ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DIAGNOSIS OF BRAIN CANCER AND CONTOUR NORMAL TISSUE FOR RADIATION THERAPY BASED ON DEEP LEARNING METHODS

Ph. D. THESIS

Navid HALILI

Department of Mechatronics Engineering

Mechatronics Engineering Programme

JULY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DIAGNOSIS OF BRAIN CANCER AND CONTOUR NORMAL TISSUE FOR RADIATON THERAPY BASED ON DEEP LEARNING METHODS

Ph. D. THESIS

Navid HALILI
(518162006)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Prof. Dr. Mustafa DOĞAN

JULY 2024

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİYLE RADYOTERAPİ İÇİN BEYİN KANSERİ TANISI VE NORMAL DOKU AYRIMI

DOKTORA TEZİ

Navid HALILI
(518162006)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mustafa DOĞAN

TEMMUZ 2024

Navid HALILI, a Ph.D. student of ITU Graduate School student ID 518162006, successfully defended the thesis/dissertation entitled "DIAGNOSIS OF BRAIN CANCER AND CONTOUR NORMAL TISSUE FOR RADIATION THERAPY BASED ON DEEP LEARNING METHODS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Mustafa DOĞAN** ....
İstanbul Technical University

Jury Members : **Prof. Dr. Mehmet Turan SÖYLEMEZ** ....
Istanbul Technical University

Assoc. Prof. Dr. Janset DAŞDEMIR ....
Yıldız Technical University

Prof. Dr. Müjde GÜZELKAYA ....
Istanbul Technical University

Assoc. Prof. Dr. Levent UCUN ....
Yıldız Technical University

Date of Submission : 21 May 2024
Date of Defense : 18 July 2024

To my family,

FOREWORD

This dissertation could not have been completed without the support, help, encouragement, and feedback of many people. Thus, I owe them a debt of gratitude.

First and foremost, my greatest appreciation goes to my advisors, Prof. Dr. Mustafa DOĞAN, for their continuous support, guidance, and encouragement from the very beginning of my graduate years. Their efforts and advice had a great influence and added value to my work.

I would also like to thank my former and current thesis committee members Prof. Dr. Mehmet Turan SÖYLEMEZ, Doç. Dr. Janset DAŞDEMIR for the valuable feedback; they have provided me throughout the progress meetings, and Prof. Dr. Müjde GÜZELKAYA and Doç. Dr. Levent UCUN for their valuable comments during the defense exam.

Special thanks to the doctors and staff of Bezmialem Alam Hospital in Istanbul for their valuable knowledge, cooperation and support during the collection of MRI data.

Endless gratitude to the people in my life who pushed me in my hardest times and gave me the strength to go on. My parents for their prayers, endless support, and love.

July 2024

Navid HALILI
(Mechatronics, MSc)

TABLE OF CONTENTS

Page

FOREWORD .... **ix**
TABLE OF CONTENTS .... **xi**
ABBREVIATIONS .... **xiii**
LIST OF TABLES .... **xv**
LIST OF FIGURES .... **xvii**
SUMMARY .... **xxi**
ÖZET .... **xxiii**
1. INTRODUCTION **1**
1.1 Background to the Research 1
1.2 Related work 5
1.3 Purpose and Scope of the Thesis 7
1.4 Thesis Outline 9
2. MACHINE LEARNING METHODS **11**
2.1 Introduction 11
2.2 Sobel Edge Detection 12
2.3 Canny Edge Detection 13
2.4 Morphological Segmentation 15
2.5 Watershed Segmentation 17
2.6 K Means Clustering 22
3. TRANFORMATION METHODS **25**
3.1 Wavelet Trsanform 25
3.2 The Contourlet Transform 29
4. DEEP LEARNING FOR IMAGES CLASIFICATION **39**
4.1 Introduction 39
4.2 Convolution Neural Network Architecture 40
4.3Residual Network 43
4.4 Resuidual Network For Clasification (ResNet-50) 47
4.4.1 ResNet50 architecture 47
4.4.2 Preparation and data augmentation 56
4.4.3 Model performance 58
5. DEEP LEARNING FOR IMAGES SEGMENATATION **61**
3.1 ResUnet Architecture 61
3.2 Data Preparation 63
3.3 Model Performance 67
6. GENERATIVE ADVERSARIAL NETWORK **71**
4.1 Introduction to GANs 71
4.2 Categories of GAN networks 74
4.3 DCGAN Model For Making MRI Images 75

7. CONCLUSIONS......**83**
7.1 Future Works......84
REFERENCES......**85**
CURRICLUM VITAE......**95**

ABBREVIATIONS

ANN	: Artificial Neural Networks
CNN	: Convolutional Neural Network
CONV	: Convolutional Layers
CT	: Computed Tomography scan
DCGAN	: Deep Convolutional Generative Adversarial Network
DNA	: Deoxyribonucleic Acid
DWT	: Discrete Wavelet Transforms
GAN	: Generative Adversarial Network
MRI	: Magnetic Resonance Imaging
RNN	: Recurrent Neural Network
SVM	: Support Vector Machines

LIST OF TABLES

Page

Table 2.1 : Sobel Mask Horizontal and Vertical matrix... **12**
Table 4.1 : The ResNet50 network architecture consists of the following layers..... **50**
Table 4.2 : Data set, train and validation set, test set **57**
Table 4.3 : Confusion matrix... **59**
Table 4.4 : Different Measures of the Confusion Matrix **59**
Table 5.1 : Data set, train and validation set, test set **63**
Table 5.2 : The specifications of all layers and learnable parameters in our U-net...**65**
Table 6.1 : The specifications of all layers in discriminator **78**
Table 6.2 : Internal structure and discriminating model layers. **78**

LIST OF FIGURES

Page

Figure 1.1 :Sample MRI and CT scan of the brain 2
Figure 1.2 :MR images of a patient with a brain tumor 3
Figure 1.3 :An example of a patient's MRI images analyzed by Siemens software ... 3
Figure 1.4: Three standard radiology imaging plates... 4
Figure 1.5 : The three majorly known standard MRI modalities 4
Figure 1.6 : Identify the location of the brain tumor to develop a treatment plan…...8
Figure 2.1 : Two sample images for preprocessing and Segmentation 11
Figure 2.2 : Sample images after preprocessing 12
Figure 2.3 : Implementation of Sobel filter... 13
Figure 2.4 : Image after applying Canny.. 15
Figure 2.5: Samples of the structuring element.. 15
Figure 2.6 : Examples of dilation.. 16
Figure 2.7 : Examples of erosion 16
Figure 2.8: Sample of closing and opening operations... 16
Figure 2.9: Implementation of morphological operations.. 17
Figure 2.10: Principle of the watershed. 18
Figure 2.11: Use the Gradient Magnitude as the Segmentation Function 18
Figure 2.12: Morphology opening 19
Figure 2.13: Morphology opening-closing.. 20
Figure 2.14: Regional maxima of opening-closing by reconstruction...................... 20
Figure 2.15: Regional maxima superimposed on original images............................ 21
Figure 2.16: Thresholding opening-closing by reconstruction 21
Figure 2.17: Results of watershed Segmentation 22
Figure 2.18: Sample segmentation with K-means 23
Figure 3.1 : The structure of a wavelet transforms 26
Figure 3.2 : Sample images 1……………………………………………………....26
Figure 2.3 : Results from the first decomposition level 25
Figure 3.4 : Results from the second decomposition level 25
Figure 3.5: Results from the first decomposition level 28
Figure 3.6 : Results from the second decomposition level 29
Figure 3.7 : The difference between the two transformations in edge detection 30
Figure 3.8: Contourlet structure 31
Figure 3.9: Visual representation of an image pyramid with five levels 32
Figure 3.10: The contourlet filter bank 32
Figure 3.11: Contourlet transformation with the different DFB and LP.................. 33
Figure 3.12: Decomposition of the first level in two directions............................... 33
Figure 3.13: Decomposition of the second level in four directions 34
Figure 3.14: Decomposition of the third level in eight directions 34
Figure 3.15: Decomposition of the first level in two directions............................... 34

Figure 3.16 : Decomposition of the second level in four directions **35**
Figure 3.17 : Decomposition of the third level in eight directions........................... **35**
Figure 3.18 : Decomposition results of the second level and find the borders and edges **36**
Figure 3.19 : Decomposition results of the second level and find the borders and edges **36**
Figure 3.20 : Decomposition results of the third level and find the borders and edges **37**
Figure 4.1 : Difference between machine learning and deep learning methods in classification **40**
Figure 4.2 : A sample CNN architecture.. **41**
Figure 4.3 : The difference in the error function in a 56-layer network compared to a 20-layer network **44**
Figure 4.4 : The phenomenon of descent or degradation in the network................. **44**
Figure 4.5 : Shortcut connections, the strength of the RESNET.............................. **45**
Figure 4.6 : Alternative path in RESNET network **47**
Figure 4.7 : The structure of VGG 19 and 34 layers plain **49**
Figure 4.8 : Sample images with nontumor label.. **56**
Figure 4.9 : Sample images with tumor label.. **57**
Figure 4.10 : Data augmentation for a sample MRI... **58**
Figure 4.11 : The result of accuracy in the ResNet50 model. **58**
Figure 4.12 : Examples of brain tumor classification results by RESNET 50.......... **60**
Figure 5.1 : Sample architecture of U-net **61**
Figure 5.2 : The Proposed RESUNET **62**
Figure 5.3 : Sample an MRI with a corresponding mask1 .... **64**
Figure 5.4 : Sample an MRI with a corresponding mask2 .... **64**
Figure 5.5 : Sample an MRI with a corresponding mask3 .... **64**
Figure 5.6 : The result of training the RESU-net mode **68**
Figure 5.7 : The result of accuracy in the RESU-net model **68**
Figure 5.8 : Image segmentation results from sample 1.. **69**
Figure 5.9 : Image segmentation results from sample 2.. **69**
Figure 6.1 : Structure of generator network **71**
Figure 6.2 : The backpropagation method is used to adjust each weight.................. **72**
Figure 6.3 : Backpropagation for update weight in discriminator network **72**
Figure 6.4 : Cost functions within the Generative Adversarial Network **73**
Figure 6.5 : Our sample data set without tumor **76**
Figure 6.6 : Our sample data set with tumor **77**
Figure 6.7 : Overview of our GAN…………………………………………………**77**
Figure 6.8 : The value of the productive and discriminating cost function in the first stage and the production sample …... **79**
Figure 6.9 : The value of the productive and discriminating cost function in the third stage and the production sample **79**
Figure 6.10 : The value of the productive and discriminating cost function in the sixth stage and the production sample **80**
Figure 6.11 : The value of the productive and discriminating cost function in the ninth stage and the production sample **80**

Figure 6.12: The value of the productive and discriminating cost function in the twelfth stage and the production sample **81**

DIAGNOSIS OF BRAIN CANCER AND CONTOUR NORMAL TISSUE FOR RADIATION THERAPY BASED ON DEEP LEARNING METHODS

SUMMARY

Brain tumors are one of the deadliest types of cancer ever identified. Rapid and accurate diagnosis of brain tumors, followed by surgical intervention or appropriate treatment, increases the probability of survival. Accurate identification of brain tumors in MRI scans allows precise location of surgical intervention or chemotherapy. Accurate segmentation of brain tumors in MRI scans is challenging due to their varied shapes and requires knowledge and accurate image interpretation.

This thesis starts with analyzing machine learning and traditional methods and focuses on the study of edge detection using the Sobel and Canny edge detector algorithm. After that, we use morphology-based techniques to segment the images and evaluate the results. We use K-means techniques for Clustering. Despite various advances, these methods still show limitations in complex situations such as tumor detection and segmentation.

In the next step, we analyze the process of dividing photos into parts using transformations. Specifically, we discuss the Wavelet and Contourlet transforms. By using these transformations, we get more detailed information about the analysis of the images. These transformations have many applications, and we can identify the borders of the image and combine them. Finally, we can use this transformation to process and generate deep learning masks using a supervised model.

In the following, we analyze new techniques using supervised and deep learning approaches in two specific areas: image classification and image segmentation. As we introduce these methods, we introduce the obstacles facing deep learning and discuss potential strategies to solve and enhance them. Using deep neural networks and the Resnet 50 model, we classify brain images into tumor and non-tumor categories and achieve a satisfactory score of 97% in the F1 criterion.

In addition, we introduce and analyze the Unet deep network in deep learning and upgrade it to a RESUNET network for segmentation. The results of this segmentation show that the proposed approach, with different criteria, such as the DICE metric with a score of 0.9434, performs exceptionally during training compared to conventional topologies and shows a faster convergence rate.

In the last part, we presented the unsupervised learning system and developed the adversarial generative network to generate brain MRI images. The adversarial generative network is an intelligent network for generating the desired data, and the results show the effectiveness of the adversarial generative network in generating new data. It is of exceptional quality.

DERIN ÖĞRENME YÖNTEMLERIYLE RADYOTERAPI IÇIN BEYIN KANSERI TANISI VE NORMAL DOKU AYRIMI

ÖZET

Beyin tümörlerinin neden olduğu kanserler en ölümcül hastalıklar arasındadır. Beyin tümörlerinin hızlı ve doğru tanısı, cerrahi müdahale veya uygun tedavi sonrasında hayatta kalma şansını arttırabilir.

MR taramasında beyin tümörlerinin kesit alanının doğru tanımlanması, cerrahi müdahale sırasında, kemoterapi ve radyoterapi tedavilerinde verimlilik açısından oldukça önemlidir. Beyin tümörlerinin MR taramalarında doğru şekilde bölütlenmesi, çok çeşitli morfolojik özellikler gösterebildiği için oldukça zordur. Bu nedenle içerilen bilginin karakteristiklerinin ve görüntünün doğru yorumlanması gerekir. Aslında bu bölütleme ne kadar doğru yapılırsa, hastanın radyasyon tedavisinin zararlı yan etkilerine maruz kalması o kadar azalır.

Son yıllarda bilgisayar biliminin yaygınlaşması ve yeni teknolojilerin ortaya çıkmasıyla birlikte bu alanda pek çok araştırma yapılmıştır. Bu çalışmaların ana odağı tümörün yerini, kesit alanını belirlemek, sınıflandırmak ve son olarak görüntüleri bölümlemektir. Temel olarak görüntü bölütleme analizi piksel bazında yapılır.

Bu tez, makine öğrenmesine karşın geleneksel yöntemlerin tanıtılması ve analizi ile başlamakta, kenar algılama algoritmalarından ikisi olan Sobel ve Canny kullanılarak örnek görüntüler üzerinde kenar tespiti gerçekleştirilmektedir. Elde edilen sonuçlar genel uygulamalar için kabul edilebilir olsa da yüksek doğruluk gerektiren beyin MR görüntüleri için kabul edilemez.

Morfolojik operatörler kullanılarak ve çıkan sonuçları değerlendirerek görüntüleri bölümlendirmek için morfoloji tabanlı tekniklerin birleşimini tasarladık. Görüntü bölütlemede morfolojik işlemler daha kabul edilebilir sonuçlara sahiptir ve bir sonraki adımda görüntüleri bölütleme için Watershed adı verilen başka bir morfolojik yöntemi denedik. Son olarak kümeleme yöntemlerinin bir örneği olan görüntüleri bölütleme için K-means algoritması uygulanarak karşılaştırma yapılmıştır. Sayısız ilerlemeye ve çok sayıda yapılan çalışmaya rağmen, bu yöntemler henüz tümör tanısı ve segmentasyonu gibi karmaşık problemleri çözmede yeterli sonuçlar ele edememişlerdir. Bu nedenle yıllar geçtikçe ve bilgisayar bilimindeki hızlı gelişmelerle birlikte, bizim de odaklandığımız birçok yeni yöntem geliştirilmiştir.

Görüntü işlemede dalgacık dönüşümünü inceledik ve Contourlet yöntemiyle karşılaştırdık. Wavelet kenar tespiti için güçlü bir araçtır; ancak Contourlet dönüşümü sadece iki yönde değil, her yönde özellik çıkarımı yaparak tümörün özgün şekline daha iyi yakınsamaktadır.

Bu dönüşümler oldukça kullanışlıdır ve görüntünün kenarlarını istenilen yönlerde tanımlayıp birleştirebiliriz. Son olarak, Contourlet dönüşümünü derin öğrenme maskelerini işlemek, etiketlemek ve sınıfları oluşturmak için de kullanabiliriz. Bu maskeler kontrollü bir derin ağa giriş olarak kullanılabilir. Dönüşümlerin fotoğrafları birleştirme ve fotoğrafların boyutunu küçültme konusunda da kısaca inceleyeceğimiz başka uygulamalar da kullanılmıştır.

Derin öğrenme yöntemlerini, geleneksel yöntemlerle karşılaştırarak farklılıklarını anlatıp, görüntülerin sınıflandırılması ve bölünmesindeki uygulama detaylarını saptayacağız. Ardından geliştirilmeye açık katmanların iç yapısını ve bilgilerini genel olarak anlatacağız. Derin sinir ağı farklı katmanlar oluşmaktadır, her katmanın kendi rolü ve görevi vardır.

Geleneksel yöntemlerden farklı olarak derin öğrenme yöntemlerinde görüntüden istenen özelliklerin çıkarılmasına gerek yoktur ve aslında gerekli özellikler bu ağların kendileri tarafından otomatik çıkartılır.

Derin sinir ağlarının tüm avantajlarına rağmen kendi sorunları da vardır. Derin sinir ağlarının karşılaştığı engelleri belirterek, RESNET kullanan derin sinir ağlarının parametrelerini iyileştirdik. Bu yöntem sayesinde birçok katman içeren derin sinir ağları oluşturabiliyoruz. Derin sinir ağlarını ve ResNet 50 modelini kullanarak beyin görüntülerini tümörlü ve tümörsüz olarak iki kategoriye ayırıyoruz. Performansı ölçmek için kullanılan metriklerden biri olan F1 kriterinde 97% gibi tatmin edici bir sonuç elde edilmiştir.

Bir sonraki bölümde görüntülerin bölütlenmesini işleyeceğiz. Görüntü bölütleme, sınıflandırmaya benzer şekilde denetimli eğitim yöntemini kullanır. Bu görev için çeşitli yöntemler ve RESNET gibi çeşitli ağ yapıları önerilmiştir.

Tezimizde görüntü bölütlemede iyi sonuçlar veren Unet denenmiştir. Fakat performans ölçütlerinde yeteri kadar iyileşme sağlamadığı gözlemlenmiştir. Bu nedenle, MR beyin görüntülerini segmentlere ayırmak için RESUNET ağını uyguladık. Segmentasyon sonuçları, önerilen yaklaşımın eğitim sırasında iyi performans gösterdiğini ve bölütleme performansını gösteren DICE metriği gibi çeşitli metriklerle geleneksel topolojilere kıyasla 94% gibi yüksek bir skorla başarılmıştır.

Son bölümde denetimsiz öğrenme sistemini, GAN ağ yapısını tanıtıyoruz ve akıllı ve denetimsiz bir ağ olarak Çekişmeli üretici ağını (GAN) kullanarak resimleri işledik. Bu ağı beyin MR görüntülerinin sayısını arttırmak için geliştirdik. Aslında bu ağ, jeneratör ve ayırıcı ağ olmak üzere iki derin sinir ağından oluşur. Bu iki ağ birbiriyle rekabet halindedir; jeneratör ağının görevi, orijinal verilere benzer veriler üretmektir. Ayırıcı ağın görevi ise bu sahte verilerle rekabet ederek doğru olanı bulmaktır. Çekişmeli üretici ağı, istenilen verinin çoğaltılabilmesi için akıllı bir ağ yapısı olup; örnek imge bulmakta zorlandığımız tıbbi görüntüleme uygulamalarında, istenilen veri kümesinin genişletilmesi için başarılı bir yöntem olarak kullanılmıştır. Üretici ağın oluşturduğu sahte verilerin orijinal verilerle benzerliği, bu ağın yeni veri üretmedeki başarısının ölçütüdür.

1. INTRODUCTION

1.1 Background to the Research

Cancer is one of the most common diseases today. Brain cancer is one of the deadliest types of cancer, and its correct and timely diagnosis will have a significant impact on a patient's life. [1]

Any group of abnormal cells that take up space in the brain produces a brain tumor. More than 40 types of brain tumors fall into two general categories: Benign tumors: These tumors grow slowly and do not spread. These tumors are not cancerous. Malignant tumors: These tumors are cancerous, overgrown, and can spread to other parts of the brain and spinal cord. There are several treatment options for people with brain tumors. Surgery, radiotherapy, and chemotherapy are some of these methods. In some patients, a combination of these treatments is needed. The type of treatment is determined according to the following: 1. Tumor type 2. The site of tumor formation in the brain 3. The size of the tumor and its spread 4. Physical health and weight of the patient. Surgery is usually the first option for treating brain tumors. In this procedure, the skull is opened, and the tumor is removed. Only the tumor is removed if it has formed near sensitive brain areas. However, removing part of the tumor also helps relieve symptoms. However, sometimes surgery is not possible to remove the tumor, and if surgery is performed, the risk of brain tissue damage increases. In such cases, radiotherapy will be considered.

Radiotherapy uses high-energy waves to kill tumor cells. Radiation therapy uses high-energy X-rays to disrupt, inhibit, or control DNA growth in cancer cells. Chemotherapy, the third approach, involves administering one or many chemotherapy medicines. Chemotherapy is typically administered to combat cancer by utilizing a blend of medications to extend the patient's lifespan, impede the spread of cancer, and alleviate specific disease symptoms as a palliative measure. In all three treatments, MRI (magnetic resonance imaging), CT scans, and computed tomography) are used to determine the size and location of the tumors. Depending on the patient's condition, one may be a better option for diagnostic imaging. For example, at the doctor's discretion, sometimes a CT scan and an MRI of the brain are prescribed. MRI is usually used for soft tissue; CT scans identify and see the bones. For example, an MRI image and CT scan of the brain are shown in Figure 1.1.

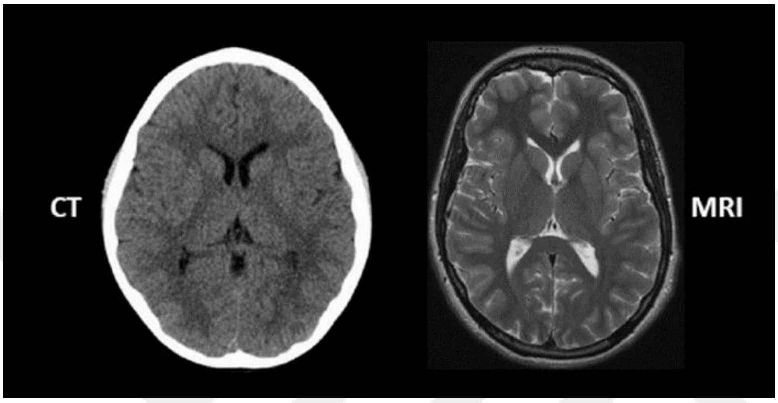

Figure 1.1 : Sample MRI and CT scan of the brain [2].

Doctors usually analyze brain tumors, but there is no way to find brain tumors that can be used as standard. In recent years, to solve this problem, radiologists have obtained information such as the location of the tumor through magnetic resonance imaging and used it to diagnose the tumor and provide a plan for surgery and radiotherapy. However, identifying the exact location and size of the tumor is not easy; sometimes, it is possible only by comparing many images from different angles, and high accuracy is required. For example, in Figure 1.2, three images were taken of a patient with a tumor, and Identification of the tumor and its location is possible only by comparing the images .We have a sequence of MRI images that should be considered for the cancer site. Today, various methods have been proposed to find the exact location of the tumor to reduce human error and increase accuracy with the development of computer science, each of which had different results.

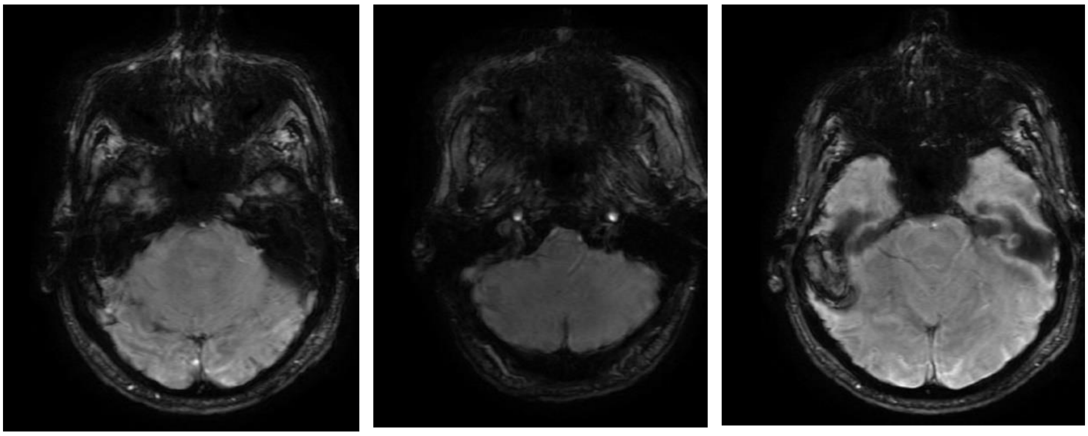

Figure 1.2 : MR images of a patient with a brain tumor.

Big medical data consists of genetic information, medical picture data, electronic health record data, etc. Most of the medical data at this point comprises medical image data. We have a large number of medical images to diagnose brain cancer, and we have at least 22 MRI images for each patient. An example of a patient's MRI images analyzed by Siemens software (syngo fastView) is shown in Figure 1. 3. The size of each image is 512 by 512 pixels, approximately the amount of information taken from Bezmialem of 65 patients, which is about 1430 photos with a size of 1.5 GB.

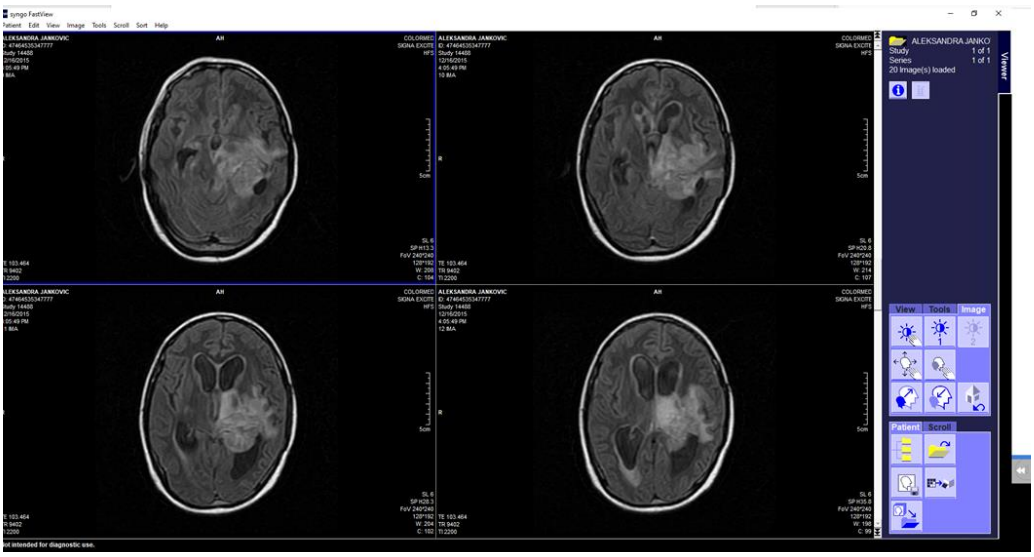

Figure 1.3 : An example of a patient's MRI images analyzed by Siemens software (syngo fastView).

Regular X-rays do not yield as high-quality data as magnetic resonance imaging; radiologists and medical professionals sometimes find it challenging to identify and segment brain tumors due to their intricacy and the rapid proliferation of cancer cells. After reviewing them, we realized that all these photos are unsuitable for us to use in training the machine learning algorithms and extracting image features. We only need MRI images taken from the axial plate. Three standard radiology imaging plates are shown in Figure 1.4.

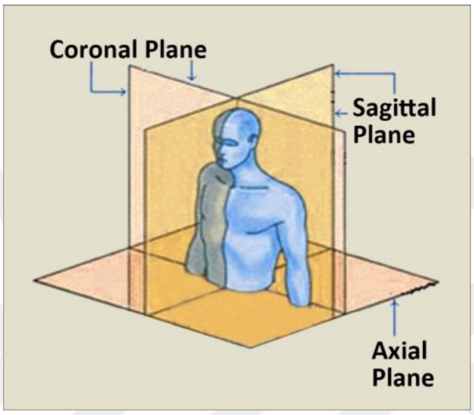

Figure 1.4 : Three standard radiology imaging plates [3].

Naturally, not every image obtained from the brain's axial plate can be used for machine learning. T1-weighted (T1), T2-flair MRI (T2), and T1-inversion recovery are the three most widely used standard MRI modalities. An example of these photos is given in Figure 1.5, and we need to start with one of these three standard photos for each patient. Choosing the right images for each patient will reduce our data size, and we can train the desired system with the correct data.

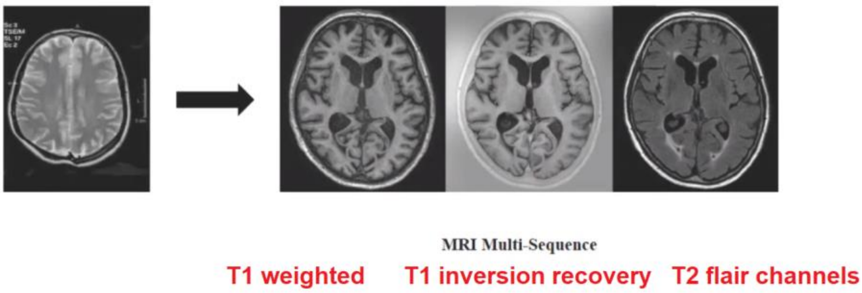

Figure 1.5 : The three majorly known standard MRI modalities.

1.2 Related work

In recent decades, much research has been done on the rapid development and growth of computers, the importance of brain cancer, and identifying the exact location and size of the tumor. Image segmentation is the key to image understanding.

Image segmentation is used today in many fields, including the design of autopilot vehicles or drones [4], [5], biology and pathology [6], [7], or even in mechanics to detect corrosion in structures [8], or computer engineering for three-dimensional reconstruction of the scene[9], [10]. There are several methods of image segmentation, such as thresholding [11], [12], [13], clustering methods, and edge detection algorithms, which are commonly referred to as classical methods. Another category of image segmentation methods is based on neural networks or artificial [14], [15], which are considered today due to their greater accuracy. However, no technique is suitable for all programs, and the choice of method is usually based on the needs of the issue.

There are many research-based edge detection algorithms for MRI images of the brain [16], [17], [18].in [18], the author compares the different edge detection algorithms for brain tumor detection. Then, the authors conclude that Sobel is suitable for the case of brain tumors between Robert, Prewitt, and Sobel. Also, in [16], try to identify the tumor's location with the Sobel accurately. In [17], [19], the Canny edge detection algorithm is used to detect brain tumors, and in [17], the tumor area is calculated approximately. Despite the widespread use of edge detection filters, there are many articles on improving these methods, mainly due to the average result [15].

Another effective way to diagnose tumors is to use morphological operators to detect edges. Morphological operators are usually used in the preprocessing stage, and there is much research on MRI images because of the acceptable results of these operators [20], [21], [22], [23]. Morphological methods have better results than others, such as Sobel or Canny [21], [24]. In[25], the authors used the clustering method, and finally, in post-processing, the tumor was extracted using morphological and region properties operations.

Clustering methods are one of the most successful ways in the Segmentation of brain images, which has been researched for many years and can be used both to diagnose the type of cancer in the [26], [27] and to identify the location of cancer in [28], [29],

[30]. However, in [26] and [31], clustering methods are used to preprocess images, and an artificial neural network is used to categorize tumors. In[32], [33], instead of using the k-means method, the Fuzzy Clustering Method (FCM) has been used, and, in FCM, any point in the image, unlike the K-means, can be associated with several centers. The major drawback of the FCM is the enormous computational time required for the convergence, and in [34], authors, to overcome this problem, modified the FCM kernel for MR brain tumors.

A review of numerous sources and articles reveals that research has been conducted since 2016 to detect brain tumors through deep learning methods, for example in the article[35], the authors use a standard database of the tumor division 2013 challenge database and propose an automated splitting method based on convolutional neural networks (CNN). Also, better results can be obtained by combining image preprocessing methods and applying the convolutional neural network (CNN) to more data. For instance, in [36], CNN has 87.42 %training accuracy. In [37], the result of CNN was used to classify the MRI images, and an accuracy of about 97.92% was achieved. In [38], multi-level thresholding is applied to segment the tumor region before applying CNN to improve network performance in learning, or in [39], image preprocessing and morphological operations are used to increase the learning speed in CNN.

With the expansion of deep learning network architectures, fresher architectures have increased network speed with better accuracy. In [40], the AlexNet model is used to classify different types of tumors; in [41], VGG-16 architecture is used to detect the tumor. Also, the outcomes of the CNN model and VGG-16 architecture are compared. In[42], Several deep learning structures are compared, and the authors conclude that ResNet-152 architecture is the most efficient for brain tumor detection. Deep learning methods are not always used for the diagnosis of the location of brain cancer but are sometimes used to classify the type of brain cancer[43], [44], [45], [46], [47]. Also, in [48], [49], the CNN was used for the classification and detection of cancer in [50]; instead of using the standard database for training the network, In order to provide an extensive data set at a lower cost, the authors employed a preprocessing technique called Deep Convolutional Generative Adversarial Network, in which a generator generates fictitious images that are intended to deceive the discriminator into thinking they are authentic images.

Another method in image segmentation is the Contourlet transformation. In[51], the authors used a probabilistic neural network and contourlet transform for dimensionality reduction and feature extraction. In recent years, contourlet transformation has also been used for image segmentation [52].

Medical image fusion integrates the essential features of different medical images into a single image to improve clinical accuracy and make better decisions [53]. Over the years, several methods have been used for the image fusion section. In [54], discrete wavelet transform is applied to the images. Deep learning methods are used in some papers [55], [56], and [57], the authors used a generative adversarial network instead of CNN. Also, Fuzzy type 2 is another method used for image fusion [53], and experimental results show that fuzzy logic type 2 performs better than the neural network-based approach.

Examining all the sources and articles, it can be said that research in the field of deep learning for cancer diagnosis is still being developed and researched.

1.3 Purpose and Scope of the Thesis

In this thesis, real-time MRIs of each patient are taken, and then, using the image preparation methods, preprocessing is done on the images. After Segmentation and deep learning, the tumor's exact location is identified and marked. Then, artificial intelligence combines MRI and CT scans to draw normal tissue and all organs and tumors accurately. As a result, after determining the tumor's location and contouring normal tissues, the radiotherapy device's radiation for different parts can be defined differently. As a result, the highest amount of radiation is applied to the tumor and the lowest amount to normal tissue, and the best treatment result is obtained with the minor side effects. Figure 1.6 shows a real example of an MRI image of a brain cancer patient in Bezmialem Hospital and how to determine the location of the tumor on the patient's body manually.

Figure 1.6: Identify the exact location of the brain tumor to develop a treatment plan.

Following a summary of the various therapies and after consulting with the Medipol Hospital's radiation physicians, we inquired about their needs and the limitations of the treatments offered. Physicians discussed the adverse effects of radiation on healthy organs and tissues. They underlined the need for an intelligent system to be used in conjunction with radiation therapy to pinpoint the malignant tumor's precise location and size and shape the patient's healthy tissues and other organs by their unique ossification.

A significant portion of machine learning and machine vision will be used in the thesis; for instance, deep learning network architecture, image registration, contouring with artificial intelligence, image fusion, edge-based, region-based, clustering-based, artificial neural network-based, and image preparation techniques will all be investigated.

1.4 Thesis Outline

The subsequent chapters are outlined as follows:

The initial chapter, known as the introduction, provides an overview of the research's context and the medical aspect. This section provides a concise overview of MR images. We will discuss this thesis's objectives and scope after thoroughly examining related literature.

The second chapter will explore conventional edge detection techniques and two essential and practical methods: the Sobel and Canny filters. Ultimately, we will present morphological operators. Subsequently, we will employ the watershed approach and, ultimately, the fuzzy clustering method to segment the image for image segmentation.

In the third chapter, two renowned image-processing transformations and their applications in edge detection are introduced.

The fourth chapter explores the implementation of deep learning techniques, highlighting their distinctions from conventional approaches, the challenges deep networks encounter, and the corresponding remedies. ReseNet 50 is utilized to classify photos into two distinct categories: cancer and non-cancer.

In Chapter 5, we will explore the segmentation of deep networks specifically designed for medical images after introducing deep networks. We will also demonstrate how to segment medical images by combining two deep neural networks.

The sixth chapter presents a new deep network that relies on unsupervised learning. We utilize this network to produce MR pictures.

Lastly, we provide a concise overview of the techniques employed.

2. MACHINE LEARNING METHODS

2.1 Introduction

This section will briefly review segmentation methods and apply some of the most critical methods to the sample images. We use the following two images to apply filters and image segmentation. Figure 2.1 displays the sample images.

Figure 2.1: Two sample images for preprocessing and Segmentation.

Image preprocessing is used to improve the quality of the image by removing irrelevant and unwanted parts of the images [34]. Contrast usually provides vital insights for accurate diagnosis, description, and treatment of tumors and is commonly used in medical imaging. Therefore, we first increase the contrast using MATLAB software. Secondly, we will sharpen the edges of the images. Noise removal is also done in the preprocessing stage. The results of increasing the contrast and sharpening the edge of the two sample images can be seen In Figure 2.2.

Figure 2.2 : Sample images after preprocessing.

Since the edge is the line that separates an object from its background, edge detection is one of the most often used techniques in image analysis. The more significant the change in level, the easier it will be to detect the edge. The image's points with sudden brightness changes are often called edges or edge points. There are several ways to do this, including Soble, Canny, Roberts, Prewitt, Zerocross, and Laplacian of Gaussian. For example, we examine the results of some of the most popular of these algorithms.

2.2 Sobel Edge Detection

This method finds the edges using derivative estimation, which returns the edges at points where the gradient of the image I is max. The horizontal Sobel mask primarily identifies the horizontal edges, and the vertical Sobel mask defines the vertical edges. The Sobel masks are shown in Table 2.1.

Table2.1: Horizontal and Vertical Sobel Mask

	G_x			G_y	
-1	**-2**	**-1**	**-1**	**0**	**1**
0	**0**	**0**	**-2**	**0**	**2**
1	**2**	**1**	**1-**	**0**	**1**

If the horizontal and vertical mask filters the G_x and G_y images, then the image $G=[G_x]+[G_y]$ shows the edges of the image better. The above procedure is called the Sobel operator or algorithm. The results of this method can be seen in the sample images in Figure 2.3.

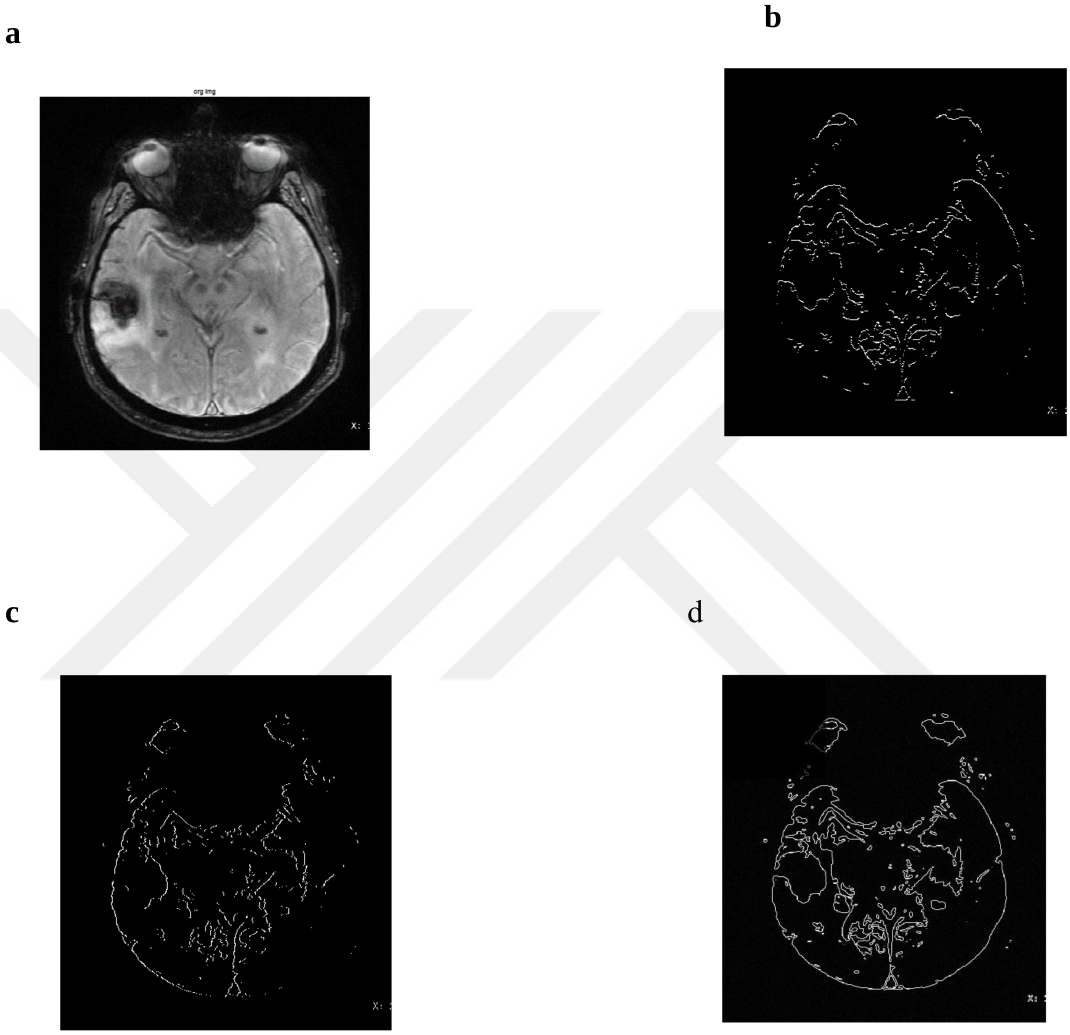

Figure 2.3: Implementation of Sobel filter. a. Original images b. Image after applying horizontal filler c. Image after applying vertical filter d. The final image.

2.3 Canny Edge Detection

The Canny edge detection method is an edge detector that locates a variety of edges in an image using a multi-step algorithm. The four steps involved in edge identification in the Canny method are as follows:

1. Noise Reduction: Noise filtering is crucial to preventing noise detection since noise in the image can easily impair all edge detection findings. A Gaussian kernel is twisted with the image to smooth it out. In order to lessen the impact of acute noise on the edge detector, this step gradually smoothest the image.

2. Gradient Computation: The edges indicate the pixel intensity variation. To enhance Identification, the most efficient approach is to employ filters that accentuate this alteration in both the horizontal and vertical orientations. Therefore, we utilize a derivative filter like Sobel. Therefore, the derivative is calculated in the horizontal direction G_X and the vertical direction G_Y. The slope intensity and edge direction calculations are performed using equations (2.1) and equations (2.2), respectively.

$$\text{Edge Gradient (G)}= \sqrt{G_x^2+G_y^2} \tag{2.1}$$

$$\text{Angle}(\alpha)= \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2.2}$$

3. Non-Max Suppression: After applying the Sobel and calculating the edge gradient, the thickness of the edges is not the same. Some edges are thick, while others are thin, and unwanted pixels may not be the answer. Non-maximum suppression is used to overcome this problem. In this step, it is checked for each pixel in the neighborhood in the direction of the maximum edge gradient. Otherwise, the pixel value is zero.

4. Hysteresis Thresholding**:** This step is used to identify pixels to determine whether they are edges .To do this, we must define two upper and lower thresholds, and each edge with an intensity more significant than the upper threshold is undoubtedly an edge. Each edge below the lower threshold is not an edge and is discarded. The edges between these two thresholds are classified based on the connection of sure-edges or non-edges, i.e., if they are connected to the boundary edge pixels, they are considered as edges. Otherwise, they are left out. The results of this method can be seen in the sample images in Figure 2.4.

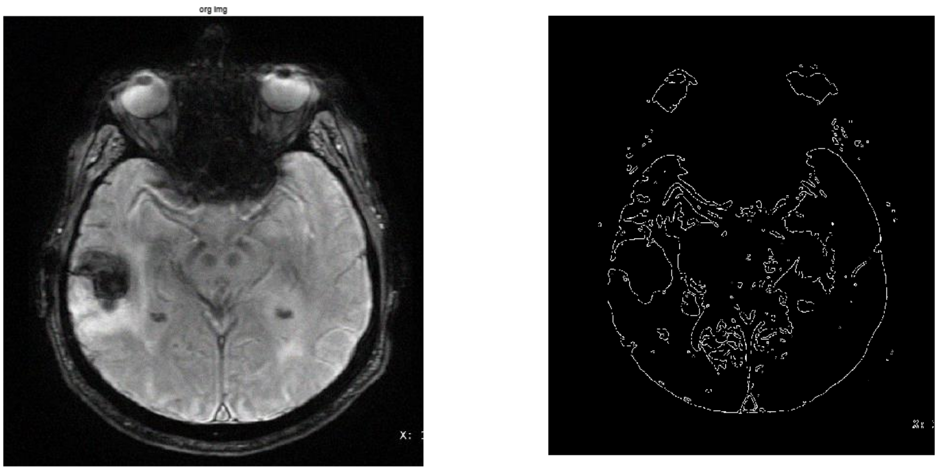

Figure 2.4 : Image after applying Canny.

2.4 Morphological Segmentation

Morphological image processing involves a collection of nonlinear procedures focusing on the shape or morphology of features within an image. Morphological techniques examine an image using a condensed shape or design referred to as a structuring element. Figure 2.5 displays various instances of this structural feature.

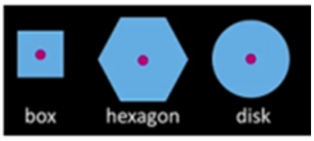

Figure 2.5 : Samples of the structuring element.

The structuring element is placed at every conceivable point inside the image and compared with the matching surrounding pixels. In this discussion, we will concisely analyze the two morphological operators: dilation and erosion. Naturally, the operations performed on the binary set include distinct formulas and relationships. To provide a concise overview, we will merely provide a broad explanation and outline the impact of each operation on the photos. The Dilation procedure enlarges the binary image while also filling up any holes and repairing any damaged borders. Figure 2.6 displays instances of this dilatation.

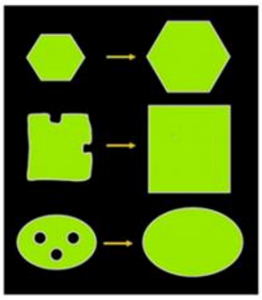

Figure 2.6: Examples of dilation[58].

Erosion is the inverse operation of the Dilation procedure. The process involves analyzing the binary image, reducing its size, and eliminating extraneous components. Figure 2.7 displays illustrations of this operation.

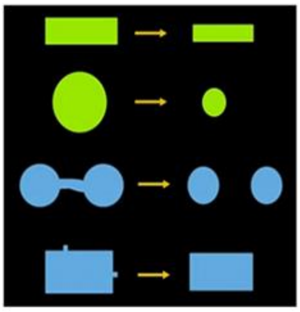

Figure 2.7 : Examples of erosion[58].

Typically, a mixture of the two procedures above is employed for another operation, the most crucial function performed on binary images. The combined operation of erosion and dilatation is referred to as opening, while the inverse of this process is shutting. Figure 2.8 illustrates this operation.

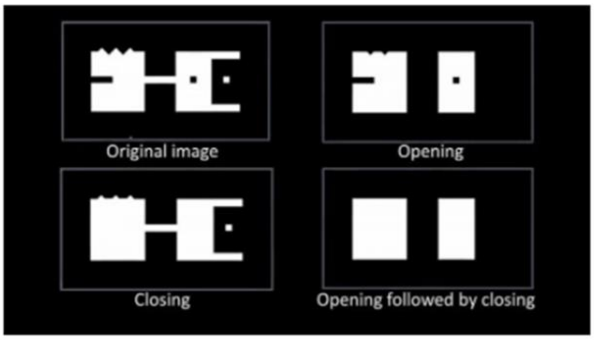

Figure 2.8 : Sample of closing and opening operations[58].

By applying the opening and closing in MATLAB, the results of this method in our sample image can be seen in the sample images in Figure 2.9**.**

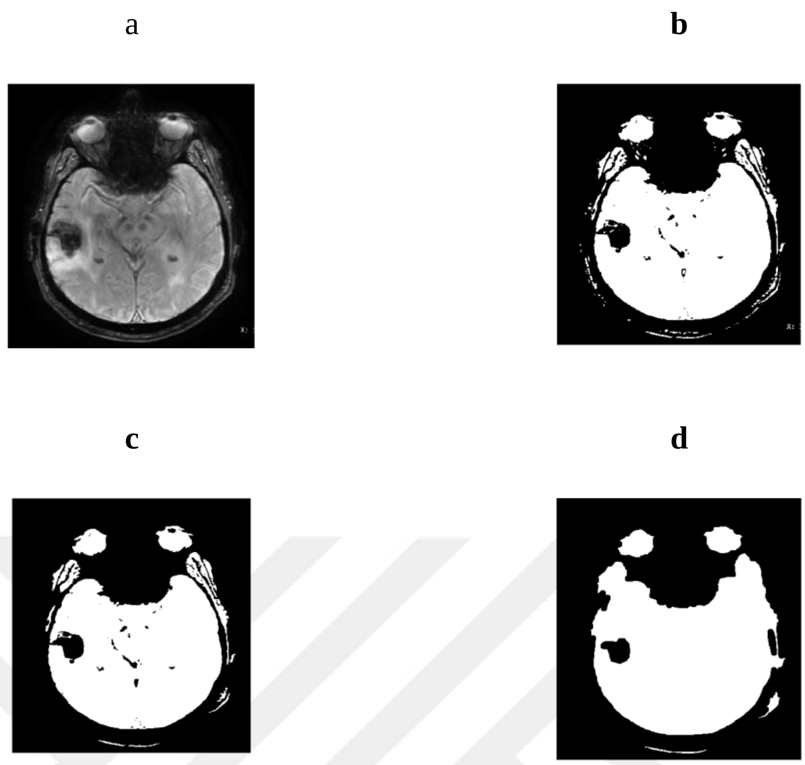

Figure 2.9: Implementation of morphological a. Original images b. Binary image after applying horizontal filler c. Image after opening d. Image after closing.

2.4.1 Watershed Segmentation

From the proposal section, we know that the results of applying morphological operators on the image were acceptable. For this purpose, we will study and apply the watershed method more.

Watershed Segmentation treats a picture as a topographic map with hills and valleys. The elevation values of the landscape are usually determined by the gray values of the pixels or their gradient magnitude. The watershed transform utilizes a 3D representation to partition an image into catchment basins. A catchment basin encompasses all sites that reach a local minimum by following the line of sharpest descent. Figure 2.10 illustrates the principle of the watershed, which demonstrates how watersheds act as boundaries between different basins. The watershed transform algorithm fully decomposes an image, categorizing each pixel as belonging to a specific region or a watershed.

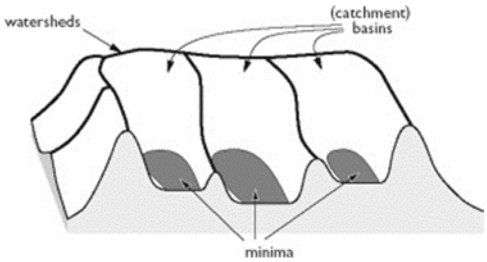

Figure 2.10 : Principle of the watershed [59] .

A watershed is where the runoff from rainfall is gathered by streams and channeled towards a discharge point, such as a river, reservoir, lagoon, lake, or sea. The watershed encompasses all the surfaces over which surface water is directed towards a particular point or location. Firstly, we compute the magnitude of the gradient. The gradient exhibits a pronounced increase near the perimeters of things and a decrease (more significant) within the objects. Applying the watershed transform directly to the gradient magnitude of the image does not produce appropriate outcomes. Using the watershed transform without extra preprocessing, such as the marker computations discussed below, can occasionally result in excessive Segmentation. The outcome is displayed in Figure 2.11.

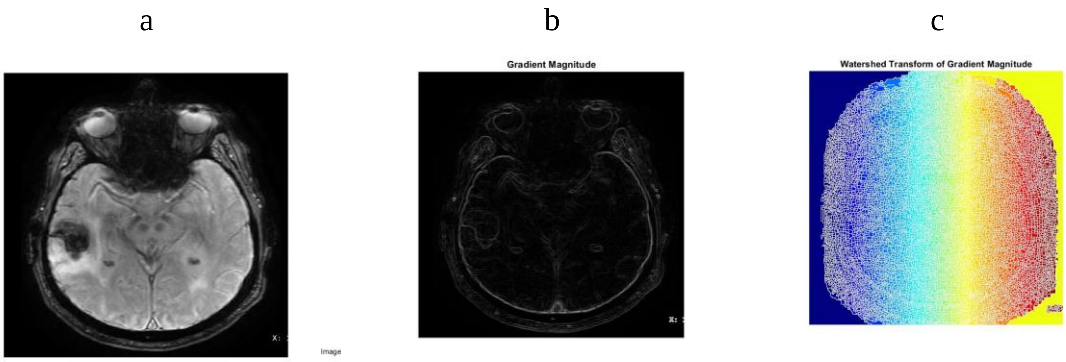

Figure 2.11: Use the Gradient Magnitude as the Segmentation Function a.Original image b. Gradient magnitude c. Over-segmentation.

The photo above shows that the presence of noise and intensity changes is a disturbing factor in highlighting and classifying sections because, in addition to the fact that there are many points, posts, and heights, the intensity of the photo is distributed in the form of area. For example, the inner part of the brain and the cortical part are classified as a class, and a total of 5 types of classes are classified, so the different parts must be separated from each other. Therefore, morphological instructions are used to improve image contrast. To prevent over-segmentation, we perform a morphological operation. In this example, we will use morphological techniques called "open with reconstruction" and "close with reconstruction" to "clean" the image. This operation creates a flat maximum inside each object. We know that opening is an erosion followed by a dilation, while opening-by-reconstruction is an erosion followed by a morphological reconstruction. The results of opening by reconstruction and standard opening are shown in Figure 2.12.

a b

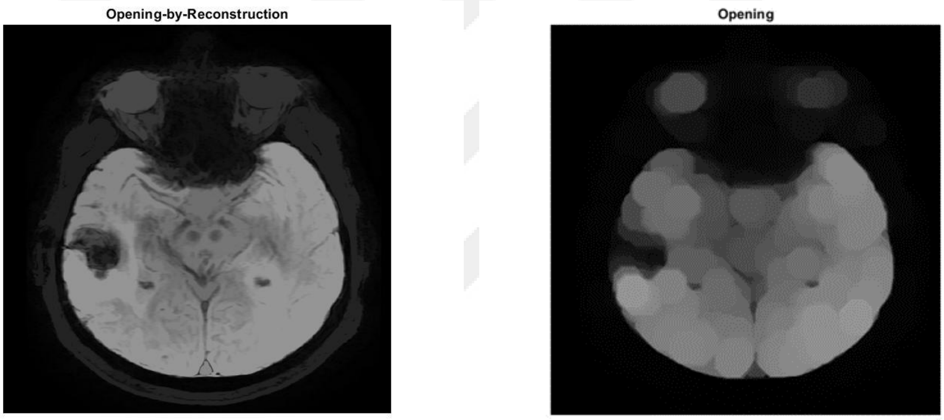

Figure 2.12: Morphology opening. a. Regular morphological closing b. Closing-by-reconstruction.

The dark blemishes and stains on the stem can be eliminated by opening and shutting the stem. A regular morphological closing can be compared to a closing-by-reconstruction. Figure 2.13 depicts.

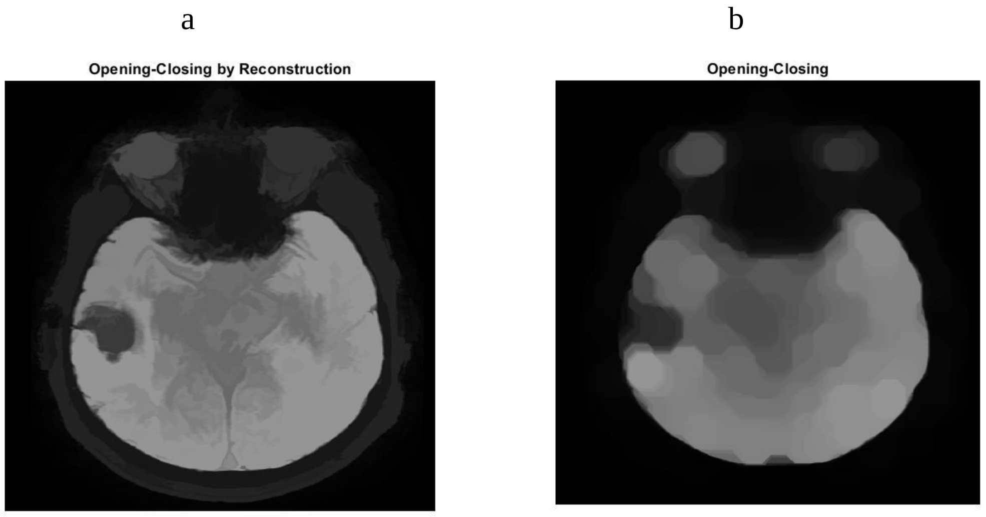

Figure 2.13: Morphology opening-closing. a.Opening-closing by reconstruction closing b. Regular opening-closing.

Upon comparing images, it becomes clear that reconstruction-based opening and closing approaches are more effective than ordinary opening and closing procedures in removing minor flaws without changing the main shapes of the objects. Currently, we employ reconstruction to accurately get foreground markers by computing the regional maxima of the opening-closing operation. The result is shown in Figure 2.14.

Figure 2.14 : Regional maxima of opening-closing by reconstruction.

In order to facilitate the interpretation of the outcome, overlay the foreground marker image onto the original image. The outcome is displayed in Figure 2.15.

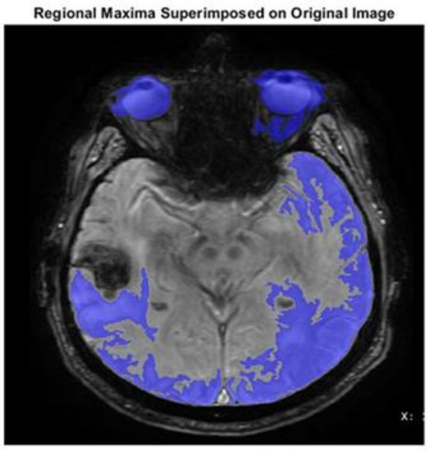

Figure 2.15 : Regional maxima superimposed on original images.

In order to determine the background markers, we can use a thresholding technique on the cleaned-up image, as we know that the dark pixels correspond to the background. The outcome is displayed in Figure 2.16.

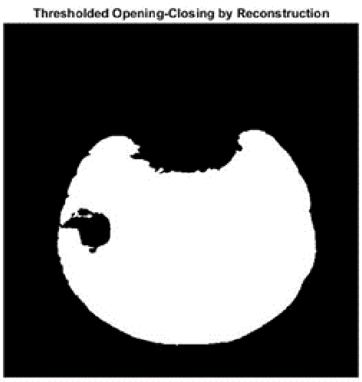

Figure 2.16 : Thresholding opening-closing by reconstruction.

At last, all the necessary preparations have been made to compute the Watershed Transform of the Segmentation Function. The outcome is depicted in Figure 2.17

a b c

Figure 2.17 : Results of watershed Segmentation. a. Markers and object boundaries b. Colored watershed label matrix c. Colored labels superimposed transparently on the original image.

This visualization illustrates how the placements of the foreground and background markers affect the result. The absence of foreground markers caused the combination of partially obscured darker objects with brighter surrounding objects. This method is inadequate for only recognizing tumors, and the differentiation between background and foreground can modify the segmentation results.

2.5 K-means Clustering

Clustering is an unsupervised learning method that divides data into separate clusters, where the data within each cluster show the most significant resemblance. On the other hand, the data gathered in separate clusters show the most notable differences. The K-means algorithm is commonly used in standard clustering approaches. The K-means algorithm is a simple and effective clustering approach. This approach employs a parameter called k, which sets the number of clusters to be obtained. The basic K-means algorithm is described as follows:

1. Select k data as the center of the cluster (Image pixels are our 3-dimensional data)

2. Repeat the third to fifth steps until there is no change in the clusters.

3. Determine the distances of the rest of the data from the center of the clusters. The object function in equations (2.3) defines this distance.

$$\sum_{i=1}^{k}\sum_{x\in s_i}||x-\mu_i|| \tag{2.3}$$

K is the number of clusters, μ_i is the mean of each cluster, s_i is the number of clusters, and X is each point of images.

4. The data closest to the center of each cluster is located in that cluster.

5. We consider the average of each cluster as the new center of the cluster

Since in this clustering method, the distance between points is calculated by the Euclidean distance function, these algorithms can be used as standard only for numerical values (and not qualitative properties). On the other hand, due to their simple and fast calculations, it is widely used and effective. On the other hand, generalized versions of the k-mean clustering method perform clustering steps with other distance functions such as Manhattan distance. The results of this method can be seen in the sample images in Figure 2.18.

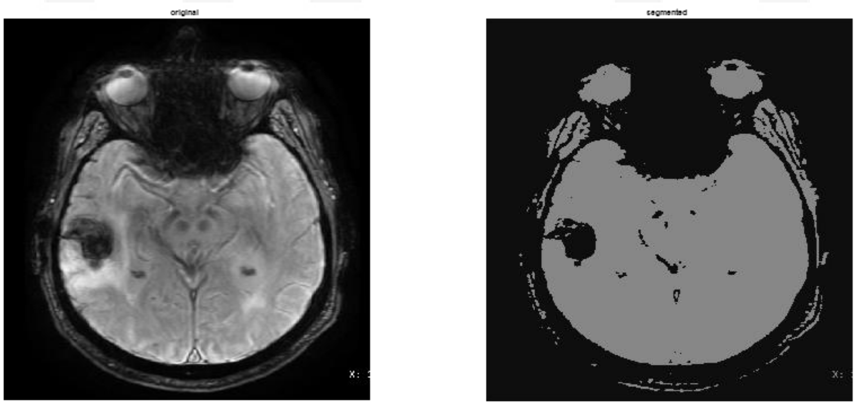

Figure 2.18 : Sample segmentation with K-means

3. TRANSFORMATION METHODS

3.1 Wavelet Transform

Signal transformations capture valuable information from any actual signal and manipulate it according to specific application requirements. The raw form of a natural signal is received and then processed in various ways, depending on the specific application. The processed signal can be utilized to extract valuable data. For example, the raw signal in the time domain can be converted to the processed signal in the frequency domain using the Fourier transform [60]. Some examples of transform techniques include Hilbert, Fourier, Radon, wavelet, etc. The selection of the transform technique can be determined by considering its benefits, drawbacks, and practical uses.

The wavelet transform is a method that combines the time and frequency domains and is well recognized as a time-frequency representation of a non-stationary signal. While the short-time Fourier transform can extract time-frequency information, the wavelet transform is a superior method since it addresses the limitations of fixed resolution seen with the short-time Fourier transform. Using the short-term Fourier transform decomposes the signal into smaller parts, which are believed to remain stationary.

Wavelets can be categorized as continuous or discrete, resulting in two types of wavelets transforms: continuous wavelet transform and discrete wavelet transform. The Discrete Wavelet Transform (DWT) is a computational method used to calculate a signal's rapid wavelet transform efficiently. The discrete wavelet transform is an efficient method for reducing processing time. It is more convenient to use and execute. [61]

The wavelet transform dissects a signal in the time domain by subjecting it to a high pass filter and a low pass filter, yielding two separate iterations: a low pass iteration and a high pass iteration. The low-pass version can be further decomposed by applying an additional set of low-pass and high-pass filters. Similarly, the High-pass version can be further analyzed by applying an additional set of low-pass and high-pass filters.

This process can be extended until a specified signal is broken down into a predetermined reference level. Subsequently, a thorough decomposition process produces signals indicating distinct frequency ranges, forming a single signal. Subsequently, signals about distinct frequency ranges can be consolidated and graphed on a three-dimensional coordinate system.

Discrete wavelet transform applies the transient filter in three directions: horizontal, vertical, and diagonal, and finds all the edges in these three directions. At the same time, an estimate of the half-dimensional image is obtained with the passage of a low-pass filter. The structure of a wavelet transform is shown in Figure 3.1.

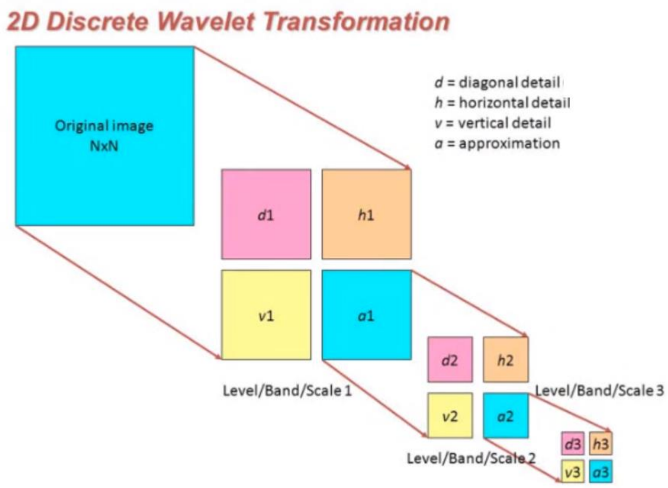

Figure 3.1: The structure of a wavelet transforms

In the following, we will examine the wavelet transform results with MATLAB 2021 b software by applying this transformation to several images. We start with simple image 3.2 and do the wavelet analysis of the two levels of the image. The result is shown in Figure 3.3.

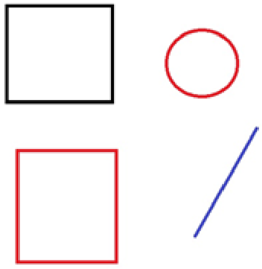

Figure 3.2 : Sample images 1.

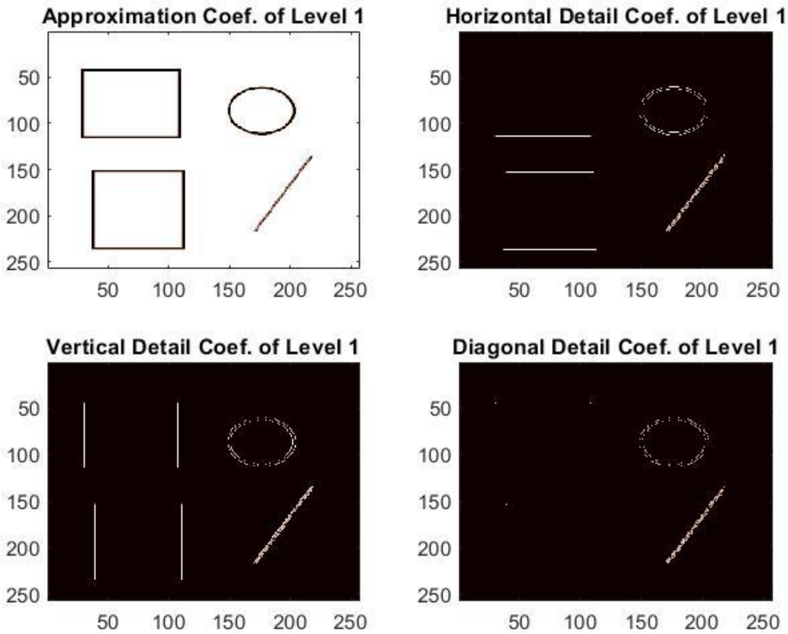

Figure 3.3 : Results from the first decomposition level.

As we can see in the results, all the boundaries are correctly identified at the level of the first decomposition in the horizontal part. However, one of the main boundaries is not. All the vertical borders are accurately identified, and we have achieved satisfactory results in the diagonal plane. However, to improve the results in the horizontal plane, we need to break down the second level in Figure 3.4.

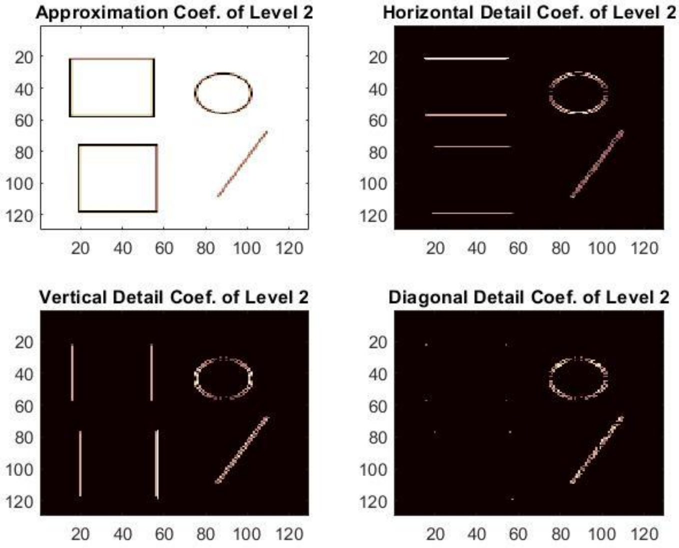

Figure 3.4 : Results from the second decomposition level.

The second decomposition level has reduced the image's dimensions by half. As we can see, the decomposition results are acceptable in all three horizontal, vertical, and diagonal levels. Multiplying the three images decomposed in the horizontal, vertical, and diagonal directions can reconstruct the original image at each level. In the following section, we will analyze the first and second levels of the brain image in Figure 3.5,3.6.

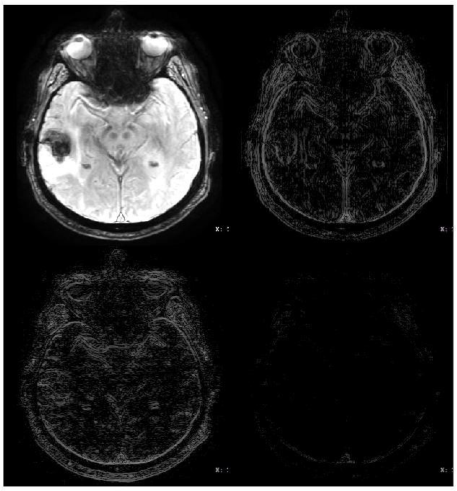

Figure 3.5 : Results from the first decomposition level.

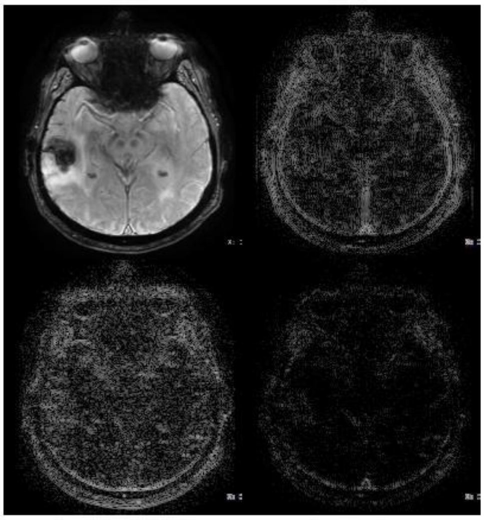

Figure 3.6 : Results from the second decomposition level.

3.2 The Contourlet Transform

Minh N. introduced contourlet transformation by Do and Martin Vetterli to improve image presentation by wavelet transform[62]. Moreover, it was later expanded by the same authors[51]. Contourlet transformation has been used in many fields, such as noise reduction in [63], [64], [65], image feature extraction in [66], image compression [67], face recognition[68], [69], [70], image fusion[71], [72], and image reconstruction in [73], [74].

The weaknesses of wavelet transformation have caused the emergence of contourlet transformation. The wavelet only deals with one-dimensional units. Moreover, it is a very suitable tool for smooth one-dimensional signals. However, for two-dimensional signals such as images, the multiplication of separate wavelets obtained from the external multiplication of one-dimensional wavelets is generally used, which is unsuitable for presenting some areas, especially along the image borders.

Inherent geometric structures in natural images characterize visual information. Due to their separable extension from 1-D bases, 2-D wavelets excel at detecting the discontinuities at edge points but cannot perceive the smoothness along the contours. Furthermore, separable wavelets can record just a restricted amount of directional information, which is a significant and distinctive characteristic of signals with multiple dimensions.

For example, if we consider the boundary of an object in Figure 3.7. Two-dimensional wavelet transforms in vertical, horizontal, and diagonal directions can describe this boundary, and the result is not ideal for us. We need a plan that the base functions are not just vertically, horizontally, and diagonally of constant size, and we have to describe the boundaries in different directions and extensions.

a b

Figure 3.7: The difference between the two transformations in edge detection a. Wavelet transform b. Contourlet Transform

Describing boundaries in contourlet transform and being a few scales can have different directions for identifying boundaries. The main feature of the transformation is its potential for effectively presenting two-dimensional units such as the peripheral boundary. In contourlet transform, each available general direction may be subdivided into more precise directions.

Finally, for a suitable transformation, we consider the following five characteristics:

1) Multiresolution refers to the ability to represent data at multiple levels of detail or resolution. The representation should enable the progressive approximation of images from low to high resolutions.

2) Localization refers to the process of adapting a product or service to meet the specific needs and preferences of a particular market or region. The crucial

components in the representation should be concentrated in both the spatial and frequency domains.

3) Sampling with a high level of importance or significance. In specific applications, such as compression, the representation must have a basis or frame with minimal redundancy.
4) Orientation. The representation should include fundamental components oriented in multiple directions well above the limited directions provided by separable wavelets.
5) Anisotropy refers to the property of a material or system that exhibits different physical properties in different directions. In order to accurately depict smooth curves in photographs, the visual representation should include necessary components utilizing a diverse range of elongated shapes with varying aspect ratios. [62]

As we examined, the wavelet transform has three cases, and the. Contourlet transform covers all cases. In the next section, we will examine the structure of a contourlet transform. The contourlet transform is derived from combining two Laplace pyramid filter banks and a Directional Filter Bank (DFB), as illustrated in Figure 3.8.

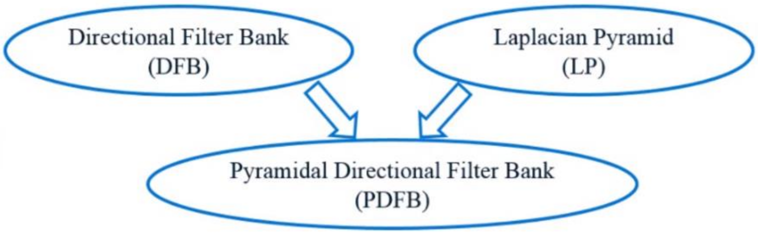

Figure 3.8 : Contourlet structure.

The Laplace pyramid is used to support multiple separations and to obtain discontinuities that we have already described in the Laplace filter, and an example of the pyramid is given in Figure 3.9. At each level, the image's dimensions are reduced by half in the pyramid.

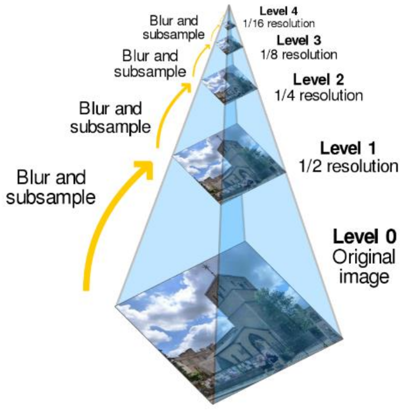

Figure 3.9 : Visual representation of an image pyramid with five levels [75].

The filter bank then presents the directional filter of the discontinuities as an image expansion with the help of essential elements called contourlets. For a better explanation, we consider Figure 3.10.

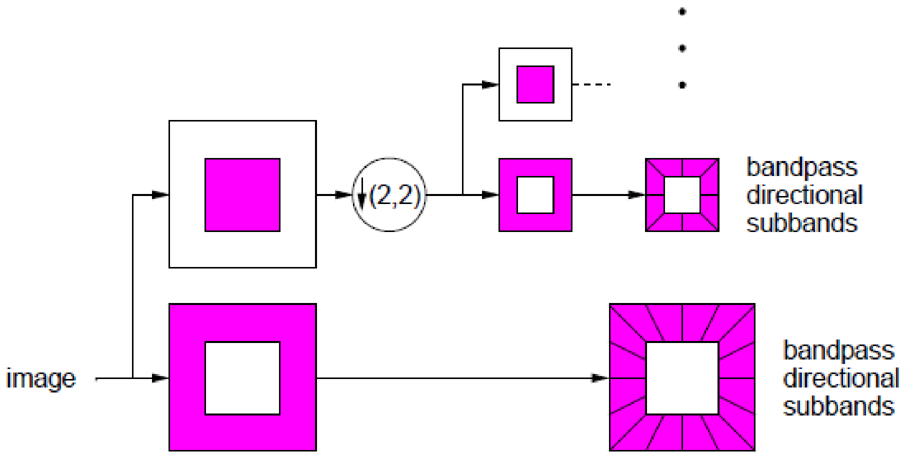

Figure 3.10 : The contourlet filter bank [62].

From the original image, the bandpass is extracted or the equivalent of this operation first, the high frequencies of the image are calculated, and then the lower frequencies of the image are subtracted from it, the middle frequencies are extracted, then directional operators are applied to these middle frequencies. We see the extraction of a large number of directions, that is, compared to the wavelet transform instead of three directions to the desired number of directions, and each of these directions helps us to find different edges of the image. On the other hand, in the low-frequency section, the previous operation is repeated by applying the sampling reduction. The middle frequencies are extracted again, and the DFB operator is applied. This process is followed as much as necessary.

The Laplacian Pyramid can be used with different support sizes in different articles, and the DFB filter can be used in different directions. For example, two examples of contourlet transformation are shown in Figure 3.11.

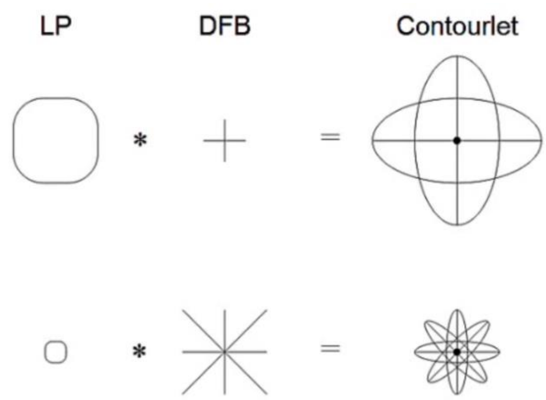

Figure 3.11 : Contourlet transformation with the different DFB and LP.

Finally, we will examine and apply contourlet transformation in sample photos at different levels and different directions with MATLAB 2021b software. The result of applying contourlet transformation in sample image one is shown in Figures 3.12, 3.13, and 3.14. As we can see, the transformation removed all the borders and edges of the image in the second level by applying filters in four directions and the third level by applying filters in 8 directions.

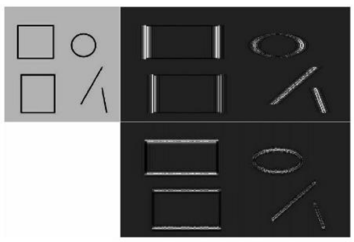

Figure 3.12 : Decomposition of the first level in two directions.

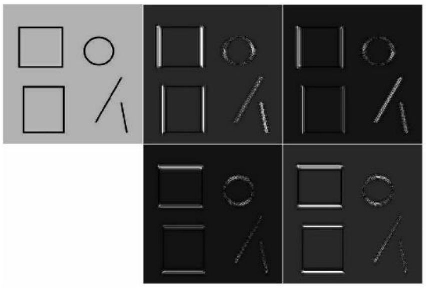

Figure 3.13 : Decomposition of the second level in four directions.

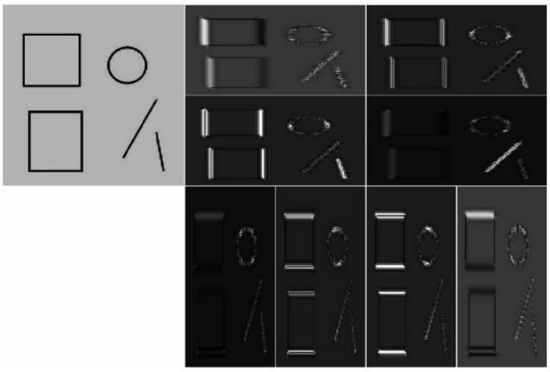

Figure 3.14 : Decomposition of the third level in eight directions.

In the following, we will apply the transformation to the second example, and the results are shown in Figures 3.15,3.16,3.17. Due to the complexity of this image, we find that in the third decomposition level and the application of filters in 8 directions, more borders and edges are identified for us.

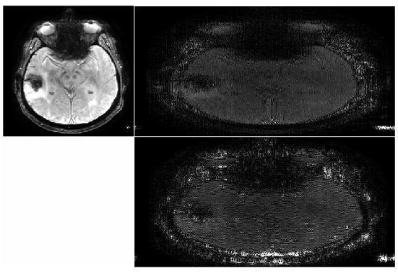

Figure 3.15 : Decomposition of the first level in two directions.

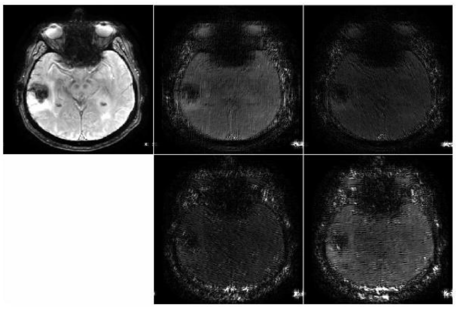

Figure 3.16 : Decomposition of the second level in four directions.

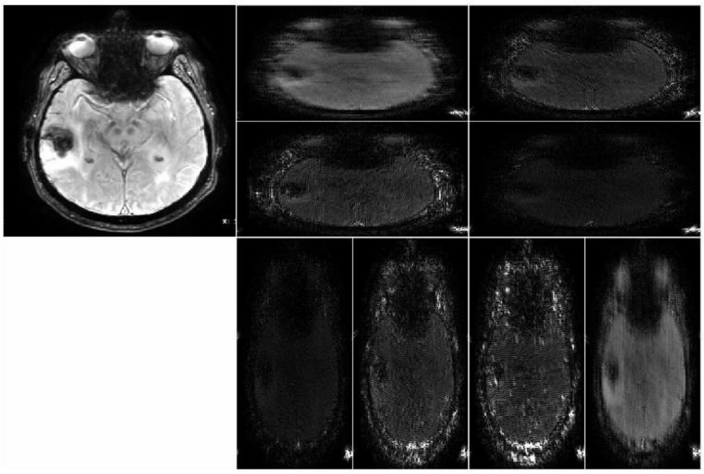

Figure 3.17 : Decomposition of the third level in eight directions.

As the introduction mentions, this transformation can compress image reconstruction and eliminate noise. Decomposition at different levels and applying filters in different directions allow us to use this method to identify image borders and contours. Image borders are the points where the contourlet coefficients are the highest in each direction. We find the borders in three sample images. Due to the simplicity of the first sample, which has two analysis levels and filters' application in four directions, we can show the borders well in Figure 3.18.

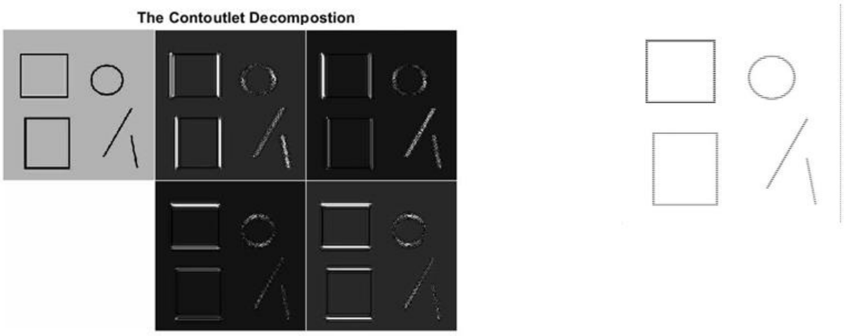

Figure 3.18 : Decomposition results of the second level and find the borders and edges

Finally, in the last example, by decomposition the image into two levels by applying four directional filters in Figure 3.19 and decomposing it into three levels and eight directions in Figure 3.20, we show the boundaries and edges in our sample images.

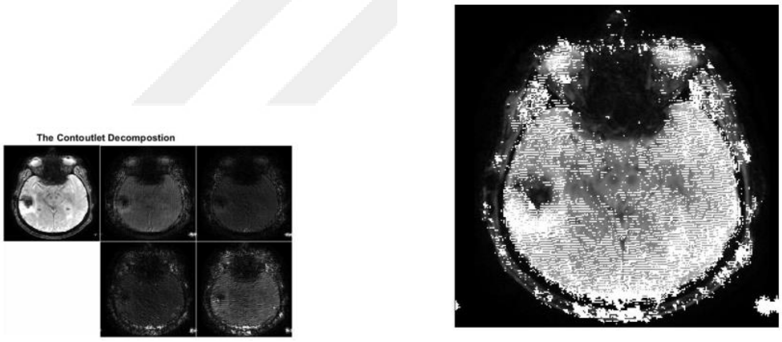

Figure 3.19 : Decomposition results of the second level and find the borders and edges.

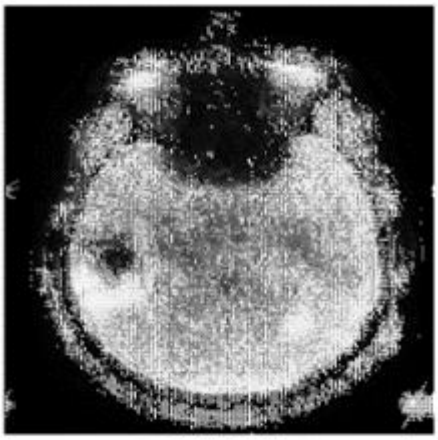

Figure 3.20 : Decomposition results of the third level and find the borders and edges.

4. DEEP LEARNING FOR IMAGE CLASSIFICATION

4.1 Introduction

Machine learning encompasses a collection of techniques that enable a machine to acquire significant patterns from data with minimal human intervention. The efficacy of a machine-learning technique is, to some extent, contingent upon human expertise. Such knowledge can enhance a machine's learning efficiency by employing suitable feature selection, transfer learning, and multitasking learning techniques. Machine learning has been effectively utilized in various applications and has achieved top-notch performance through this symbiotic relationship[76], [77], [78].

Due to the rapid advancement in computer power and the abundance of data, deep learning has emerged as the primary machine-learning technique. Deep learning can understand highly complex patterns that traditional machine-learning techniques cannot. Deep learning methods offer a significant advantage over traditional machine-learning techniques by streamlining the feature engineering process. Specific deep learning methods can be immediately applied to raw data without preprocessing. Obtaining sufficient subject expertise for accurate feature determination in medical imaging analysis can be time-consuming and requires years of training. Therefore, this enables researchers to capitalize on novel concepts with incredible speed and simplicity. Illustrated in Figure 4.1 is a comparison between traditional machine learning techniques and deep learning in categorization.

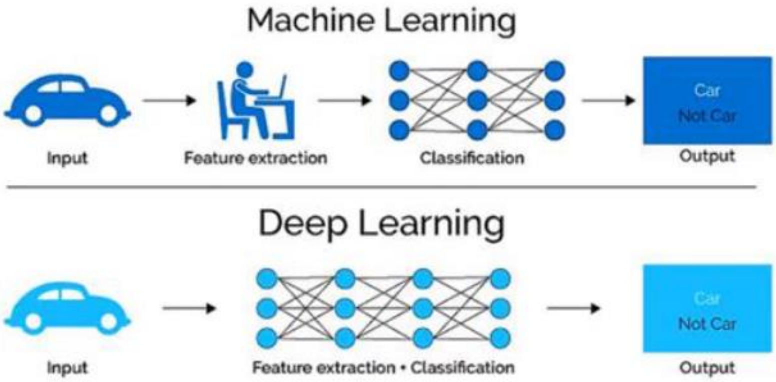

Figure 4.1: Difference between conventional machine learning methods and deep learning in classification.

Convolution Neural Network (CNN) is one of the concepts of deep learning neural networks, which was used in the late '90s but has become very popular in recent years with very significant results in image recognition or recognition in computer vision projects. Convolutional networks were initially introduced in the 1940s and 1960s but were not given serious attention due to the lack of necessary hardware, and again in 2006 due to hardware upgrades and solving existing problems. Deep neural networks have many applications and are generally used in three areas: 1. Classification [79], [80], [81], [82], 2. object detection [83], 3. Image semantic[84], [85], [86] 4. image generation[87].The ability of deep learning to process large volumes of data and overcome the difficulties of feature extraction in the old methods and high accuracy, including the advantages of deep learning networks.

4.2 Convolution neural network architecture

These convolution neural networks closely resemble neural networks, consisting of neurons with adjustable parameters for weight and bias. However, a different feature of CNN is that it explicitly assumes that the inputs are images. This allows us to encode features to identify the elements in the images. To understand how these neural networks work, let us consider how objects are detected. For example, if we see a face, we recognize it by having ears, eyes, nose, and hair. To decide whether something is a face, we compare and mark each feature with the items in our mind box. Sometimes, the ears of a face may be covered with hair, but seeing factors such as the eyes, nose, and mouth make sure it is a face. To do this, we must first know what the ear or nose looks like to recognize them in an image. First, we must identify lines, edges, tissues, or shapes resembling ears or noses.

Moreover, this is what the layers of a CNN do. However, simply identifying these elements is not enough to name something. We must also identify how face components relate to each other and their relative sizes. Otherwise, the face will not look like what we have always seen. A sample CNN architecture is shown in Figure 4.2.

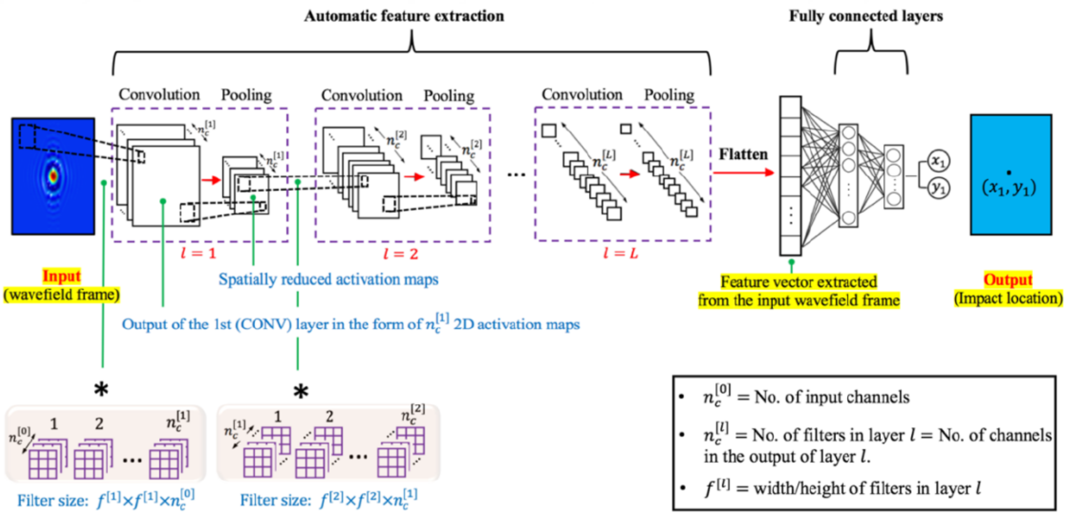

Figure 4.2 :A sample CNN architecture **[**88].

A CNN consists of four types of layers:

1. Convolutional Layers (CONV): The essential components of CNN are the main building blocks. Each convolutional layer consists of a set of filters, also known as kernels, with trainable parameters, i.e., weights. A per-layer filter can be thought of as a three-dimensional volume of neurons. The primary convolutional layer corresponds to the number of channels of the input image, which is 3 for the color image and 1 for the grayscale image. At the heart of the CNN, the convolution process performs feature extraction using filters in early layers to extract low-level information. Conversely, deeper layers look for higher-level features. Unlike standard computer vision systems, which require manually designing filter parameters, CNNs can learn these filter

weights via backpropagation with sufficient training data. Each layer only needs the specifications for the number and size of the filters. It is essential to understand that a filter is only connected to a specific local part of the input at any given time, meaning that the connections are sparse. Furthermore, the parameters learned for a single filter are applied to the entire spatial range of the image, indicating that parameter sharing is involved. Due to these two features, CNNs are a suitable choice for processing image data, unlike fully connected Artificial Neural Networks (ANNs), where each neuron in a given layer is connected to each neuron in its adjacent layers. As a result, many trainable parameters are created.

Pooling (POOL) layers: The output of the convolutional layer is often passed into pooling layers. The pooling layers aim to progressively reduce the representation's spatial dimensions (height, breadth) while maintaining depth. This decreases the number of trainable parameters in the network, leading to reduced computational expenses and enhanced control over overfitting. The convolutional and pooling layers constitute the CNN feature extraction module.

The pooling layer does not possess any trainable parameters. The algorithm utilizes a basic yet efficient sample pooling technique, like a convolution operation, where a window traverses the image. Max pooling and average pooling are the most prevalent instances of pooling.

3. wholly connected (FC) layers: The feature extractor component of the CNN produces a flattened output that serves as a condensed representation of the initial input. Subsequently, this data is inputted into one or many fully connected layers, which are ultimately succeeded by an output layer. A vital function of the FC layer is to serve as a classifier. The attributes obtained through convolutional layers are ultimately converted into a vector. The attribute vector is ultimately provided to an FC classifier to determine the accurate class.

4. Activation Function layer: CNN uses a nonlinear activation function after the convolutional layer, like other neural networks. Employing a nonlinear function introduces a nonlinear attribute into the neural network, which holds significant importance. Some frameworks, in the exact definition of the convolution layer, allow the specify the type of nonlinear function. In some frameworks, we also need to create

a separate layer. By defining a nonlinear function independently from the convolutional layer, greater flexibility is achieved. The ReLU function is the most popular among all nonlinear functions. The ReLU function returns values less than zero (negative) to zero and values greater than zero to the output without any change.

4.3 Residual network

There has been notable advancement in computer vision in recent years, particularly with the emergence of deep CNN. We have achieved outstanding outcomes in picture classification and image recognition. Over time, researchers have increasingly developed deeper neural networks by adding additional layers to address and enhance challenging tasks. However, the problem arises as additional layers are incorporated into the neural network, as it becomes challenging to effectively train them, resulting in a decline in the network's performance accuracy. ResNet assists and effectively addresses this issue.

What are the challenges of designing and training deep neural networks? In general, adding more layers and parameters to deep networks faces two significant problems:

Gradient fade: Gradient fading is a phenomenon that occurs due to the deepness of the neural network, in which the gradients from the loss function do not reach the primary layers and become less and less with successive multiplications, and this causes the weights in the layer the initial ones are not updated, and therefore the network does not have the desired output.

Typically, when faced with a complex situation, we enhance the design of our neural network by incorporating several layers and increasing the depth of our network. This ultimately leads to improved accuracy and network performance. The primary justification for incorporating additional layers is that as these layers are introduced, the network progressively acquires more intricate characteristics.

For example, in the problem of image recognition, the first layer may recognize the edges of the image, and the second layer may learn the background. Similarly, the third layer may learn the entire structure of the object or face. By conducting numerous experiments, researchers have realized that there is a threshold for increasing depth in the convolutional neural network model. Figure 4.3 shows the percent error for training and testing data for 20-layer and 56-layer networks, respectively.

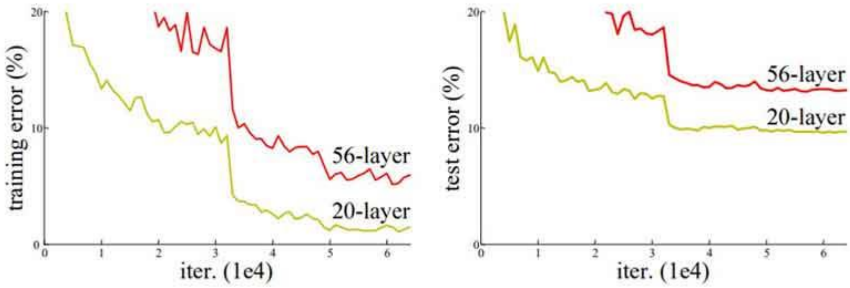

Figure 4.3 : The difference in the error function in a 56-layer network compared to a 20-layer network [89].

As shown in Figure 5, the error percentage for both test and training data is higher for a 56-layer architecture than a 20-layer network. Figure 4.3 indicates that adding more layers to a network does not always mean improving its performance but may degrade its performance. As the network gets deeper, its performance becomes weaker, which can be found in several factors, such as the optimization function, the initial weighting of the network, and, most importantly, the gradient vanishing problem.

Degradation: In the phenomenon of degradation, the network does not have the problem of fading the gradient, and naturally, when more layers are placed, it is expected that the network will perform better than networks with shallow layers, or at least its performance equal to shallow networks, which does not happen!

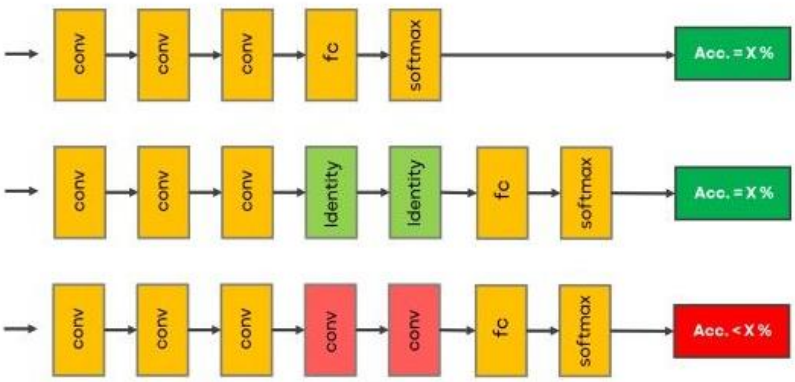

Figure 4.4 : The phenomenon of descent or degradation in the network [89], [90].

Typical deep learning networks, such as AlexNet, ZFNet, and VGGNet, often have convolutional and fully connected layers for classification without shortcut connections. Here, we call them plain networks. When plain networks are deeper (that is, layers increase), the problem of vanishing gradient or exploding gradient occurs; therefore, deepening the network is not accessible by just adding layers. This is where ResNet was introduced to solve this problem. This network can have up to 152 layers.

Residual Network is a specific neural network architecture first presented in [89]. The ResNet models have achieved remarkable success. One of these achievements was securing the top position in the ILSVRC 2015 Classification competition. The primary breakthrough of this network was the introduction of a novel block called the Residual Block, which effectively addresses the issue of layer proliferation. The details of this block will be elucidated. The ResNet architecture employed Residual blocks, as depicted in Figure 4.5

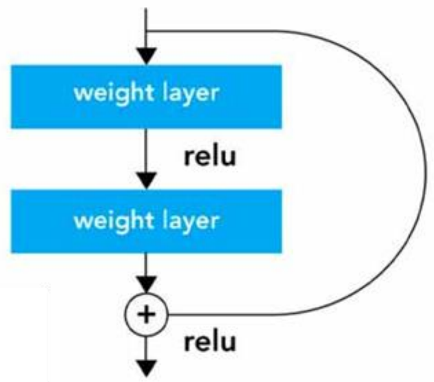

Figure 4.5 : Shortcut connections, the strength of the ResNet [91].

The primary distinction is that a direct link circumvents the intermediary layer (in the broader framework, the intermediary layers). The link is called a "Skip Connection" and serves as the primary component of the remaining blocks. Due to this correlation, the layer's output differs from other networks. The input "x" is multiplied by the weight of the layers, and then the bias value is applied to it. Subsequently, the given expression undergoes the activation function F(x), resulting in the final output denoted as H(x), which is subsequently displayed. Furthermore, it is explicitly stated in Equations 4.1.

$$H(x)=F(x)+x \quad (4.1)$$

In conventional neural networks, the layers are trained to learn the input distribution. However, in the case of residual blocks, the network is trained to learn the distribution between the output and input, hence the name "residual blocks."
The ResNet network utilizes Skip Connections to address the Vanishing gradient issue in deep neural networks. These connections enable the gradient to flow along Shortcut pathways, effectively resolving the problem. Furthermore, through these connections, the model can acquire the same function that guarantees the upper layer's performance is equal to, if not better, the lower layer's performance. Suppose we have a shallow network and a deep network that maps input "x" to output "y" using the function H(x). Our goal is that the deep network performs at least as well as the shallow network and that the network performance does not degrade as the number of layers increases. One way to achieve this goal is for the additional layers in the deep network to learn the same function so that their output equals their input. Finally, the model's performance is not degraded by additional layers.
For the network to learn the same function, it must be F(x)=0, and the relationship will be as follows in Equations 4.2.

$$H(x)=x \quad (4.2)$$

In the optimal scenario, the additional layers of the deep neural network may more accurately approximate the relationship between the input "x" and the output "y" compared to shallow networks, resulting in a substantial reduction in error. Consequently, we anticipate that ResNet will outperform deep neural networks. We can say in this way that when the network cannot learn the Identity layer directly, we can help him in this field with the help of a skip connection or, in other words, an alternative route! Figure 4.6 shows this. As shown in this Figure, in addition to the main path that includes convolution layers and Batch Normalization and activation function, the input data is added directly and without any changes to the output of the used nonlinear layers. In this case, we have helped the network to learn. In other words, in this case, the network can consider the weights of the convolutional layers to be zero, as if the weighted layers do not exist and the input data has gone directly to the output. In other words, it is as if there is no layer. The interpretation of this issue is that the network can consider weighted layers wherever it likes and bring their effect to the output. Wherever it does not like, it cannot believe it, which makes us any number. Let us add a layer and consider the network of different layers as a weighted layer according to your request or apply the input without any change to the output; this is the solution proposed by the network that enabled us to create a network with 100 and Even design 1000 layers.

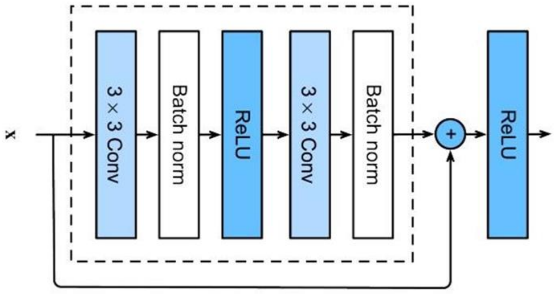

Figure 4.6 : Alternative path in RESNET network [91].

This part discusses and examines the brilliant idea of ResNet architecture. We saw how adding a side path can significantly help simplify the cost function, solve the degradation problem, make networks of any depth design, and solve the network depth problem. In the next step, we will use this idea in the other two parts to design a deep network for categorizing and segmenting brain cancer images.

4.4 Residual Network for Classification (ResNet-50)

4.4.1 ResNet 50 architecture

Machine learning professionals add layers to address a computer vision challenge when utilizing deep convolutional neural networks. Including supplementary layers enhances the ability to efficiently address intricate challenges, as distinct layers can be taught for different tasks to attain exact outcomes. Increasing the number of stacked layers in a model can enhance its features. However, a deeper network can encounter a degradation issue. When the number of layers in a neural network increase, the accuracy levels may reach a maximum and gradually decline beyond a certain point.

Consequently, the model's performance declines in the training and testing data. This degeneration is not caused by overfitting. Alternatively, it can stem from the network setup, the optimization function, or, most significantly, the vanishing or ballooning gradients issue.

ResNet was developed to address this issue. Deep residual networks employ residual blocks to enhance the precision of models. The fundamental power of this neural network lies in the concept of "jumping connections" inside the remaining blocks. ResNet encompasses numerous variations that adhere to the same fundamental Principle but possess varying quantities of layers. ResNet50 refers to the version of the neural network capable of operating with 50 layers.

The initial ResNet architecture, or ResNet-34, incorporated shortcut connections to transform a regular network into its residual network equivalent. In this instance, the simple network drew inspiration from VGG neural networks (VGG-16, VGG-19), employing convolutional networks with 3×3 filters. The distinction between the two models is illustrated in Figure 4.7.

ResNet adhered to two fundamental design principles: ensuring that the layers had the same number of filters for the same output feature map size and doubling the number of filters when the feature map size was halved to maintain the temporal complexity per layer. The structure comprised 34 layers, each with a specific weight. Although the input and output dimensions were equal, the identity shortcuts were employed directly. Shortcut links have been incorporated into this simple network. Given the increase in size, two options needed to be considered. Initially, the shortcut would continue to execute an identity mapping while additional zero entries would be added to accommodate higher dimensions. A choice was to utilize the projection shortcut to align dimensions.

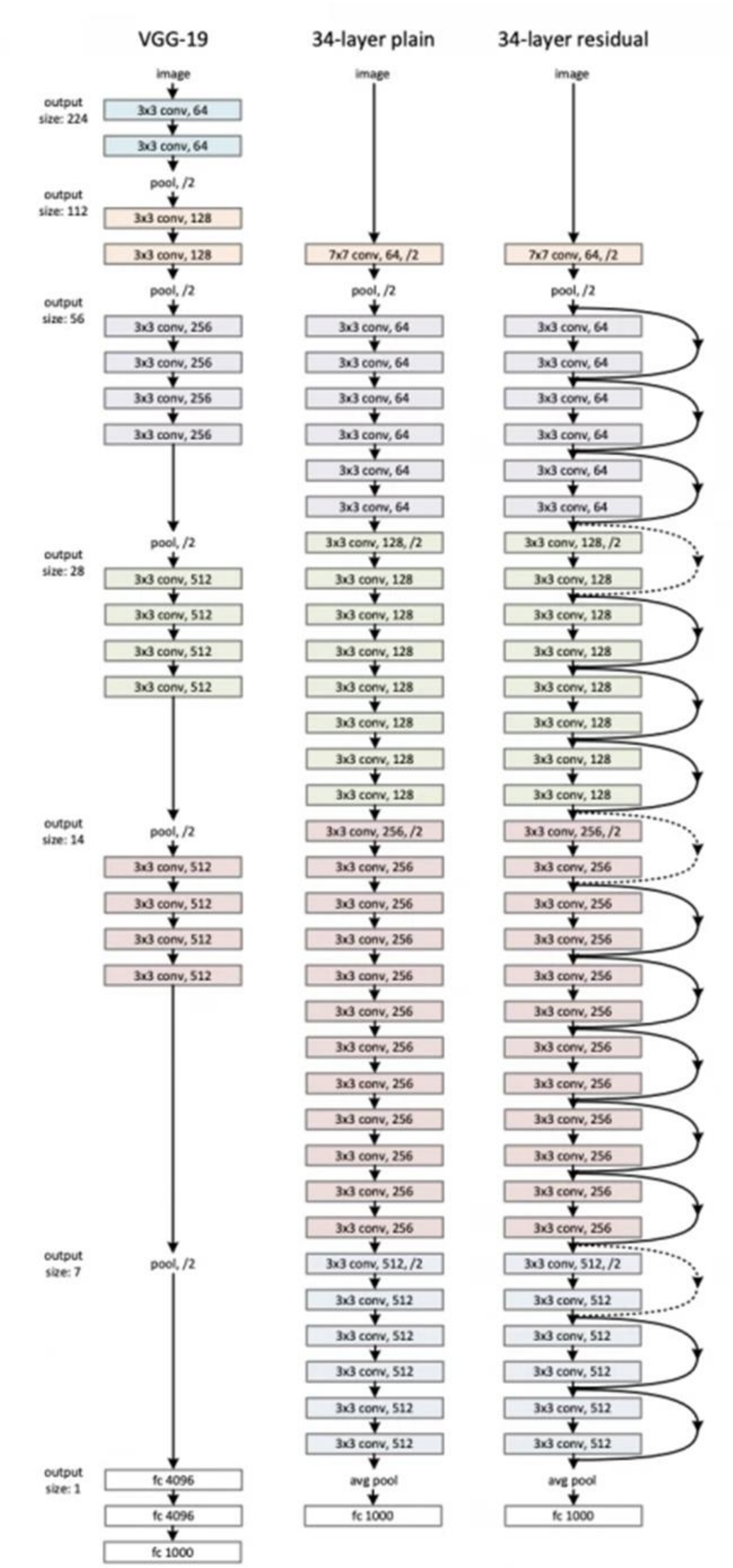

Figure 4.7 : The structure of VGG 19 and 34 layers plain [92].

Although the ResNet50 architecture is derived from the model mentioned above, it has one notable distinction. In this instance, the fundamental component was altered into a bottleneck configuration due to concerns regarding the duration required to train the layers. This utilized a stack consisting of three levels instead of the previous two. Consequently, the ResNet34's 2-layer blocks were substituted with 3-layer bottleneck blocks, creating the ResNet 50 architecture. The accuracy of this model is significantly superior to that of the 34-layer ResNet model.

Finally, we build the ResNet50 network model using the Keras library, learn the weights, and evaluate this network in tumor detection. The characteristics of each layer and the number of its parameters are given in Table 4.1.

Table 4.1: The ResNet50 network architecture consists of the following layers

Layer	**Layer structure**	**Shape**
1	Input Layer	256,256,3
2	Zero Padding	262, 262, 3
3	Convolutional	128, 128, 64
4	Batch Normalization	128, 128, 64
5	Activation	128, 128, 64
6	Zero Padding	130, 130, 64
7	Max Pooling	64, 64, 64
8	Convolutional	64, 64, 64
9	Batch Normalization	64, 64, 64
10	Activation	64, 64, 64
11	Convolutional	64, 64, 64
12	Batch Normalization	64, 64, 64
13	Activation	64, 64, 64
14	Convolutional	64, 64, 256
15	Convolutional	64, 64, 256
16	Batch Normalization	64, 64, 256
17	Batch Normalization	64, 64, 256

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

18	Add	64, 64, 256
19	Activation	64, 64, 256
20	Convolutional	64, 64, 64
21	Batch Normalization	64, 64, 64
22	Activation	64, 64, 64
23	Convolutional	64, 64, 64
24	Batch Normalization	64, 64, 64
25	Activation	64, 64, 64
26	Convolutional	64, 64, 256
27	Batch Normalization	64, 64, 256
28	Add	64, 64, 256
29	Activation	64, 64, 256
30	Convolutional	64, 64, 64
31	Batch Normalization	64, 64, 64
32	Activation	64, 64, 64
33	Convolutional	64, 64, 64
34	Batch Normalization	64, 64, 64
35	Activation	64, 64, 64
36	Convolutional	64, 64, 256
37	Batch Normalization	64, 64, 256
38	Add	64, 64, 256
39	Activation	64, 64, 256
40	Convolutional	32, 32, 128
41	Batch Normalization	32, 32, 128
42	Activation	32, 32, 128
43	Convolutional	32, 32, 128
44	Batch Normalization	32, 32, 128
45	Activation	32, 32, 128
46	Convolutional	32, 32, 512

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

47	Convolutional	32, 32, 512
48	Batch Normalization	32, 32, 512
49	Batch Normalization	32, 32, 512
50	Add	32, 32, 512
51	Activation	32, 32, 512
52	Convolutional	32, 32, 128
53	Batch Normalization	32, 32, 128
54	Activation	32, 32, 128
55	Convolutional	32, 32, 128
56	Batch Normalization	32, 32, 128
57	Activation	32, 32, 128
58	Convolutional	32, 32, 512
59	Batch Normalization	32, 32, 512
60	Add	32, 32, 512
61	Activation	32, 32, 512
62	Convolutional	32, 32, 128
63	Batch Normalization	32, 32, 128
64	Activation	32, 32, 128
65	Convolutional	32, 32, 128
66	Batch Normalization	32, 32, 128
67	Activation	32, 32, 128
68	Convolutional	32, 32, 512
69	Batch Normalization	32, 32, 512
70	Add	32, 32, 512
71	Activation	32, 32, 512
72	Convolutional	32, 32, 128
73	Batch Normalization	32, 32, 128
74	Activation	32, 32, 128
75	Convolutional	32, 32, 128

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

76	Batch Normalization	32, 32, 128
77	Activation	32, 32, 128
78	Convolutional	32, 32, 512
79	Batch Normalization	32, 32, 512
80	Add	32, 32, 512
81	Activation	32, 32, 512
82	Convolutional	16, 16, 256
83	Batch Normalization	16, 16, 256
84	Activation	16, 16, 256
85	Convolutional	16, 16, 256
86	Batch Normalization	16, 16, 256
87	Activation	16, 16, 256
88	Convolutional	16, 16, 1024
89	Convolutional	16, 16, 1024
90	Batch Normalization	16, 16, 1024
91	Batch Normalization	16, 16, 1024
92	Add	16, 16, 1024
93	Activation	16, 16, 1024
94	Convolutional	16, 16, 256
95	Batch Normalization	16, 16, 256
96	Activation	16, 16, 256
97	Convolutional	16, 16, 256
98	Batch Normalization	16, 16, 256
99	Activation	16, 16, 256
100	Convolutional	16, 16, 1024
101	Batch Normalization	16, 16, 1024
102	Add	16, 16, 1024
103	Activation	16, 16, 1024
104	Convolutional	16, 16, 256

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

105	Batch Normalization	16, 16, 256
106	Activation	16, 16, 256
107	Convolutional	16, 16, 256
108	Batch Normalization	16, 16, 256
109	Activation	16, 16, 256
110	Convolutional	16, 16, 1024
111	Batch Normalization	16, 16, 1024
112	Add	16, 16, 1024
113	Activation	16, 16, 1024
114	Convolutional	16, 16, 256
115	Batch Normalization	16, 16, 256
116	Activation	16, 16, 256
117	Convolutional	16, 16, 256
118	Batch Normalization	16, 16, 256
119	Activation	16, 16, 256
120	Convolutional	16, 16, 1024
121	Batch Normalization	16, 16, 1024
122	Add	16, 16, 1024
123	Activation	16, 16, 1024
124	Convolutional	16, 16, 256
125	Batch Normalization	16, 16, 256
126	Activation	16, 16, 256
127	Convolutional	16, 16, 256
128	Batch Normalization	16, 16, 256
129	Activation	16, 16, 256
130	Convolutional	16, 16, 1024
131	Batch Normalization	16, 16, 1024
132	Add	16, 16, 1024
133	Activation	16, 16, 1024
134	Convolutional	8, 8, 512

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

135	Batch Normalization	8, 8, 512
136	Activation	8, 8, 512
137	Convolutional	8, 8, 512
138	Batch Normalization	8, 8, 512
139	Add	8, 8, 512
140	Convolutional	8, 8, 2048
141	Convolutional	8, 8, 2048
142	Batch Normalization	8, 8, 2048
143	Batch Normalization	8, 8, 2048
144	Add	8, 8, 2048
145	Activation	8, 8, 2048
146	Convolutional	8, 8, 512
147	Batch Normalization	8, 8, 512
148	Activation	8, 8, 512
149	Convolutional	8, 8, 512
150	Batch Normalization	8, 8, 512
151	Activation	8, 8, 512
152	Convolutional	8, 8, 2048
153	Batch Normalization	8, 8, 2048
154	Add	8, 8, 2048
155	Activation	8, 8, 2048
156	Convolutional	8, 8, 512
157	Batch Normalization	8, 8, 512
158	Activation	8, 8, 512
159	Convolutional	8, 8, 512
160	Batch Normalization	8, 8, 512
161	Activation	8, 8, 512
162	Convolutional	8, 8, 2048
163	Batch Normalization	8, 8, 2048

Table 4.1(continued): The ResNet50 network architecture consists of the following layers

164	Add	8, 8, 2048
165	Activation	8, 8, 2048

Our ResNet50 ends with almost a hundred iterations and more than 25 million parameters per iteration with Nvidia RTX3050 graphics card hardware after about 16 hours, and our desired model is made. In the next part, we will examine the performance of this network

4.4.2 Preparation and data augmentation

We operate in this network as before for a better comparison with the previous networks. To implement this network, we will first re-introduce brain MRI images. We have 253 photos for training and testing the ResNet50, 98 of which have a non-tumor label, and the other 155 have a tumor label. Sample photos labeled with tumors and non-tumors are shown in Figure 4.8,4.9.

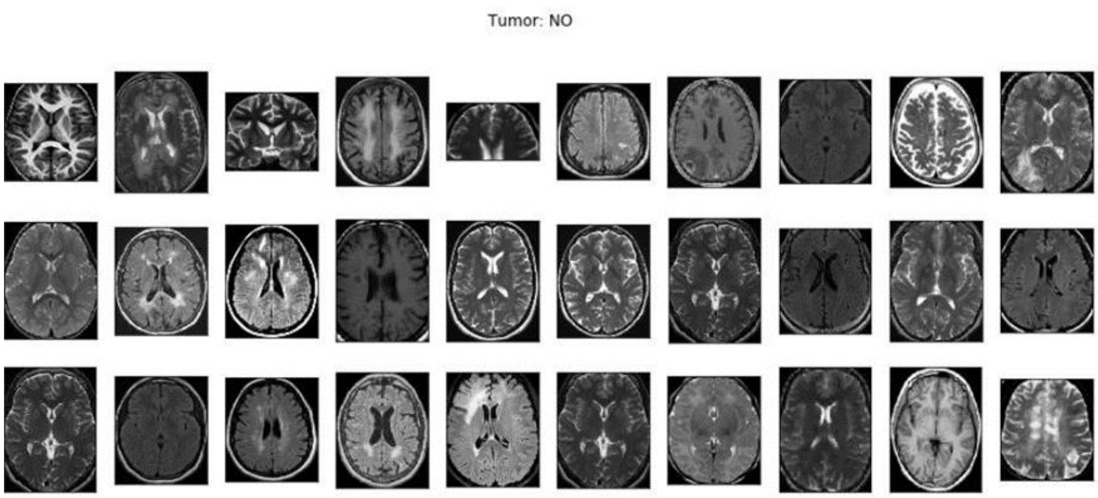

Figure 4.8 : Sample images with nontumor label.

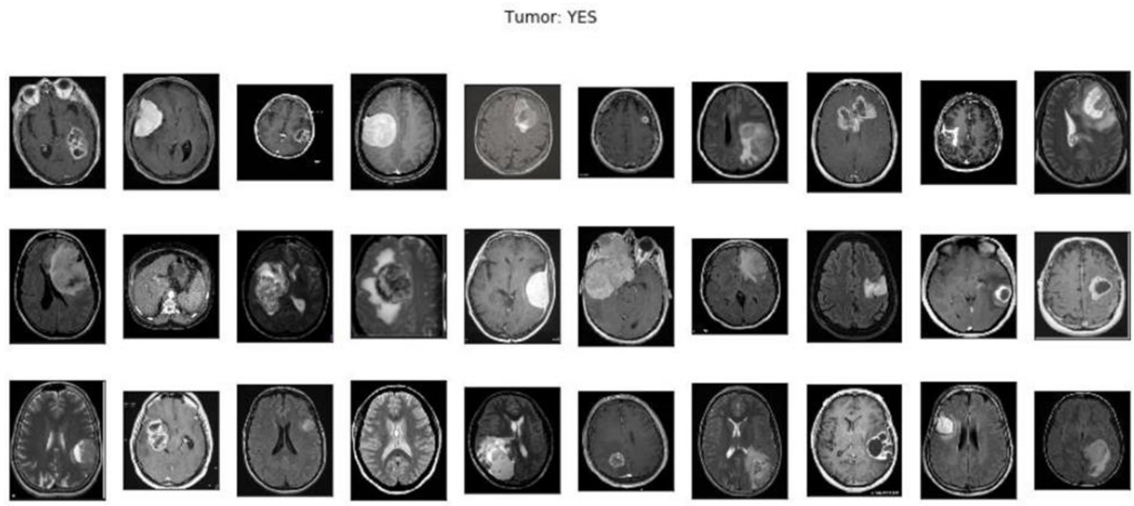

Figure 4.9 : Sample images with tumor label.

11. 70% of the total data (177 images) are used for convolutional neural network training, and 15% (37 images) are used for network validation. We use the remaining 16% of the data (39 images) to test ResNet50. The data preparation summary is given in Table 4.2.

Table 4.2 : Data set, train and validation set, test set.

All datasets	Train set	Validation set	Test set
253 images	77 images with labels yes or no	37 images with labels yes or no	39 images without label

As we saw in the previous sections, most computer vision systems will perform better as data volumes increase. However, in many cases, collecting more data is a complex and costly task. Therefore, data augmentation is a technique often used to improve the performance of computer vision systems. It does not matter if we are using transfer learning or training our model; data augmentation usually helps.
As before, we use the image data generator in the Keras library. The data augmentation technique [93] helps us to have new images for training by producing new images artificially from the original images. There are many techniques for doing this, including the following: Vertical and horizontal image shifting techniques [94], Flip

images from left to right and up and down, Random image rotation technique [95], Color space change technique, and Random image magnification technique[96]. In this part, using the Keras library and the same as what we did in the VGG16 network, we use the random image rotation Technique and increase the training data. We add approximately 21 new images for each MRI image, as shown in Figure 4.10.

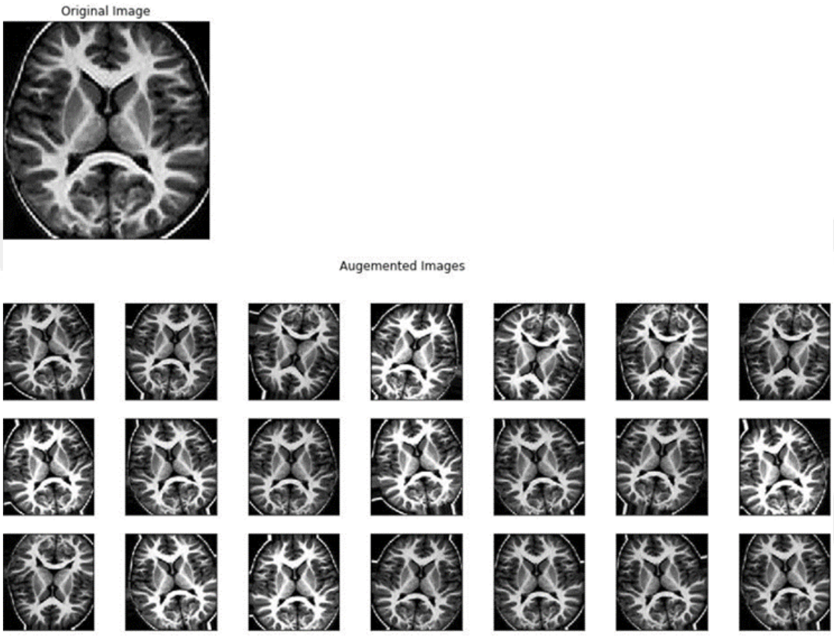

Figure 4.10 : Data augmentation for a sample MRI

4.4.3 Model performance

Loss function loss is one of the essential components in deep learning methods based on medicine. Over the years, more than 20 loss functions have been used for various issues in Segmentation and classification. Using the Keras library and based on this function's good results and experiences in classification problems, we use conventional functions such as Mean Squared Error(MSE) instead of the entropy crossover function given in equation (4.3). In MSE, we measure the difference in distance between all network output samples and the labeled sample and set it as a loss, but in entropy crossover, we set the total logarithm of the output samples that should have been one as a loss. The results of training with this function are shown in Figure 4.11.

$$L(z,y) = -\left[y\log(z) + (1-y)\log(1-z)\right] \tag{4.3}$$

In the relationship above, y is the actual output value and z is the predicted output value

We employ the confusion matrix to assess the system's performance and accuracy. This matrix consists of four states: The sample is classified as a positive category and is correctly identified as belonging to the same class (True Positive). The sample belongs to the positive class but is incorrectly classified as a member of the hostile class, resulting in a false negative. The sample belongs to an antagonistic category and is correctly identified as a category member (True Negative). Ultimately, the sample belongs to the antagonistic category and is acknowledged as a positive category (False Positive) member. The confusion matrix for our example is given in Table 4.3. All the properties of this matrix are calculated in Table 4.4.

Table 4.3 : Confusion matrix.

		Predicted condition	
		Positive (Cancer)	Negative (Non-cancer)
Actual condition	Positive (Cancer)	20 (TP)	0 (FN)
	Negative (Non-cancer)	1 (FP)	17 (TN)

Table 4.4 : Different Measures of the Confusion Matrix.

Measure	Value	Derivations
Sensitivity	0.9524	TPR = TP / (TP + FN)
Specificity	1.0000	SPC = TN / (FP + TN)
Precision	1.0000	PPV = TP / (TP + FP)
Negative Predictive Value	0.9444	NPV = TN / (TN + FN)
False Positive Rate	0.0000	FPR = FP / (FP + TN)
False Discovery Rate	0.0000	FDR = FP / (FP + TP)
False Negative Rate	0.0476	FNR = FN / (FN + TP)
Accuracy	0.9737	ACC = (TP + TN) / (P + N)
F1 Score	0.9756	F1 = 2TP / (2TP + FP + FN)

Accuracy is the most common, essential, and straightforward measure of classification quality. As we remember, the system's accuracy is obtained from numbers on the primary diameter divided by the total numbers. For our ResNet 50, accuracy is equal to 97.37%. The accuracy chart for each iteration is also shown in Figure 4.11.

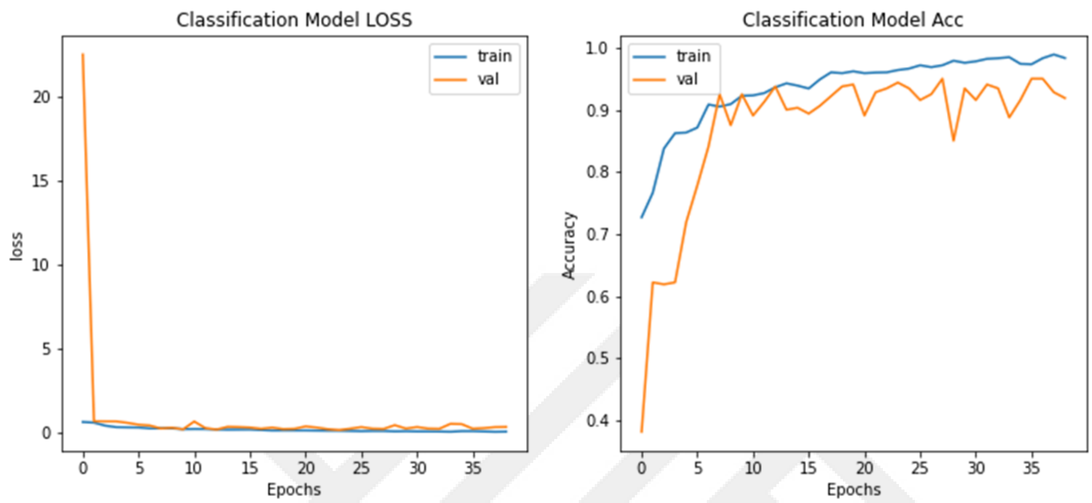

Figure 4.11 : The result of accuracy and model loos in the ResNet50 model.

Finally, in the test data output, some results are given in Figure 4.12, in which class one corresponds to the tumor, and class zero corresponds to the absence of cancer.

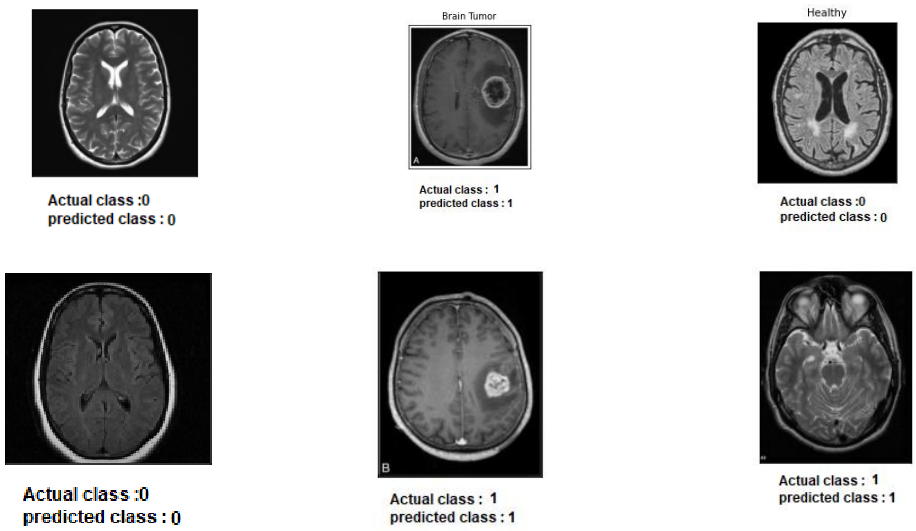

Figure 4.12 : Examples of brain tumor classification results by ResNet 50.

5. DEEP LEARNING FOR IMAGES SEGMENTATION

5.1 RESUNET Architecture

The U-Net method [97] has gained popularity in recent years among deep learning algorithms because of its precise response, exceptional precision, efficient processing speed, and lack of requirement for intricate and costly hardware. This network's architecture comprises two components: The left side represents the compression path, while the right represents the expansion path. The compression path comprehends the image's content, while the expansion path is crucial in precisely locating objects. The structural design of the U-Net is depicted in Figure 5.1.

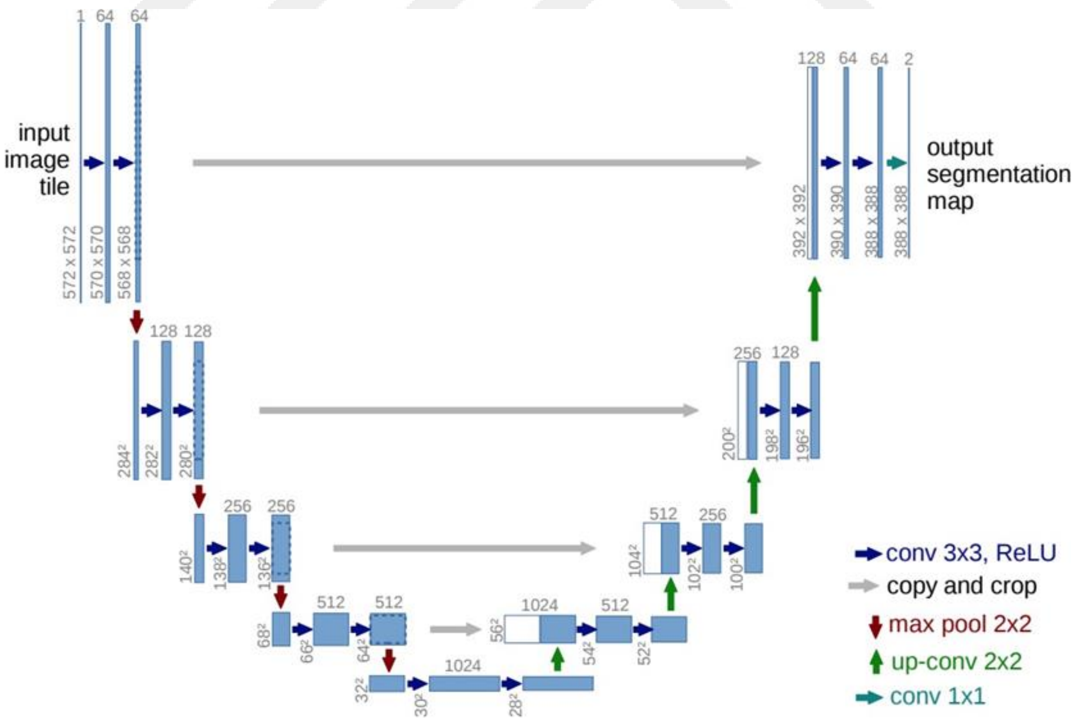

Figure 5.1 : Sample architecture of U-net [98].

This section uses a combination of the U-Net and the RES networks to optimize the outcomes. The RESUNET design integrates the U-Net backbone with residual blocks to address the limitations of the U-Net architecture. RESUNET combines the U-Net architecture with deep residual learning to maximize its benefits. RESUNET is a convolutional neural network that is specifically built to provide excellent performance while using a smaller number of parameters. The architecture of the proposed RESUNET is shown in Figure 5.2.

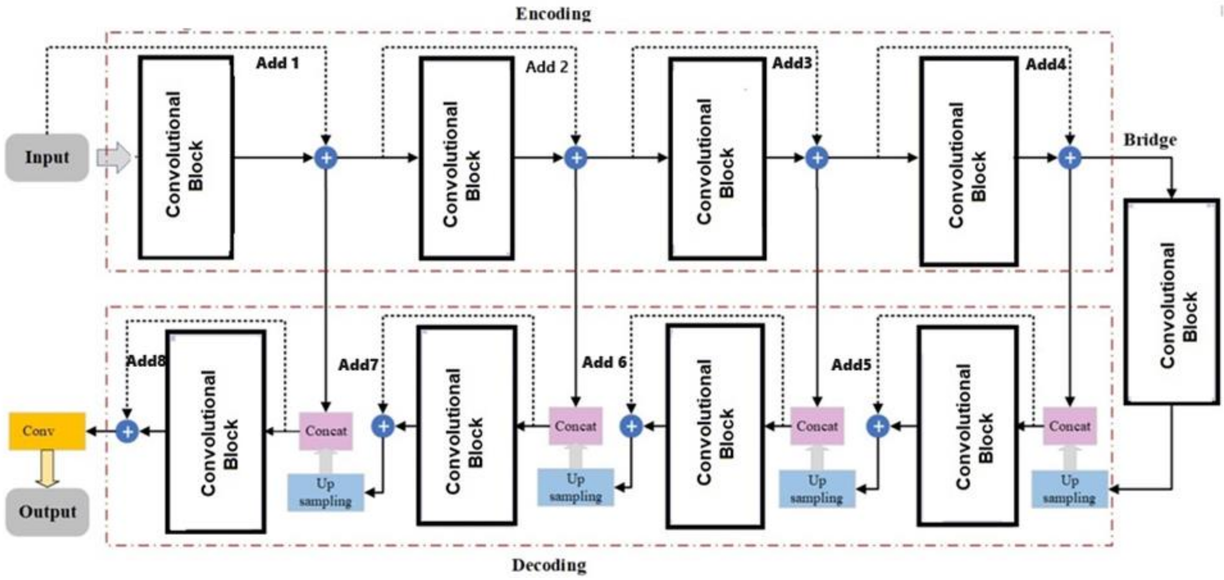

Figure 5.2 : The Proposed RESUNET.

RESUNET consists of three elements: an encoder, a bottleneck, and a decoder. The input image is sent through multiple encoder blocks to allow the network to obtain a conceptual representation. The encoder consists of three encoder blocks built using the pre-activated residual block. The output of each encoder block functions as a skip connection for the corresponding decoder block. The second and third encoder blocks employ a stride of 2 in the initial 3×3 convolution layer to reduce the spatial dimensions of the feature maps. A stride value 2 reduces the spatial dimensions by half, from 256 to 128. The bridge also includes a pre-activated residual block with a stride value of 2. The decoder utilizes the feature map acquired from the bridge. The model employs skip connections from several encoder blocks to enhance the semantic representation and provide a segmentation mask. The decoder consists of three decoder blocks. Following each block, the size of the feature map is doubled in spatial dimensions while the number of feature channels is reduced. Each decoder block begins with a 2×2 up sampling process, which leads to doubling the spatial dimensions of the feature maps. Afterward, the feature maps are combined with the matching skip connection from the encoder block. The skip connections enable the decoder blocks to acquire the features of the encoder network. Afterward, the feature maps of the concatenation operation move via a pre-activated residual block.

5.2 Data Preparation and Mask Preparation

To implement this network, we use Kaggle brain MRI images. We have 3929 images for training, validation, and testing RESUNET, 1373 of which have a non-tumor label, and the other 2556 have a tumor label. 75% of the data (2990 images) are used for RESUNET training and 22% (900 images) for network validation. We use the remaining 10% of the data (39 images) to test RESUNET. The data preparation summary is given in Table 5.1

Table 5.1 : Data set, train and validation set, test set.

All datasets,	Train and validation set	Test set
3929 images	1362 image with tumor 2528 image without tumor	39 images

Like other supervised deep learning algorithms, supervised segmentation approaches necessitate extensive annotated preprocessed data for training. The necessary annotations differ depending on the segmentation type performed by the model, ranging from specific annotations needed for panoptic segmentation tasks to basic annotations needed for semantic segmentation tasks. Here, we can use different methods to annotate and create masks to fit each photo. Here, we can use different methods to annotate and create masks to fit each picture. For instance, we can use the label me [99] software introduced by MIT University. Examples of input images to the RESUNET network after image preprocessing and preparation of the corresponding mask are shown in Figure 5.3,5.4,5.5.

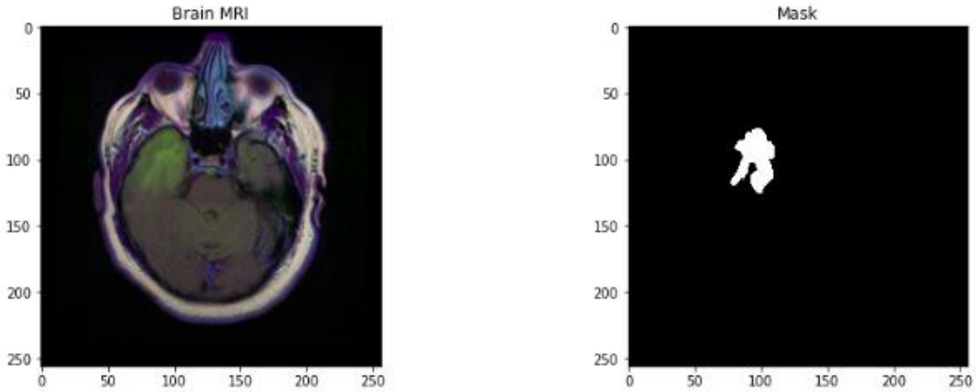

Figure 5.3 : Sample 1 an MRI with a corresponding mask.

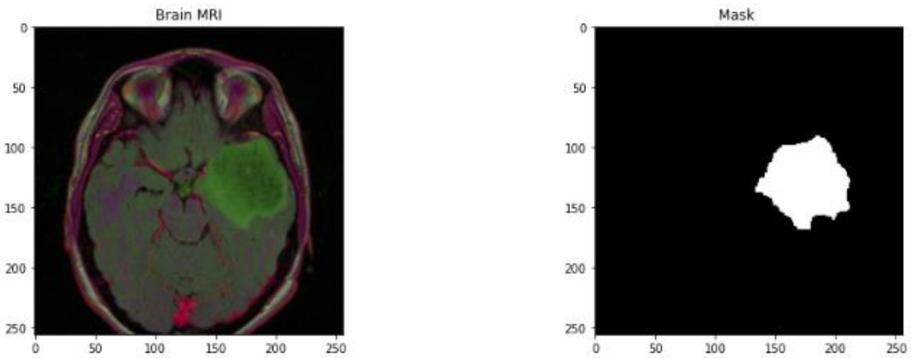

Figure 5.4 : Sample 2 an MRI with a corresponding mask.

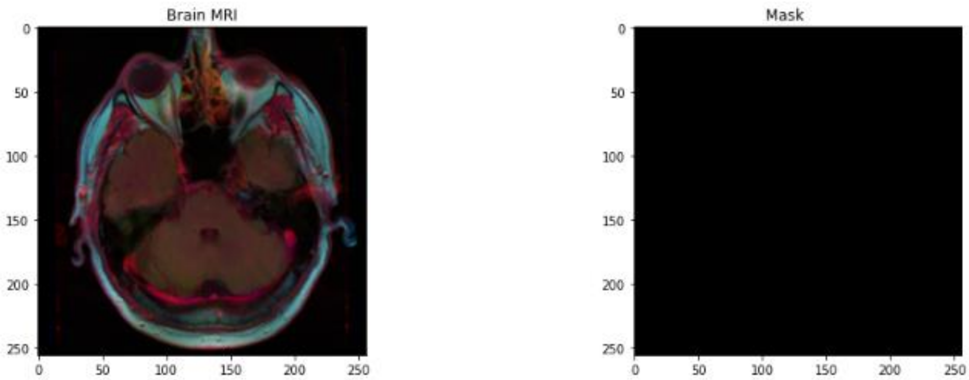

Figure 5.5 : Sample 3 an MRI with a corresponding mask.

We train our RESUNET from the beginning. Specifications of all layers and the parameters that can be trained are given in Tab 5.2. Here, we have vast experience building a complex network with all the controllable parameters in our hands, and the weights are randomly selected and finally optimized.

Table 5.2 : The specifications of all layers and learnable parameters in our U-net

Layer	Layer structure	Shape
1	Input Layer	256, 256, 3
2	Convolutional	256, 256, 16
3	Batch Normalization	256, 256, 16
4	Convolutional	256, 256, 16
5	Batch Normalization	256, 256, 16
6	MaxPooling	128, 128, 16
7	Convolutional	128, 128, 32
8	Batch Normalization	128, 128, 32
9	Activation	128, 128, 32
10	Convolutional	128, 128, 32
11	Convolutional	128, 128, 32
12	Batch Normalization	128, 128, 32
13	Batch Normalization	128, 128, 32
14	Add	128, 128, 32
15	Activation	128, 128, 32
16	MaxPooling	64, 64, 32
17	Convolutional	64, 64, 64
18	Batch Normalization	64, 64, 64
19	Activation	64, 64, 64
20	Convolutional	64, 64, 64
21	Convolutional	64, 64, 64
22	Batch Normalization	64, 64, 64
23	Batch Normalization	64, 64, 64
24	Add	64, 64, 64
25	Activation	64, 64, 64
26	MaxPooling	32, 32, 64
27	Convolutional	32, 32, 128
28	Batch Normalization	32, 32, 128
29	Activation	32, 32, 128
30	Convolutional	32, 32, 128
31	Convolutional	32, 32, 128
32	Batch Normalization	32, 32, 128
33	Batch Normalization	32, 32, 128
34	Add	32, 32, 128
35	Activation	32, 32, 128
36	MaxPooling	16, 16, 128
37	Convolutional	16, 16, 256
38	Batch Normalization	16, 16, 256
39	Activation	16, 16, 256

Table 5.2 (continued) : The specifications of all layers and learnable parameters in our U-net

40	Convolutional	16, 16, 256
41	Convolutional	16, 16, 256
42	Batch Normalization	16, 16, 256
43	Batch Normalization	16, 16, 256
44	Add	16, 16, 256
45	Activation	16, 16, 256
46	UpSampling	32, 32, 256
47	Concatenate	32, 32, 384
48	Convolutional	32, 32, 128
49	BatchNormalization	32, 32, 128
50	Activation	32, 32, 128
51	Convolutional	32, 32, 128
52	Convolutional	32, 32, 128
53	BatchNormalization	32, 32, 128
54	Batch Normalization	32, 32, 128
55	Add	32, 32, 128
56	Activation	32, 32, 128
57	UpSampling	64, 64, 128
58	Concatenate	64, 64, 192
59	Convolutional	64, 64, 64
60	Batch Normalization	64, 64, 64
61	Activation	64, 64, 64
62	Convolutional	64, 64, 64
63	Convolutional	64, 64, 64
64	Batch Normalization	64, 64, 64
65	Batch Normalization	64, 64, 64
66	Add	64, 64, 64
67	Activation	64, 64, 64
68	UpSampling	128, 128, 64
69	Concatenate	128, 128, 96
70	Convolutional	128, 128, 32
71	Batch Normalization	128, 128, 32
72	Activation	128, 128, 32
73	Convolutional	128, 128, 32
74	Convolutional	128, 128, 32
75	Batch Normalization	128, 128, 32
76	Batch Normalization	128, 128, 32
77	Add	128, 128, 32
78	Activation	128, 128, 32
79	UpSampling	256, 256, 32
80	Concatenate	256, 256, 48
81	Convolutional	256, 256, 16
82	Batch Normalization	256, 256, 16
83	Activation	256, 256, 16

Table 5.2 (continued) : The specifications of all layers and learnable parameters in our U-net

84	Convolutional	256, 256, 16
85	Convolutional	256, 256, 16
86	Batch Normalization	256, 256, 16
87	Batch Normalization	256, 256, 16
88	Add	256, 256, 16
89	Activation	256, 256, 16
90	Convolutional	256, 256, 1

Our RESU network ends with almost 50 iterations and more than 1million parameters per iteration with Nvidia RTX3050 graphics card hardware after about 30 hours, and our desired model is made. In the next part, we will examine the performance of this network.

5.3 Model Performance

Ultimately, we assess the system's effectiveness and precision by employing the DSC criteria and evaluating the accuracy of our RESU-net model. The Dice or DSC criterion is commonly employed to quantify the similarity between two photographs. Indeed, it is versatile and may be applied to various forms of data. The dice coefficient is a crucial metric for assessing the outcomes of medical picture segmentation. The Dice coefficient quantifies the degree of similarity between the outputs of the segmentation model and the ground truth texture mask. The Differential Scanning Calorimetry (DSC) equation is provided in equation (5.1), equation (5.2). A dice represents a numerical value ranging from 0 to 1, where a higher value indicates a more remarkable similarity between the two images. When dealing with dice in articles, multiplying them by 100 typically results in a value ranging from 0 to 100 (% DSC). In order to utilize this criterion as a cost function during model training, it is necessary to employ it in order to converge towards the minimal value or -1 during the optimization process.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{5.1}$$

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{5.2}$$

X represents the mask of each image, whereas Y represents the mask that our RESU-net predicts in the image. The coefficient of similarity in the Segmentation is 0.95. Furthermore, Figure 5.7 illustrates the precision of the test and validation data for each iteration of our networks. Additionally, Figure 5.6 illustrates the rate of change of the cost function during each iteration. In each chart, the orange line represents the validation data, while the blue line represents the test data.

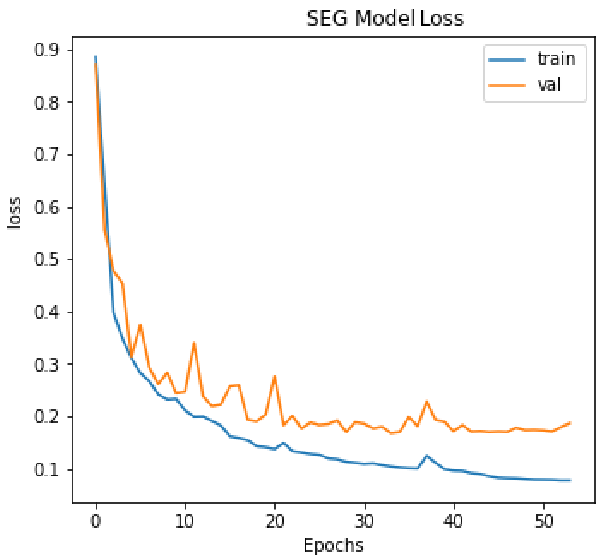

Figure 5.6 : The result of training the RESU-net mode.

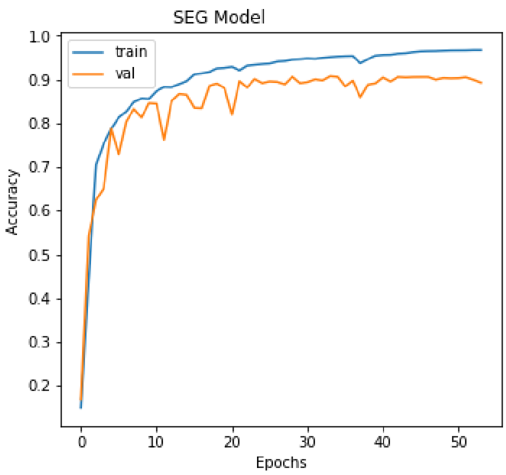

Figure 5.7 : The result of accuracy in the RESU-net model.

Finally, in the output of our RESU-net, the result of some sample Brain MRI images is shown in Figure 5.8, 5.9.

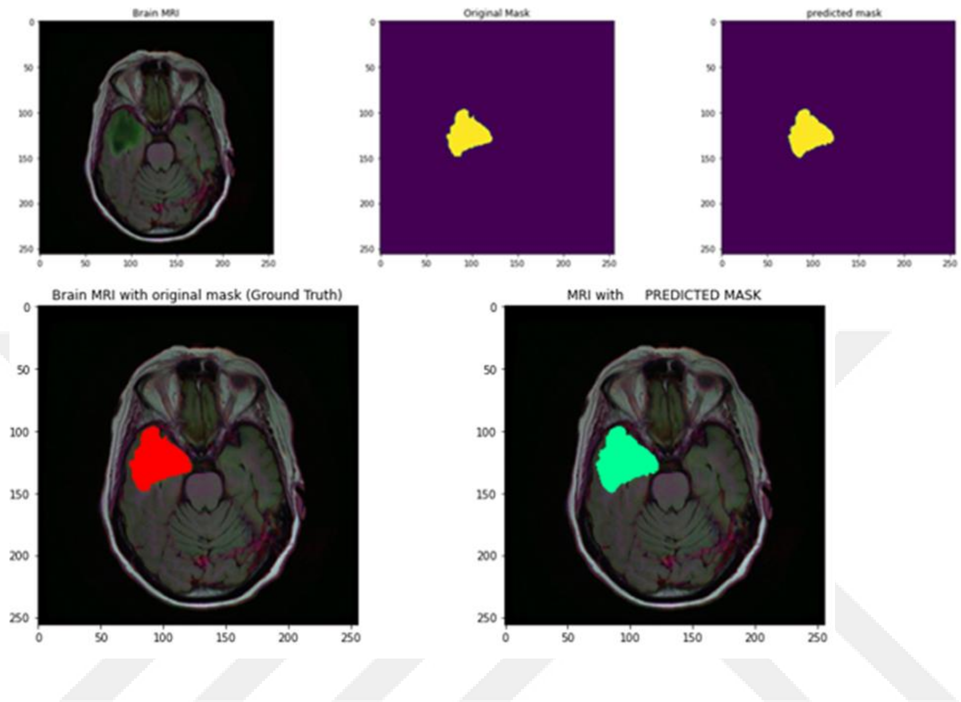

Figure 5.8 : Image segmentation results from sample 1.

.

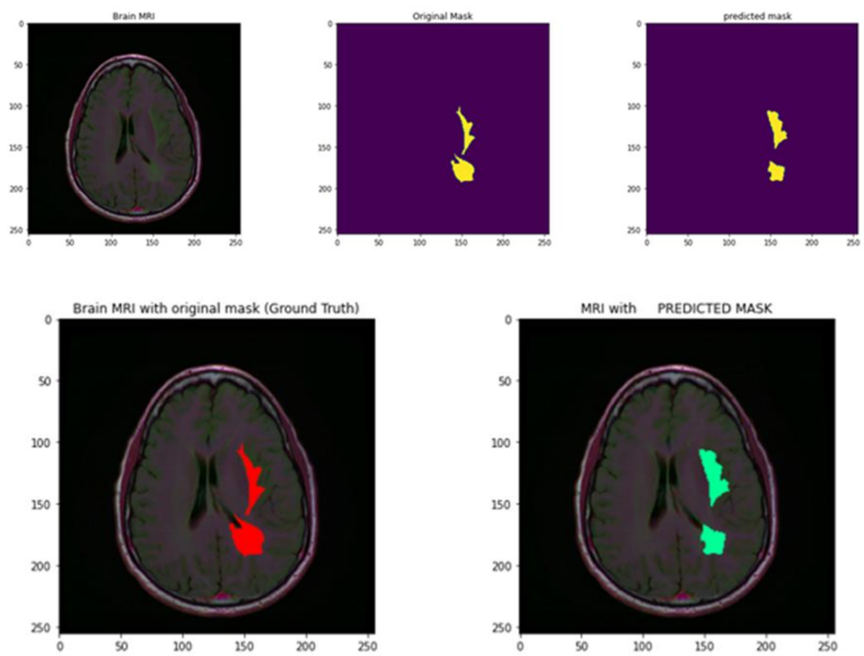

Figure 5.9 : Image segmentation results from sample 2.

6. GENERATIVE ADVERSARIAL NETWORK

6.1 Introduction to GAN

The Generative Adversarial Network (GAN), developed in 2014 by Ian J. Goodfellow and his colleagues, is a machine-learning model [100], [101]. GANs are utilized for unsupervised learning problems in the field of machine learning. These networks consist of two models capable of autonomously detecting and learning patterns in the input data. The generator and the discriminator vie to analyze, document, and reproduce alterations within the dataset. GANs can produce novel samples that closely resemble the data present in the original dataset [102].

The generator is a neural network that generates synthetic data to train the discriminator. The generator acquires the ability to provide data that meets the required standards. The cases generated are negative educational instances for the discriminator. The generator inputs a random noise vector of a specific length and produces a sample. The structure of the generator network is shown in Figure 6.1.

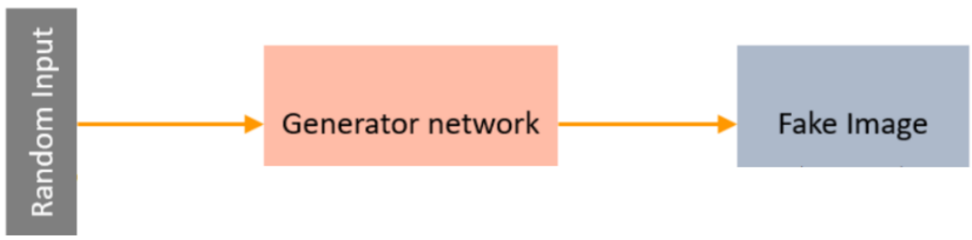

Figure 6.1 : Structure of generator network [103].

The primary purpose of the discriminating deception generator is to classify its output as "authentic."

The GAN network performs the following operations

1. Input vector representing noise.

2. The generative network transforms a random input into data samples.

3. A distinct network that classifies the generated data

4. Production losses occur when the producer is penalized because the auditor is unintelligent.

The backpropagation method adjusts each weight appropriately by assessing its impact on the output. This procedure is also employed to acquire gradients, which can be utilized to adjust the production weights. The generating network and the discriminator are trained using this approach, as seen in Figure 6.2.6.3.

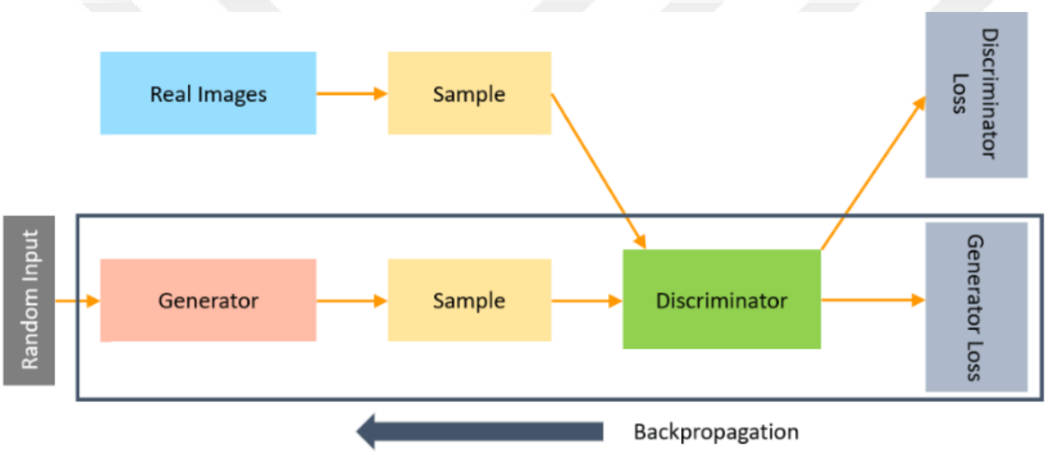

Figure 6.2 : The backpropagation method is used to adjust each weight [103].

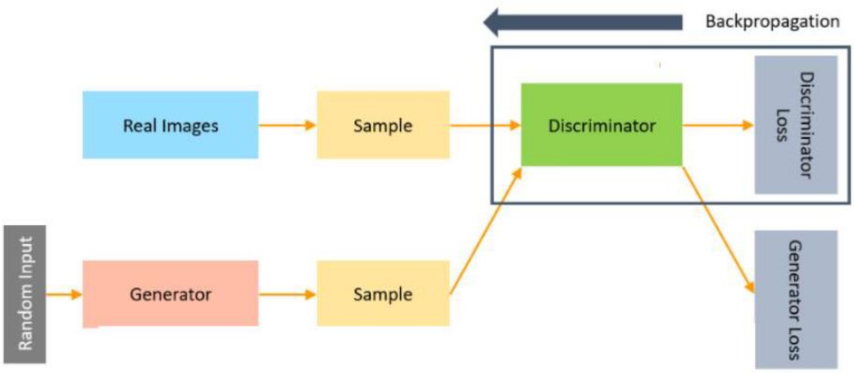

Figure 6.3 : Backpropagation for update weight in discriminator network [103].

A GAN network consists of two distinct cost functions: one for the generator and one for the discriminator. The discriminator error determines both of these expenses. Figure 6.4 displays the summary.

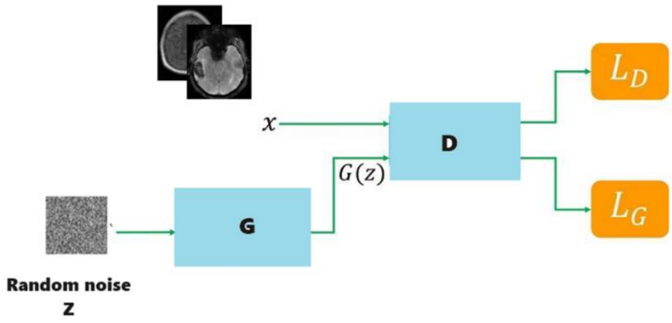

Figure 6.4 : Cost functions within the Generative Adversarial Network.

GAN is typically composed of a dual set of neural networks. There are two functions, G(x) and D(x). Both individuals participate in a competitive game. The purpose of G(x) is to deceive the discriminator by producing data that closely matches the data in the training set. The D(x) will strive to prevent deception by differentiating spurious data from authentic data. Both collaborate to obtain and enhance complex data, such as audio, video, or video files. The GAN takes an input sample and produces a fabricated sample of the data.[1]

The D(x) is associated with two separate loss functions during the training procedure. During discriminant network training, the primary emphasis is optimizing discriminant losses while neglecting productive losses.

The discrimination loss function assesses the precision of the D network by examining the outcomes of both correct and incorrect predictions, as depicted below. The quantity of errors in network D directly influences the magnitude of losses. This illustrates the binomial loss function. The input values are categorized into two terms: "true input" for the x value and "fake input" for the G(z) value. equation (6.1) represents the first component of the loss function for the provided input.

$$l_{\mathrm{Discriminator}_1} = \log \delta\big(D(x)\big) \tag{6.1}$$

The symbol σ denotes a sigmoid excitation function that generates an output value ranging from 0 to 1. A value of 1 in the output signifies the precision of the data. As the sigmoid output number approaches 1, the logarithm converges to 0. Network D has accurately determined that the input x belongs to the actual data, whereas the second claim (G(z)) is a false input. Based on equation (6.2), we anticipate that network D will accurately identify these inputs as fake and will not produce any output specifically for this sort of input.

$$l_{Discriminator_2} = \log\left(1 - \delta\left(D\big(G(z)\big)\right)\right) \tag{6.2}$$

Next, we must combine the two formulas above and formulate the discriminator loss function as equation(6.3).[1]

$$\begin{aligned} L_{Discriminator} &= l_{Discriminator_1} + l_{Discriminator_2} \\ &= \log\delta\big(D(x)\big) + \log\left(1 - \delta\left(D\big(G(z)\big)\right)\right) \end{aligned} \tag{6.3}$$

The generator's loss performance is decided by the discriminator's capacity to differentiate between the synthetic data produced by the generator and the actual data, as depicted in the provided image. equation (6.4) defines the loss function for the generator.

$$l_{\text{Generator}} = \log\delta\left(D\big(G(z)\big)\right) \tag{6.4}$$

6.2 Categories of GAN Networks

Numerous variations of GAN networks exist, and we will introduce a few.

1- The Vanilla GAN, created by Ian Goodfellow, is the original and groundbreaking network in the area [104]. Their optimization strategy, the minimum-maximum formula, entails the discriminator doing binary classification and employing sigmoid cross-entropy losses for optimization. The generator and discriminator in vanilla GANs are implemented as multilayer perceptron’s. This technology improves the optimization of the mathematical equation that controls GANs by employing a stochastic gradient descent technique.

2. Deep Convolutional GANs [105], often called DCGANs, employ convolutional neural networks in both the generator and discriminator instead of standard neural networks. These networks have excellent stability and produce images of higher quality. The generator comprises a sequence of convolutional layers that employ fractional stride convolutions or transpose convolutions. This enables the generator to enhance the resolution of the input image at every convolutional layer. The discriminator is composed of a sequence of convolutional layers that use stride convolutions to progressively reduce the size of the input image at each layer. Conditional GANs [106]are an enhanced version of Vanilla GANs that use additional information to produce better results. The Conditional Generative Adversarial Network (CGAN) incorporates an additional parameter, denoted as "y," into the generator to ensure the generation of relevant data. Labels are inputted into the differentiator to aid in distinguishing between authentic data and artificially generated data. The network can be designed to create data with particular characteristics. Consider a hypothetical scenario with a database of numerical values from 0 to 9. Each number can generate random numerical representations within a consistently hostile network. However, we can build a specific criterion inside this network arrangement by inputting C to obtain the necessary output.

Indeed, while several additional types of GAN networks have been established, these three are the most important.

6.3 DCGAN Model for Making MR Images

To prepare a DCGAN network, we must go through the following steps:

1. define the problem 2. select GAN architecture 3. discriminator training with real data 4. production of fake inputs by the generator 5. discriminator training with fake data, 6. productive training with distinctive output.

In defining the problem, as we saw in the previous sections, having more MR images is crucial for training the network. In classifying our images, we used image augmentation techniques, but these methods generally have drawbacks, for example, as we saw in Figure 12. These images are only obtained by rotating or shifting the original photo and are not independent. To solve this problem, we will try to use the DCGAN network.

To implement this network, we use 3000 brain MRI images; generally, half of the photos have tumors, and the other half do not. This data set is a combination of our data set in our classification problem and its combination with the second data set in our segmentation problem. Examples of this dataset are also given in Figure 6.5,6.6.

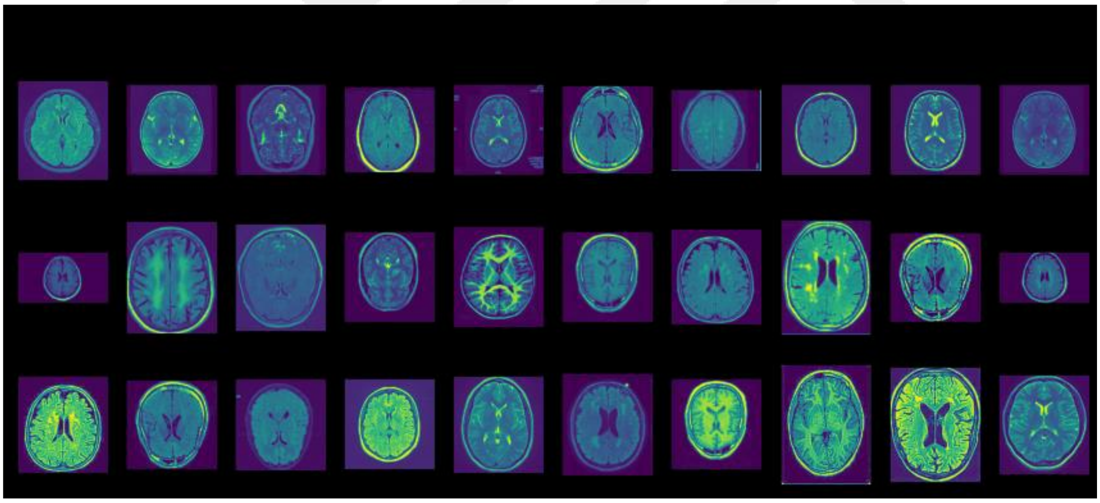

Figure 6.5 : Our sample data set without tumor.

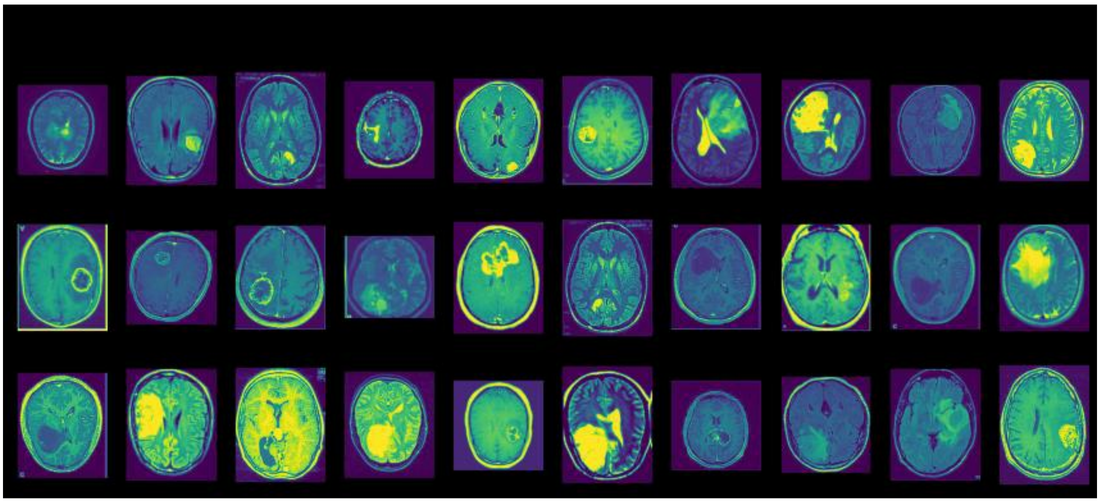

Figure 6.6 : Our sample data set with tumor.

As shown in Figure 6.7, our goal in this network is to create new MR images similar to real MRI.

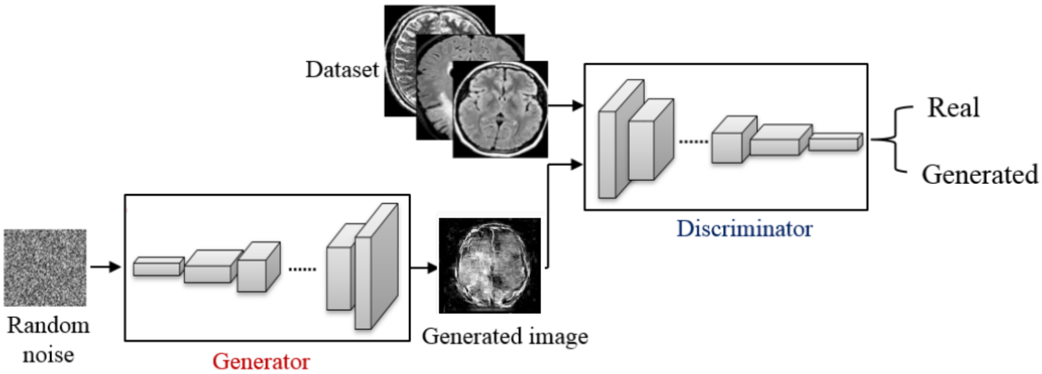

Figure 6.7 : Overview of our GAN.

Here, we train our DCGAN neural network from the beginning, and the specifications of all layers and the parameters that can be trained are given in Table 6.1,6.2. Table 6.1 lists all the layers and specifications of our generator network, and Table 6.2. lists all parts of the discriminator network. As in the previous section, we consider the rest of the network settings and cost functions for simplicity and better results.

Table 6.1 : The specifications of all layers in discriminator[1].

Layer	width	Hight	Depth	Stride	Layer info
Dense	1	1	262144	-	-
Activation	1	1	262144	-	Leaky ReLU
Reshape	32	32	256	-	-
Convolutional Transpose	64	64	256	2	-
Activation	64	64	256	-	Leaky ReLU
Convolutional Transpose	128	128	256	2	-
Activation	128	128	256	-	Leaky ReLU
Convolutional	128	128	1	-	-
Activation	128	128	1	-	Tanh

Table 6.2 : Internal structure and discriminating model layers.[1]

Layer	width	Hight	Depth	Stride	Layer info
Convolutional	128	128	64	-	-
Activation	128	128	64	-	Leaky ReLU
Convolutional	64	64	128	2	-
Activation	64	64	128	-	Leaky ReLU
Convolutional	32	32	128	2	-
Activation	32	32	128	-	Leaky ReLU
Convolutional	16	16	256	2	-
Activation	16	16	256	-	Leaky ReLU
Flatten	1	1	65536	-	-
Dropout	1	1	65536	-	Rate 0.4
Dense	1	1	65537	-	-
Activation	1	1	65537	-	Sigmoid

Finally, we will examine several images created by the generator network. In order not to be long and tedious, only five results out of 12 iterations of the program and images created in our network are examined, and the results with different iterations are given in Figures 6.8, 6.9, 6.10, 6.11, and 6.12.

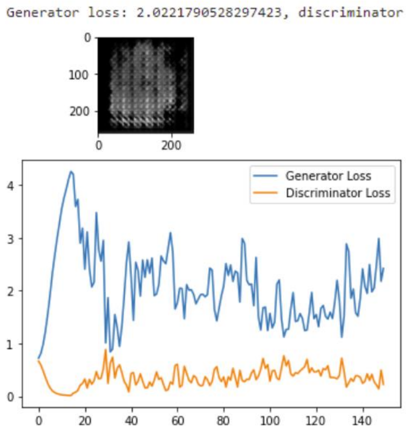

Figure 6.8 : The value of the productive and discriminating cost function in the first stage and the production sample.

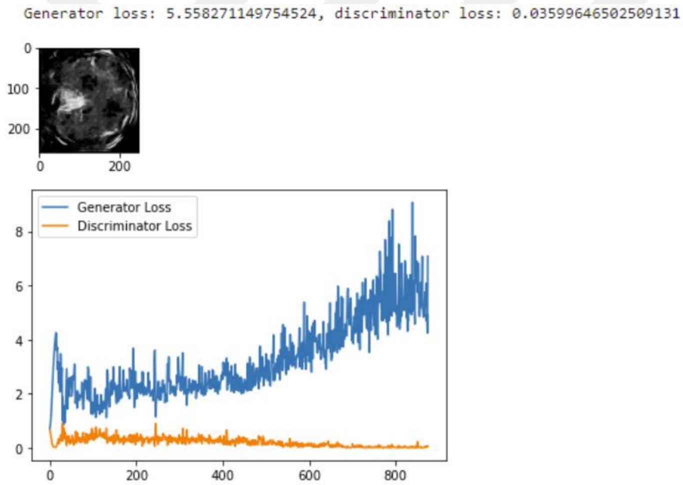

Figure 6.9 : The value of the productive and discriminating cost function in the third stage and the production sample.

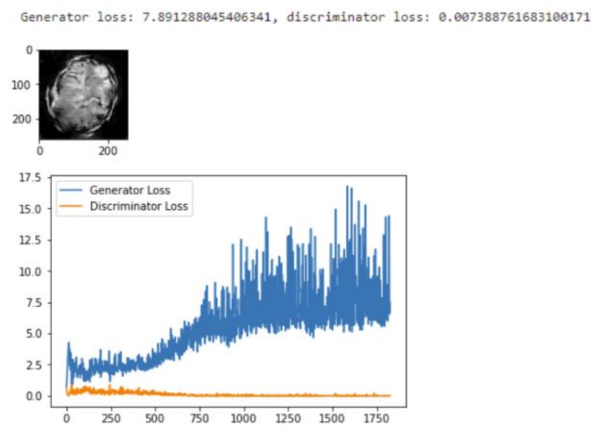

Figure 6.10 : The value of the productive and discriminating cost function in the sixth stage and the production sample.

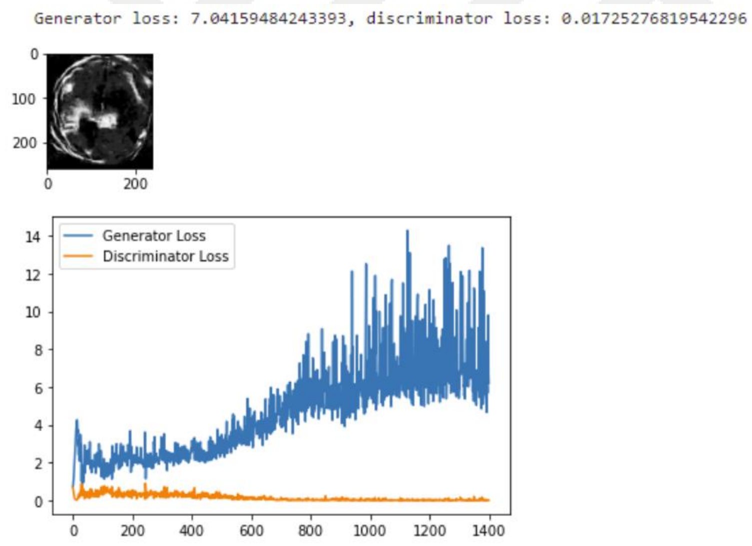

Figure 6.11 : The value of the productive and discriminating cost function in the ninth stage and the production sample.

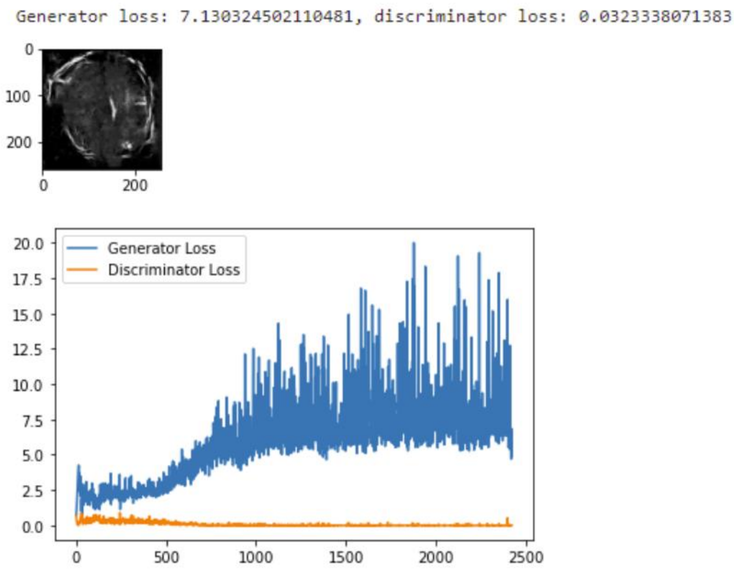

Figure 6.12 : The value of the productive and discriminating cost function in the twelfth stage and production sample.

7. CONCLUSIONS

This thesis critically examines the limitations of many conventional machine learning techniques, such as Sobel and Canny edge detector filters, for sharping and segmentation of brain MR images. Despite the advantages and ease of using these traditional methods in medical images, it is clear that these techniques do not always bring satisfactory results and highlight the need for innovation and change in our approach.

Contrary to conventional machine learning approaches, new techniques, such as deep neural networks, including RESNET in classification and RESUNET in segmentation, bring satisfactory and impressive results in brain MR datasets despite the complex and lengthy implementation procedure. In the deep learning section, we presented RESUNET by combining the residual networks with the deep neural network. This network solves one of the most fundamental problems of deep networks, i.e., the vanishing gradient. It shows improved performance during training, and we obtained a faster convergence rate than traditional Unet topologies. The remaining (residual) blocks effectively propagate the local features of the segmentation system to the whole system, making them suitable for the automatic solution of combinatorial segmentation problems. The results show the success of the proposed network with a dice score of 94% in the segmentation of brain MR medical images.

In the transformations section, by comparing the three transformations, contourlet, wavelet and curvelet, we implement the most suitable transformation for brain medical images, i.e. contourlet, in order to identify the edges and lines of the image in different directions and then use this transformation to reduce the size of medical images. In addition, we introduce the combination of this transformation with morphological operators as a new technique to generate the ground truth (mask) for deep neural network segmentation training. The "ground truth (mask)" is a reference for supervised networks and is used to train deep neural networks such as the RESUNET image segmentation network.

In the last part of this research, the adversarial generator network is introduced as an unsupervised network. In this chapter, we solve one of biomedical images fundamental problems: providing suitable and sufficient data for training deep networks. We also generate high-quality brain MR data that can be used as input in future research. We can use the images of this neural network to segment or categorize medical images and other related tasks, and we can even use this network as a new technique for data augmentation.

Overall, this study presents a comprehensive automated method for brain image segmentation. This system combines a supervised deep learning model, such as RESUNET, with an unsupervised deep learning model, such as a generative adversarial network. As a result, by combining these two deep networks, this system can be applied to many datasets, including using some with a small number of samples and presenting an utterly automatic image segmentation method in different areas.

7.1 FUTURE WORKS

In the continuation of this thesis, the following topics are suggested for future works.

In the image segmentation section, we can use 3D RESUNET models to calculate the volume of the tumor area and more accurately determine the tumor’s location.

In the section on generative adversarial networks, instead of the DCGAN model, we can use a conditional adversarial generative network. This method can produce images that closely resemble authentic images.

Practical implementation of this research and segmentation of brain MRI images online can be one of the most vital tasks of this research. If this system is implemented and designed, there will be no existing similar model and it can be replaced with the old models used in hospitals with lower cost and more accuracy.

REFERENCES

[1] **N. Khalili Dizaji and M. Doğan**, “A Comprehensive Brain MRI Image Segmentation System Based on Contourlet Transform and Deep Neural Networks,” *Algorithms*, vol. 17, no. 3, 2024, doi: 10.3390/a17030130.

[2] **A. Kentab**, “Further Brain Magnetic Resonance Imaging (MRI) Delineation of Aicardi-Goutières Syndrome (AGS),” *Open Access Journal of Neurology & Neurosurgery*, vol. 9, Feb. 2019, doi: 10.19080/OAJNN.2019.09.555772.

[3] **P. S. Kumar and D. Vs**, “A Study of MRI Segmentation Methods in Automatic Brain Tumor Detection,” *International Journal of Engineering and Technology*, vol. 8, pp. 609–614, Apr. 2016.

[4] **K. C. Peng *et al*.**, “Unmanned Aerial Vehicle for infrastructure inspection with image processing for quantification of measurement and formation of facade map,” in *2017 International Conference on Applied System Innovation (ICASI)*, May 2017, pp. 1969–1972. doi: 10.1109/ICASI.2017.7988578.

[5] **P. Fang, W. Zecong**, and X. Zhang, “Vehicle automatic driving system based on embedded and machine learning,” in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Jul. 2020, pp. 281–284. doi: 10.1109/CVIDL51233.2020.00-85.

[6] **J. de Ruijter**, J. Muijsers, F. van de Vosse, M. Sambeek, and R. Lopata, “A Generalized Approach for Automatic 3-D Geometry Assessment of Blood Vessels in Transverse Ultrasound Images Using Convolutional Neural Networks,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. PP, pp. 1–1, Jun. 2021, doi: 10.1109/TUFFC.2021.3090461.

[7] **M. Antico *et al*.**, “Deep Learning-Based Automatic Segmentation for Reconstructing Vertebral Anatomy of Healthy Adolescents and Patients With Adolescent Idiopathic Scoliosis (AIS) Using MRI Data,” *IEEE Access*, vol. 9, pp. 86811–86823, 2021, doi: 10.1109/ACCESS.2021.3084949.

[8] **I. Hathout, K. Callery**, T. Hathout, and U. Sivagnanenthirarajah, “Digital image expert system for corrosion analysis of steel transmission structures,” in *2017 IEEE Power & Energy Society General Meeting*, Jul. 2017, pp. 1–5. doi: 10.1109/PESGM.2017.8274521.

[9] **X. Chai, G. Tang, S. Wang, K. Lin, and R. Peng**, "Deep Learning for Irregularly and Regularly Missing 3-D Data Reconstruction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–22, Aug. 2020, doi: 10.1109/TGRS.2020.3016343.

[10] **L. Zhang and L. Zhang**, "Deep Learning-Based Classification and Reconstruction of Residential Scenes From Large-Scale Point Clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 1887–1897, Apr. 2018, doi: 10.1109/TGRS.2017.2769120.

[11] **H. Aydın and B. Dizdaroğlu**, "Medical image segmentation with edge based level sets: Mobile client-server application," in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct. 2017, pp. 994–999. doi: 10.1109/UBMK.2017.8093412.

[12] **Panda and Rosenfeld**, "Image Segmentation by Pixel Classification in (Gray Level, Edge Value) Space," *IEEE Transactions on Computers*, vol. C–27, no. 9, pp. 875–879, Sep. 1978, doi: 10.1109/TC.1978.1675208.

[13] **J. S. Weszka and A. Rosenfeld**, "Threshold Evaluation Techniques," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 622–629, 1978.

[14] **D. Bouzidi, F. Ghozzi, and A**. Fakhfakh, "MR Brain Image Segmentation Optimized by Using Ant Colony Algorithm," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, Mar. 2021, pp. 1230–1236. doi: 10.1109/SSD52085.2021.9429465.

[15] **S. Kaur and P. Kaur,** "An Edge detection technique with image segmentation using Ant Colony Optimization: A review," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Nov. 2016, pp. 1–5. doi: 10.1109/GET.2016.7916741.

[16] **Manisha, B. Radhakrishnan,** and L. P. Suresh, "Tumor region extraction using edge detection method in brain MRI images," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, Apr. 2017, pp. 1–5. doi: 10.1109/ICCPCT.2017.8074326.

[17] **G. Parthasarathy, R. Lakshmanan**, A. K, and J. Y Y, "Predicting Source and Age of Brain Tumor Using Canny Edge Detection Algorithm and Threshold Technique," *Asian Pacific Journal of Cancer Prevention*, vol. 20, pp. 1409–1414, May 2019, doi: 10.31557/APJCP.2019.20.5.1409.

[18] **R. Rulaningtyas and K. Ain,** "Edge detection for brain tumor pattern recognition," in *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009*, Nov. 2009, pp. 1–3. doi: 10.1109/ICICI-BME.2009.5417299.

[19] **Xie Mei, Zhen Zheng, Wu Bingrong, and Li Guo,** "The edge detection of brain tumor," in *2009 International Conference on Communications, Circuits and Systems*, Jul. 2009, pp. 477–479. doi: 10.1109/ICCCAS.2009.5250487.

[20] **E. E. Ulku and A. Y. Camurcu,** "Computer aided brain tumor detection with histogram equalization and morphological image processing techniques," in *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, Nov. 2013, pp. 48–51. doi: 10.1109/ICECCO.2013.6718225.

[21] **M. R. Islam, M. R. Imteaz, and Marium-E-Jannat**, "Detection and analysis of brain tumor from MRI by Integrated Thresholding and Morphological Process with Histogram based method," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, Feb. 2018, pp. 1–5. doi: 10.1109/IC4ME2.2018.8465663.

[22] **S. S. Hunnur, A. Raut, and S. Kulkarni,** "Implementation of image processing for detection of brain tumors," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, Jul. 2017, pp. 717–722. doi: 10.1109/ICCMC.2017.8282559.

[23] **Y. Sharma and Y. K. Meghrajani,** "Brain tumor extraction from MRI image using mathematical morphological reconstruction," in *2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking*, Dec. 2014, pp. 1–4. doi: 10.1109/ET2ECN.2014.7044982.

[24] **A. Soni and A. Rai,** "CT Scan Based Brain Tumor Recognition and Extraction using Prewitt and Morphological Dilation," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–6. doi: 10.1109/ICCCI50826.2021.9402677.

[25] **Z. Zulkoffli and T. A. Shariff**, "Detection of Brain Tumor and Extraction of Features in MRI Images Using K-means Clustering and Morphological Operations," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Jun. 2019, pp. 1–5. doi: 10.1109/I2CACIS.2019.8825094.

[26] **A. Biswas and M. S. Islam**, "Brain Tumor Types Classification using K-means Clustering and ANN Approach," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Jan. 2021, pp. 654–658. doi: 10.1109/ICREST51555.2021.9331115.

[27] **R. Agrawal, M. Sharma**, and B. Singh, "Segmentation of Brain Tumour Based on Clustering Technique: Performance Analysis," *Journal of Intelligent Systems*, vol. 28, Jul. 2017, doi: 10.1515/jisys-2017-0027.

[28] **B. Srinivas and G. S. Rao,** "Unsupervised learning algorithms for MRI brain tumor segmentation," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, Jan. 2018, pp. 181–184. doi: 10.1109/SPACES.2018.8316341.

[29] **M. -N. Wu, C. -C. Lin, and C. -C. Chang,** "Brain Tumor Detection Using Color-Based K-Means Clustering Segmentation," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, Nov. 2007, pp. 245–250. doi: 10.1109/IIHMSP.2007.4457697.

[30] **N. Singh,** “Automatic Detection of Brain Tumor Using K-Means Clustering,” *International Journal for Research in Applied Science and Engineering Technology*, vol. V, pp. 114–121, Nov. 2017, doi: 10.22214/ijraset.2017.11017.

[31] **P. Birgani,** M. Ashtiyani, and S. Asadi, *MRI Segmentation Using Fuzzy C-means Clustering Algorithm Basis Neural Network*. 2008, p. 5. doi: 10.1109/ICTTA.2008.4530110.

[32] **M. Thilagam,** K. Arunesh, and A. Rajeshkanna, “Analysis of brain MRI images for Tumor segmentation using Fuzzy C means Algorithm,” in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, pp. 183–186. doi: 10.1109/ICESC48915.2020.9155646.

[33] **R. Preetha and G. R. Suresh,** “Performance Analysis of Fuzzy C Means Algorithm in Automated Detection of Brain Tumor,” in *2014 World Congress on Computing and Communication Technologies*, Mar. 2014, pp. 30–33. doi: 10.1109/WCCCT.2014.26.

[34] **J. Shedbalkar, K. Prabhushetty, and A. Inchalc,** “A Comparative Analysis of Filters for Noise Reduction and Smoothening of Brain MRI Images,” in *2021 6th International Conference for Convergence in Technology (I2CT)*, Apr. 2021, pp. 1–6. doi: 10.1109/I2CT51068.2021.9417979.

[35] **S. Pereira, A. Pinto, V. Alves, and C. A. Silva,** “Brain tumor segmentation using convolutional neural networks in MRI images,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.

[36] **C. Rajeshkumar, K. R. Soundar, M. Sneha, S. S. Maheswari, M. S. Lakshmi, and R. Priyanka**, “Convolutional Neural Networks (CNN) based Brain Tumor Detection in MRI Images,” in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Jan. 2023, pp. 976–979. doi: 10.1109/ICSSIT55814.2023.10060968.

[37] **I. Khan, K. Ahsan, Md. A. Hasan, and A. Sattar**, *Brain Tumor Analysis Using Deep Neural Network*. 2021, p. 1754. doi: 10.1109/ICICCS51141.2021.9432235.

[38] **A. Sinha, A. R P, M. Suresh, N. Mohan R, A. D, and A. G. Singerji,** “Brain Tumour Detection Using Deep Learning,” in *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*, Mar. 2021, pp. 1–5. doi: 10.1109/ICBSII51839.2021.9445185.

[39] **N. Bnouni, H. B. Amor, I. Rekik, M. S. Rhim, B. Solaiman, and N. E. B. Amara**, “Boosting CNN Learning by Ensemble Image Preprocessing Methods for Cervical Cancer Segmentation,” in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, Mar. 2021, pp. 264–269. doi: 10.1109/SSD52085.2021.9429422.

[40] **R. Ezhilarasi and P. Varalakshmi,** *Tumor Detection in the Brain using Faster R-CNN*. 2018, p. 392. doi: 10.1109/I-SMAC.2018.8653705.

[41] **S. Grampurohit, V. Shalavadi, V. R. Dhotargavi, M. Kudari, and S. Jolad,** "Brain Tumor Detection Using Deep Learning Models," in *2020 IEEE India Council International Subsections Conference (INDISCON)*, Oct. 2020, pp. 129–134. doi: 10.1109/INDISCON50162.2020.00037.

[42] **D. P. L, N. Ponraj, and S. V. G,** "A Comparative Analysis of Deep Neural Networks for Brain Tumor Detection," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021, pp. 36–40. doi: 10.1109/ICSPC51351.2021.9451742.

[43] **A. Saleh, R. Sukaik, and S. S. Abu-Naser**, "Brain Tumor Classification Using Deep Learning," in *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)*, Aug. 2020, pp. 131–136. doi: 10.1109/iCareTech49914.2020.00032.

[44] **N. Abiwinanda, M. Hanif, S. Hesaputra, A. Handayani, and T. Mengko,** "Brain Tumor Classification Using Convolutional Neural Network," 2019, pp. 183–189. doi: 10.1007/978-981-10-9035-6_33.

[45] **M. M. Badža and M. Č. Barjaktarović,** "Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network," *Applied Sciences*, vol. 10, no. 6, 2020, doi: 10.3390/app10061999.

[46] **K. DAĞLI and O. EROĞUL,** "Classification of Brain Tumors via Deep Learning Models," in *2020 Medical Technologies Congress (TIPTEKNO)*, Nov. 2020, pp. 1–4. doi: 10.1109/TIPTEKNO50054.2020.9299231.

[47] **W. Ayadi, W. Elhamzi, and M. Atri,** "A new deep CNN for brain tumor classification," in *2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Dec. 2020, pp. 266–270. doi: 10.1109/STA50679.2020.9329328.

[48] **C. L. Choudhury, C. Mahanty, R. Kumar, and B. K. Mishra,** "Brain Tumor Detection and Classification Using Convolutional Neural Network and Deep Neural Network," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, Mar. 2020, pp. 1–4. doi: 10.1109/ICCSEA49143.2020.9132874.

[49] **M. Yasir, U. Hayat, A. U. Rahman, and F. Riaz,** "Classification and Detection of Glioblastoma Tumor from MRI Images," in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, Jan. 2021, pp. 322–327. doi: 10.1109/IBCAST51254.2021.9393176.

[50] **B. Sandhiya, R. Priyatharshini, B. Ramya, S. Monish, and G. R. Sai Raja,** "Reconstruction, Identification and Classification of Brain Tumor Using Gan and Faster Regional-CNN," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021, pp. 238–242. doi: 10.1109/ICSPC51351.2021.9451747.

[51] **M. N. Do and M. Vetterli,** "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on image processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[52] **M. Abdoli, R. Dierckx, and H. Zaidi,** “Contourlet-based active contour model for PET image segmentation,” *Medical physics*, vol. 40, p. 082507, Aug. 2013, doi: 10.1118/1.4816296.

[53] **H. R. Ramya and B. K. Sujatha,** “A novel approach for medical image fusion using fuzzy logic type-2,” in *2016 International Conference on Circuits, Controls, Communications and Computing (I4C)*, Oct. 2016, pp. 1–5. doi: 10.1109/CIMCA.2016.8053286.

[54] **P. Prasad, S. Subramani, V. Bhavana, and H. K. Krishnappa,** “Medical Image Fusion Techniques Using Discrete Wavelet Transform,” in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2019, pp. 614–618. doi: 10.1109/ICCMC.2019.8819672.

[55] **A. D. Olayemi, A. Dharmaratne, and M. F. Pasha,** “Deep Learning and Late Fusion Technique in Medical X-ray Image,” in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Dec. 2020, pp. 460–465. doi: 10.1109/ICARCV50220.2020.9305346.

[56] **A. S. Yousif, Z. Omar, U. U. Sheikh, and S. A. Khalid**, “A New Scheme of Medical Image Fusion Using Deep Convolutional Neural Network and Local Energy Pixel Domain,” in *2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, Mar. 2021, pp. 384–388. doi: 10.1109/IECBES48179.2021.9398840.

[57] **Z. Le, J. Huang, F. Fan, X. Tian, and J. Ma,** *A Generative Adversarial Network For Medical Image Fusion*. 2020, p. 374. doi: 10.1109/ICIP40778.2020.9191089.

[58] **J. E. Rodriguez Rojas and D. Ayal**a, *Erosion and Dilation on 2-D and 3-D Digital Images: A New Size-Independent Approach* . 2001.

[59] **W. Zhang and D. Jiang,** “The marker-based watershed segmentation algorithm of ore image,” in *2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 472–474. doi: 10.1109/ICCSN.2011.6014611.

[60] **N. Albassam, V. Ramachandran, and S. Parameswaran,** “Wavelet Theory and Application in Communication and Signal Processing,” 2021. doi: 10.5772/intechopen.95047.

[61] **A. Prakash,** “Wavelet and its Applications,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, pp. 95–104, Nov. 2018, doi: 10.32628/CSEIT183820.

[62] **M. N. Do and M. Vetterli,** “Contourlets: a directional multiresolution image representation,” presented at the Proceedings. International Conference on Image Processing, IEEE, 2002, p. I–I.

[63] **H. Sadreazami, M. O. Ahmad, and M. N. S. Swamy**, “Contourlet domain image denoising using the alpha-stable distribution,” in *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2014, pp. 141–144. doi: 10.1109/MWSCAS.2014.6908372.

[64] **H. Yan, F. Chen, and W. -w. Li,** "Inter-scale Correlation Image Denoising Based on Non-aliasing Contourlet Transform," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, Mar. 2010, pp. 516–519. doi: 10.1109/ICMTMA.2010.241.

[65] **J. M. Mejía Muñoz *et al.*,** "SAR image denoising using the non-subsampled contourlet transform and morphological operators," presented at the Advances in Artificial Intelligence: 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part I 9, Springer, 2010, pp. 337–347.

[66] **S. Saman and S. Jamjala Narayanan,** "Survey on brain tumor segmentation and feature extraction of MR images," *International journal of multimedia information retrieval*, vol. 8, pp. 79–99, 2019.

[67] **X. Bi, X. Chen, and X. Li,** "Medical image compressed sensing based on contourlet," presented at the 2009 2nd International Congress on Image and Signal Processing, IEEE, 2009, pp. 1–4.

[68] **H. Masood, M. Asim, M. Mumtaz, and A. B. Mansoor,** "Combined Contourlet and Non-subsampled Contourlet Transforms Based Approach for Personal Identification Using Palmprint," in *2009 Digital Image Computing: Techniques and Applications*, Dec. 2009, pp. 408–415. doi: 10.1109/DICTA.2009.73.

[69] **H. Sajedi and M. Jamzad,** "A contourlet-based face detection method in color images," presented at the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, IEEE, 2007, pp. 727–732.

[70] **Yu Hai-Long, Ren Huo-Rong, and Yan Kai,** "Contourlet transform based algorithm of shadow compensation for face recognition," in *2010 International Conference on Image Analysis and Signal Processing*, Apr. 2010, pp. 671–674. doi: 10.1109/IASP.2010.5476183.

[71] **M. Kumar, N. Ranjan, and B. Chourasia,** "Hybrid Methods of Contourlet Transform and Particle Swarm Optimization for Multimodal Medical Image Fusion," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Mar. 2021, pp. 945–951. doi: 10.1109/ICAIS50930.2021.9396021.

[72] **X. Ji,** "An Improved Image Fusion Method of Infrared Image and SAR Image Based on Contourlet and Sparse Representation," in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Aug. 2015, pp. 282–285. doi: 10.1109/IHMSC.2015.11.

[73] **L.-W. Liu, X.-W. Chen, and Z.-W. Ying,** "Texture image retrieval algorithm with dual tree complex contourlet and three statistical features," *2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, May 2011, doi: 10.1109/ICCSN.2011.6013609.

[74] **R. Romdhane, H. Mahersia, and K. Hamrouni,** "A Novel Content Image Retrieval Method Based on Contourlet," in *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, Apr. 2008, pp. 1–5. doi: 10.1109/ICTTA.2008.4529999.

[75] **E. Adelson, C. Anderson, J. Bergen, P. Burt, and J. Ogden,** "Pyramid Methods in Image Processing," *RCA Eng.*, vol. 29, Nov. 1983.

[76] **E. Brynjolfsson and T. Mitchell,** "What can machine learning do? Workforce implications," *Science*, vol. 358, pp. 1530–1534, Dec. 2017, doi: 10.1126/science.aap8062.

[77] **G. Hinton *et al.*,** "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.

[78] **M. Ibnkahla,** "Applications of neural networks to digital communications-survey," *Signal Processing*, vol. 80, pp. 1185–1215, Jul. 2000, doi: 10.1016/S0165-1684(00)00030-X.

[79] **T. Agrawal, R. Gupta, and S. Narayanan**, *On Evaluating CNN Representations for Low Resource Medical Image Classification*. 2019, p. 1367. doi: 10.1109/ICASSP.2019.8682397.

[80] **T. Agrawal, R. Gupta, and S. Narayanan,** "On Evaluating CNN Representations for Low Resource Medical Image Classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 1363–1367. doi: 10.1109/ICASSP.2019.8682397.

[81] **Xiang Li, W. Li, Xiaodong Xu, and Wei Hu,** "Cell classification using convolutional neural networks in medical hyperspectral imagery," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Jun. 2017, pp. 501–504. doi: 10.1109/ICIVC.2017.7984606.

[82] **Y. Yu, E. Favour, and P. Mazumder,** "Convolutional Neural Network Design for Breast Cancer Medical Image Classification," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Oct. 2020, pp. 1325–1332. doi: 10.1109/ICCT50939.2020.9295909.

[83] **S. Kido, Y. Hirano, and N. Hashimoto,** "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN)," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018, pp. 1–4. doi: 10.1109/IWAIT.2018.8369798.

[84] **B. Sushma, C. K. Raghavendra, and J. Prashanth,** "CNN based U-Net with Modified Skip Connections for Colon Polyp Segmentation," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, pp. 1762–1766. doi: 10.1109/ICCMC51019.2021.9418037.

[85] **K. Brahma, V. Kumar, A. E. Samir, A. P. Chandrakasan, and Y. C. Eldar,** "Efficient Binary Cnn For Medical Image Segmentation," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2021, pp. 817–821. doi: 10.1109/ISBI48211.2021.9433901.

[86] **L. Liu, F. -X. Wu, Y. -P. Wang, and J. Wang,** "Multi-Receptive-Field CNN for Semantic Segmentation of Medical Images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3215–3225, Nov. 2020, doi: 10.1109/JBHI.2020.3016306.

[87] **P. Sedigh, R. Sadeghian, and M. T. Masouleh,** "Generating Synthetic Medical Images by Using GAN to Improve CNN Performance in Skin Cancer Classification," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, Nov. 2019, pp. 497–502. doi: 10.1109/ICRoM48714.2019.9071823.

[88] **S. Zargar,** *Introduction to Convolutional Neural Networks*. 2021. doi: 10.13140/RG.2.2.11572.17282.

[89] **K. He, X. Zhang, S. Ren, and J. Sun,** "Deep residual learning for image recognition," presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[90] **B. S. Negara, E. Satria, S. Sanjaya, and D. R. Dwi Santoso,** "ResNet-50 for Classifying Indonesian Batik with Data Augmentation," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, Jul. 2021, pp. 1–4. doi: 10.1109/ICOTEN52080.2021.9493488.

[91] **M. Shafiq and Z. Gu,** "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences*, vol. 12, no. 18, 2022, doi: 10.3390/app12188972.

[92] **N. Zakaria, F. Mohamed, R. Abdelghani, and K. Sundaraj,** "VGG16, ResNet-50, and GoogLeNet Deep Learning Architecture for Breathing Sound Classification: A Comparative Study," in *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, Nov. 2021, pp. 1–6. doi: 10.1109/AI-CSP52968.2021.9671124.

[93] **J. S. Athertya and G. S. Kumar,** "Data augmentation techniques for classifying vertebral bodies from MR images," presented at the Data Science Analytics and Applications: First International Conference, DaSAA 2017, Chennai, India, January 4-6, 2017, Revised Selected Papers 1, Springer, 2018, pp. 38–45.

[94] **R. Alcala, Z. Arceo, J. Baterisna, J. Morada, and J. O. Ramirez,** *Image Processing: Orthogonal Image Rotation and Flipping Using Matrix Operations*. 2017. doi: 10.13140/RG.2.2.10186.26561.

[95] **T. Asano, S. Bitou, M. Motoki, and N. Usui,** "In-place algorithm for image rotation," presented at the Algorithms and Computation: 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007. Proceedings 18, Springer, 2007, pp. 704–715.

[96] **S. K. Mitra, C. Murthy, and M. K. Kundu,** "A technique for image magnification using partitioned iterative function system," *Pattern recognition*, vol. 33, no. 7, pp. 1119–1133, 2000.

[97] **A. S. Akbar, C. Fatichah, and N. Suciati,** "Single level UNet3D with multipath residual attention block for brain tumor segmentation," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 3247–3258, 2022.

[98] **S. Ji,** "Multi-classification of Human Action Based on ResNet," *Highlights in Science, Engineering and Technology*, vol. 23, pp. 98–105, Dec. 2022, doi: 10.54097/hset.v23i.3203.

[99] **A. Torralba, B. C. Russell, and J. Yuen,** "LabelMe: Online Image Annotation and Applications," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1467–1484, Aug. 2010, doi: 10.1109/JPROC.2010.2050290.

[100] **I. Goodfellow *et al.***, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[101] **T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen,** "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[102] **J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky,** "Adversarial Learning for Neural Dialogue Generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2157–2169. doi: 10.18653/v1/D17-1230.

[103] **K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. -Y. Wang,** "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017, doi: 10.1109/JAS.2017.7510583.

[104] **K. Simonyan and A. Zisserman,** "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv 1409.1556*, Sep. 2014.

[105] **A. Chapagain**, *DCGAN--Image Generation*. 2019. doi: 10.13140/RG.2.2.23087.79523.

[106] **G. G. Chrysos, J. Kossaifi, and S. Zafeiriou,** "RoCGAN: Robust Conditional GAN," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2665–2683, Nov. 2020, doi: 10.1007/s11263-020-01348-5.

CURRICULUM VITAE

Name Surname **:Navid HALILI**

EDUCATION

- **B.Sc.** **:** 2010**,** Azad University of Tabriz, Electronics, Electrical Engineering
- **M.Sc.** **:** 2013, Azad University of Tabriz, Mechatronics, Mechatronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- Two-year Academic Scholarship 2008-2010 in Azad University of Tabriz for My B.Sc. Degree
- Expert Engineer in " Aryan Machines company"; Tabriz, Iran; 23 August 2013 - 14 March 2015.
- Lab assistant, Research Laboratory in Electronic 1, Azad University of Tabriz, Sep. 2011 - Jan. 2013.
- Iranian organization for engineering order of building province East Azerbaijan; Field: Electrical Engineering: From September 2013.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Khalili Dizaji, Navid, and Mustafa Doğan. 2024. "A Comprehensive Brain MRI Image Segmentation System Based on Contourlet Transform and Deep Neural Networks" Algorithms 17, no. 3: 130. https://doi.org/10.3390/a17030130