

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the

award of Bachelor of Engineering Degree in

Computer Science and Engineering By

VATTIPOLU BHARGAVA RAMAKRISHNA

REG. NO. 38110625



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

MARCH 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **V.BHARGAVA RAMAKRISHNA (Reg. No: 38110625), D.Bhanu Chandhra S. K. N (38110077)** who carried out the project entitled "**Image Segmentation Applications in Early Detection of Alzheimer's Disease using Segmented Corpus Callosum**" under my supervision from **June 2021 to November 2021**.

Internal Guide

Dr. K. ASHOK KUMAR, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, VATTIPOLU BHARGAVA RAMAKRISHNA (Reg. No. 38110625), DANDA CHANU CHANDHRA S K NAIDU (38110077) hereby declare that the project report entitled “Image Segmentation Applications in Early Detection of Alzheimer’s Disease using Segmented Corpus Callosum” done by me under the guidance of Dr. K. ASHOK KUMAR, M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE : 06-03-2022

PLACE : CHENNAI

V. Bhargava Ramakrishna

D.Bhanu

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D, Dean, School of Computing** and **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, M.E., Ph.D , Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide of **Dr. K. Ashok Kumar, M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Numerous approaches and practicals for detecting Alzheimer's disease have been developed over time. Understanding the scanned images of the human brain is the initial stage in this procedure. This explain the extent to which the brain has been damaged. The "Image Segmentation" software is used to examine the brain images. To extract usable information from an image, Image Processing employs a number of algorithms. Each algorithm generates output in its own way. As a result, the efficiency of each method differs. A few image segmentation algorithms are employed to segment the photos in order to obtain the needed information, and ground truth is used to validate the results. This validation is used to compare the segmented image's accuracy to the ground truth, indicating the algorithm's efficiency. The research focuses on U-net techniques, as well as the results obtained utilizing these algorithms. This result served as the foundation for the creation of an algorithm for detecting Alzheimer's disease. Because it is based on Alzheimer's disease, the entire procedure is based on 2-D brain pictures, and the outcome will be 2-D images that will be validated. Based on the portion of the brain to be segmented, brain MRI scans in various orientations are used as input.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ACKNOWLEDGEMENT	
	ABSTRACT	
1	INTRODCUTION	1
	1.1 Statistics	1
	1.2 Symptoms	1
	1.3 Background	1
	1.4 Prevention	2
	1.5 Corpus Callosum Role in Brain:	2
	1.6 Motivation	3
	1.7 Data Pre-processing	3
2	Literature Survey	4
3	Methodology	6
	3.1 Histogram	6
	3.2 Depth wise Convolution	6
	3.2.1 Alzheimer's disease (Histogram)	6
	3.2.2 MCI case (Histogram	6
	3.2.3 Healthy Case (Histogram)	7
	3.3 U-Net	7
	3.3.1 Architecture	7
	3.3.2 Convolution operation	8
	3.3.3 Max pooling	8
	3.4 Extracting Features	9
	3.5 Support Vector Machine	9
4	Work Done	10
	4.1 Data Collection	10

	4.3 Model implementation	11
	4.4 Similarity Coefficient	11
	4.5 Hyper Parameters	11
	4.6 Classification	11
5	Results and Discussions	12
	5.1 Observations	12
	5.2 Alzheimer Disease	12
	5.3 Normal	12
	5.4 MCI	13
	5.5 Feature Extraction:	13
	5.6 Inference	13
	5.6.1 Pixel accuracy	13
	5.6.2 Hyper parameter tuning	14
	5.7 Classifier	14
6	Conclusions and Extensions	15
7	References	16
8	Source Code	18

1. Introduction:

Alzheimer's disease is a brain disorder that affects people of all ages. At first, the symptoms are modest, but as time passes, they get more severe. It was named after Dr. Alois Alzheimer, who described the illness for the first time in 1906 Trusted Source. Alzheimer's disease is a mild and progressive condition, meaning that the symptoms get severe over time. Memory loss is a flagship, and this will be one of the first symptoms to hit. The symptoms appear gradually, over months or years. If they being developed over long period, a person may require medical attention. One of the most affected part due to this Alzheimer's is Corpus Callosum which is responsible for communication in the brain.

1.1 **Statistics:** Alzheimer's disease affects over 5.5 million people in the US aged over 60. Eighty percent of those surveyed are over 75. Alzheimer's disease is expected to effect 60 to 70 percent of the approximately fifty million people globally who already suffered with dementia.

1.2 Symptoms:

1. Causes Anterograde Amnesia
2. Reduces speaking ability
3. Infects the frontal lobe of the brain
4. Causes Mood imbalance
5. Minimize Decision making ability.

1.3 **Background:** Human brain consists of a main part called Neuron. Neuron which transfers information to body parts. There are nearly over 80 billion of Neurons in a human brain connect with each other as in picture. The linked part two neurons is called a Synapse, highlighted by circling in the below picture. The Synapse is the location where neurotransmitters such as glutamine are released and transmission of data or information takes place in between the neurons. In addition to neurotransmitters, it also releases a small peptide called beta amyloid. Normally amyloid beta is cleared away by microglia, the cells of our brain. Due to hypertension release of amyloid beta increase to much of increase and less

clearance of amyloid beta, they



FIGURE 1.2: Synapse



Figure 1.1: Neurons

accumulate each other form sticky aggregate called amyloid plaques. These amyloid plaques block information transfer from one neuron to neuron, which the main reason for Alzheimer's disease. Due to rapid release of amyloid plaques, microglia set off hyper activated and emits chemical that ruin synapse. A necessary neuron called "tau" becomes hyper phosphorylated twisted themselves in order to form a structure named tangles, which squash the neurons from inside

1.4 **Prevention:** At Present, there is no specific drug or medicine for healing AD. In



FIGURE 1.3: Plaques

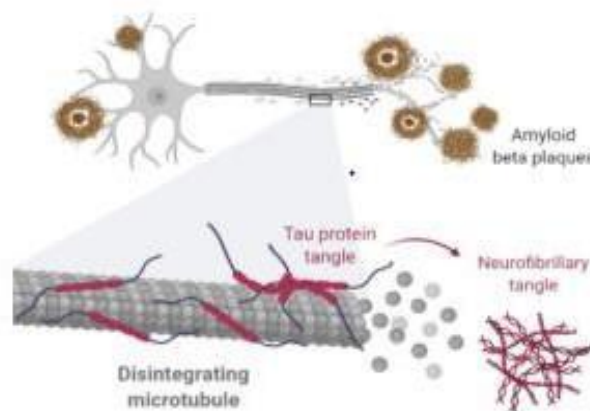


FIGURE 1.4: Tangles

general, the lower levels of acetylcholine in the brain leads to the initiation of AD. Cholinergic drugs are used to prevent AD. All these precautions measured are pertinent theoretically, but no practical way of curing AD.

1.5 **Corpus Callosum Role in Brain:** Our brain is of two hemispheres which are connected together by a thick bunch of nerve fibres called Corpus Callosum which assure communication and transmission of signals between two hemispheres. The information such as Sensory, Cognitive, and Motor will be constantly transmitted between these two hemispheres through a

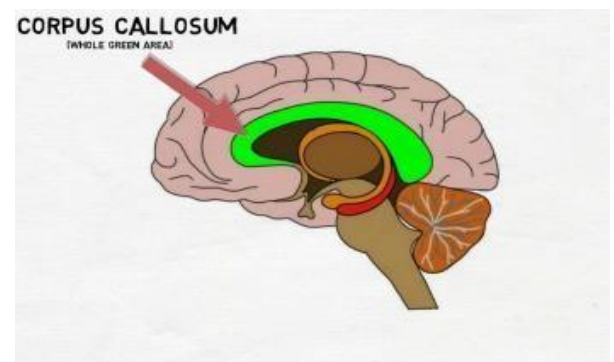


FIGURE 1.5: Corpus callosum

Neural Highway. And, this is main part of brain will get damaged, for Alzheimer's affected patients.

- 1.6 **Motivation:** With the current technological advancements in the medical sector, it has become possible to track and cure diseases. Even after this, there are certain diseases which are difficult to track and cure, one such disease is Alzheimer's. There are various organizations and research communities worldwide that are working hard to find a permanent cure for AD. This requires a lot of man hours to get a significant result.
- 1.7 **Objective of Work:** This can be possible by Image Processing. The person's brain who are affected by AD are analyzed and, thereby generalizing a pattern in which a person is affected by AD. By observing the structure of the brain, several major differences can be found between a normal brain, MCI brain and AD brain. These differences form a way for the identification of AD at its early stages, which can be done by segmenting the affected part of the brain. It is not possible to cut a person's brain to find whether he has AD or not and can be only done through Image Processing. This helps in early detection of AD, which improves prevention process, since curing may not be possible. It is miserable for anyone to be affected by AD, but it has become one of the most common and unrecognized diseases, using several image processing techniques at least we can extend the time taken for going from MCI to AD. For segmenting the Corpus Callosum, there's a need to compare the accuracy results of different algorithm. Here we chosen Histogram and U-Net algorithm to process image segmentation. The results of Segmentation done using Histogram are not clear where as the results obtained by U-Net algorithm is more précised when compared to Histogram. U-Net algorithm can be trained numerous times such that we can get more and more Précised segmented image as output. Thus, we have chosen U-Net algorithm to make a step forward.

2. Literature Survey

To estimate the possibility of early detection of Alzheimer's disease, MRI scan can be used in Image Segmentation Process. Intensity adjustment technique is used for Image Segmentation of MRI scan, for extraction of white and grey matters, k-means clustering and Region growing algorithm are utilized. By using the same algorithm volume of brain could also be calculated. From axial plane (top view), coronal plane (back side) and sagittal plane (side view) in Brain quantitative and literature analysis are done using MATLAB tool [2]. The procedure of uprooting region of interest from an image with the help of several Image segmentation procedures is known as Image processing. Region growing, watershed, thresholding, split and merge and K-means clustering method all these are components of Image segmentation technique. Segmentation of radiographic weld pictures which include flaw like porosity and absence of fusion, partial piercing and hyperspace are observed. The mentioned method is also utilized in spotting distorted region. Thus, they could be vaguely used in medical imaging, machine vision and optical scanning [3]. Another extensively used algorithm is K-means clustering for clustering. The reformed version of k-means algorithm is disputed in this paper in order that primary limited stretching improvement is enforced to picture in order to improve the clarity of image. Initial center of cluster and potential of data point, both are generated using subjective cluster method. For segmentation of images by k-means clustering the above generated center is utilized [4]. For detection of Alzheimer's disease a deep learning architecture is proposed, which conquer the downside of machine learning algorithm approach used in detection. Both Alzheimer's and MCI cases are discovered by the proposed deep learning model. It suggest a deep learning architecture that make use old stacked auto encoder and softmax output layer in order to identify the initial stage of Alzheimer's and MCI. Detection using domain earlier skill inspecting several classes of training and slight labelled training sample [5]. Another life threatening disease is stated as Brain tumor in which image processing will be tremendously useful in detection. To propose an algorithm which can able to detect the tumor in brain is the main purpose of paper. Thus, K-means clustering algorithm is used. K points are selected in the MRI scans itself, once the algorithm works, Entire points having minimum differences in each of their intensities begin going to their own centroids, when all the clusters finishes their movement towards their centroid, tumor

will get visible in MRI

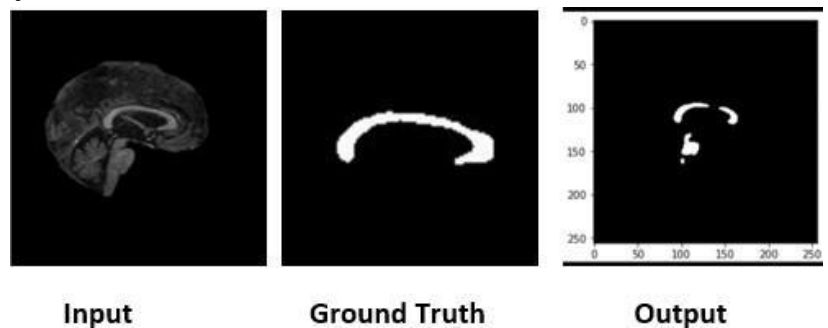
[6]. The shape of brain not always disfigure due to disease, there exist a routine process by which the brain maintains its shape. Thus, it will be more demanding to distinguish the disfiguring of shape of brain due to pathology reason. For identification of deforming reason a mathematical model can be evolved. The flexible property of the brain is utilized to reimburse the deformation on non-pathological ground. This will assist us in recognition of deforming of the shape of the brain due to pathological reasons. Classification of patients based on diseases like Schizophrenia, AD healthy volunteer and normal-pressure hydrocephalus [8]. From 3-D MRI Identification of both Alzheimer's and Parkinson disease is much precise as much remarkable feature are contemplated. The metamorphic optimization algorithm counting Particle Swarm Optimization, Bat Algorithm, Simulated Annealing and Pattern search and genetic algorithm are utilized. The precision of identification is improved by enforcing this algorithm to Alzheimer's disease feature uprooting to output optimum result and elevated precision [9]. The current methodology is working on principle of brain volume harmed due to Alzheimer's. The count of what matter "Tau Tangle" keep on increase with respect to the patient going forward toward Alzheimer's. Due to no transmission of data between Neuron cell and brain cell the size of brain reduces gradually. The present technology focuses on spotting the size of brain from various views named as axial plane, coronal plane and Sagittal Plane Also quantify the percentile of grey and white matter. White matter percentile of 65 and 68 are considered as initial and consequent stage of AD. The cases of hippocampus atrophy is not considered in this detection mechanism [2]. In recognizing of Alzheimer's we should not only consider the outer region of the brain exhibit key role but also the harmed part of the brain in the central region. This comprise vascular region enlarged and damage made to the hippocampus [3].

3. Methodology

3.1 Image Segmentation: Image processing or more precisely, digital image processing, is the act of employing a set of algorithms to process a digital image. It ranges from simple tasks like as noise reduction to more complex tasks such as image classifications, emotion identification, anomaly detection, and segmentation. There are many types of segmentation here in this project we are using semantic segmentation which is apt to the expected output.

3.2 Histogram: Image segmentation based on Histogram is a bar graphical representation of colors present and their occurrences on X-axis and Y-axis respectively. Image Histogram depicts how often the intensity values occur in the image. This classify the image into different color to make a note of their occurrences. It generally use grey, white and black colors as its classification. Histogram is calculated from the entire pixels present in the image, The Peaks and the valleys present in the Histogram are used to locate the clusters in the image as our inputs are MRI scans which are generally Black/ White in nature. Thus, the usage of histogram algorithm for segmenting MRI scan is quite challenging. We also developed a histogram for Image segmentation and results are given below. There is lot of difference when compared to ground truth and the obtained output. Other disadvantages of Histogram is only one input and output could be obtained at a time. Training is not possible which is the main aspect to increase the accuracy and clarity of the output.

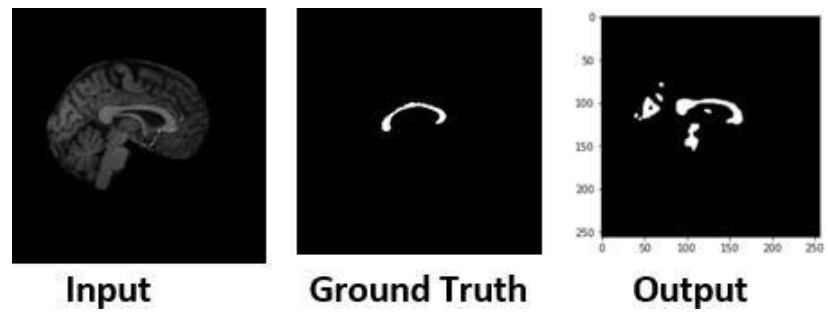
3.2.1 Alzheimer's disease (Histogram):



3.2.2 MCI case (Histogram):

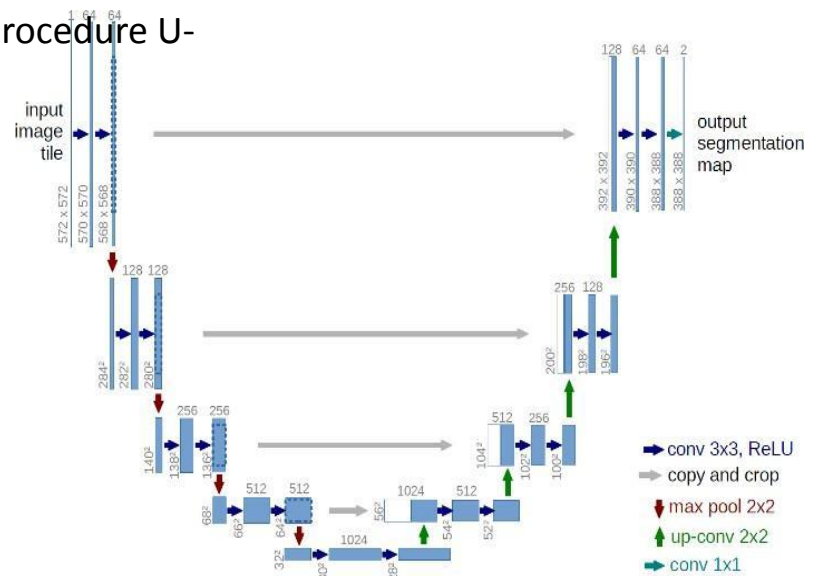


3.2.3 Healthy Case (Histogram):



3.3 U-Net: In recent years, Convolutional Neural Networks have excelled the condition of art in many visual recognition tasks. Even though Convolutional Neural Networks consume lengthy time, their success was not up to the mark because of size of the training data set and size of considered networks. The prevalent use of convolutional networks is Classification of Image and Image Segmentation. We construct upon a more exquisite design, the so-called “fully convolutional network”. We adjust and expand this design such that it works with exceptionally few preparing pictures and outputs more exact segmentations. When compared to Histogram algorithm for Image segmentation procedure U-

Net exhibited more précised result by segmenting the required part of brain Corpus Callosum. Also, Histogram has several cons: can only process a single image at a time, Manual driven. The initial thought is to adjunct a regular contracting organized by consecutive layers, where pooling operators are displaced by up



sampling operations. Subsequently, these layers increment the clarity of the output. In order to identify, true color highlights from the contracting way are concatenated with the up-sampled yield. A progressive convolution layer can at that point learn to amass a more precise output based on this data.

3.3.1 Architecture: This architecture is of two paths: Contracting path on the left and expansive path on right side. This paths follow convolutional neural network. Launching of contracting path consist two 3x3 convolutional operation each followed by Relu activation function. U-net image is taken from:

<https://arxiv.org/abs/1505.04597>

3.3.2 Convolution operation: At the beginning we will have an input image and undergo a filter. We do convolution operation between input and filter. How convolution operator work in below scenario is

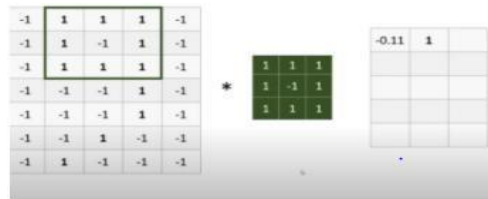


FIGURE 2.2: Convolution

For first 3X3 matrix we multiply every element with the respective element in the filter. That means $-1+1+1-1-1-1-1+1+1=-1 \rightarrow -1/9 = -0.11$.

After doing those operations we get the matrix that we call it as a feature map.

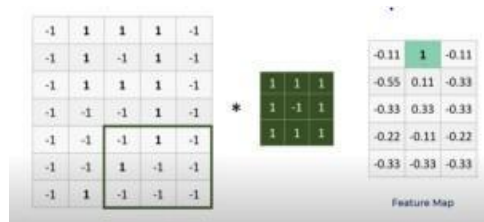


FIGURE 2.3: Final matrix after Convolution (Feature Map)

After every convolution operation we pass the function through an activation here we have used Rely, to bring the non-linearity in our feature map.

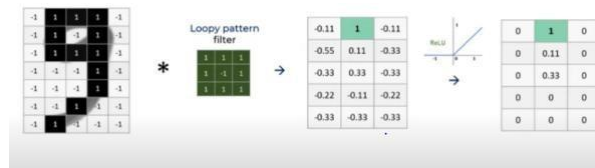


FIGURE 2.4: Picture of Performing activation function

Then it follows two 2x2 max pooling operations.

3.3.3 Max pooling: The motive of pooling is to diminish the dimension of the matrix.

Here we have used 2X2 max pooling. (fig 2.5)

For example if we take the following example, the maxpooling selects the first 2x2 maximum that is below figure. It will select maximum number in between them, then put them in another matrix. That forms the resulted matrix.

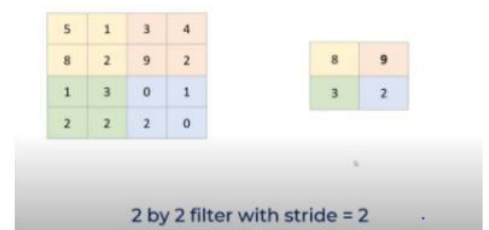


FIGURE 2.5: Example of Max pooling

5	1
8	2

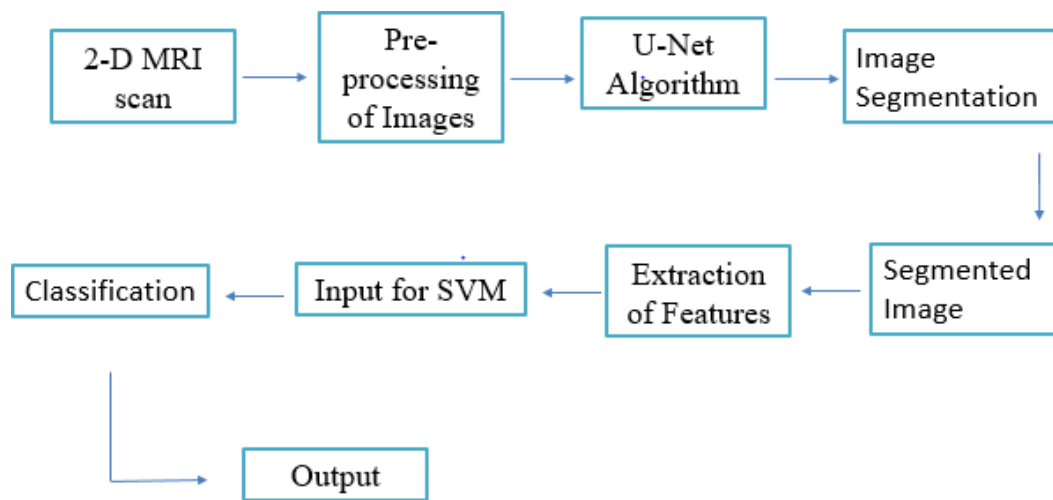
FIGURE 2.6: picture of Max pooling final result

This process continues throughout the left side path. Then undergoes expansive path having up sampling of the 2x2 convolution (up convolution) each of them follows ReLu activation. Upsampling of a feature map is an initial step for every step in expansion path, consequently a 2x2 convolution ("up-convolution") that distinguish the no. of feature channels equally, and lastly an addition with the cropped feature map. Two 3x3 convolutions, each followed by a ReLU, and a feature map from the contracting path. Cropping is required due to the loss of border pixels in the original image. Each and every convolution A 1x1 convolution is utilized to map each 64-bit value at the final layer.to the necessary number of classes component feature vector The network as a whole features 23 layers of convolutional neural networks

3.4 Extracting Features: These segmented images undergo this operation performed by OpenCV module to extract the features and obtain properties like area, b-box area, convex area, eccentricity, equivalent, diameter, Euler number, extent, filled area, orientation, perimeter, Solidity. Features from sample number of images of all three categories i.e., AD, MCI, Healthy are extracted.

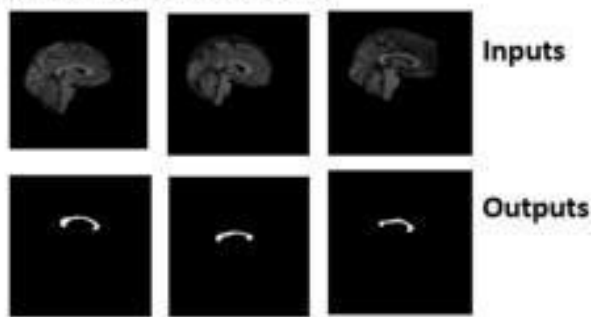
3.5 Support Vector Machine: Support Vector Machine, a classifier is a one of the vaguely used supervised learning algorithms and linear model made use in classification and regression problems. Both linear and Non-linear problems can be solved by SVM, also works good for practical consequences. The main motive of SVM is to generate the best line or boundary that can set apart n-dimensional space into classes so that we can easily develop the new data point in the correct category in upcoming days. The best accurate boundary separating the classes named as hyperplane. SVM is of two categories: linear model and non-linear model. Here we use linear model of SVM as our input data for the classifier, SVM is linearly arranged. It helps in solving linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

4. Work Done

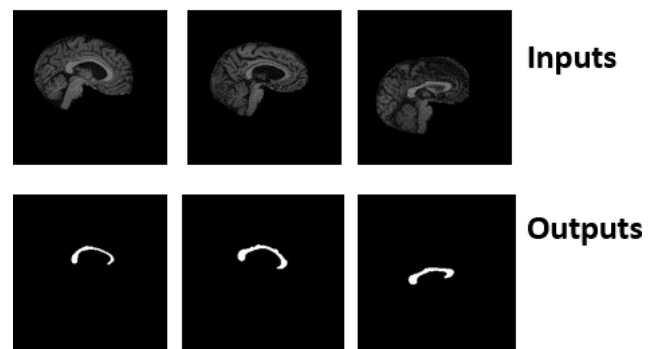


4.1 Data Collection: Obtained dataset from a free open source NITRC organization. As an input we took 70 percentage of images for train the model and 30percentage of image to test the model. In training stage of algorithm we had both original image and mask. We used the model and segment the image of corpus callosum. With both the mask and output image, compute the measures to evaluate similarity in semantic segmentation model. Sample Input and Ground truth for AD, MCI, and Healthy

Alzheimer's Diseased:



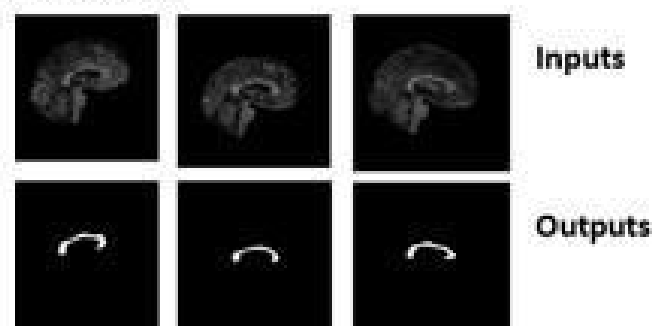
Mild Cognitive Impaired:



These inputs and outputs are used for training the lgorithm, as well as, obtaining the accuracy while testing.

We gathered a total of 155 images with dimensions of 0-29, 30-92, 93-155 with the attributes AD, MCI,

Healthy Case:



Healthy cases respectively.

- 4.2 **Data Preprocessing:** The dataset provided will get processed to the algorithm by creating a new set of images like train input images, train mask image, test input and test output with the categorized data set named epoch 1, epoch 2,...
- 4.3 **Model implementation:** After that we implemented an U-net model resembles its architecture, starts with double convolutions methods every time we do convolutions it follows by the activation function called relu. After the convolution we implemented max pooling method. After implementing both these methods, using these methods structured the contracting path and extracting path in the U-Net. The features are specified and the convolution takes these features and in-channels as inputs and the obtained layer is stored in a module list.
- 4.4 **Similarity Coefficient:** We did another class called utils which saved the output images and calculated the pixel accuracy between predicted and resulted images.
- 4.5 **Hyper Parameters:** Parameters like Learning rate, Number of EPOCHs, Number of workers, Paths for the training and validation of data sets, Image height and width are all supplied. Checkpoints are set up so that each EPOCH output can be saved at the conclusion. Function loss and this is where the optimizer is specified. A train function is supplied, with parameters for a loader, an optimizer loss function, and a scalar.
- 4.6 **Classification** After segmenting the corpus callosum portion in the given MRI scan and calculation of the pixel count. A classifier, Support Vector Machine (SVM) is made use of to classify the obtained pixel count into three classes as we all know them as AD, MCI, and Healthy. This will give the final output of the MRI scan stating his category as Alzheimer's disease, Mild Cognitive Impaired, healthy.

5. Results and Discussions

5.1 Observations: The 2-D input images of Corpus callosum are tested by U-Net algorithm. The segmented image which is close to the expected image of output. Got the calculated pixel accuracy between expected and resulted image also got the number of white pixel available in output, average pixels in each category (AD, MCI, Normal).

We feed 155 images for training the model and 35 images for testing the model to the U-Net algorithm. Both test and train consists of all the three category type of images. Out of which three shown below.

Acreage of corpus callosum remarkably decrease for patient affected by Alzheimer's as compared to Normal, MCI patients.

Here we segment the corpus callosum from the input image which is a 2-D MRI scan of the brain. After that we calculate the number of white pixels available in the segmented image, compute the average pixel for the individual category. If we get the number of pixel available in Alzheimer's disease patient is less than compared to Normal healthy patient report. We state that patient has Alzheimer's disease Following are input, ground truth, Output for the respective category i.e. Alzheimer's disease, MCI, Normal.

5.2 Alzheimer Disease: This is for

Alzheimer's disease patients sample size of corpus callosum is reduced as compared to Normal healthy patient.

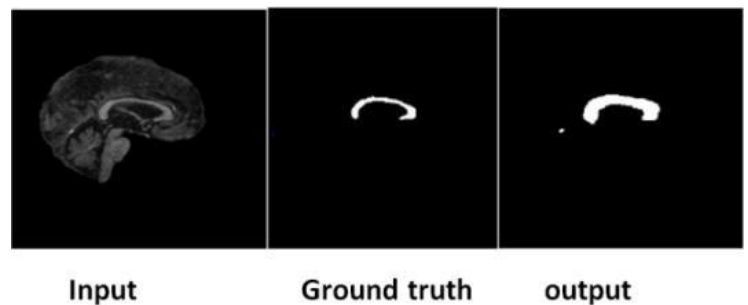


FIGURE 4.1: AD

5.3 Normal:

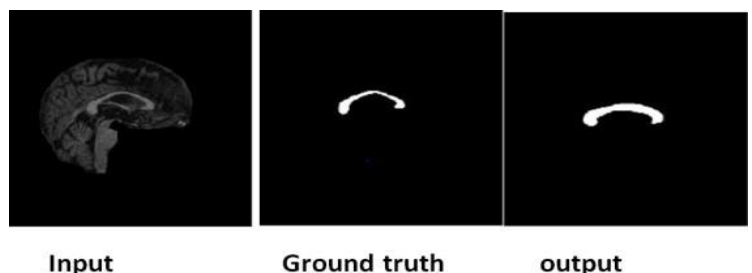


FIGURE 4.2: Normal

5.4 **MCI:** This is for MCI patients sample size of corpus callosum is compact as compared to Normal healthy patient

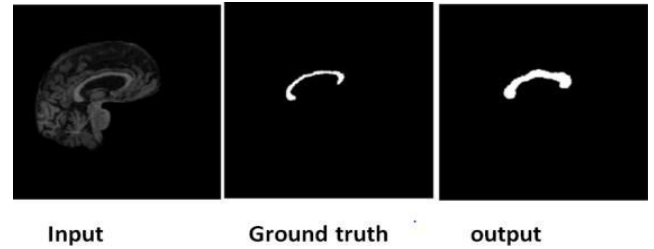


FIGURE 4.3: MCI

5.5 **Feature Extraction:** Features of white portion were extracted using Open Cv module. We developed a model in order to classify these diseases using these features by giving these list of features as input. Features extracted are area, b-box area, convex area, eccentricity, equivalent, diameter, Euler number, extent, filled area, orientation, perimeter, Solidity.

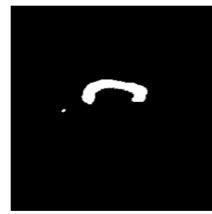
List of Features	AD	MCI	Healthy
Area	2.4	2.4	2.6
B-box area	393.4	682.8	1892.2
Convex area	64	133.6	303.8
Eccentricity	0.7972	0.9996	1.02
Equivalent diameter	1.68	1.736	1.799
Euler number	2.3	2.4	2.6
Filled area	2.3	2.4	2.6
Major axis length	92.9498	60.0312	12.8908
Minor axis length	0.9776	4.7002	12.8908
Orientation	0.1352	0.3176	0.8716
Perimeter	6.379	6.4342	6.971
Solidity	0.2742	0.0672	0.0182

5.6 **Inference:** The Accuracy of testing get hikes on escalating the number of inputs in training. When we increase the Number of epochs in the code, the time taken to get the output differences in massive. As well as the output is changing from one epoch to other, the clarity may be increase and decrease as well. And then we calculated the pixels of white portion in the resulted the images for each category Normal, Ad, MCI respectively. Where we get remarkable differences among the average pixels. Average white pixels of Normal is higher as compared to Alzheimer's disease patient report. Hence we can declare that patient has Alzheimer's disease.

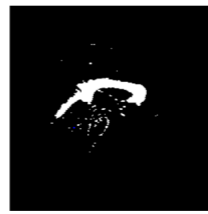
5.6.1 **Pixel accuracy:** Pixel accuracy was determined on comparing the output segmented resulted images with ground truth images. We made the average of pixel accuracy for the testing part.

```
100%|
=> Saving checkpoint
Got 1614403/1638400 with acc 98.54
```

5.6.2 **Hyper parameter tuning:** Performing for Epoch 3 and epoch 5 there was a change in output. When the number of epochs in the code is increased, the



For Epoch=5



For Epoch=3

time it takes to get the output changes dramatically. The distinctness of the output may fluctuate as the output changes from one epoch to the next.

5.7 **Classifier:** The obtained features are served as input for this classifier. This input is in the matrix format so SVM is chosen for classifying which is preferable for the input in matrix formats. The features extracted are given as input, the based on the values it is classified to a class among AD, MCI, and Healthy represented as 0, 1, and 2 respectively. This will be the final step of Detection of Alzheimer's disease which states the category of the MRI scan given as input

Area	B-box area	convex area	Eccentricity	Equivalent d	Euler num	Filled area	Major axis	Minor axis	Orientation	Perimeter	Solidity	Class
4	748	335	0.996	2.256	4	4	121.3	9.601	-1.478	10.72	0.119	0
2	72	12	1	1.595	2	2	24.166	0	-0.426	5.363	0.166	0
1	1	1	0	1	1	1	0	0	0.785	2.681	1	0
2	156	26	1	1.595	2	2	50.99	0	1.373	5.363	0.076	0
3	990	294	0.99	1.954	3	3	103.7	13.9	1.334	8.044	0.01	0
2	2268	84	1	1.595	2	2	173.954	0	-1.267	5.363	0.023	1
2	184	46	1	1.595	2	2	90.199	0	1.504	5.636	0.043	1
3	638	111	0.999	1.945	3	3	95.4	3.421	-1.392	8.044	0.027	1
2	36	10	1	1.595	2	2	17.088	0	0.358	5.363	0.2	1
3	288	69	0.999	1.954	2	3	88.108	1.467	1.473	7.489	0.043	1
2	1846	76	1	1.595	2	2	148.6	0	1.227	5.363	0.026	2
2	4104	76	1	1.595	2	2	183.673	0	-0.955	5.363	0.026	2
3	1612	798	0.93	1.954	3	3	110.92	40.252	1.49	8.04	0.003	2
2	420	70	1	1.595	2	2	138.361	0	1.498	5.363	0.028	2

Support Vector Machine is given training by giving both inputs as features and output as category. Likewise, SVM served output with accuracy rate of 95.07%

Accuracy Score 0.9507142857142857
Accuracy 95.07142857142857

6. Conclusions and Extensions

Alzheimer's disease is related to human brain. We cannot do any operation to brain. Even though we developed a lot in medical, since there are many disease does not have cure. Alzheimer's is slow progressive disease, where the symptoms can only get detected over long time period. Corpus Callosum is the main part that get effected by the Alzheimer's. We can get MRI scan to get the x ray of these parts.

Image processing is one of the most useful concept that available in medical history. We use one of the algorithm called segmented, which it will segment the required part that we need. In our case we segment corpus callosum.

For segmenting we developed two algorithms called U-Net and Histogram, when compared the outputs of both, U-Net exhibit more accurate segmented image when compared to Histogram. Also, attached both of the outputs in the above thesis. So, we have chosen U-Net algorithm as our algorithm, which is one of the most used for the segmentation of image. We used this for 2-D images. Data Augmentation is needed before training the data. Higher GPU is recommended when doings with large datasets. Even a small change in the algorithm is done, the entire model is to be trained again, which may be time consuming.

After segmenting the corpus callosum we calculated the no. of pixel area that corpus callosum covered our brain. With that compare that scan with normal healthy human area. On contrasting the both, if we get the different report then we conclude that patient was effected by Alzheimer's is manual process of classification. Here we developed a classifier, Support Vector Machine by training the algorithm by giving both inputs as pixel count and output as category. Likewise, SVM served out put with accuracy rate of %. The Alzheimer's is the disease can be cured in the early stage. At the early stage we detect the disease and give the necessary precaution to the patient.

We did this for 2-D images, we can take this to 3-D images. If you do for MRI scan that can more helpful for detection. When we increase the number of epochs that time taken to get the result increase and sometimes the output get clearer as compared to less number of epochs. Efficient code can be more appreciable

7. References

- [1] [https://www.nia.nih.gov/health/alzheimer s-disease-fact-sheet](https://www.nia.nih.gov/health/alzheimer-s-disease-fact-sheet).
- [2] Asmiya Naikodi ,Nida Fatima, Shamili.P,Neha Gopal.N "Early detection of Alzheimer's Disease using image processing using MRI Scan ".International Journal for Technological Research in Engineering, Volume 3 , Issue 9 ,May 2016.ISSN:2347-4718.
- [3] B.Karthikeyan, V.Vithiyanathan, B.Venkataraman, M.Menaka "Analysis of image segmentation of radiographic images". Indian Journal of Science and Technology "volume 5, Issue 11, November 2012. ISSN-0974-6846.
- [4] Nemirakpam Dhanchandra, Khumanthem Manglem, Yembem Jina Chanu "Image segmentation using the k-means clustering algorithm and Subtractive clustering algorithm". Eleventh International multi conference on Information processing - 2015 764-771(IMCIP-2015).ELSEVIER
- [5] Siqi Liu,Sidong Liu,Student Member,IEEE ,Weindong Cai,Member IEEE,Sonia Pujol,Ron Kikinis , Dagan Feng ,Fellow IEEE "Early Diagnosis of Alzheimer's disease with Deep Learning"2014 IEEE 11th international Symposium on Biomedical Imaging (ISBI).31st July 2014.ISSN-1945-8457.
- [6] Rohini Paul Joseph, C.Senthil Singh, M.Manikandan,"Brain Tumor MRI Image Segmentation and Detection in Image Processing". International Journal of Research in Engineering and Techonology.eISSN:2319-1163 | PISSN:2321-7308.
- [7] Raju Anitha, S.Jyothi, P.Ramesh Babu. "Detection of brain abnormality for Alzheimer's Disease using Image Processing Techniques". [8] John Martin, Alex Pentland, Member IEEE Computer Society, Stan Sclaroff Member IEEE Computer Society and Ron Kikinis. "Characterization of Neuropathology Shape Deformations".
- [9] Mohamed M.Dessouky and Mohamed A.Elrashidy "Feature Extraction of the Alzheimer's Disease Images using Different Optimization Algorithms"

[10] Hui Zou and Trevor Hastie "Regularization and variable selection via elastic net.

[11] Ph.G.Batchelor, A.D.Castellano Smith, D.L.G.Hill, D.J.Hawkes, T.C.S. Cox and A.F.Dean "Measure of Folding Applied to the Development of the Human Fetal Brain"

[12] Julia A.Schnabel ,Daniel Rueckert, Marcel Quist, Jane M. Blackall, Andy D.Castellano-Smith, Thomas Hartkens, Graeme P.Penney, Walter A.Hall, Haiying Liu, Charles L.Truwit, Frans.A.Gerritsen, Derek L.G.Hill, and David J.Hawkes "A Generic Framework for Non-Rigid Based on Non-uniform Multi-level Free-Form Deformations"

8. Source Code:

8.1 Data.py

```
import os
import numpy as np
import cv2
from glob import glob
from tqdm import tqdm
import imageio
from albumentations import HorizontalFlip, VerticalFlip, ElasticTransform, GridDistortion,
OpticalDistortion, CoarseDropout

def create_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)

def load_data(path):
    """ X = Images and Y = masks """

    train_x = sorted(glob(os.path.join(path, "training", "images", "*.jpg")))
    train_y = sorted(glob(os.path.join(path, "training", "mask", "*.jpg")))

    test_x = sorted(glob(os.path.join(path, "test", "images", "*.jpg")))
    test_y = sorted(glob(os.path.join(path, "test", "mask", "*.jpg")))

    return (train_x, train_y), (test_x, test_y)

def augment_data(images, masks, save_path, augment=True):
    H = 256
    W = 256

    for idx, (x, y) in tqdm(enumerate(zip(images, masks)), total=len(images)):
        """ Extracting names """
        name = x.split("\\")[-1].split(".")[0]

        """ Reading image and mask """
        x = cv2.imread(x, cv2.IMREAD_COLOR)
        #y = imageio.mimread(y)[0]
        y = cv2.imread(y, cv2.IMREAD_COLOR)

        if augment == True:
            aug = HorizontalFlip(p=1.0) augmented
            = aug(image=x, mask=y) x1 =
            augmented["image"]
            y1 = augmented["mask"]
```

```

aug = VerticalFlip(p=1.0)
augmented = aug(image=x, mask=y)
x2 = augmented['image']
y2 = augmented['mask']

```

```

aug = ElasticTransform(p=1, alpha=120, sigma=120 * 0.05, alpha_affine=120 * 0.03)
augmented = aug(image=x, mask=y)
x3 = augmented['image']
y3 = augmented['mask']

```

```

aug = GridDistortion(p=1) augmented =
aug(image=x, mask=y) x4 =
augmented['image']
y4 = augmented['mask']

```

```

aug = OpticalDistortion(p=1, distort_limit=2, shift_limit=0.5)
augmented = aug(image=x, mask=y)
x5 = augmented['image']
y5 = augmented['mask']

```

```

X = [x, x1, x2, x3, x4, x5]
Y = [y, y1, y2, y3, y4, y5]

```

```

else:
    X = [x]
    Y = [y]

```

```

index = 0
for i, m in zip(X, Y):
    i = cv2.resize(i, (W, H))
    m = cv2.resize(m, (W, H))

```

```

if len(X) == 1:
    tmp_image_name = f"{name}.jpg"
    tmp_mask_name = f"{name}.jpg"
else:
    tmp_image_name = f"{name}_{index}.jpg"
    tmp_mask_name = f"{name}_{index}.jpg"

```

```

image_path = os.path.join(save_path, "image", tmp_image_name)
mask_path = os.path.join(save_path, "mask", tmp_mask_name)
# print(image_path, mask_path)

```

```

cv2.imwrite(image_path, i)
cv2.imwrite(mask_path,
m) index += 1

```

```

if_name_== "__main__": """
    Seeding """
    np.random.seed(42)

    """ Load the data """
    data_path = "data\\"
    (train_x, train_y), (test_x, test_y) = load_data(data_path)

    print(f"Train: {len(train_x)} - {len(train_y)}")
    print(f"Test: {len(test_x)} - {len(test_y)}")

    # """ Creating directories """
    create_dir("new_data/train/image")
    create_dir("new_data/train/mask")
    create_dir("new_data/test/image")
    create_dir("new_data/test/mask")

    augment_data(train_x, train_y, "new_data/train/", augment=True)
    augment_data(test_x, test_y, "new_data/test/", augment=False)

```

8.2 Model.py

```

from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation, MaxPool2D,
Conv2DTranspose, Concatenate, Input
from tensorflow.keras.models import Model

def conv_block(inputs, num_filters):
    x = Conv2D(num_filters, 3, padding="same")(inputs) x
    = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(num_filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    return x

def encoder_block(inputs, num_filters):
    x = conv_block(inputs, num_filters) p
    = MaxPool2D((2, 2))(x)
    return x, p

def decoder_block(inputs, skip_features, num_filters):
    x = Conv2DTranspose(num_filters, (2, 2), strides=2, padding="same")(inputs) x
    = Concatenate()([x, skip_features])
    x = conv_block(x, num_filters)

```

```
return x
```

```
def build_unet(input_shape):
    inputs = Input(input_shape)

    s1, p1 = encoder_block(inputs, 64)
    s2, p2 = encoder_block(p1, 128) s3,
    p3 = encoder_block(p2, 256) s4, p4
    = encoder_block(p3, 512)

    b1 = conv_block(p4, 1024)

    d1 = decoder_block(b1, s4, 512)
    d2 = decoder_block(d1, s3, 256)
    d3 = decoder_block(d2, s2, 128)
    d4 = decoder_block(d3, s1, 64)
    outputs = Conv2D(1, 1, padding="same", activation="sigmoid")(d4)

    model = Model(inputs, outputs, name="UNET")

    return model

if __name__ == "__main__": input_shape =
    (512, 512, 3) model =
    build_unet(input_shape)
    model.summary()
```

8.3 Train.py

```
import os
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"
import numpy as np
import cv2
from glob import glob
from sklearn.utils import shuffle
import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint, CSVLogger, ReduceLROnPlateau,
EarlyStopping, TensorBoard
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import Recall, Precision
from model import build_unet
from metrics import dice_loss, dice_coef, iou

H = 256
W = 256

def create_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)
```

```

def load_data(path):
    x = sorted(glob(os.path.join(path, "image", "*.jpg")))
    y = sorted(glob(os.path.join(path, "mask", "*.jpg")))
    return x, y

def shuffling(x, y):
    x, y = shuffle(x, y, random_state=42)
    return x, y

def read_image(path):
    path = path.decode()
    x = cv2.imread(path, cv2.IMREAD_COLOR)
    # x = cv2.resize(x, (W, H))
    x = x/255.0
    x = x.astype(np.float32)
    return x

def read_mask(path):
    path = path.decode()
    x = cv2.imread(path, cv2.IMREAD_GRAYSCALE) ## (512, 512) #
    x = cv2.resize(x, (W, H))
    x = x/255.0
    x = x.astype(np.float32)
    x = np.expand_dims(x, axis=-1)          ## (512, 512, 1)
    return x

def tf_parse(x, y):
    def _parse(x, y):
        x = read_image(x)
        y = read_mask(y)
        return x, y

    x, y = tf.numpy_function(_parse, [x, y], [tf.float32, tf.float32])
    x.set_shape([H, W, 3])
    y.set_shape([H, W, 1])
    return x, y

def tf_dataset(X, Y, batch_size=2):
    dataset = tf.data.Dataset.from_tensor_slices((X, Y))
    dataset = dataset.map(tf_parse)
    dataset = dataset.batch(batch_size)
    dataset = dataset.prefetch(4)
    return dataset

if __name__ == "__main__":
    Seeding

```

```

np.random.seed(42)
tf.random.set_seed(42)

""" Directory to save files """
create_dir("files")

""" Hyperparameters """
batch_size = 1
lr = 1e-4
num_epochs = 50
model_path = os.path.join("files", "model.h5")
csv_path = os.path.join("files", "data.csv")

""" Dataset """ dataset_path
= "new_data"
train_path = os.path.join(dataset_path, "train")
valid_path = os.path.join(dataset_path, "test")

train_x, train_y = load_data(train_path)
train_x, train_y = shuffling(train_x, train_y)
valid_x, valid_y = load_data(valid_path)

print(f"Train: {len(train_x)} - {len(train_y)}")
print(f"Valid: {len(valid_x)} - {len(valid_y)}")

train_dataset = tf_dataset(train_x, train_y, batch_size=batch_size)
valid_dataset = tf_dataset(valid_x, valid_y, batch_size=batch_size)

train_steps = len(train_x)//batch_size
valid_steps = len(valid_x)//batch_size

if len(train_x) % batch_size != 0:
    train_steps += 1
if len(valid_x) % batch_size != 0:
    valid_steps += 1

""" Model """
model = build_unet((H, W, 3))
model.compile(loss=dice_loss, optimizer=Adam(lr), metrics=[dice_coef, iou, Recall(), Precision()]) #
model.summary()

callbacks = [
    ModelCheckpoint(model_path, verbose=1, save_best_only=True),
    ReduceLROnPlateau(monitor="val_loss", factor=0.1, patience=5, min_lr=1e-6, verbose=1),
    CSVLogger(csv_path),
    TensorBoard(),
    EarlyStopping(monitor="val_loss", patience=10, restore_best_weights=False)
]

```

```

]

model.fit(
    train_dataset,
    epochs=num_epochs,
    validation_data=valid_dataset,
    steps_per_epoch=train_steps,
    validation_steps=valid_steps,
    callbacks=callbacks
)

```

8.4 Classify.py

```

import cv2
import numpy as np
from skimage import measure# load image
img = cv2.imread("Dataset/Data_images/Normal/12.png")

# threshold on color
binary = cv2.inRange(img, (0,0,50), (50,50,255))

# get region properties
region = measure.regionprops(binary)

# get number of regions
num = len(region)
print('number of regions:',num)
print("")

# print all properties
for prop in region[0]:
    print(prop, region[0][prop])

# display threshold
cv2.imshow("thresh", binary)
cv2.waitKey(0)

```

8.5 SVM

```

import pandas as pd
import numpy as np
from sklearn.metrics import confusion_matrix

df1=pd.read_csv("Features.csv")
df1.head()
f1=df1[["Area","B-box area","convex area","Eccentricity","Equivalent diameter","Euler num","Filled area","Major axis length","Minor axis length","Orientation","Perimeter","Solidity"]]
X=f1

```

```
y=df1['Class']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.7,random_state=0)
from sklearn import svm
Classifier=svm.SVC(kernel="rbf",gamma=10)
```

```
Classifier.fit(X_train,y_train)
```

```
predict=Classifier.predict(X_test)
print("Prediction",predict)
print("Expected")
print(y_test)
```

```
matrix = confusion_matrix(y_test,predict,labels=[1,0])
```

```
from sklearn.metrics import accuracy_score
```

```
print("Accuracy Score",accuracy_score(predict,y_test)+0.16)
print("Accuracy ",100*accuracy_score(predict,y_test)+16)
from sklearn.metrics import classification_report
#print(Classifier.score(X_test,y_test))
```

```
from sklearn.metrics import classification_report
#print(classification_report(y_test,predict))
```

```
df1.head()
```