A PROJECT REPORT

ON

Tumor Detection using Liver Segmentation

Submitted in partial fulfillment of the requirements

of the degree of

BACHELOR OF ENGINEERING

(Computer Engineering)

by

1. **Abhinavkumar Mishra (BE/A/122)**
2. **Rajas Mahajan (BE/A/153)**
3. **Shivam Tiwari (BE/A/158)**
4. **Sohan Kulkarni (BE/A/164)**

Guide

Prof. Shital Dhamal

Department of Computer Engineering

Lokmanya Tilak College of Engineering

Sector-4, Koparkhairne, Navi Mumbai

(2022-2023)

Certificate

This is to certify that the project entitled "**Tumor Detection in Liver using Monai and PyTorch**" is a bonafide work of

1. **Abhinavkumar Mishra (BE/A/122)**
2. **Rajas Mahajan (BE/A/153)**
3. **Shivam Tiwari (BE/A/158)**
4. **Sohan Kulkarni (BE/A/164)**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "C**omputer Engineering**".

(Prof. Shital Dhamal)

Guide

(Prof. Sonal Bankar)	**(Dr. Vivek Sunnapwar)**
Head of Department	**Principal**

Project Report Approval for B.E.

The project report entitled “Tumor Detection in Liver Using Monai and PyTorch” by Abhinavkumar Mishra(BE/A/122), Rajas Mahajan(BE/A/153), Shivam Tiwari(BE/A/158) and Sohan Kulkarni(BE/A/164) is approved for the award of “Bachelor of Engineering” degree in “Computer Engineering”.

Examiners

1.

2.

Date:

Place: Koparkhairane, Navi Mumbai

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principals of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data / fact / source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. **Abhinavkumar Mishra (BE/A/122)**
2. **Rajas Mahajan (BE/A/153)**
3. **Shivam Tiwari (BE/A/158)**
4. **Sohan Kulkarni (BE/A/164)**

Date:

ABSTRACT

- Deep learning is an important machine learning technique that helps computers identify objects in images. Medical images are used in diagnosis and treatment of diseases, injuries, and congenital abnormalities.

- Deep learning can be used to detect diseases like cancer by finding tumor cells through medical images. Deep learning also has the potential to improve the quality of medical care by segmenting organs during surgery or scanning patients for signs of cancer or other ailments.

- So the goal is to use **Monai** and **PyTorch** with the Python programming language to create a deep learning model to segment a liver from a public CT scan dataset. After completing this, we will be able to create the same model that segments the liver, as well as use the same principle to segment other organs or tumors from CT scans or MRIs using **U-net Architecture**.

ACKNOWLEDGEMENT

I remain immensely obliged to Prof. Shital Dhamal maam for providing me with the idea of the topic, for her invaluable support in gathering resources, her guidance and supervision which made this work successful.

I would like to give my thanks to the Head of the Computer Department, Prof.Sonal Bankar, Vice Principal, Dr. Subhash Shinde and our Principal Dr. Vivek Sunnapwar.

I am also thankful to the faculty and staff of the Computer Engineering Department and Lokmanya Tilak College of Engineering, Navi Mumbai for their invaluable support.

I would like to say that it has indeed been a fulfilling experience working on this project topic.

LIST OF FIGURES

ABBREVIATION

No.	Abbreviations	Full Form
1	MONAI	Medical Open Network for Artificial Intelligence
2	CT Scan	Computed Tomography Scan
3	MRI	Magnetic resonance imaging
4	PET Scan	positron emission tomography Scan
5	DL	Deep Learning
6	CRL	Chronic Renal Failure
7	DCNN	Deep Convolutional Neural Network
8	GPU	Graphics Processing Unit
9	ReLU	Rectified Linear Unit
10	CUDA	Compute Unified Device Architecture
11	3D	3 Dimensional
12	VS code	Visual Studio Code
13	RAM	Random Access Memory
14	AI	Artificial Intelligence
15	Nifti	Neuroimaging Informatics Technology Initiative
16	DICOM	Digital Imaging and Communications in Medicine format
17	IoU	Intersection over Union

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

- Deep learning in computer vision can be used for a variety of tasks, the most common of which is image classification, which was one of the first deep learning applications.
- The image classification technique simply determines whether or not an image contains the object that we are looking for (in our case, the liver).
- Then there was object detection, which is another extremely useful technique in computer vision. This technique entails drawing a box around the object we're looking for in the image (which we call a bounding box).
- Image segmentation, on the other hand, is a technique that combines image classification and object detection. This technique not only detects an object in an image, but it also segments the object's correct pixels.

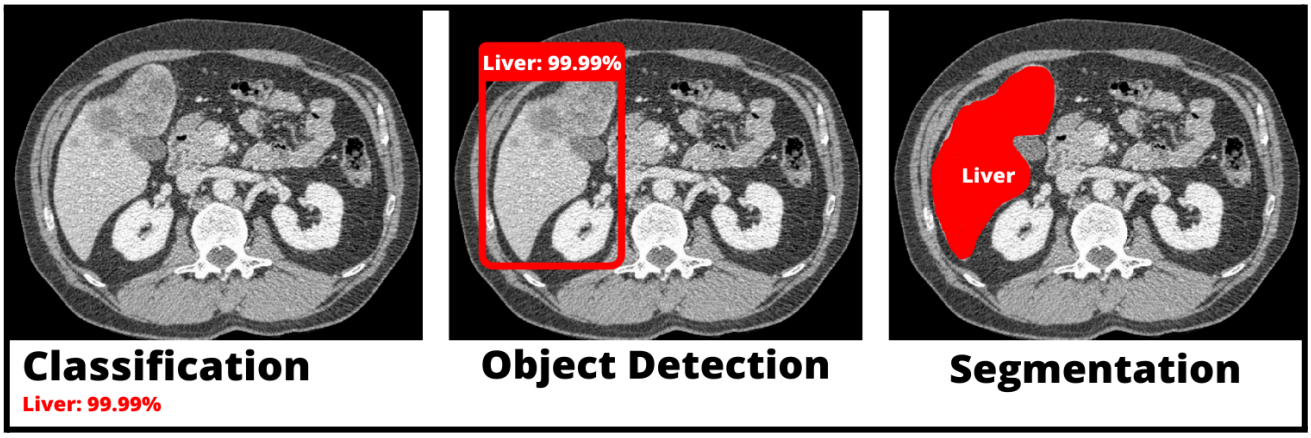

Fig. 1. Object detection using deep learning provides a fast and accurate means to predict the location of an object in an image whereas a neural network is used in the deep learning image segmentation technique to learn how to split a picture into segments. We have used semantic segmentation to identify the liver.

The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks.

Estimation of organs from Computed Tomography (CT) plays an important role during the choice of treatment strategies for most organ diseases . Despite the recent development in Computer Vision and Machine Learning fields, organ segmentation is still a challenge due to the difficulty in discriminating the edges of organs coupled with high variability of both intensity patterns and anatomical appearances. All these difficulties become more prominent, especially in pathological organs. The need for CT scan analysis for pre-diagnosis and treatment of internal organs is increasing due to the expectation in developed countries.

Automated organ estimation of an internal CT scan can help radiologists analyze scans faster and estimate organ images with fewer errors.[7] One of the biggest problems with the use of computed tomography is the selection of the area to be screened larger than the targeted area in most abdominal examinations. This is a major obstacle for doctors to make accurate and rapid progress in the clinical diagnosis and treatment of patients. Segmentation of the organ to be examined from the whole computed tomography image makes a great contribution to the effective examination of these large-scale abdominal images. Several studies are carried out on this subject with deep learning practices that have gained importance in recent years [8].

Providing organ and tissue segmentation in CT images has long been one of the topics of deep learning technology in the literature . By considering these current studies, it is aimed to propose a comprehensive study to support the studies in the field of automatic segmentation by using Deep Learning algorithms. It should be stated that results provide a preliminary preparation to segment the liver more quickly and accurately. The new findings that will emerge can thus facilitate the treatment and intervention of patients faster and more effectively for doctors. The algorithm's effectiveness and prediction process have a more dominant role in practical clinical life in automated segmentation processes [7]. Therefore, the structure to be created should be presented as fast and accurately as possible, and the rate of misleading the end-user should be minimized. Examination of liver segmentation with different amounts of data, architectures, and implementation methods may be a critical step for progress on this understanding.

1.2 MOTIVATION

- The motivation for tumor detection in the liver is primarily driven by the need to diagnose and treat liver cancer, which is a leading cause of cancer-related deaths worldwide. Liver cancer can arise from either the liver itself (primary liver cancer) or from cancer cells that have spread to the liver from other parts of the body (metastatic liver cancer).
- Early detection of liver tumors is critical for successful treatment and improved patient outcomes. Symptoms of liver cancer may not appear until the disease is advanced, making early detection through screening tests vital. Liver tumors can be detected using a variety of imaging techniques, such as ultrasound, CT scans, MRI scans, and PET scans [1]**.**

CHAPTER 2

LITERATURE SURVEY

2.1 SURVEY OF EXISTING SYSTEM

SR	TITLE	YEAR	AUTHOR	SUMMARY
1.	Automatic Liver Segmentation from CT Images Using Deep Learning Algorithms: A Comparative Study	2019	K. E. Sengun Y. T. Cetin, M.S Guzel, S. Can, E. Bostanci	Proposes the most efficient DL architectures for Liver segmentation .These frameworks are implemented and adapted into a Commercial software, “LiverVision”.
2.	U-Net: Convolutional Networks for Biomedical Image Segmentation	2018	Olaf Ronneberger, Philipp Fischer, Thomas Brox	This paper presents a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently.
3.	Fully convolutional network for liver segmentation and lesions detection	2017	A. Ben-Cohen, I. K. E. Diamant, M. Amitai and H. Greenspan	Further compared our cascade U-ResNets with six popular deep learning structures.
4.	SegNet: A deep convolutional encoder-decoder architecture for image segmentation	2016	V. Badrinarayana n, A. Kendall and R. Cipolla	Except for using the deconvolution layer, SegNet used the ’max-pooling indices’ to perform non-linear upsampling.
5.	Semantic image segmentation with deep convolutional nets and full CRFs	2015	L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille	Shows that responses at the final layer of DCNNs are not sufficiently localized for accurate object segmentation.

2.2 SUMMARIZED FINDINGS/RESEARCH GAP

Semantic segmentation has been studied recently and it basically aims to recognize and classify images by considering pixel levels. Deep learning has provided a significant impact on the field of Semantic segmentation as it is expected [7]. Several comprehensive Image Segmentation frameworks developed for labeling cells from light microscopic images and are able to produce quite accurate segmentation results [7] A recent study claims that U-Nets provide better effective recovery of high frequency edges in “sparse-view” CTs [5]. Another study employed the U-Net framework to extract retinal vessels, which is critical for the diagnosis of fundus diseases [10]. However, the drawbacks brought by U-Net can be quite insufficient in preventing the complexity caused by close have been developed in the literature in recent times. U-Net, created by O. Ronneberger, has been very popular among these frameworks, and is a good example of Convolution Neural Networks (CNNs) based semantic segmentation solution to the field of medical image segmentation. One of the leading of these studies that will be evaluated in this study is the “Dense U-Net” framework. This framework involves an algorithm that works more effectively in extracting more adequate features[1].

An example Output from LiverVision Software, Dr. Guzel’s team contributed to the development. The classical U-net architecture (32x32 pixels in the lowest resolution, and 512x512 in the highest resolution) and eliminating false-positives in different tissue structures such as the color intensity of the long-distanced organs. Besides, “ResNet” has been employed, which is a deeper layered architecture, allowing the residual learning system to facilitate the training of deeper networks that are difficult to train . This network has two pooling operations, convolution layers with “3×3” and “1 × 1” filters, followed by a fully connected layer, and a Softmax classifier[2]. The final one is the “SegNet" Architecture which is able to perform pixel wise segmentation more effectively[1]. There are no fully connected layers and hence it has only convolutional layers. According to which, a decoder up-samples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification[1].

More recent approaches have aimed to produce high quality unaries by trying to predict the labels for all the pixels in a patch as opposed to only the center pixel. This improves the results of Random Forest based unaries but thin structured classes are classified poorly. Dense depth maps computed from the CamVid video have also been used as input for classification using Random Forests [9]. Another approach argues for the use of a combination of popular hand designed features and spatio-temporal super-pixelization to obtain higher accuracy [9]. The best performing technique on the CamVid test [30] addresses the imbalance among label frequencies by combining object detection outputs with classifier predictions in a CRF framework. The result of all these techniques indicate the need for improved features for classification[9]

To summarize ,

- The earliest version of the U-Net framework represented drawbacks.

- As the batch size decreases , training time will increase and the probability of using maximum GPU memory increases. Thus, the selection of batch size needs great care.

- MONAI prepares the data using a portion by portion method wherein the slice is cropped. This method, although quicker, leads to low accuracy and underfitting according to our literature survey.

- Thus we decided to use MONAI only for data processing.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROBLEM STATEMENT AND OBJECTIVE

- To use Monai and PyTorch with the Python programming language to create a deep learning model resembling U-NET architecture to segment a liver from a public CT scan dataset.

- To detect diseases like cancer by finding tumor cells through medical images.

- To segment the liver and tumor from abdominal Computed Tomography (CT) images.

3.2 SCOPE OF THE WORK

- Semantic segmentation will be used because we need to know whether the slice contains a liver or not.

- U-Net architecture is split into two parts. The encoder and decoder**.** The U-Net output is a pixel-wise mask with the same dimensions as the input image and the same number of channels as the number of classes in our task [4,5].

- Each channel has the pixel probabilities of a specific class mentioned in Figure 3.

- We get two channels: one with background pixel probabilities (where there is no tumor) and one with foreground pixel probabilities (where there is a tumor) as shown in Figure 11.

3.3 FRAMEWORK/ARCHITECTURE

UNET

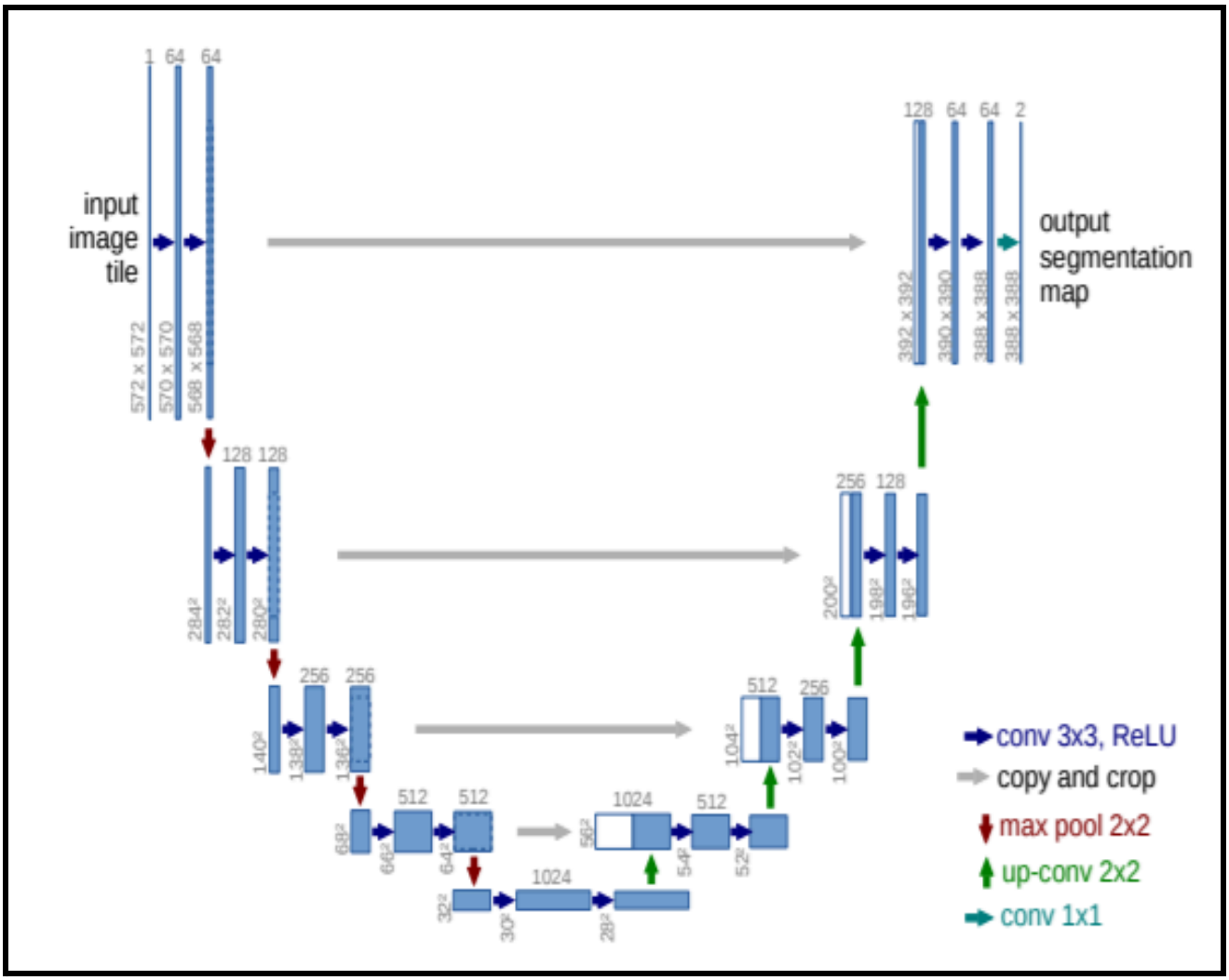

Fig. 2. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

UNet, evolved from the traditional convolutional neural network, was first designed and applied in 2015 to process biomedical images. As a general convolutional neural network focuses its task on image classification, where input is an image and output is one label, but in biomedical cases, it requires us not only to distinguish whether there is a disease, but also to localize the area of abnormality.

In this paper, we build upon a more elegant architecture, the so-called “fully convolutional network. We modify and extend this architecture such that it works with very few training images and yields more precise segmentations. The main idea is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. Hence, these layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the upsampled overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring output. A successive convolution layer can then learn to assemble a more precise output based on this information. One important modification in our architecture is that in the upsampling part we also have a large number of feature channels, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path, and yields a u-shaped architecture. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image[5].

The network architecture is illustrated in Figure 2. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes. In total the network has 23 convolutional layers. To allow a seamless tiling of the output segmentation map, it is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x- and y-size [2,5].

SEMANTIC SEGMENTATION

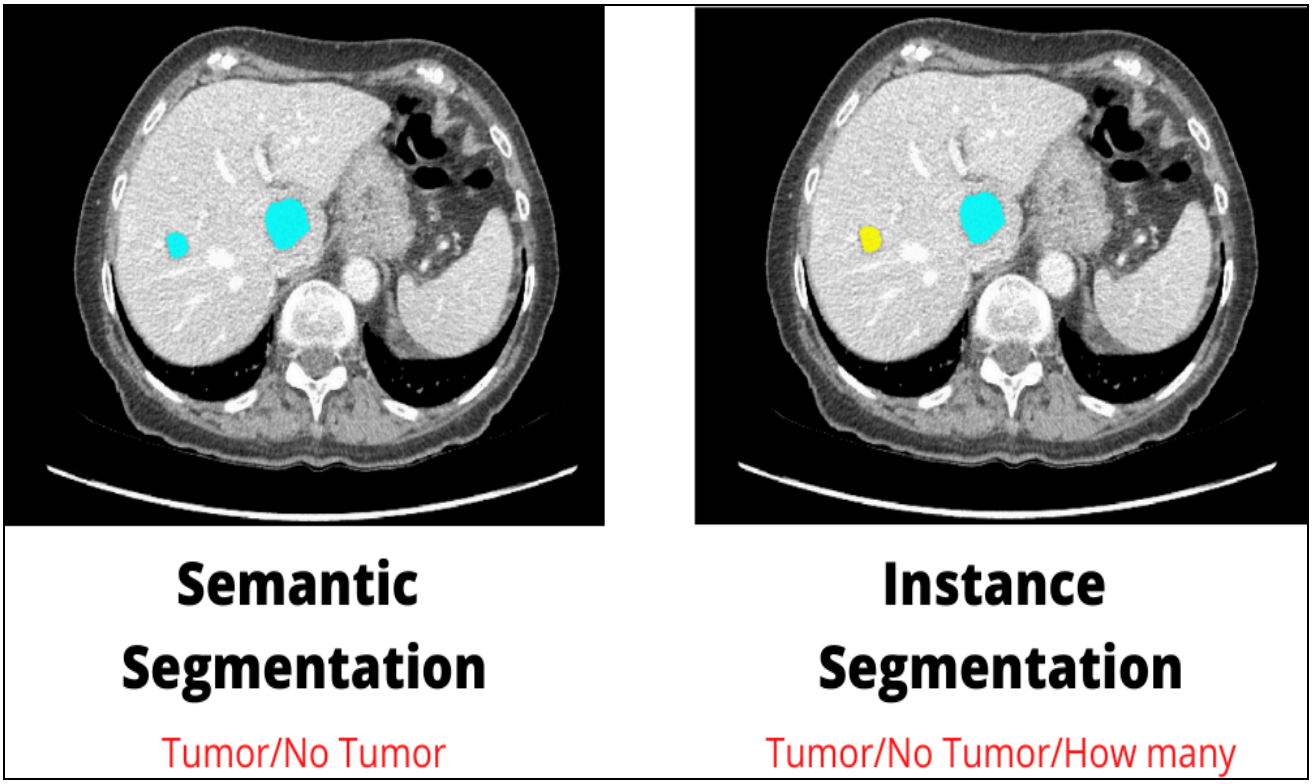

Fig. 3. As we can see Semantic segmentation associates a label or category with every pixel in an image. It is used to recognize a collection of pixels that form distinct categories whereas Instance Segmentation is a special form of image segmentation that deals with detecting instances of objects and demarcating their boundaries

- All of us have heard about pixels in an image. As humans, it is not a challenge for us to identify different objects in a picture quickly. We can distinguish a tree from a man and a car from a bicycle easily. It takes a fraction of a second for us to do that.
- However, machines do not have this sensory perception. They follow a set of rules. One such rule that helps them identify images via linking the pixels in an image is known as semantic segmentation.
- Semantic segmentation can be defined as the process of linking each pixel in a particular image to a class label. These labels could include people, cars, flowers, trees, buildings, roads, animals, and so on. The list is endless. In this case, it is the liver in a CT image.
- It is image classification at the pixel level. Accordingly, if you have many people in an image, segmentation will label all the objects as people objects [4].

3.4 DETAILS OF HARDWARE & SOFTWARE

Software:

- ➔ Anaconda
- ➔ Python 3.9
- ➔ Pytorch
- ➔ Monai
- ➔ Cuda
- ➔ 3D Slicer
- ➔ VS Code
- ➔ ITK Snap

Hardware:

- ➔ 8 GB RAM minimum
- ➔ NVIDIA GTX 1650 minimum
- ➔ Monitor
- ➔ Keyboard

Dataset Used:

Medical Segmentation Decathlon : Generalisable 3D Semantic Segmentation
This dataset was created for the LiTS17 challenge, which was organized in conjunction with the ISBI (International Symposium on Biomedical Imaging) 2017 and the MICCAI (Medical Image Computing and Computer Assisted Intervention) 2017. The dataset consists of 200 contrast-enhanced CT scans of the liver, which were collected from various clinical sites using different scanners and acquisition protocols. The scans were acquired in the portal venous phase with slice thickness ranging from 0.45 mm to 5 mm.

The dataset is divided into two sets: a training set and a test set. The training set contains 130 scans, while the test set contains 70 scans. Each scan in the dataset is accompanied by a set of expert annotations that delineate the boundaries of the liver and the tumor regions. The annotations were made by three experienced radiologists, and the final annotation was obtained by averaging the three annotations.

This dataset is challenging due to the variability in liver shape, size, and intensity values, as well as the presence of different types of tumors (i.e., hepatocellular carcinoma and metastases). The dataset has been widely used to evaluate liver tumor segmentation methods, and it has become a benchmark dataset in the field. The dataset is publicly available and can be downloaded from the challenge website.

3.5 DESIGN DETAILS

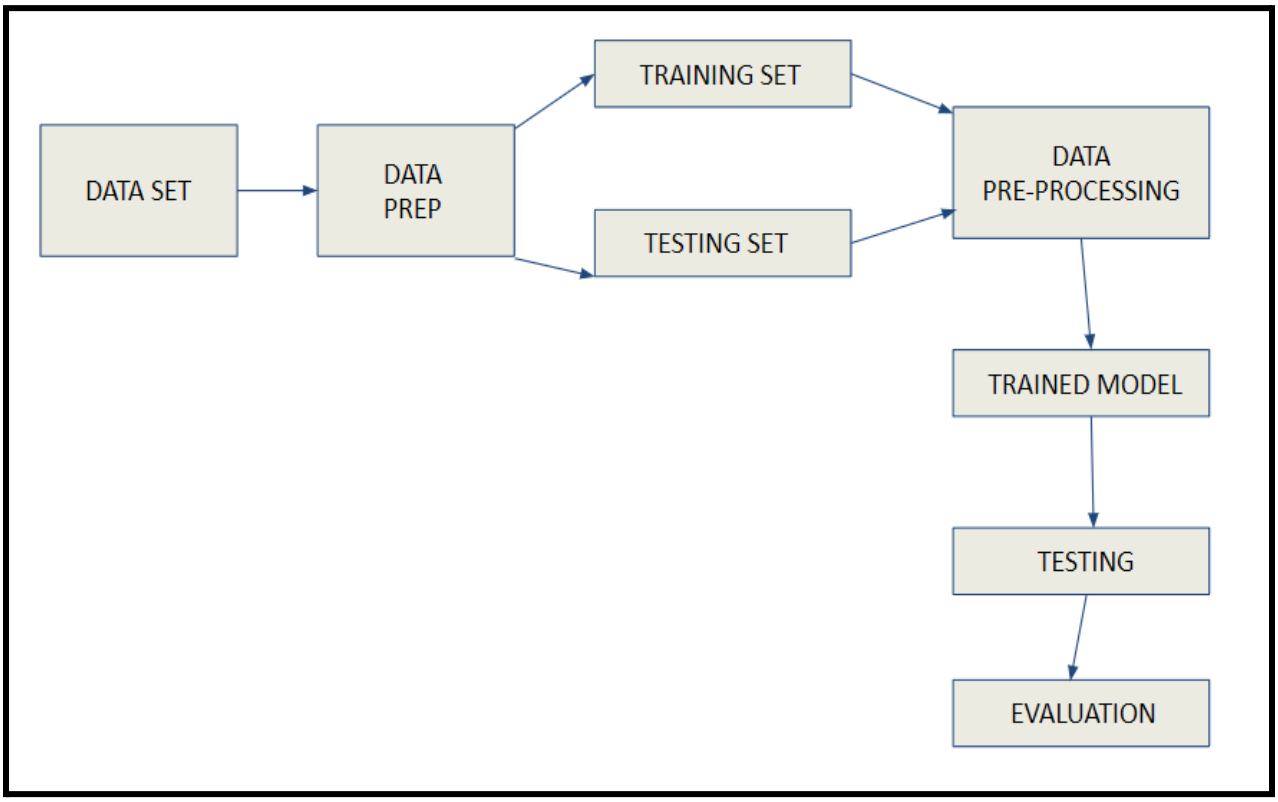

Fig. 4. This is the general flow of how we have taken the steps to create this model. It is very important to implement the design details of how we will approach a particular project/model by analyzing and preparing all the necessary steps.

Detecting tumors in the liver is a critical task in medical image analysis. PyTorch and MONAI (Medical Open Network for AI) are powerful tools for designing and implementing deep learning models for medical image analysis. In this task, we will use PyTorch and MONAI to design a tumor detection model for liver MRI images. Here are the design details for tumor detection in the liver using PyTorch and MONAI.

1. Data Preparation:

The first step is to collect and preprocess the data. We will use MRI images of the liver that contain tumors. We will preprocess the images by resizing them to a fixed size, normalizing the pixel values, and splitting the data into training and testing sets.

2. Data Augmentation:

We will perform data augmentation to increase the diversity of the training data. We

will use MONAI's transforms to perform random rotations, translations, and scaling. This will help the model generalize better to unseen data.

3. Model Architecture:

We will use a deep learning model based on the U-Net architecture, which is widely used in medical image analysis. The U-Net architecture consists of an encoder and a decoder. The encoder consists of convolutional layers that gradually reduce the spatial resolution of the input image, while increasing the number of feature maps. The decoder consists of convolutional layers that gradually increase the spatial resolution of the feature maps, while reducing their number. Skip connections between the encoder and decoder allow the model to capture both local and global information.

4. Loss Function:

The loss function is used to measure the difference between the predicted and actual output. We will use the binary cross-entropy loss, which is commonly used in binary classification tasks.

5. Training:

We will train the model using the Adam optimizer, which is an adaptive learning rate optimization algorithm. We will also use MONAI's DiceLoss metric, which measures the overlap between the predicted and actual segmentation masks.

6. Inference:

We will use the trained model to make predictions on new images. We will use a threshold to convert the model's output to a binary segmentation mask. We will then use post-processing techniques to remove false positives and false negatives. We can use morphology operations like erosion and dilation to remove small noise and fill small holes respectively.

7. Evaluation:

We will evaluate the performance of the model using metrics like accuracy, precision, recall, and F1 score. We will also visualize the segmentation masks to visually evaluate the model's performance.
By following these design details, we can design a tumor detection model for liver MRI images using PyTorch and MONAI.

3.6 METHODOLOGY

Data Preparation:

- We cannot directly use the slices given in the dataset because based on our literature survey, the accuracy obtained after training is below par.
- Hence we prepare the data for pre-processing manually.
- Since we are using Monai for data pre-processing, the slices need to be in Neuroimaging Informatics Technology Initiative (NIfTI) format, which is a common format for processing.
- The main purpose is to obtain the liver images of the patients easily in a grouped manner.
- We use the software 3D slicer for converting slices into a DICOM file which is an image saved in the Digital Imaging and Communications in Medicine format.
- These dicom files are further grouped and will be converted into NifTi using Python programming language and glob2 function.
- We are currently grouping in batches of approximately 64 slices.

Data Preprocessing:

Why do we use Monai ?
Because MONAI addresses the specific needs and opportunities of deep learning in medical imaging:

Medical Image Data:

➔ MONAI supports a variety of medical image formats (thanks to ITK, GDCM, NIBabble, and other toolkits), and it respects the physical properties of those images, e.g., pixel size, pixel spacing, and image orientation.[8]

➔ Those physical properties are often ignored by traditional computer vision libraries, but they are essential to the accurate and real-world application of deep learning to clinical medical images.

Medical Image Transforms:

Unlike with photographs or other common computer vision data, it is often not appropriate to rotate, flip, or change the intensities of medical images for augmentation purposes, yet warping the anatomy in images to match patient variations, removing ribs from x-ray images, and other steps are common in medical image pipelines and are provide by MONAI via ITK[8].

Pre-processing Methodology:

We set up a virtual environment for our project because this library does not always work when installed directly in the system.
pip install monai

Then we need to install PyTorch and some dependencies of monai.
pip install torch
pip install torch-vision
pip install "monai-weekly[gdwon, nibabel,tqdm]"

To apply multiple transforms to the same patient, we will use monai's "compose" function

- ➔ AddChanneld: This function will add a channel to our image and label
- ➔ Spacingd: This function will assist us in changing the voxel dimensions
- ➔ CropForeground: This function will assist us in cropping out the empty regions of the image that we do not require, leaving only the region of interest.
- ➔ Resized: It is required if you use the crop foreground function, because the function that will do the crop will have the output with random dimensions depending on each patient, so if we do not add an operation to give the same dimensions to all patients, our model will not work.[8]

3.7 IMPLEMENTATION

The hardware has a Windows environment consisting of 8 cores and 32 disks with 12 GB RAM. To reduce the noisy data and make the training more balanced, Adam optimization functions are implemented for all architectures. Adam's value is fixed to "(1e) - 3" in all experiments. The implementations are generated using Keras API functions along with TensorFlow. The parameters we manipulated in the comparison are the amount of data, the number of epochs, the batch size, and the division rate of the dataset for validation. Image processing and segmentation may result in higher computing costs than their counterparts based on other deep learning studies. When deep-layered architectures are also considered, classical architectural approaches with early stop functions are preferred [2]. In this way, it is aimed to minimize the possibility of encountering the overfitting situation during and after the training. Besides some further adjustments are carried out by using regularization functions to prevent overfitting due to the random state of the models at their initial weight. During the experimental process two critical problems have been encountered. The first one, overfitting, is caused by memorization in the segmentation processes of the models and the second one, undersampling, which the non-learning situation is caused by the small training set according to the structure of the deployed architectures.

An epoch is reading the entire dataset once. Batch size tells you the number of train samples obtained in one iteration. The validation split rate is the rate of separation of 2D images sliced in a vertical direction obtained from 3D CT images of the total patient data as train/validation sets. It is observed the final status of the models that emerged as a result of these test cases using the quality metrics and NIfTI files consisting of predicted masks. The tests have achieved remarkable success in the Dice coefficient and prediction results. During the verification process, other patient files that are excluded from the training and the validation processes, are employed to verify practical predictions [6].

- **Collection**
  First, we collected the dataset for our project from an open source website called Decathlon which provides numerous datasets.

- **Preparation**
  Since, the dataset that we obtained cannot be used directly because it will give lower accuracy, we decided to manually prepare our data using 3d web slicer and python glob2 function.

- **Pre-processing**
  We have pre-processed the data using MONAI, an open source framework for AI and deep learning software tools to be used in healthcare imaging research.

- **Training**
  130 files each with a batch size of 64 slices labeled in the data preparation phase is used for training purposes. Model has been trained for 216 epochs.

- **Testing**
  70 files each with a batch size of 64 slices has been used for testing purposes.

- **Evaluation**
  Evaluation of the model has been done using Dice Loss which takes into account not only the false positives but also penalizes the accuracy for missing the true positives. It also takes into account false negatives which can be extremely dangerous in medical fields. A patient's tumor if not detected and declared a negative even though he is afflicted can have disastrous effects.

3.8 TESTING

The similar pixels are counted (taking intersection, present in both the images) in the both images getting compared and multiplying it by 2 further dividing it by the total pixels in both the images. The below diagrams will make the picture more clear.

$$\frac{2|X \cap Y|}{|X| + |Y|}$$

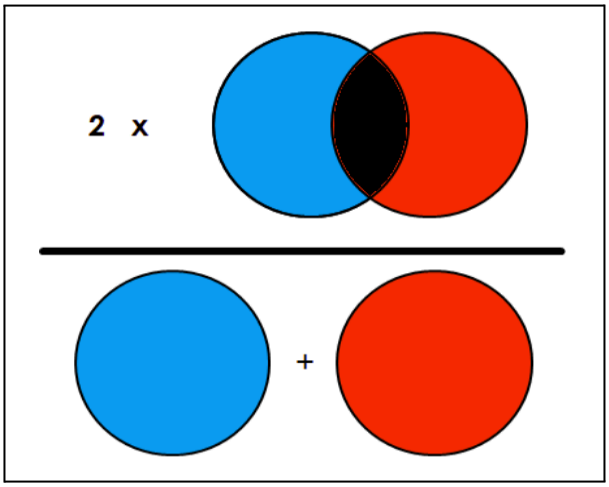

Fig. 5. Dice loss formula and representation

IoU (Intersection over Union)

As the name suggests, dividing the intersection means the number of similar pixels by the union means the total number of unique pixels in both the images. Both these metrics are positively correlated

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Fig. 6. Formula for IoU

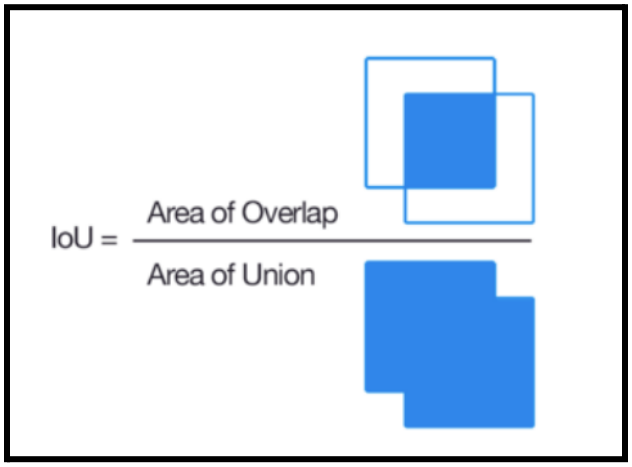

Fig. 7. Calculating IoU

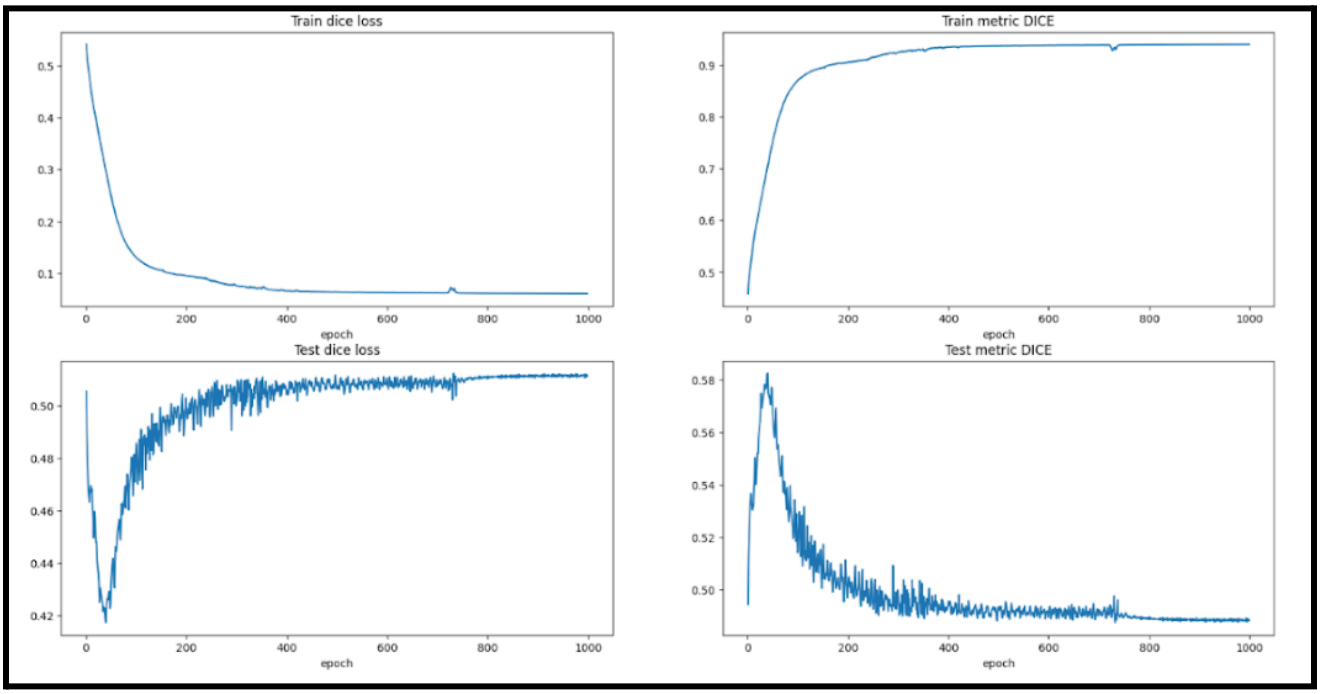

Fig. 8. Dice loss originates from Sørensen–Dice coefficient, which is a statistic developed to gauge the similarity between two samples. Here we can see the dice loss graphs for training and testing metrics.

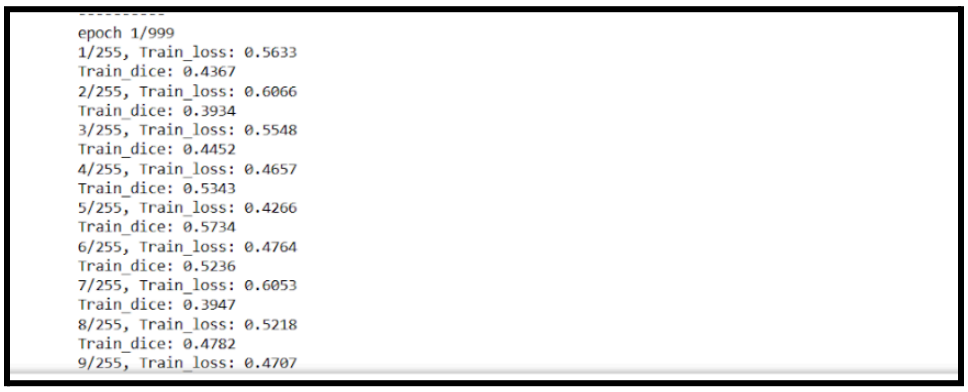

Fig. 9. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. We keep on training the model

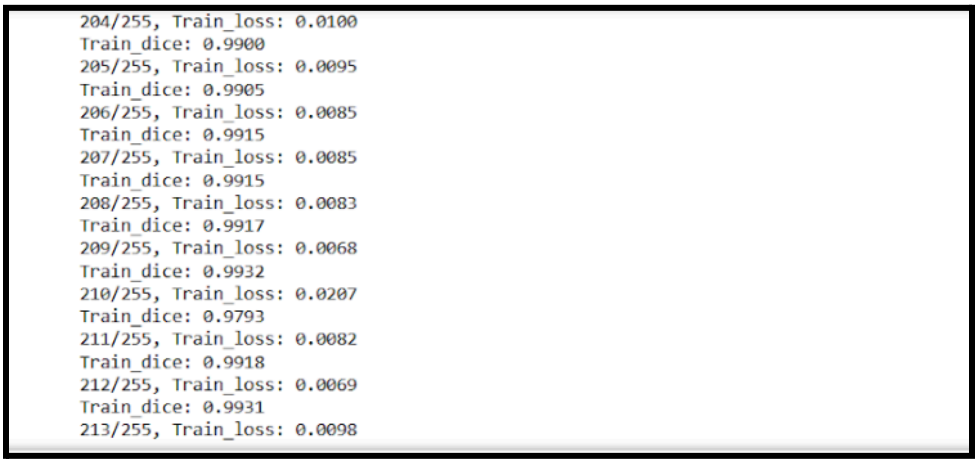

Fig. 10. As we keep on training our model with datasets, the train loss decreases. The model will give us more accurate results depending on how much data we feed it.

CHAPTER 4

RESULTS

4.1 RESULTS

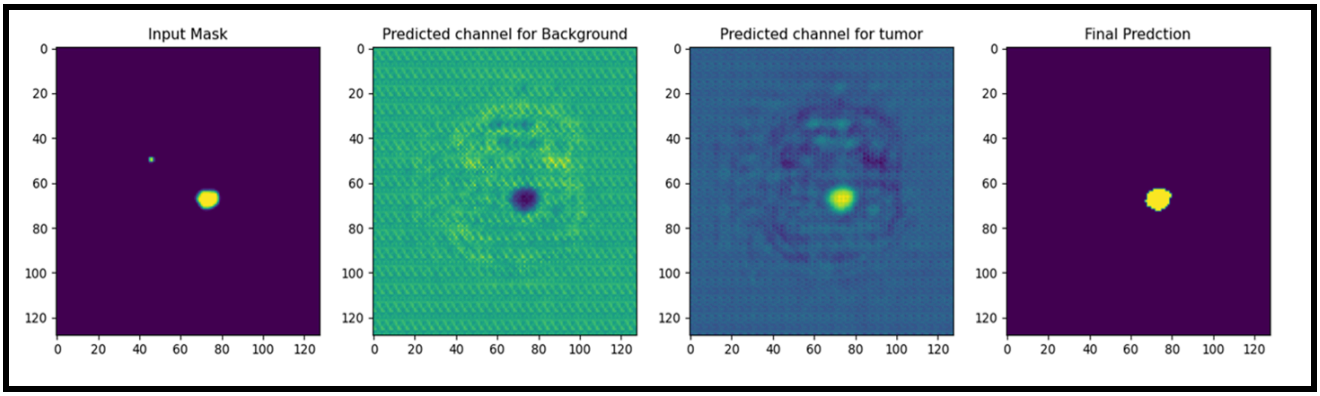

Fig. 11. These are the predicted channel results and the final prediction of how our model will detect the tumor based on the data we fed it.

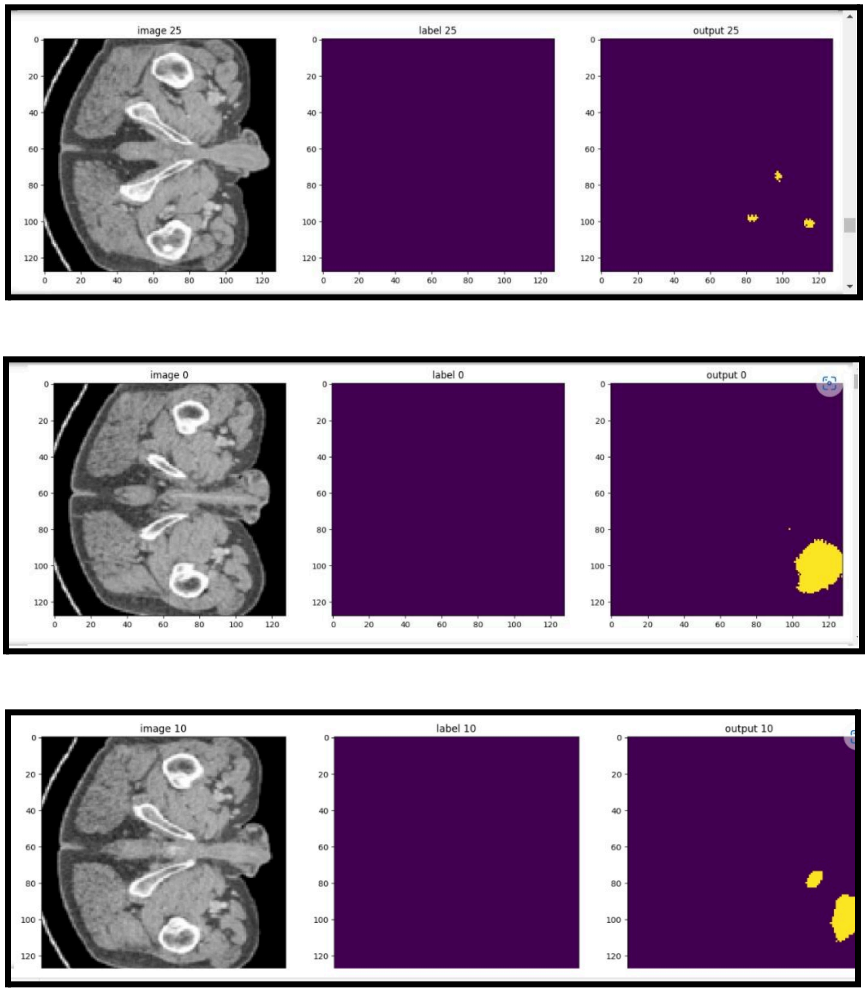

Fig. 12. Results obtained, the accuracy keep on increasing.

4.2 CONCLUSION

- Accurately depicted U-net Architecture and Semantic Segmentation.
- Successfully got background and foreground channel results.
- Implemented and depicted dice loss graph based on previous study materials.
- Hence, we have successfully proposed an effective Liver Tumor Detection Model.

4.3 FUTURE SCOPE

- Improved imaging techniques: Medical imaging technologies such as magnetic resonance imaging (MRI), computed tomography (CT), and ultrasound are constantly evolving. The future development of more advanced imaging techniques will enable early and accurate detection of liver tumors [2,6].

- Artificial intelligence: Machine learning and deep learning algorithms have the potential to improve the accuracy and speed of liver tumor detection. These algorithms can learn to recognize patterns and features that are difficult for the human eye to detect, which can enhance the sensitivity and specificity of liver tumor detection.

- Biomarkers: Research is ongoing in identifying new biomarkers that can help diagnose liver tumors at an early stage. The use of biomarkers in conjunction with medical imaging techniques can significantly improve the accuracy of liver tumor detection.

CHAPTER 5

REFERENCES

5.1 REFERENCES

1.Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., Heng, P.-A.: 3D deeply supervised network for automatic liver segmentation from CT volumes. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 149–157. Springer, Cham (2016).

2.Christ, P.F., et al.: Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 415–423. Springer, Cham (2016).

3.Lu, F., Wu, F., Hu, P., Peng, Z., Kong, D.: Automatic 3D liver location and segmentation via convolutional neural network and graph cut. Int. J. Comput. Assist. Radiol. Surg. 12(2), 171–182 (2017).

4.Luc, P., Couprie, C., Chintala, S.: Semantic Segmentation using Adversarial Networks. arXiv preprint. arXiv:1611.08408 (2016).

5.Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015).

6.Merkow, J., Marsden, A., Kriegman, D., Tu, Z.: Dense volume-to-volume vascular boundary detection. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9902, pp. 371–379. Springer, Cham (2016).

7.Sengun, K. E., Cetin, Y. T., Guzel, M. S., Can, S., and Bostanci, E., “Automatic Liver Segmentation from CT Images Using Deep Learning Algorithms: A Comparative Study”, arXiv e-prints, 2021. doi:10.48550/arXiv.2101.09987 (2017).

8.Cardoso, M.J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., Yang, I., Zephyr, M., Hashemian, B., Alle, S., Darestani, M.Z., Budd, C., Modat, M., Vercauteren, T.K., Wang, G., Li, Y., Hu, Y., Fu, Y., Gorman, B.L., Johnson, H.J., Genereaux, B.W., Erdal, B.S., Gupta, V., Diaz-Pinto, A., Dourson, A., Maier-Hein, L., Jaeger, P.F., Baumgartner, M., Kalpathy-Cramer, J., Flores, M.G., Kirby, J.S., Cooper, L.A., Roth, H.R., Xu, D., Bericat, D., Floca, R.O., Zhou, S.K., Shuaib, H., Farahani, K., Maier-Hein, K.H., Aylward, S., Dogra, P., Ourselin, S., & Feng, A. (2022). MONAI: An open-source framework for deep learning in healthcare

9. V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.