising tasks. Convolutional Neural Network Denoising Autoencoders (CDAEs) leverage spatial correlations in images, outperforming traditional methods.

We test how well CDAEs remove noise from the mini-MIAS and Panoramic Dental X-rays databases in this chapter, both with and without median and Gaussian preprocessing filters. Before the CDAE processes the images, these filters aim to reduce noise. Additionally, we explore handling speckle noise using logarithmic transformation to convert it from multiplicative to additive, followed by inverse transformation after denoising. However, this approach did not yield significant improvements.

We present quantitative and qualitative results, including visual comparisons and metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). The findings highlight the marginal benefits of preprocessing and discuss the implications of dataset size on denoising model performance.

5.2 Types of Datasets

For training and testing the denoising process of medical images, we use the mini-MIAS and a Panoramic Dental X-rays dataset.

5.2.1 mini-MIAS database

The Mammographic Image Analysis Society has produced a digital mammography database with a resolution of 1024×1024. It contains 161 pairs of films, examples of abnormalities commonly encountered in screening, as well as a comprehensive set of normal cases. We carefully selected the mammograms from the United Kingdom National Breast Screening Programme to ensure the highest quality of exposure and patient positioning. We digitized each medio-lateral oblique view using a scanning microdensitometer with a linear response in the optical density range of 0.0 to 3.2, representing each pixel with 8 bits. The entire database, when compressed, occupies less than 2 GBytes, fitting onto a single 8-mm magnetic tape. For research purposes, copies are available [38].

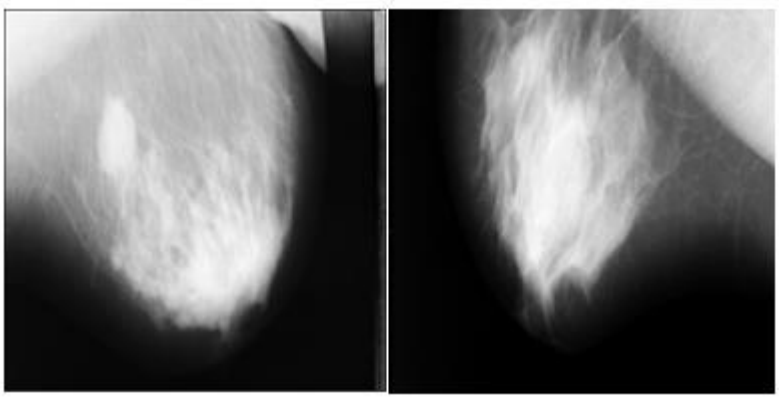

Figure 5- 1 Random samples of medical images taken from mini-MIAS dataset [38]**.**

5.2.2 Panoramic Dental X-rays database

This dataset consists of anonymized and deidentified panoramic dental X-rays of 116 patients with a format of 2900×1250 pixels taken by the Soredex CranexD digital panoramic x-ray unit at Noor Medical Imaging Center, Qom, Iran.

The subjects cover a wide range of dental conditions, from healthy to partial and complete edentulous cases. Figures 5.1 and 5.2, respectively, display a sample of images from the two datasets [39].

Figure 5- 2 Random samples of medical images taken from the Panoramic Dental X-rays database [39].

5.3 Noise Types

Developing a system for universally reducing noise in images is a complex challenge that requires a solution. Gaining an understanding of the diverse patterns generated by the various types of noise offered valuable insight into other approaches for solving this challenge. Figure 5.3 and 5.4 illustrates the impact of different types of noise on medical images from both datasets.

When we closely observe Figure 5.3, we notice the following:

1. Versions of Gaussian noise include salt and pepper noise.
2. One can represent poisson noise as a version of speckle noise.
3. Gaussian noise can represent speckle noise as a function.

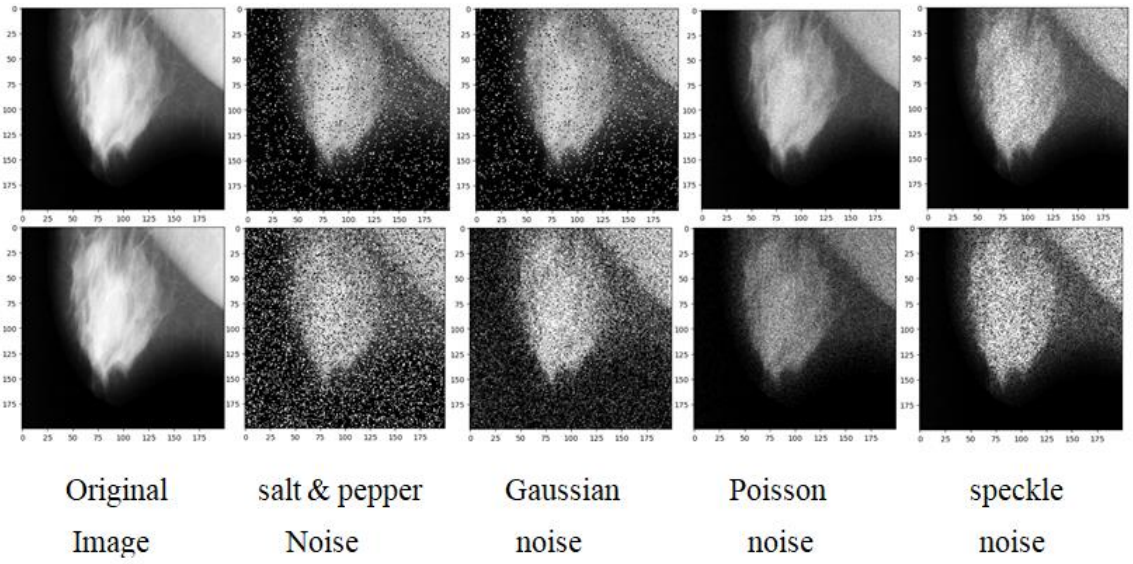

Figure 5- 3 Effect of different type of noise on the original image, taken from the mini-MIAS dataset (first row show the minimal level of noise second row show a higher noise level).

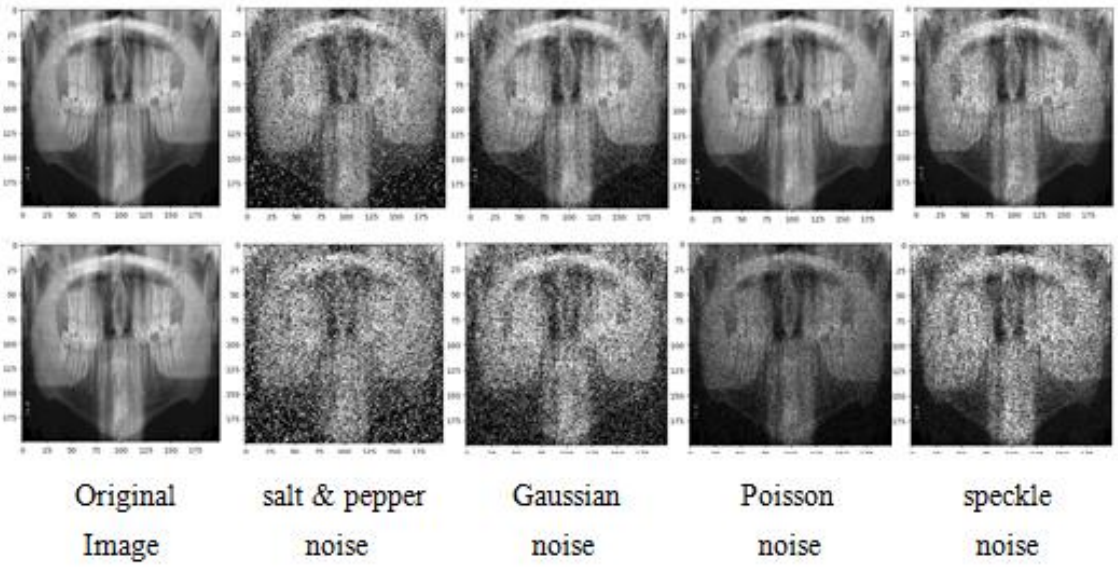

Figure 5- 4 Effect of different type of noise on the original image, taken from the Panoramic Dental X-rays dataset (first row show the minimal level of noise second row show a higher noise level).

5.4 Types of Losses

To train such as model, there are made use of various types of loss functions:

- **L_1 Loss:**

$$L_1 = \sum |\mathrm{I}_{\mathrm{orig}} - \mathrm{I}_{\mathrm{denoised}}| \tag{5.1}$$

where I_{orig} is the original image and $I_{denoised}$ is the output of the model.

- **L_2 Loss:** Mean Squared Error (MSE)

$$L_2 = MSE = \frac{1}{m}\sum |\mathrm{I}_{\mathrm{orig}} - \mathrm{I}_{\mathrm{denoised}}|^2 \tag{5.2}$$

where I_{orig} is the original image and $I_{denoised}$ is the output of the model.

- **Elastic Loss with λ = 0.25, 0.5, and 0.75:**

$$ElasticNet = \lambda . L_1 + (1 - \lambda) . L_2 \tag{5.3}$$

To train our model, we have made use of the MSE loss function.

5.5 Optimizer Types

We have made use of two types of optimizers in our training and testing process: Mini Batch Gradient Descent and Adam Optimizer. We found that both optimizers perform optimally, but Adam Optimizer was a better optimizer.

5.6 preprocessing

5.6.1 Loading and Resizing Images

The images are loaded from a specified path, in grayscale mode. Each image is resized to a fixed dimension of 200x200 pixels to ensure uniformity across the dataset. The resized images are then flattened and stored in a list ‘img_single’.

5.6.2 Splitting Dataset

The dataset is divided into training and testing sets. The first 10% of the images, are assigned to the testing set (test_y), while the remaining images are used for training (train_y).

5.6.3 Normalizing Images

The training and testing images are normalized to a range of [0, 1], ensuring the pixel values are suitable for training the autoencoder. The images are then reshaped to include a channel dimension. resulting in arrays of shape (number of images, width, height, channel).

5.6.4 Adding Noise

The flattened images are normalized to the range [0, 1]. and instead of corrupting a single image at a time, flattened dataset with each row representing an image was corrupted, simultaneously perturbing all images. Different parameters detailed in Table 1 were used for corruption.

Noise type	Corruption parameters
Salt and Pepper	Density = 0.1, proportion = 0.5
Salt and pepper	density= 0.3, proportion=0.5
Gaussian	mean=0, variance = 0.01
Gaussian	mean=0, variance = 0.08
Poisson	Proportion = 0.2, mean=1
Poisson	Proportion = 0.2, mean=5
Speckle	mean = 0, variance = 0.04
Speckle	mean = 0, variance = 0.2

Table 5- 1 For each type of noise in the, there are both low and high levels of perturbation.

5.6.5 Preparing Noisy Images for Training

The noisy images are reshaped back to their original 200x200 dimensions. Similar to the ground truth images (the free-noise images in 'test_y' and 'train_y'), the first 10% of the noisy images are used for testing (test_x), while the remaining images are used for training (train_x). Both the noisy training and testing images are reshaped to include a channel dimension, resulting in arrays of shape (number of images, width, height, channel).

5.6.6 median and gaussian filter

We fed the photos into the CDAE after applying a mix of Gaussian and median filters. This preprocessing phase aims to improve the input image quality and minimize noise.

5.6.6.1 Median Filter: A median filter with a kernel size of 5x5 pixels was used. The median filter is effective in removing impulsive noise, such as salt-and-pepper noise, while preserving the edges of the image.

5.6.6.2 Gaussian Filter: A Gaussian filter with a kernel size of 5×5 pixels was applied. with a standard deviation for the Gaussian distribution was set to 1.0. The Gaussian filter smooths the image by averaging the pixel values within the kernel, weighted by a Gaussian function, effectively reducing Gaussian noise.

5.7 Evaluation metrics

To quantify the difference in change between the two images, before and after passing it through our proposed model, we need relevant metrics that would score any improvement or deterioration in the image. We have used metrics that capture the structure of the image in both cases, while also using metrics that capture the effectiveness of a machine learning model by only changing the input to the model. An ideal evaluation technique would be where our proposed architecture could be exposed to various types of noise and image sizes. If any system works consistently in this setting, we would have an ideal and optimal blind denoising and enhancement system.

Making use of metrics like PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure) are being used to evaluate changes in the structural changes of an image. Both these metrics use very different aspects of an image to determine the difference with respect to the original, non-noisy, ideal image. The lesser difference in the images equates to higher scores. PSNR primarily looks into the mean square error between the two images, whereas SSIM calculates changes in the luminance, contrast, and structure difference between them.

5.7.1 PSNR (Peak Signal-to-Noise Ratio)

This term is defined as a ratio between the maximum possible signal strength and the amount of noise that affects the quality of an image. This ratio is calculated in the logarithmic scale due to existence of wide dynamic[40].

range in images. The formula to calculate PSNR is expressed below:

$$PSNR = 20.\log_{10}(\frac{Sig_{max}}{\sqrt{MSE}})$$

Where: (5.4)

$$MSE = \frac{1}{M.N}\sum_{rows=1}^{m}\sum_{columns=1}^{n}||\mathrm{I}_{\mathrm{orig}} - \mathrm{I}_{\mathrm{degraded}}||^2$$

Where:

Sig_{max}: is the maximum signal strength of the original image,

MSE: is the Mean Squared Error,

I_{orig}: is the image data of the original image,

$I_{degraded}$: is the image data of the degraded image,

m: is the number of rows in the image,

n: is the number of columns in the image [40].

5.7.2 SSIM (Structural Similarity Index)

It is a method to measure the similarity between two images in terms of factors like contrast, luminance, and structural context. It is viewed as a quality measure for an image being compared to the original image,

which in these cases, is regarded as an ideal image. This metric evaluates the images on structural metrics instead of the absolute difference in pixel values[40], as seen in PSNR. SSIM can be calculated using the following formulae:

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}\,[s(x,y)]^{\gamma} \tag{5.5}$$

where α, β and γ> 0 control the relative significance of each of three terms in SSIM and l, c and s are luminance, contrast and structural components calculated as follow:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$
$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{5.6}$$
$$s(x,y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + +C_3}$$

where μ_x and μ_y represents the mean of original and coded image, $б_x$ and $б_y$ are standard deviation and $б_{xy}$ is the covariance of two images [40].

5.8 Tools

5.8.1 Software Tools

Developing the different deep learning models such as CDAE were done in python 3 programming language and main libraries such as Tensorow and Keras were used.

- TensorFlow: is an open-source machine learning system that runs at large scale and in heterogeneous environments. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multi-core CPUs (Central Processing Units), general-purpose GPUs (Graphics Processing Units), and TPUs (Tensor Processing Units). TensorFlow enables developers to test novel optimizations and training algorithms. It also supports a variety of applications, with a focus on training and inference on deep neural networks.

- Keras : is a high-level neural networks API that is developed in Python and can run on top of software libraries such as TensorFlow, CNTK, or Theano. It can run smoothly on both CPUs and GPUs. Keras was implemented with a focus on enabling fast experimentation [37].

5.8.2 Hardware Tools

The CDAE modelling were done on an ASUS Vivobook

-Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz.

-16.0 GB RAM.

-no GPU.

5.9 Fine tuning

To maximize the performance of the model, the autoencoder was fine-tuned independently for each database (mini-MIAS database and the Panoramic Dental X-rays database) with salt and pepper noise effect. Important hyperparameters were also adjusted. The primary hyperparameters tuned were the number of epochs, batch size, and image size. For each tuning process, the model's performance was evaluated using SSIM (Structural Similarity Index Measure) and PSNR (Peak Signal-to-Noise Ratio) metrics. The following consecutive steps were used to carry out the fine-tuning process:

5.9.1 Batch Size Tuning

Process: The batch size was varied across the values of 10, 20, 30, 40, and 100, while keeping the number of epochs fixed at 50 and the image size at 64x64 pixels.

Outcome: The batch size that produced the best SSIM and PSNR scores was selected for the next stage.

5.9.2 Epoch Tuning

Process: With the optimal batch size determined from the previous step, the number of epochs was then varied among 50, 100, 200, 300, 400, and 500.

Outcome: The best-performing number of epochs was chosen for the subsequent tuning step.

5.9.3 Image Size Tuning

Process: Finally, the image size was adjusted among 64x64, 100x100, and 200x200 pixels, while keeping the batch size and number of epochs fixed at their optimal values obtained from the previous stages.

Outcome: The image size that resulted in the highest SSIM and PSNR was selected as the optimal configuration.

By systematically adjusting and evaluating these hyperparameters, the model was fine-tuned to achieve the best denoising performance for each database, ensuring robust and reliable results.

5.10 Fine tuning results

5.10.1 mini-MIAS database results

5.10.1.1 Batch Size Tuning

Batch size	10	20	30	40	100
SSIM	0.853	0.846	0.814	0.822	0.774
PSNR	26.445	25.057	19.381	22.940	22.828

Table 5- 2 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels).

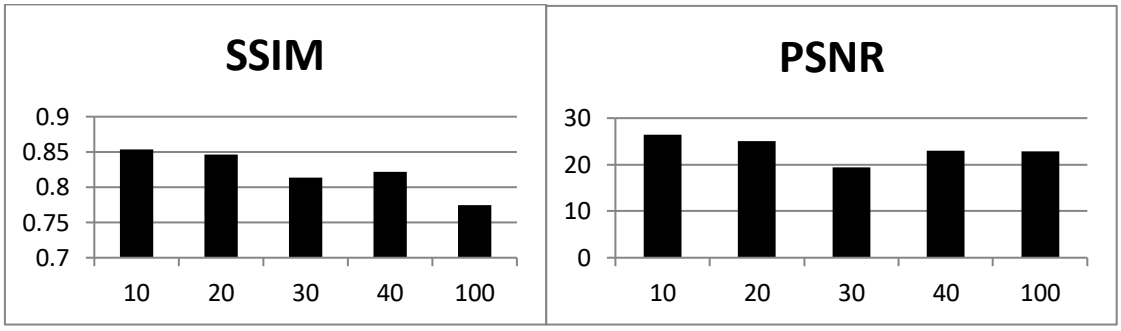

Figure 5- 5 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels).

5.10.1.2 Epoch Tuning

Epoch	50	100	200	300	400	500
SSIM	0.856	0.878	0.889	0.894	0.891	0.890
PSNR	22.407	23.495	27.993	28.328	27.179	27.514

Table 5- 3 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels).

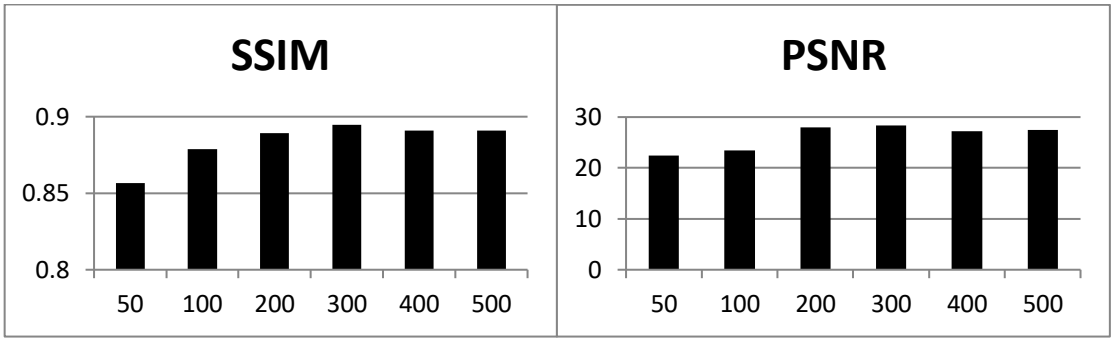

Figure 5- 6 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels).

5.10.1.3 image size Tuning

Image size in pixel	64×64	100×100	200×200
SSIM	0.883	0.903	0.913
PSNR	27.263	29.246	30.598

Table 5- 4 SSIM and PSNR results with different image sizes (fixing batch size in 10 and number of epochs in 300).

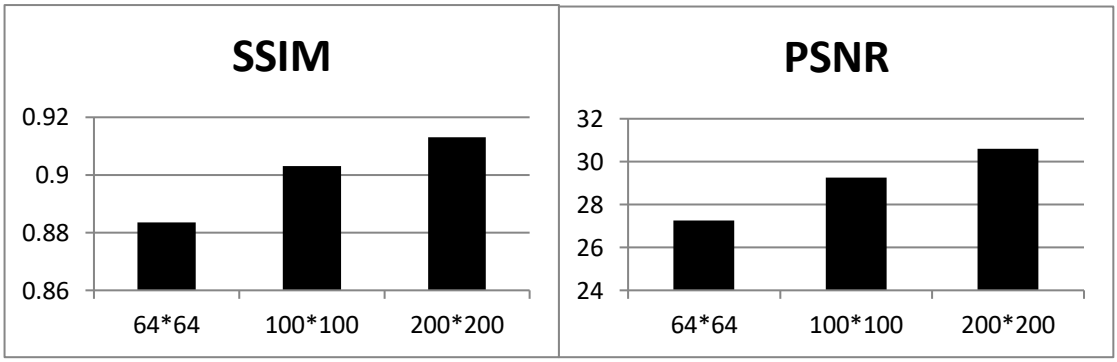

Figure 5- 7 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 300).

The best hyperparameters for mini-MIAS database are 10 for batch size, 300 for the number of epochs and 200×200 pixels for image size.

5.10.2 Panoramic Dental X-rays database results

5.10.2.1 Batch Size Tuning

Batch size	10	20	30	40	100
SSIM	0.737	0.705	0.704	0.690	0.595
PSNR	21.328	20.824	20.623	20.061	19.095

Table 5- 5 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels).

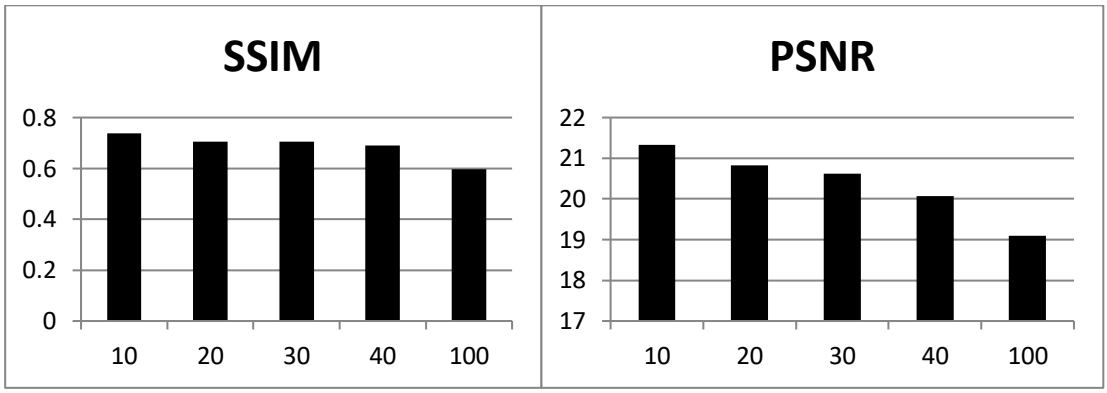

Figure 5- 8 SSIM and PSNR results with different batch size values (fixing epochs in 50 and image size in 64×64 pixels).

5.10.2.2 Epoch Tuning

Epoch	50	100	200	300	400	500
SSIM	0.725	0.787	0.809	0.801	0.806	0.784
PSNR	19.937	23.131	23.796	22.952	23.353	21.980

Table 5- 6 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels).

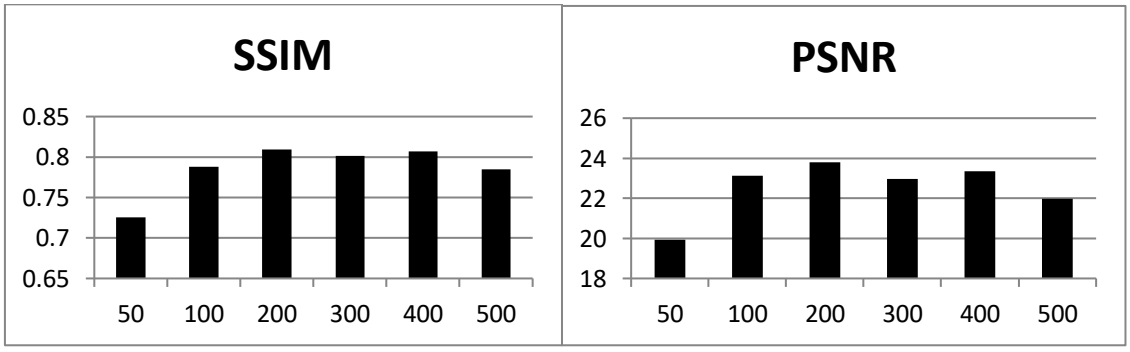

Figure 5- 9 SSIM and PSNR results with different numbers of epochs (fixing batch size in 10 and image size in 64×64 pixels).

5.10.2.3 image size Tuning

Image size	64×64	100×100	200×200
SSIM	0.807	0.835	0.868
PSNR	23.042	22.925	25.937

Table 5- 7 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 200).

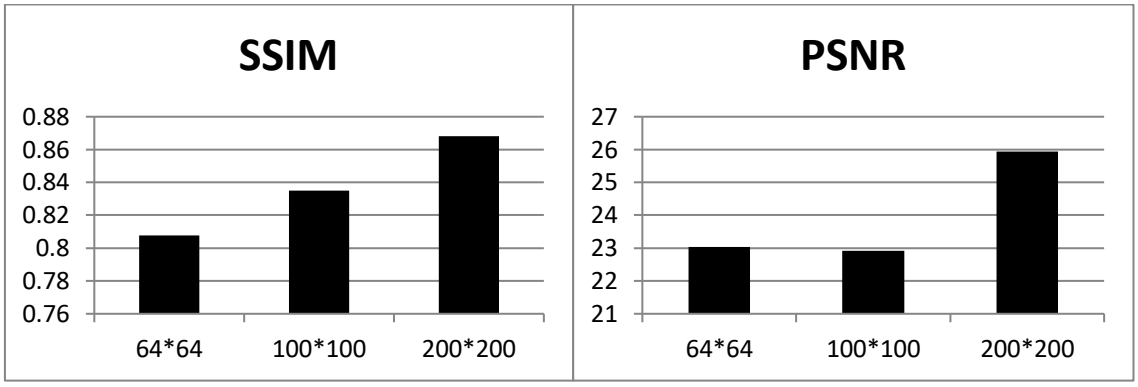

Figure 5- 10 SSIM and PSNR results with different image sizes (fixing batch size in 10 and epochs in 200).

The best hyper parameters for the Panoramic Dental X-rays database are: 10 for batch size, 200 for the number of epochs and 200×200 pixels for image size.

5.11 Empirical evaluation

For the baseline comparison, we explore the impact of incorporating median and Gaussian filtering in the preprocessing step of the CDAE.

Our objective with this addition is to improve the performance of the autoencoder for denoising images.

To keep similar noise effect, we corrupt the images of the two datasets with same noise level as shown in table 1.

Using a batch size of 10 and 300 epochs, with an image size of 200×200 pixels for the mini-MIAS database, and similarly for the Panoramic Dental X-rays database (with the number of epochs changed to 200), the denoising results are presented in Figure 5.11,5.12, and Table 7"

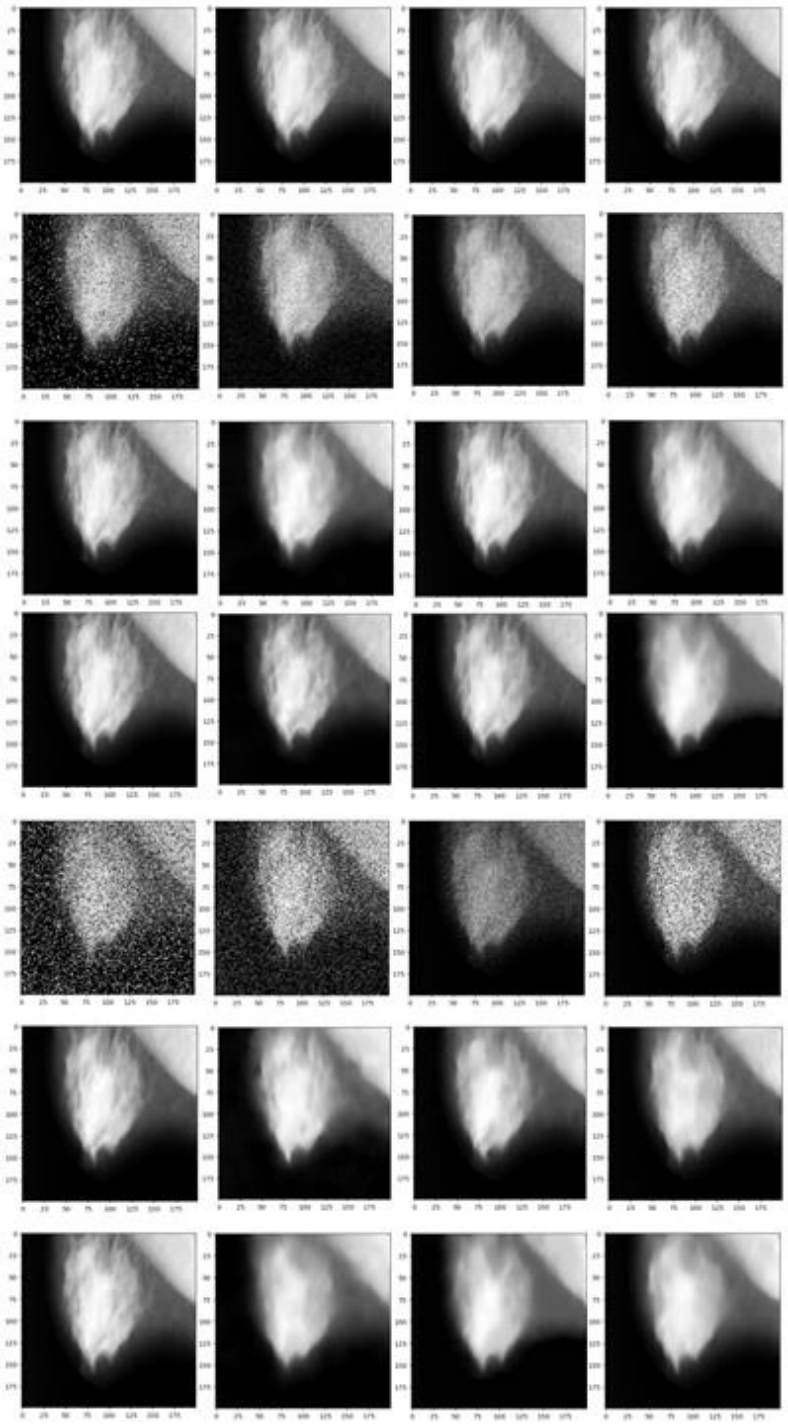

Figure 5- 11 denoising performance of CDAE on the mini-MIAS database with and without filtering in the preprocessing. The top row displays the real images. The second and fifth rows show the noisier versions with minimal and higher noise levels, respectively. The third and sixth rows present the denoising results of CDAE without filtering. The fourth and eighth rows show the results of CDAE with filtering in the preprocessing.

Mammogram database with minimal noise levels								
Noise types	S&P noise		Gaussian noise		Speckle noise		Poisson noise	
Metrics	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
CNNAE	0.913	30.598	0.846	28.546	0.868	28.955	0.887	29.441
CNNAE+Filtering	0.903	32.265	0.840	30.152	0.860	30.356	0.851	31.109
Mammogram database with higher noise levels								
Noise types	S&P noise		Gaussian noise		Speckle noise		Poisson noise	
Metrics	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
CNNAE	0.882	30.822	0.750	21.697	0.840	29.070	0.854	29.483
CNNAE+Filtering	0.889	28.740	0.777	28.515	0.842	26.680	0.758	28.416

Table 5- 8 comparing mean SSIM and PSNR scores using denoising CDAE without and with filtering for mini_MIAS database.

we investigated the denoising performance of a Convolutional Neural Network Denoising Autoencoder (CDAE) on the mini-MIAS database, both with and without the application of Gaussian and median filtering in the preprocessing stage. The results are presented in a comprehensive Figure 5.12 and table 8.

The results reveal that with minimal noise levels, there is a slight improvement in PSNR scores when using CDAE with filtering compared to without_ filtering, while the SSIM values remain close. Conversely, at higher noise levels, CDAE with filtering shows a slight improvement in SSIM scores, with PSNR values being similar between the two approaches.

These findings suggest that the preprocessing filters only slightly improve denoising performance. The filters appear to help the CDAE better maintain image details, as evidenced by the minor gains in PSNR at low noise levels. The increase in SSIM at higher noise levels suggests that the filters help preserve the pictures' structural integrity.

To test if less sample size by using Panoramic Dental X-rays database would have an impact on denoising performance.

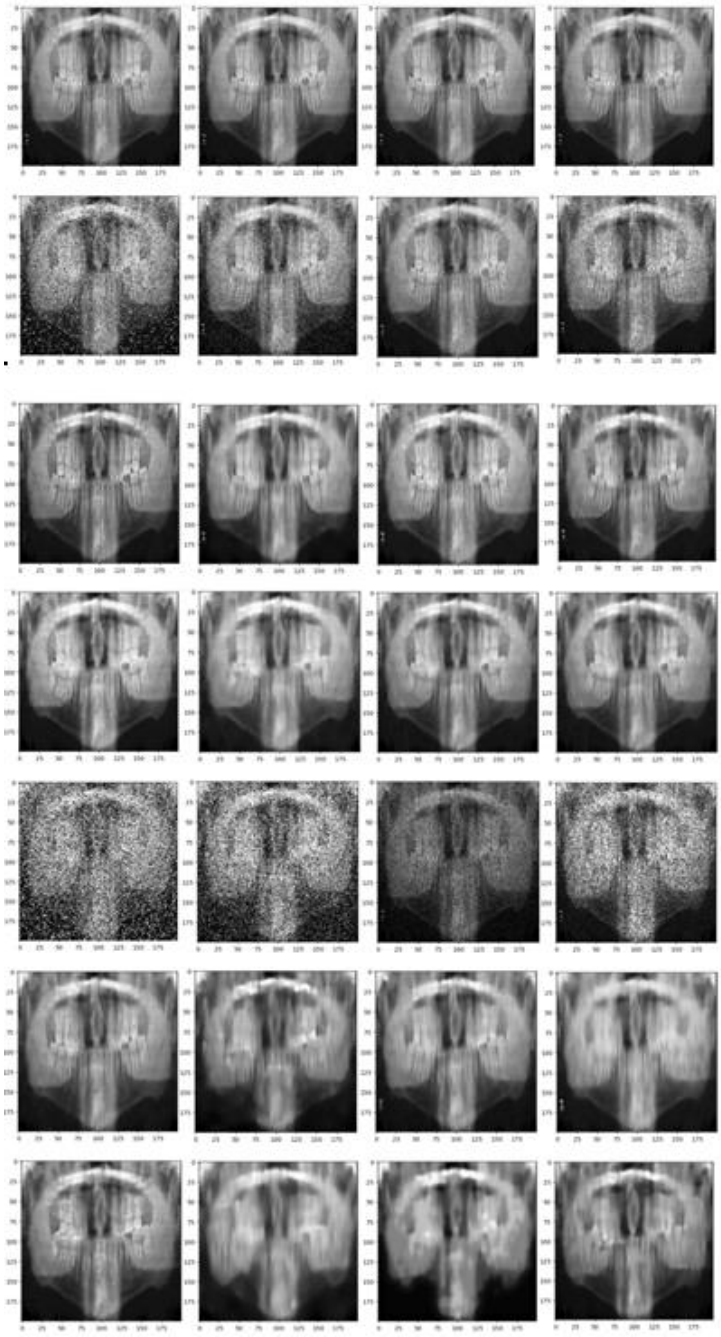

Figure 5- 12 denoising performance of CNNDAE on the Panoramic Dental X-rays database with and without filtering in the preprocessing. The top row displays the real images. The second and fifth rows show the noisier versions with minimal and higher noise levels, respectively. The third and sixth rows present the denoising results of CNNDAE without filtering. The fourth and eighth rows show the results of CNNDAE with filtering in the preprocessing.

Panoramic Dental X-rays database with minimal noise levels								
Noise types	S&P noise		Gaussian noise		Speckle noise		Poisson noise	
Metrics	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
CNN-AE	**0.868**	**25.937**	**0.802**	**25.621**	**0.813**	**26.615**	**0.868**	**27.779**
CNNAE+Filtering	0.826	24.977	0.753	23.614	0.769	24.674	0.803	27.648
Panoramic Dental X-rays database with higher noise levels								
Noise types	S&P noise		Gaussian noise		Speckle noise		Poisson noise	
Metrics	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
CNN-AE	**0.815**	**27.740**	**0.662**	**24.212**	**0.718**	**25.478**	**0.766**	**26.635**
CNNAE+Filtering	0.805	25.938	0.655	24.063	0.690	23.441	0.655	24.018

Table 5- 9 comparing mean SSIM and PSNR scores using denoising CDAE without and with filtering for Panoramic Dental X-rays database.

The results reveal that with minimal and higher noise levels, there is no significant improvement in PSNR and SSIM scores when using CDAE with filtering compared to without filtering.

These findings suggest that the preprocessing filters do not provide substantial benefits in denoising performance for this particular dataset. The lack of improvement can be attributed to the limited size of Panoramic Dental X-rays database compared to the mini-MIAS which is almost double the size of the other database, which provides fewer images for training the autoencoder.

We conclude that the capability of CDAE to learn and denoise is strong, making the added preprocessing filters less impactful but still beneficial under certain noise conditions.as we can see in figure 15 the model converged nicely for the given noise levels and sample size (292 training images). But as the noise level is changes with less sample size (106 training images) the network has trouble converging. As shown in figure 29.

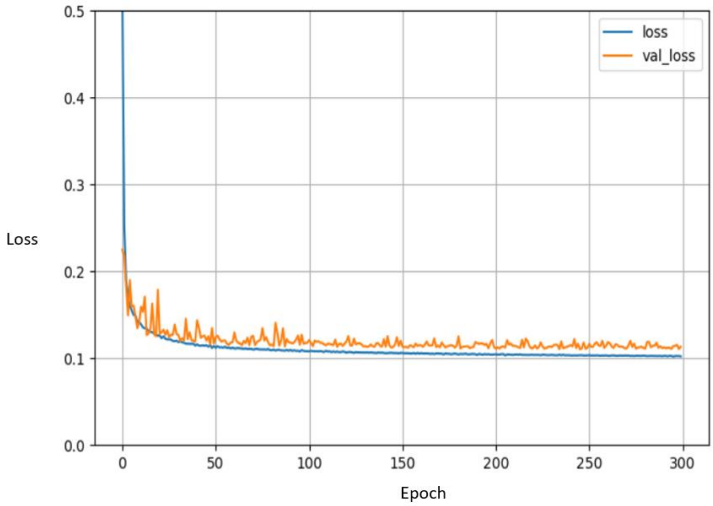

Figure 5- 13 loss and validation loss from 300 epochs using a batch size of 10 and 200*200 pixels image size with salt and pepper noise (density=0.3, proportion=0.5).

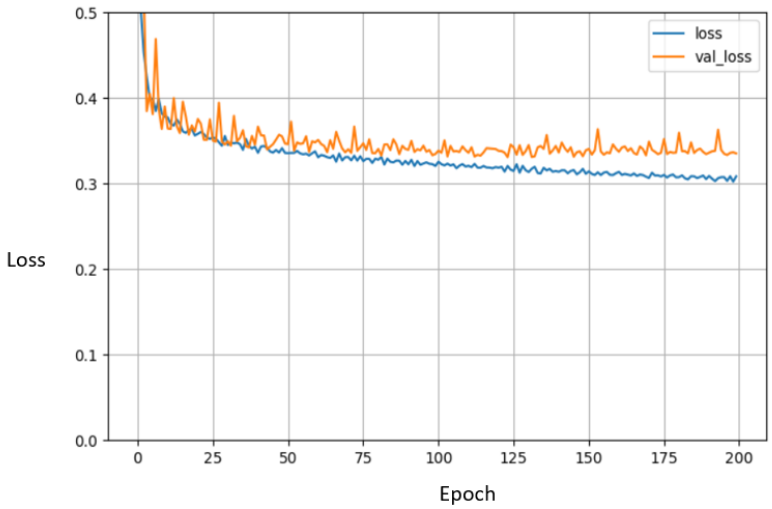

Figure 5- 14 loss and validation loss from 200 epochs using a batch size of 10 and 200×200 pixels image size with Gaussian noise (mean=0, variance=0.08).

Speckle noise

we also explored a specific approach for handling speckle noise, which is a type of multiplicative noise that poses significant challenges in image denoising. We suggested incorporating a logarithmic transformation before the autoencoder (AE) to convert the multiplicative noise into additive noise, followed by an inverse logarithmic transformation

after the AE processing. This method aimed to simplify the noise characteristics, hoping to improve the denoising performance.

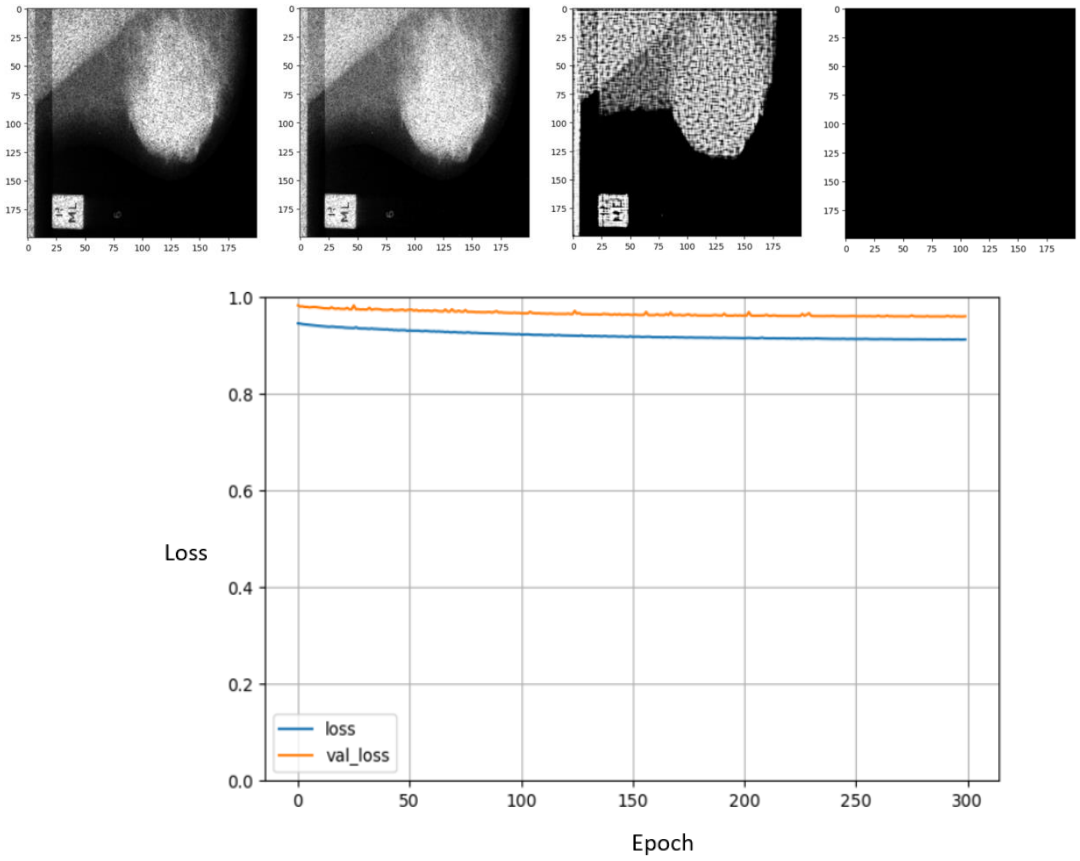

Figure 5- 15 loss and validation loss from 300 epochs using a batch size of 10 and 200*200 pixels image size with Speckle noise (mean=0, variance=0.04) with a log transformation in the preprocessing.

Average SSIM: 0.20129868

Average PSNR: 13.300069

However, contrary to our expectations, the results did not show any improvement when compared to the standard approach without the logarithmic transformation.

5.12 Result and discussion

The proposed medical image denoising model is trained using two medical datasets as in section 5.2; These images were processed before passing to the denoising system. Pre-processing consists of resizing all images to have same dimensions (64×64, 100×100 and 200×200), to use computational resources more efficiently. Then, a noise is added to images with different levels, as in Figure 5.3 and 5.4. The processed images are split into 90% for

training and 10% testing. as indicated in section 5.2, 322 images from mini-mias dataset, and 116 images from Panoramic Dental X-rays database.

The training data are then encoded via the CDAE. When the data is encoded, the input is transformed into a set of features. These features are identified by the encoder to act as markers for decoding the output at the next stage. Figure 5.13 presents the loss curve of model training and validation. The training has stopped at 300 epochs, as the loss starts to stabilize to its minimum after 250 epochs.

The 10 % images test dataset is then passed to the trained model, to evaluate the performance of the system in reconstructing medical mammography and X-ray panoramic dental noisy images. Table 8 and 9 shows the average results without and with applying filtering on the image test dataset using PSNR and SSIM metrics.

They show that the trained model with images that have minimal noise levels added noise, has the best average results for the salt& paper noise: **0.913** for SSIM, **30.598 dB** for PSNR without filtering, and **0.903** for SSIM, **32.265 dB** for PSNR with filtering, but the trained model with images that have high noise levels added noise, has the best average results for the Gaussian noise: **0.777** for SSIM, **28.515 dB** for PSNR.

The conclusions drawn from the data shown in Tables 8 and 9 indicate that the performance of the proposed model is significantly influenced by the level and nature of noise, as well as the preprocessing procedures implemented. An optimal solution would be to select an SSIM value greater than 0.88, while maintaining a PSNR of approximately 28 dB.

5.13 Conclusion and future work

We have demonstrated that Convolutional Neural Network Denoising Autoencoders (CDAEs) can effectively denoise medical images. Contrary to common belief, our results indicate that good denoising performance can be achieved even with small training datasets, as few as 300 samples.

The gains in denoising performance were negligible even after experimenting with a number of preprocessing methods, such as logarithmic adjustment for speckle noise and median and Gaussian filtering.

This implies that although preprocessing can have certain advantages, CDAEs' built-in capabilities are strong enough to effectively manage noise on their own, given enough training data.

We propose a CDAE denoising system to effectively reduce noise in mini-MIAS and X-ray panoramic dental images. We train this system using small medical datasets, demonstrating its efficiency with two metrics: PSNR and SSIM.

Future work can use larger datasets with a mixture of different kinds of noise. Furthermore, we can apply this model to other types of images, including DNA and microscopic images.

Chapter 6 Conclusion

Noise in images is a well-known problem. In every image, noise is predominant in some form or another. Noisy images pose challenges and frequently lead to their removal from datasets, as they lead the model to incorrectly identify features. Images with lower resolution do not hold enough context and information due to the small number of pixels in the image. Various stages in the image processing pipeline introduce different types of noise.

Identifying and using the noise artifacts from the images to enhance the images can help provide better-looking images. Denoising would also improve the image's details, which can help machine learning models learn better features.

This enhancement in the existing datasets has the potential to enhance the precision and functionality of contemporary cutting-edge algorithms. Software-based denoising and image enhancement can serve as a remedy to sensor noise, which is primarily a hardware issue. By using a denoising and augmentation technique, we can effectively utilize all the intricate details included in an image. Historically, we have relied on empirical methods to ascertain the most effective approaches for addressing different forms of noise in photographs. We implemented noise reduction techniques tailored to the specific type of noise identified by the user in the image.

During image processing applications, the CAD system serves as a highly powerful diagnostic tool. Numerous factors can introduce noise into natural images. However, the primary source of noise addition occurs during acquisition and transmission. Appropriate denoising filters can suppress or detach this noise. Therefore, we should implement the denoising process to enhance the image quality for a more accurate diagnosis.

This master thesis critically examines the benefits and constraints of many published publications on approaches for denoising dataset images. In order to achieve improved results while working with noisy photos, the denoising method consistently necessitates prior knowledge of the noise map and its ability to adjust.

This master thesis provides a comprehensive summary of the process of reconstructing the dataset picture, identifying the noise present in the dataset image, applying CDAE

denoising methods, and conducting a comparative analysis based on the resulting output. The evaluation of the output is done using numerical performance metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

However, the findings of the experimental study indicate that the performance of the suggested model is strongly affected by the level and kind of noise, as well as the preprocessing processes employed. To achieve the best outcome, it is advisable to choose an SSIM value that is higher than 0.88, while also ensuring that the PSNR remains around 28 dB.

The imaging techniques and diverse noise reduction algorithms are subject to daily improvements. Thus, it is crucial to improve the denoising techniques as well. The objective of this review is to present a comprehensive analysis of the presence of noise in dataset photos and the many denoising approaches that are now accessible. Researching noise can assist developers in devising novel denoising techniques for dataset photos.

References

[1] M. Ghosh and T. Arunachalam, "Introduction to Artificial Intelligence," 2021, pp. 23-44.

[2] S. K. M. S. Islam, M. A. A. Nasim, I. Hossain, D. M. A. Ullah, D. K. D. Gupta, and M. M. H. Bhuiyan, "Introduction of Medical Imaging Modalities," in *Data Driven Approaches on Medical Imaging*, B. Zheng, S. Andrei, M. K. Sarker, and K. D. Gupta Eds. Cham: Springer Nature Switzerland, 2023, pp. 1-25.

[3] L. Gondara, "Medical Image Denoising Using Convolutional Denoising Autoencoders," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 12-15 Dec. 2016 2016, pp. 241-246, doi: 10.1109/ICDMW.2016.0041.

[4] G. Abdul and S. Usman, "Convolutional Autoencoder for Image Denoising," *UMT Artificial Intelligence Review*, vol. 1, no. 2, 12/31 2021, doi: 10.32350/AIR.0102.01.

[5] D. Bank, N. Koenigstein, and R. Giryes, *Autoencoders*. 2020.

[6] "Généralités sur le traitement d'images," in *Université Mohamed Khider Biskra*, ed.

[7] "INTRODUCTION TO DIGITAL IMAGE," in *Master of Computer Application(MCA) (MCA)*, ed. Indira Gandhi National Open University, 2022/2023.

[8] "chapter 1: Digital Image Basics," ed. Clemson University south carolina USA.

[9] S. shica and D. Gupta, "Various Raster and Vector Image File Formats," *IJARCCE*, pp. 268-271, 03/30 2015, doi: 10.17148/IJARCCE.2015.4364.

[10] "UNIT -I: Introduction to Digital Image processing," ed. VEMU INSTITUTE OF TECHNOLOGY.

[11] N. Dilmen. Medical X-rays [Online] Available: https://commons.wikimedia.org/wiki/File:Medical_X-Ray_imaging_JUP05_nevit.jpg#filelinks

[12] Cerevisae. CT scan of fluid collection at gastro-oesophageal junction [Online] Available: https://commons.wikimedia.org/wiki/File:CT_scan_of_fluid_collection_at_gastro-oesophageal_junction.png

[13] P. 16. One frame of an MRI scan of the head showing the eyes and brain [Online] Available: https://commons.wikimedia.org/wiki/File:MRI_Scan_General_Illustration.jpg

[14] O. work. Ultrasound images of pancreas [Online] Available: https://commons.wikimedia.org/wiki/File:Ultrasound_image_of_pancreas_110316091246_0916180.jpg

[15] M. Han. whole body scan for thyroid cancer evaluation [Online] Available: https://commons.wikimedia.org/wiki/File:Iodine_wb_scan.jpg

[16] A. Adler. CT of human thorax showing current paths for EIT and equipotentials [Online] Available: https://commons.wikimedia.org/wiki/File:CT_of_human_thorax_showing_current_paths_for_EIT_corrected.jpg

[17] Bionerd. MRI slice of my heart [Online] Available: https://commons.wikimedia.org/wiki/File:Cardiac_mri_slice_bionerd.jpg

[18] K. Ahmad, J. Khan, and M. S. U. D. Iqbal, "A comparative study of Different Denoising Techniques in Digital Image Processing," in *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 15-17 April 2019 2019, pp. 1-6, doi: 10.1109/ICMSAO.2019.8880389.

[19] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art,* vol. 2, no. 1, p. 7, 2019/07/08 2019, doi: 10.1186/s42492-019-0016-7.

[20] S. Swamy and P. Kulkarni, "A basic overview on image denoising techniques," *Int. Res. J. Eng. Technol,* vol. 7, no. 5, pp. 850-857, 2020.

[21] Z. Al Qadi, "Salt and Pepper Noise: Effects and Removal," *International Journal on Electrical Engineering and Informatics,* vol. 2, 07/15 2018.

[22] S. Hasinoff, "Photon, Poisson Noise," 2021, pp. 980-982.

[23] D. R. Newlin and C. S. Christopher, "Medical Image Denoising Using Different Techniques," *International Journal of Scientific & Technology Research,* vol. 9, pp. 1061-1066, 2020.

[24] "Generative AI : Neural Networks Overview." https://medium.com/@avicsebooks/generative-ai-neural-networks-overview-8a1efc438143 (accessed.

[25] "NEURAL NETWORKS," ed. MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY, 2022-2023.

[26] R. U. a. S. F. Richard Zemel, "Lecture 10: Neural Networks I," ed. University of Toronto.

[27] M. L. Dahhan, "Building a Clustering Model Using Neural Networks for Supporting Electronic Marketing Operation " Master Degree in Management Information Systems, Dept. of Statistics and Information System, University of Aleppo, 2015. [Online].

Available: https://www.academia.edu/23153102/Building_a_Clustering_Model_Using_Neural_Networks_for_Supporting_Electronic_Marketing_Operation

[28] S. K. Tareen and F. Khan Tareen, "Convolutional Neural Networks for Beginners," 09/07 2023, doi: 10.2139/ssrn.4566310.

[29] D. Merzougui, "Chapitre 3 : Réseau de neurones convolutionel CNN," ed. Univ-batan2.

[30] "Chapter 18 Convolutional Neural Networks," ed.

[31] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental Concepts of Convolutional Neural Network," in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, V. E. Balas, R. Kumar, and R. Srivastava Eds. Cham: Springer International Publishing, 2020, pp. 519-567.

[32] A. Sharma. "Introduction to Autoencoders." https://pyimg.co/ehnlf (accessed.

[33] G. Zerveas, *Improving Clinical Predictions through Unsupervised Time Series Representation Learning*. 2018.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481-2495, 2017.

[35] J. Xie, L. Xu, and E. Chen, "Image Denoising and Inpainting with Deep Neural Networks," *Advances in Neural Information Processing Systems*, vol. 1, 01/01 2012.

[36] C. Qi and F. Su, "Contrastive-center loss for deep neural networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, 17-20 Sept. 2017 2017, pp. 2851-2855, doi: 10.1109/ICIP.2017.8296803.

[37] M. A. P. Nicolas Anderson, Johnny Sognnes, "On the use of Denoising Autoencoders and Deep Convolutional Adversarial Networks for Automated Removal of Date Stamps," Department of ICT, University of Agder, 2019. [Online]. Available: https://uia.brage.unit.no/uia-xmlui/handle/11250/2618711

[38] J. Suckling, Parker, J., Dance, D., Astley, S., Hutt, I., Boggis, C., Ricketts, I., Stamatakis, E., Cerneaz, N., Kok, S., Taylor, P., Betal, D., & Savage, J. *Mammographic Image Analysis Society (MIAS) database v1.21*. [Online]. Available: https://www.repository.cam.ac.uk/items/b6a97f0c-3b9b-40ad-8f18-3d121eef1459

[39] A. K. Abdi, Shohreh. *Panoramic Dental X-rays With Segmented Mandibles*, doi: 10.17632/hxt48yk462.2.

[40] S. T. Gandhi, "Context Sensitive Image Denoising and Enhancement using U-Nets," Computer Science (MS), Computer Science (GCCIS), Rochester Institute of Technology, 2020. [Online]. Available: https://repository.rit.edu/theses/10588/

MEDICAL IMAGE DENOISING. AN AUTO ENCODERS BASED APPROACH

Abstract:

Artificial intelligence (AI) has significantly enhanced medical diagnostics, particularly through medical imaging. However, these images often suffer from noise due to various factors like reduced radiation exposure. Efficient noise reduction is crucial for accurate diagnosis. This master thesis addresses the problem of denoising medical images using Convolutional Autoencoders (CAEs). CAEs leverage the power of Convolutional Neural Networks (CNNs) to effectively distinguish between noise and essential diagnostic information. By training on large datasets, CAEs learn intricate noise patterns specific to different imaging modalities, preserving critical anatomical details. The proposed method aims to improve image clarity, ensuring reliable diagnostics and better healthcare outcomes. This study demonstrates the potential of CAEs in enhancing the quality of medical imaging, thereby supporting more precise medical evaluations.

Key words: *Medical Image Denoising, Autoencoders, Convolutional Autoencoders (CAEs), Convolutional Neural Networks (CNNs), Noise Reduction, Medical Imaging, Image Processing, Diagnostic Imaging, Artificial Intelligence.*

DÉBRUITAGE D'IMAGES MEDICALES PAR UNE APPROCHE BASÉE SUR LES AUTO- ENCODEURS

Résumé :

L'intelligence artificielle (IA) a considérablement amélioré le diagnostic médical, en particulier grâce à l'imagerie médicale. Cependant, ces images souffrent souvent de bruit en raison de divers facteurs tels que l'exposition réduite aux rayonnements. Une réduction efficace du bruit est essentielle pour un diagnostic précis. Cette thèse aborde le problème de l'identification des images médicales à l'aide d'auto-encodage convolutif (CAEs). Les CAEs tirent parti du pouvoir des réseaux neuronaux convolutifs (CNN) pour distinguer efficacement le bruit et les informations essentielles de diagnostic. En s'entraînant sur de grands ensembles de données, les CAE apprennent des motifs de bruit complexes spécifiques à différentes modalités d'imagerie, tout en préservant les détails anatomiques critiques. La méthode proposée vise à améliorer la clarté de l'image, en assurant un diagnostic fiable et de meilleurs résultats de soins de santé. Cette étude démontre le potentiel des ECA dans l'amélioration de la qualité de l'imagerie médicale, soutenant ainsi des évaluations médicales plus précises.

Key words: *Débruitage d'images médicales, Autoencodeurs, Autoencodeurs Convolutionnels (CAEs), Réseaux de Neurones Convolutionnels (CNNs),Réduction du Bruit, Imagerie Médicale, Traitement d'Image, Imagerie Diagnostique, Intelligence Artificielle*.

إزالة الضوضاء من الصور الطبية باستخدام نهج يعتمد على المشفرات الذاتية

ملخص :

أحدث الذكاء الاصطناعي تحسينات كبيرة في تشخيصات الطبية، وخاصة من خلال التصوير الطبي. ومع ذلك، غالبًا ما تعاني هذه الصور من الضوضاء بسبب عوامل مختلفة مثل تقليل التعرض للإشعاع. يعد تقليل الضوضاء بكفاءة أمرًا حاسمًا للتشخيص الدقيق. تتناول هذه الأطروحة مشكلة إزالة الضوضاء من الصور الطبية باستخدام المشفرات الذاتية الالتفافية .تستفيد المشفرات الذاتية الالتفافية من قوة الشبكات العصبية الالتفافية للتمييز الفعال بين الضوضاء والمعلومات التشخيصية الأساسية. من خلال التدريب على مجموعات بيانات كبيرة، تتعلم المشفرات الذاتية الالتفافية أنماط الضوضاء المعقدة الخاصة بطرائق التصوير المختلفة، مع الحفاظ على التفاصيل التشريحية الحرجة. تهدف الطريقة المقترحة إلى تحسين وضوح الصورة، مما يضمن تشخيصًا موثوقًا ونتائج رعاية صحية أفضل. تُظهر هذه الدراسة إمكانات المشفرات الذاتية الالتفافية في تحسين جودة التصوير الطبي، مما يدعم التقييمات الطبية الأكثر دقة.

الكلمات المفتاحية : *إزالة الضوضاء من الصور الطبية، المشفرات الذاتية، المشفرات الذاتية الالتفافية، الشبكات العصبية الالتفافية، تقليل الضوضاء، التصوير الطبي، معالجة الصور، التصوير التشخيصي، الذكاء الاصطناعي.*

.