Prediction of Cardiovascular disease based on Image Segmentation of CT Scans

Dr A. Maheshwari*[a], Chinmaya Purohit**[b], Milind Pruthi***[b]

[a]Assistant Professor, Department of Computational Intelligence, SRMIST, Chennai, Tamil Nadu, India.
[b]CSE-AIML, CINTEL, SRMIST, Chennai, Tamil Nadu, India.

Abstract— **Machine learning is often considered an advanced technology that only highly trained experts can access. This prevents many doctors and biologists from using this tool in their research. This article aims to dispel that outdated notion. Heart failure (HF) has been shown to be one of the most common causes of death, so accurate and timely prediction of the risks of heart failure is necessary and of utmost importance. Cardiovascular diseases are a broad category of several diseases that affect the heart and blood vessels. Early predictive methods for cardiovascular disease helped make decisions about changes in at-risk patients that reduced risks. The healthcare industry contains a lot of medical data, so machine learning algorithms are needed to predict heart disease. We propose using U-net and deep learning to segment cardiac CT scan images to check the area of the heart that may cause/ is causing problems in the foreseeable future/currently. A comparative analysis was made of selected publications that were published in the years 2016 to 2022.**

Keywords—Machine Learning, Cardiovascular disease, U-net, CNN, Neural Networks, CT Scans, Image masks, Deep Learning.

I. Introduction

Machine Learning Models are mostly used for classifications and regressions models, and are rarely used in the medical professions.

This prevents many doctors and biologists from using this tool in their research. The goal of this paper is to eliminate this outdated perception. Heart disease is the leading cause of death in the world, and it is surprising that research into this is relatively new, with most cited and up to date papers from being around the year 2021 to mid 2022.

We believe that more research into this subject must be made, to make sure that the high numbers of casualties can be reduced to some extent by easy access to preventive measures. For this reason, we are looking ahead to add functionality to the model by allowing the user to access a user-friendly mode of sending their data to the dataset and check if they are at a risk of contracting CVD.

II. Innovation Idea

Cardiovascular diseases are the leading cause of global deaths anually . Most of these cases are due to lack of awareness and inability to understand the symptoms in the body in time. These symptoms can be inherited or mutated; Either way, it's important to diagnose these symptoms before they can lead to more serious health risks.

High blood pressure, smoking, high cholesterol, diabetes or even inactivity can become risk factors that can lead a person to cardiovascular disease.

We intend to help solve this problem by implementing a highly accurate deep learning model that alerts the user to immediate health threats by tracking and segmenting CT scans to ultimately infer whether a person is at risk of disease. CVD or not.

III. Clinical Motivation

We have gone through several IEEE transaction and access papers, and we saw that the majority of these research papers have done image segmentation on the brain and other organs of the body but very few of them have done the same on the heart.

The data set we used is quite exclusive, extensive and has been acquired through a painstakingly long process through multiple sources and from expert systems.

Since the architecture of the deep learning model used is quite intricate, and the data set we are using is quite large, the model takes a lot of computational time. Therefore, we plan to run it on AWS, using cutting edge technology.

*78mahee@gmail.com
**cp6935@srmist.edu.in
***mp7563@srmist.edu.in

IV. Requirement Gathering

The model being used in this project is a deep learning architecture called the U-net. U-net requires a lot of computational power, which can be easily acquired through a lot of services that offer Infrastructure as a Service, namely but not including AWS, google colab, Kaggle etc. For having most accurate results within the nominal amount of time, we look forward to using AWS with a minimal fee and a GPU enabled accelerator.

CT scans are not as mainstream as ultrasound scans, but are steadily gaining more traction as the need for more accurate and more detailed scans for study are being needed increasingly. Any hospital with radiology can provide the patient with CT scans, and are readily available.

The model should also be able to read specific medical files, which are used specifically for bio-imaging. These images contain more data about the patient than regular images, and for 2D images, most popular images are stored in DICOM format, which is needed in this project. Hence, we will need to import a lot of libraries, including numpy, os, and pydicom.

V. Literature Survey

There has been quite some research done into U-net and Deep Learning. These are:

1. Nikolaos Zioluis et al, proposed a biologically inspired long-range skip connection for the UNet architecture that relies on the perceptual illusion of hybrid images, being images that simultaneously encode two images. Coupling features from earlier encoders deeper into the decoder allows UNet models to generate fine-grained density prediction. While proven in segmentation tasks, the network's benefits are down-weighted for dense regression tasks as these long-range skip connections additionally result in texture transfer artifacts, 2022.

2. Yiming Bao et al, introduced a novel network called Comprehensive 3D UNet (C3D-UNet). Compared to 3D-UNet, an intact encoding (IE) strategy is proposed to extract features from wider receptive fields, designed as extended convolutional blocks with a higher expansion rate. Moreover, a local attention (LA) mechanism is applied in skip connections for more robust and effective information fusion, 2021.

3. Song-Toan Tran et al, proposed a Un-Net, an n-fold network architecture, was proposed based on the traditional U-Net. In the Un-Net model, the output characteristics of the convolution units are taken as the bypass interface.. Therefore, the Un-Net network exploits the output features of the convolution units in the nodes. They investigated a U2 -Net and a U3 -Net for segmentation of the liver and liver tumors, 2021.

4. Nahian Siddique et al, discussed the many innovations that have advanced in deep learning and discuss how these tools facilitate U-net. In addition, they reviewed the different image modalities and application areas that have been enhanced by U-net, 2021.

5. Yuichi Wakamatsu et al, proposed method consists of supranational muscle localization using a scapula segmentation result and supranational muscle segmentation based on the localization result. U-Net is used for scapula and supranational muscle segmentation. In the experiment, they used torso CT images and pseudo-chest CT images which were generated from the scans of the same patient, 2021.

6. Rania Ramadan et al, proposed three novel variants of U-Net model with single, dual, and triple inputs, namely,
   Single Input Color U-Net (SICU-Net), Dual Input Color U-Net (DICU-Net) and Triple Input Color U-Net (TICU-Net) are proposed. The structure of SICU-Net, DICU-Net and TICU-Net contains subnets of one, two and three encoders connected by only one decoder path.. A different color space of the input image is directed to each subnet of the encoder.

7. Yasmin M. Kassim et al, proposed a novel pipeline for red blood cell detection and counting in thin blood smear microscopy images, named RBCNet, using a dual deep learning architecture. RBCNet consists of a U-Net first stage for cell cluster super pixel segmentation, followed by a second refinement stage Faster R-CNN for detecting small cell objects within the connected component clusters.

8. Neil Micallef et al, proposed a variation of the U-Net++ model, which is itself an adaptation of U-Net, and evaluated its brain tumor segmentation capabilities. The proposed approach obtained dice ratios of 0.7192, 0.8712, and 0.7817 in the Tumor Enhancement, Whole Tumor, and Tumor Core categories on the BraTS 2019 challenge data set. The proposed approach differs from the standard U-Net model in several ways, including the loss function, the number of convolutional blocks, and the method of using depth monitoring.

9. Chuan Zhou et al, developed a recursive training strategy to train a deep learning model for kernel detection and segmentation using incomplete annotation. High-quality U-Net segmented objects were selected using a semi-automatic method. High-quality U-Net segmented objects were selected using a semi-automatic method. Combining the newly selected high-quality objects with the annotated cores and previously generated negative samples, the U-
   Net model was retrained recursively until the stopping
   criteria were met.

10. Alyaa Amer et al, propose a fully automated left ventricle segmentation method that can overcome those challenges. The method performs accurate delineation for the ventricle boundaries despite the ill-defined borders and shape variability of the left ventricle. U-net's well-known deep learning segmentation model met some of these challenges with excellent performance. However, this ignores the contribution of all semantic information throughout the segmentation process. Hence, they proposed a novel deep learning segmentation method based on U-net, named ResDUnet.

11. Yuli Sun Hariyani, et al, proposed dual attention deep learning based on U-Net for nailfold capillary segmentation, named DA-CapNet. Automated segmentation of capillaries in nail folds is a difficult task due to image variability due to noise and insufficient focus and poor visibility of capillaries. This task can be useful to detect and estimate the severity of autoimmune diseases of connective tissues or learning the status of white blood cells based on the cells' blood flow on the nailfold capillary.

12. Mohammad D. Alahmadi proposed a Multi-access Attention U-Net (MSAU-Net) for skin lesion segmentation. In particular, we improve the typical U-net by inserting an attention mechanism at the bottleneck of the network to model the hierarchical representation.

13. Andreas Kofler et al, claimed that their method outperforms a 2D spatially trained U-net and a 2D spatiotemporal U-net. Compared with the 3D spatiotemporal U network, our method provides comparable results, but requires shorter training time and less training data. Compared to the compressed detection-based methods kt-FOCUSS and total variation regularized reconstruction, our method improves image quality in all reported metrics. In addition, it achieves competitive results compared to the iterative reconstruction method based on adaptive regularization with dictionary learning and full variation and methods based on sequential networks, but requires only a fraction of the computation and training time.

14. Yunjiao Deng et al, support the fact that U-Net was developed into an advanced U-network by designing an architecture with nested and dense bypass links, and U-Net 3 was developed into an advanced U-network using full-scale. bypass connections and depth monitoring with full-scale composite map feature maps. A proposed ELU Network with different loss functions is discussed to improve the learning effect of brain tumor including WT (whole tumor), TC (tumor core) and ET (tumor enhancement), and a new loss function DFK is proposed. The effectiveness of the proposed method was demonstrated on brain tumors used in the BraTS 2018 challenge and liver data used in the ISBI LiTS 2017 challenge.

15. An-Cin Li et al proposed that a deep learning (DL) method based on the smallest number of measurements could be used for isotropic qDPC microscopy. We use a commonly used convolutional neural network, namely the U-network architecture, which is trained to produce 12-axis isotropic reconstructed cell images from 1-axis anisotropic cell images. To further expand the number of training images, the U-network model is trained on a patch basis.

VI. DEEP LEARNING ARCHITECTURE

The architecture being used for deep learning is U-net. Evolving from traditional convolutional neural networks, UNet was first developed and applied in 2015 to process biomedical images. A typical convolutional neural network focuses its task on image classification, where the input is an image and the output is a label, so in the biomedical case it not only distinguishes whether a disease is present, but also identifies its region(abnormality).

UNet is committed to solving this problem. The reason is that they are able to define and distinguish boundaries by classifying each pixel so that inputs and outputs share the same size.

Let's take an overview about U-NET.

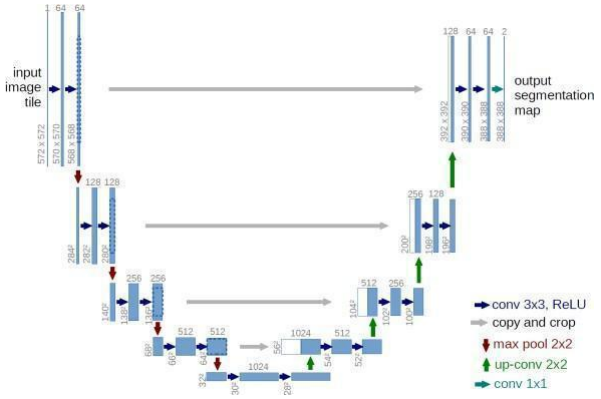

Figure 1 : U-net Architecture(https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/)

At first glance, it is shaped like the letter "U". The structure is symmetrical and consists of two main parts: the left part, called the contraction path, is formed by the general folding process; The right side is the stretch path, and consists of transported 2D convolutional layers.

I. CONTRACTING PATH

The formula for contracting path in UNET is as follows, in that order:

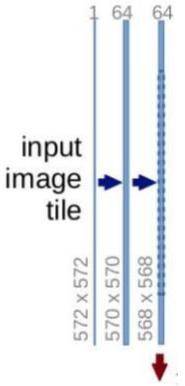

Figure 2: Convolutional layer

Conv-layer1, conv_layer2, max_pooling, dropout (optional)

Note that each process forms two convolution layers and the number of channels changes from 1 → 64 as the convolution process increases the image depth. The red arrow pointing down is the largest merging process, which reduces the image size by half (the reduced size 572x572 → 568x568 is due to padding issues, but this implementation uses the padding="same" parameter).

This process is repeated 3 more times.

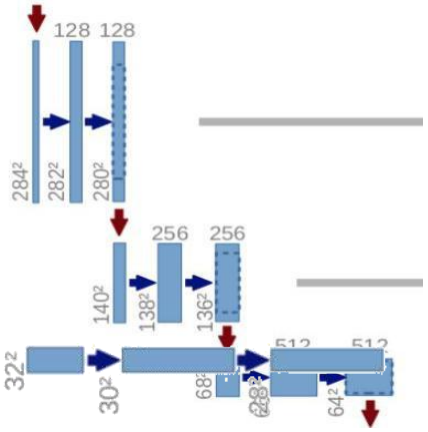

Figure 3: Encoding blocks

Then we reach at the bottom that is,

Figure 4: Bottom layer of U-net

At this stage we perform two more convolutions but there is no max pooling done. The point in the image has been resized to 28x28x1024. Let's go to the expansive path.

II. EXPANSIVE PATH

The expansion path expands the image to its original size. The formula is: conv_2d_transpose, concatenate, conv_layer1 and conv_layer2.

Figure 5: Expansive Upsampling path

Transposed convolution is an upsampling technique that increases the size of the image. Basically, the original image is padded, followed by a convolution operation. After transposed convolutions, the image is scaled to 28 x 28 x 1024 → 56 x 56 x 512 and this image is concatenated with the corresponding image from the reduction pass and together yields

an image of size 56 x 56 x 1024 . The reason here is to combine information from previous layers to get a more accurate prediction.

Now that we've reached the topmost stage of the architecture, the last step is to reformat the image to meet our prediction requirements.

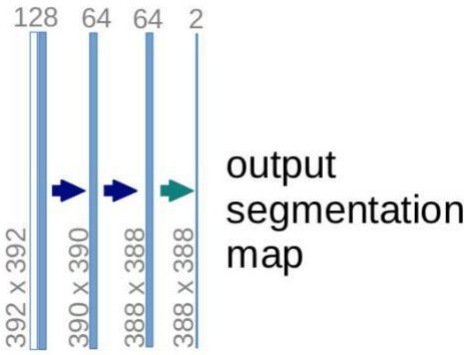

Figure 6: Output segmentation

The last layer is a convolutional layer with 1 filter of size 1x1 (note that there is no dense layer in the whole network). And the rest is the same for neural network training.

Conclusion

UNet is able to perform image localisation by predicting the image pixel by pixel. UNet has many applications for image segmentation.

The usage of U-net and MRI has been increasing over the years, as Jiong Wu et al. Have demonstrated in their brain MRI white matter hyperintensities Segmentation research paper.

VII. About the Modules

The model is divided into four major modules, each encompassing a major milestone in this project. The model uses deep learning and U-net architecture to localize the image and perform pixel by pixel analysis by utilizing both the image and the mask.

1) Module 1: Data Preprocessing

This module revolves around data pre-processing and cleaning to make the data more presentable to the model, without affecting the model due to impurities and other outliers that can have an effect on the predictive power of the model.

The data set has been taken from various sources, both online and offline. The primary data set has been taken from kaggle, an online data science platform. This dataset has been through multiple additions, from other various online sources, and through an expert system offline, from the actual CT database.

After the dataset has been gathered, we need to clean the data, which requires resizing the images to fit the input image size for the U-net. After the completion of this task, we need to split the data into test and train. However, we already have the data split, which was already provided with the data set. Therefore, no splitting the data into test and train is necessary. We will, however, split the train data into train and validation, to confirm that our model is working correctly. The data is split into an 80:20 ratio, confirming that we obtain around 4000 images for training and around 800 for validation.

The input size of the U-net in question will be an image of dimensions 512 x 512 x 3. The images in training folder will iteratively go through the process of resizing such that each image is the aforementioned size to be used in the U-net without any problems.

2) Module 2: Implementing U-net

Implementing U-net is an arduous task in of itself, because of the complexity of the architecture compared with other deep learning models like Alex-Net and VCC Net. The U-net consists of a multitude of convolutional layers, which help in decreasing the size and increasing the features in the first half of the model, and doing the opposite in the other half of the model.

The process consists of four encoder blocks, after which it gets passed through a convolutional block. The decoders are responsible for deconvoluting the features, which is then passed as the output.

Breaking down the encoder block into its components, we have two main parts - convolution block and the pooling layer. We will come to the convolutional block after we discuss the pooling layer.

The pooling layer is used to reduce the dimensions of feature maps. This is especially necessary when we are dealing with large amounts of dimensions, because we need to reduce the number of parameters that we can train, and in turn, reduce the computational power.

Since we are working with thousands of images, we need to have a pooling layer to reduce training time and focus on only specific parts of the images, due to which U-net was selected. U-net has been described as a very good model for localization of images.

We now come on the decoder block, which reduces features to upsample the image, creating less blur and more recognizable pattern, in contrast to the encoder block which localizes the images down to the pixels. The decoder transposes the convolutional layer, and the result is then added to the encoder block at the same level, resulting in an image having two times the features than the encoder at the same level.

The convolutional layer here is of the utmost importance, and is used to blur or sharpen images in the U-net. The convolutional block here consists of a series of processes, in order to get the desired output.

The layer first convolutes the input image with the amount of filter the next layer of the U-net requires, for example, if the input image is 572 x 572 x 1, it will be convoluted to 568 x 568 x 64, and then the next filter will be 280 x 280 x 128, the double of the last filter. The filters also decide what the size of the image in the next layer will be. The same can be said for upsampling of the images.

After the layer has been convoluted, we normalize the layer to prevent excessive computational resource wastage. The layer is then finally available for activation, and uses the ReLu function.

We repeat the process once more, reduce the size further, and then finally return the whole image for the next iteration through the U-net layers.

Figure 7 shows the trainable parameters.

```
activation_15 (Activation)     (None, 256, 256, 12  0          ['batch_normalization_15[0][0]']
                               8)

conv2d_transpose_3 (Conv2DTran  (None, 512, 512, 64  32832     ['activation_15[0][0]']
spose)                         )

concatenate_3 (Concatenate)    (None, 512, 512, 12  0          ['conv2d_transpose_3[0][0]',
                               8)                               'activation_1[0][0]']

conv2d_16 (Conv2D)             (None, 512, 512, 64  73792      ['concatenate_3[0][0]']
                               )

batch_normalization_16 (BatchN  (None, 512, 512, 64  256       ['conv2d_16[0][0]']
ormalization)                  )

activation_16 (Activation)     (None, 512, 512, 64  0          ['batch_normalization_16[0][0]']
                               )

conv2d_17 (Conv2D)             (None, 512, 512, 64  36928      ['activation_16[0][0]']
                               )

batch_normalization_17 (BatchN  (None, 512, 512, 64  256       ['conv2d_17[0][0]']
ormalization)                  )

activation_17 (Activation)     (None, 512, 512, 64  0          ['batch_normalization_17[0][0]']
                               )

conv2d_18 (Conv2D)             (None, 512, 512, 1)  65         ['activation_17[0][0]']

==================================================================================================
Total params: 31,055,297
Trainable params: 31,043,521
Non-trainable params: 11,776
```

Figure 7: U-net implementation

This process is the main one, and utilizes the most time and resources to complete, because we are running around 4500 images and their masks through the U-net for every time, and comparing it against the h5 and the csv file to check for measurements.

The train function is responsible for checking the image along with its mask, and determining the correct Region of Interest (R.O.I). A correct determination will cause the increase in accuracy and vice versa.

In the current model, as stated before, the activation function is ReLU, and the optimizer being used is Adam.

Finally, we have callbacks, which is basically hyper parameter tuning, which will increase the accuracy if correctly handled with, and is generally used in advanced deep learning models. There are five parameters at play here, which are ModelCheckpoint, ReduceLROnPlateau, CSVLogger, TensorBoard and EarlyStopping. Each of these hyper parameters play a role, for example, early stopper keeps a lookout for val_loss in the model, which if it does not decrease, stops the model early to prevent overuse of the computational resources when the model is clearly not progressing in its accuracy.

After the last parameters have been declared, we can start trying to fit the model. We will pass the train dataset, with 5 epochs, the validation dataset, the declared callbacks and keep shuffling out of the loop.

Figure 8 shows the epochs after training. Notice that the loss is reduced significantly, and that the precision is increased substantially. If the precision does not improve, it is not stored in the hierarchical format for reducing computations.

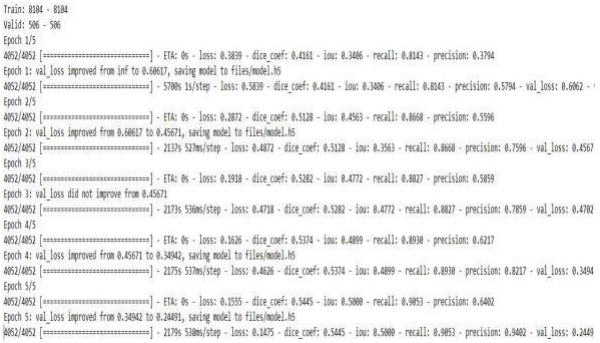

Figure 8: Training on U-Net

3) MODULE 3: TRAINING DATA ON U-NET

We go back to our training and validation data sets discussed in module 1. We now have our model that we are going to use. The next step is to train the model on the data so that it is able to predict the area with the problems in the test DICOM files.

There are two more things to mention here, and that is the presence of evaluation and metrics. Both of these programs are essential for the smooth working of the model. Metrics is used to measure the dice loss of the model, while evaluation is used to find a multitude of measures which are Accuracy, F1 score, Jaccard, Recall and Precision.

Evaluation metrics not only capture the previously mentioned values, they store every value for each image in every epoch. This data is stored in a csv file which is used by the training model to decide at which point it would be a good idea to dropout, as well other hyper parameter tuning and an insight into what the final accuracy and recall would look like.

Finally, we have the main function called train, which is responsible for binding everything together.

Figure 9 through 12 on the other hand, show some sample input images along with their masks used for training, respectively.

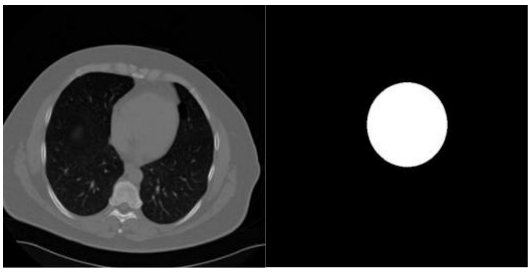

Figure 9: Patient 100072 slice 43

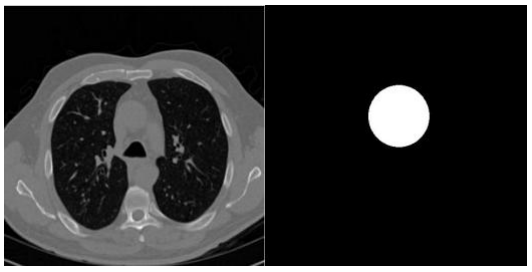

Figure 10: Patient 100093 Slice 58

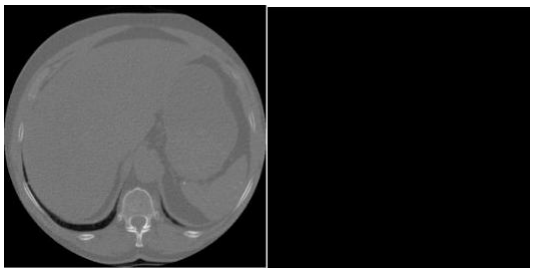

Figure 11: Patient 100053 Slice 07

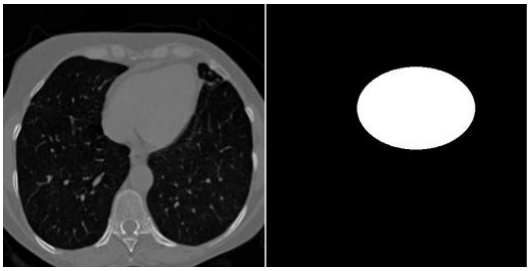

Figure 12: Patient 100091 Slice 89

4) MODULE 4: TESTING DATA AND RESULT SHOWCASE

After the model has been trained on the data and has been validated, we move onto the testing phase. The test folder consists of around 800 high resolution DICOM files, which are a standard in medical imaging. To use DICOM files in our python program, we import pydicom, which is a python open source library for reading and working on DICOM files.

The file predict goes through several hundred of these DICOM files and checks each image for the area which can be a problem, according to the training it received, with all the CT Scans and their masks. The file predicts the anomaly and appends it to the original DICOM file and saves it as a new PNG file altogether.

We get two results from the model. One is the result obtained from validation data and the other is test data itself. The validation data showcases the results along with their masks and the predicted mask in the image. Test data on the other hand carries over values that are completely standalone, with just the original DICOM image and the predicted mask for the image based on the model.

Figure 13 to 15 show the validation result while Figure 16 to 18 show test results.

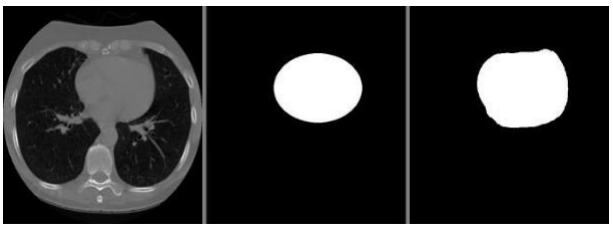

Figure 13: Patient 100056 Slice 73

Figure 14: Patient 100072 Slice 17

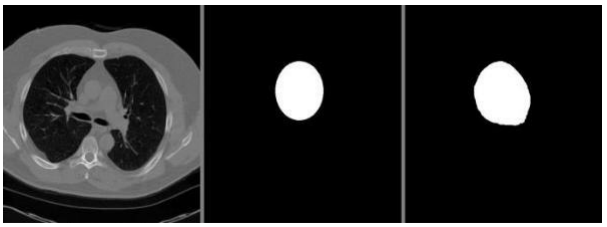

Figure 15: Patient 100073 Slice 50

The test results look something like:

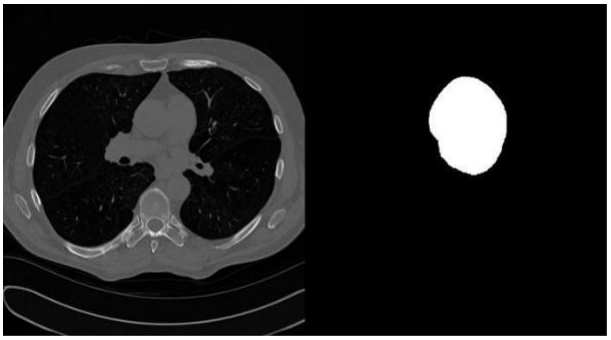

Figure 16: Patient 100253 Slice 84

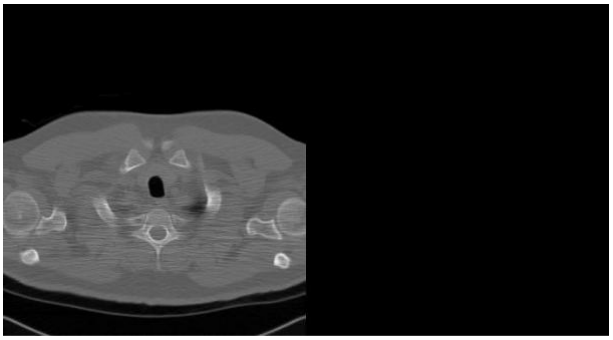

Figure 17: Patient 100258 Slice 02

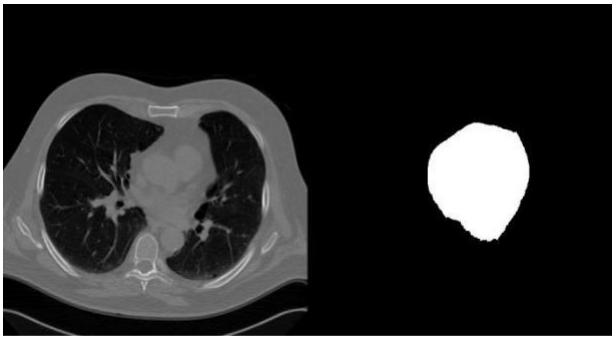

Figure 18: Patient 100254 Slice 70

VIII. CONCLUSION

To conclude, we would like to state that U-net is indeed one of the best deep learning models to implement for localization of images and to find the area in the images which show the abnormality in the heart. With an accuracy of **97%** and a high F1, recall score, we have successfully achieved our objective of obtaining a high accuracy low loss model capable of giving out results with an excellent chance of the model giving out results which are close to the ideal results.

ACKNOWLEDGMENT

We would like to express our gratitude and appreciation to Mrs A. Maheshwari, Mrs Abhirami and other members of the panel for allowing us to take this opportunity and attempt to build this project.

We would also like to thank SRMIST for providing us with the resources and the platforms we had the opportunity to use in this project.

Finally, we would like to thank our classmates and our faculties for their valuable support and input to the project.

REFERENCES

[1] M. F. Mushtaq et al., "BHCNet: Neural Network-Based Brain Hemorrhage Classification Using Head CT Scan," in IEEE Access, vol. 9, pp. 113901-113916, 2021, doi: 10.1109/ACCESS.2021.3102740.

[2] V. Ravi, M. Alazab, S. Srinivasan, A. Arunachalam, and K. P. Soman, ‘‘Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning,’’ IEEE Trans. Eng. Manag., early access, Mar. 12, 2021, doi: 10.1109/TEM.2021.3059664.

[3] J. Tan, Y. Gao, Z. Liang, W. Cao, M. J. Pomeroy, Y. Huo, L. Li, M. A. Barish, A. F. Abbasi, and P. J. Pickhardt, ‘‘3D-GLCM CNN: A 3-dimensional gray-level co-occurrence matrix-based CNN model for polyp classification via CT colonography,’’ IEEE Trans. Med. Imag., vol. 39, no. 6, pp. 2013–2024, Jun. 2020..

[4] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. C. Mok, L. Shi, and P. A. Heng, ‘‘Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks,’’ IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1182–1195, May 2016

[5] Y. Yuan, M. Chao, and Y.-C. Lo, ‘‘Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance,’’ IEEE Trans. Med. Imag., vol. 36, no. 9, pp. 1876–1886, Sep. 2017.

[6] M. J. J. P. van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sanchez, ‘‘Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images,’’ IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1273–1284, May 2016

[7] H. Park, T. Schoepflin, and Y. Kim, ‘‘Active contour model with gradient directional information: Directional snake,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 2, pp. 252–256, Feb. 2001.

[8] P. Zhang et al., "3D Urban Buildings Extraction Based on Airborne LiDAR and Photogrammetric Point Cloud Fusion According to U-Net Deep Learning Model Segmentation," in IEEE Access, vol. 10, pp. 20889-20897, 2022, doi: 10.1109/ACCESS.2022.3152744.

[9] S. -T. Tran, C. -H. Cheng and D. -G. Liu, "A Multiple Layer U-Net, Un-Net, for Liver and Liver Tumor Segmentation in CT," in IEEE Access, vol. 9, pp. 3752-3764, 2021, doi: 10.1109/ACCESS.2020.3047861.

[10] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, ''H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes,'' IEEE Trans. Med. Imag., vol. 37, no. 12, pp. 2663–2674, Dec. 2018.

[11] A. Amer, X. Ye and F. Janan, "ResDUnet: A Deep Learning-Based Left Ventricle Segmentation Method for Echocardiography," in IEEE Access, vol. 9, pp. 159755-159763, 2021, doi: 10.1109/ACCESS.2021.3122256.

[12] O. Bernard, J. G. Bosch, B. Heyde, M. Alessandrini, D. Barbosa, S. Camarasu-Pop, F. Cervenansky, S. Valette, O. Mirea, M. Bernier, and P. M. Jodoin, ''Standardized evaluation system for left ventricular segmentation algorithms in 3D echocardiography,'' IEEE Trans. Med. Imag., vol. 35, no. 4, pp. 967–977, Apr. 2016.

[13] Y. S. Hariyani, H. Eom and C. Park, "DA-Capnet: Dual Attention Deep Learning Based on U-Net for Nailfold Capillary Segmentation," in IEEE Access, vol. 8, pp. 10543-10553, 2020, doi: 10.1109/ACCESS.2020.2965651.

[14] H. Seo, C. Huang, M. Bassenne, R. Xiao, and L. Xing, ''Modified U-Net (mU-Net) with incorporation of object-dependent high level features for improved liver and liver-tumor segmentation in CT images,'' IEEE Trans. Med. Imag., vol. 39, no. 5, pp. 1316–1325, May 2020.

[15] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, ''UNet++: Redesigning skip connections to exploit multiscale features in image segmentation,'' IEEE Trans. Med. Imag., vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[16] Y. M. Kassim et al., "Clustering-Based Dual Deep Learning Architecture for Detecting Red Blood Cells in Malaria Diagnostic Smears," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 5, pp. 1735-1746, May 2021, doi: 10.1109/JBHI.2020.3034863.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[18] S. M. Ibrahim, M. S. Ibrahim, M. Usman, I. Naseem and M. Moinuddin, "A Study on Heart Segmentation Using Deep Learning Algorithm for MRI Scans," *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 2019, pp. 1-5, doi: 10.1109/MACS48846.2019.9024793.

[19] N. Micallef, D. Seychell and C. J. Bajada, "Exploring the U-Net++ Model for Automatic Brain Tumor Segmentation," in IEEE Access, vol. 9, pp. 125523-125539, 2021, doi: 10.1109/ACCESS.2021.3111131.

[20] Y. An, N. Huang, X. Chen, F. Wu and J. Wang, "High-Risk Prediction of Cardiovascular Diseases via Attention-Based Deep Neural Networks," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 1093-1105, 1 May-June 2021, doi: 10.1109/TCBB.2019.2935059.

[21] C. P. Loizou, E. Kyriacou, M. B. Griffin, A. N. Nicolaides and C. S. Pattichis, "Association of Intima-Media Texture With Prevalence of Clinical Cardiovascular Disease," in *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 9, pp. 3017- 3026, Sept. 2021, doi: 10.1109/TUFFC.2021.3081137.

[22] J. Pedrosa, S. Queirs, O. Bernard, J. Engvall, T. Edvardsen, E. Nagel, and J. D'hooge, ''Fast and fully automatic left ventricular segmentation and tracking in echocardiography using shape-based b-spline explicit active surfaces,'' IEEE Trans. Med. Imag., vol. 36, no. 11, pp. 2287–2296, Nov. 2017

[23] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. De Marvao, T. Dawes, D. P. O'Regan, and B. Kainz, ''Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation,'' IEEE Trans. Med. Imag., vol. 37, no. 2, pp. 384–395, Feb. 20