APPLICATIONS OF EXPLAINABLE AI IN DISEASE DIAGNOSIS

A Report Submitted
in Partial Fulfilment for the Degree
of
BACHELOR OF TECHNOLOGY

in
Department of Computer Science and Engineering

by

Pranjal Singh Katiyar
(CSB20046)

Ramakrishnananda
(CSB20048)

Under the Supervision of
Dr. Rosy Sarmah
Associate Professor
Department of Computer Science and Engineering, Tezpur University

School of Engineering
Department of Computer Science and Engineering
Tezpur University
Napaam – 784028, Assam, India

June, 2024

Department of Computer Science and Engineering

Tezpur University

Napaam – 784028, Assam, India

DECLARATION

We hereby declare that the work presented in this report entitled "APPLICATIONS OF EXPLAINABLE AI IN DISEASE DIAGNOSIS", is carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not our original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Place:	Pranjal Singh Katiyar	Ramakrishnananda
Date:	CSB20046	CSB20048
	Signature:	Signature:

Department of Computer Science and Engineering

Tezpur University

Napaam – 784028, Assam, India

CERTIFICATE

This is to certify that the dissertation entitled "**Applications of Explainable AI in Disease Diagnosis**" submitted by **Pranjal Singh Katiyar** bearing enrolment number **CSB20046** and **Ramakrishnananda** bearing enrolment number **CSB20048** have carried out their work under my supervision and guidance for partial fulfilment of the requirements and the regulations for the award of the degree of Bachelor of Technology in Computer Science & Engineering during the session 2020 – 2024 at Tezpur University. To the best of my knowledge, the matter embodied in the dissertation has not been submitted to any other university/institute for the award of any Degree or Diploma.

Dr. Rosy Sarmah

Associate Professor

Place: Department of Computer Science & Engineering

Date: Tezpur University

Department of Computer Science and Engineering

Tezpur University

Napaam – 784028, Assam, India

CERTIFICATE

This is to certify that the dissertation entitled "**Applications of Explainable AI in Disease Diagnosis**" is submitted by **Pranjal Singh Katiyar** bearing enrolment number **CSB20046**, and **Ramakrishnananda** bearing enrolment number **CSB20048**. They have completed their project work successfully as needed for partial fulfilment of the requirements and the regulations for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the session 2020-2024 at Tezpur University. To the best of my knowledge, the matter embodied in the dissertation has not been submitted to any other university/institute for the award of any Degree or Diploma.

Head of the Department

Department of Computer Science & Engineering

Date**:** Tezpur University

Department of Computer Science and Engineering

Tezpur University

Napaam – 784028, Assam, India

Certificate by the Examiner

This is to certify that the report entitled "**Applications of Explainable AI in Disease Diagnosis**" submitted by **Pranjal Singh Katiyar** bearing enrolment number **CSB20046**, and **Ramakrishnananda** bearing enrolment number **CSB20048** in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering during the session 2020-2024 at Tezpur University has been examined by me and is found satisfactory for the award of the degree.

This approval does not necessarily endorse or accept every statement made, opinion expressed, or conclusion drawn as recorded in the report. It only signifies the acceptance of this report for the purpose for which it is submitted.

(Examiner)

Place:

Date**:**

PLAGIARISM UNDERTAKING

We hereby declare that the project work presented in the dissertation report titled "Applications of Explainable AI in Disease Diagnosis" is solely our work with no significant contribution from any other person/report. Any small contributions or help received have been duly acknowledged. The complete report has been written by us.

We understand the zero-tolerance policy of Tezpur University towards plagiarism. Therefore as the authors of the above-titled thesis, we declare that no portion of our report has been plagiarized. Any material used as a reference is properly cited.

We undertake that if we are found guilty of any formal plagiarism in the above-titled report even after the award of our B.Tech (CSE) degree Tezpur University reserves the right to revoke our B.Tech (CSE) degree.

Student Signature:

Name: Pranjal Singh Katiyar

Register Number: CSB20046

Student Signature:

Name: Ramakrishnananda

Register Number: CSB20048

ACKNOWLEDGEMENTS

We take this opportunity to extend our heartfelt gratitude to Dr. Rosy Sarmah, Associate Professor, Dept. of Computer Science and Engineering, Tezpur University, for giving us the opportunity to carry out our work under her and providing us ample guidance and support through the course of the project.

We would also like to thank the Head of the Department, Dept. of Computer Science and Engineering, and all the faculty members and Staff of the dept. of CSE, Tezpur University for the valuable guidance and cooperation throughout the project.

Our sincere thanks and appreciation go to our friends and family members who have directly or indirectly helped us out with their abilities.

(Signature) (Signature)

Pranjal Singh Katiyar Ramakrishnananda

ABSTRACT

In the realm of medical imaging, convolutional neural networks (CNNs) have shown exceptional performance in tasks such as disease detection, localization, and segmentation. However, their "black-box" nature impedes clinical trust and reliability, as the rationale behind their predictions remains opaque. To address this, our project, "Applications of Explainable AI in Disease Diagnosis," integrates Gradient-weighted Class Activation Mapping (Grad-CAM) into deep learning models to enhance interpretability and trustworthiness. We applied U-Net, MultiResU-Net, DCU-Net, and our novel VU-Net model across three datasets—Heart CT scans, Breast Ultrasound images, and the LiTS17 dataset for multiclass segmentation. Each model was evaluated using Binary Cross Entropy, Focal Loss, and Dice Loss functions. Our experiments revealed that MultiResU-Net generally excelled in segmentation tasks, with VU-Net showing promising results in the LiTS17 dataset. Grad-CAM was implemented to generate heatmaps, offering visual explanations of the model's focus areas in medical images. These visualizations were crucial in understanding and validating the models' decision-making processes, thereby enhancing transparency and clinical trust. The heatmaps confirmed the models' effectiveness in accurately targeting relevant features, aiding clinicians in making more informed decisions. Our comprehensive application of multiple models, combined with advanced visualization tools like Grad-CAM, underscores the potential of explainable AI in medical imaging. This approach bridges the gap between high computational accuracy and clinical interpretability, fostering greater confidence and reliability in AI-driven diagnostic tools.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviation	Description
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence
XAI	Explainable Artificial Intelligence
CAM	Class Activation Map
Grad-CAM	Gradient Class Activation Map
BFL	Binary Focal Loss
BCE	Binary Cross Entropy
PNG	Portable Network Graphics
NII	Neuroimaging Informatics Technology Initiative

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Deep learning, a powerful subset of machine learning, has emerged as a transformative force in the realm of artificial intelligence. At its core, deep learning revolves around the concept of neural networks, inspired by the intricate structure and functioning of the human brain. Unlike traditional machine learning algorithms, deep learning excels at handling vast amounts of unstructured data by automatically learning hierarchical representations through multiple layers of neural networks.

1.1 Why Deep Learning

The necessity for deep learning arose from the limitations of conventional algorithms in addressing complex, high-dimensional data. Tasks such as image and speech recognition, natural language processing, and medical image analysis demanded a more sophisticated approach. Earlier, the various general machine learning techniques existed which were not using neural networks for their model, they were required to be trained on large amount of data with varieties of features which is gathered by human experts having domain knowledge in the particular field (Manual feature extraction) for getting the more accurate and precise result but with advent of Neural Networks, the Deep learning's ability to automatically extract intricate features from raw data became a driving force behind its adoption, the tedious process of the picking up the various low level features became a task to be done by machine itself.

1.2 History

The foundation of Neural Networks was laid back in 1943 by (McCulloch & Pitts, 1943). They led the foundational principles of artificial neurons. Inspired by the binary logic of the nervous system, they presented a mathematical model of neurons as binary switches. This model, known as the McCulloch-Pitts neuron (MCP), demonstrated how networks of such artificial neurons could perform logical computations by encoding some logical proposition in them but in a very basic and simplified way . A group of MCP neurons when connected form an Artificial Neural Network. Their main aim was to simulate the neural activity in brain in computers. A breakthrough came in the year 1957 when Frank Rosenblatt published a seminal paper (Rosenblatt, 1958) on the Perceptron (a unit of Neural Network), it was inspired by the functioning of biological neurons, Rosenblatt's perceptron aimed to replicate learning and decision-making processes. The perceptron, a single-layer neural network, demonstrated the ability to learn binary classifications through iterative adjustments of weights. While limited to linearly separable problems, this work laid the foundation for future neural network development. The paper described perceptron as output value of a function with y as its output for a given input vector X=(x1,x2,. . . ,xn) and corresponding weight vector W=(w1,w2,. . . ,wn) for whom we compute a weighted sum, add a bias b to it and put it in a non-linear function.

Mathematically it is computed as follows:

$$Y = g\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{1.1}$$

x_i represents the input features individually.
w_i represents the weights associated with each input.Every piece of data is not equally important, thus weights allow us to decide that. During training, they are changed to allow the neural network to learn from data and adapt to its behavior.
b is the bias term, increasing the flexibility of the model. This enables it to better fit the training data and prevents non-zero outputs.

Later, in the paper (Rumelhart et al., 1986) in 1986 they talked how to provide a non-linear nature to the perceptrons by introducing the concept of activation functions.

Mathematically it is computed as follows:

$$y = g\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{1.2}$$

g is a non-linear function called as activation function.

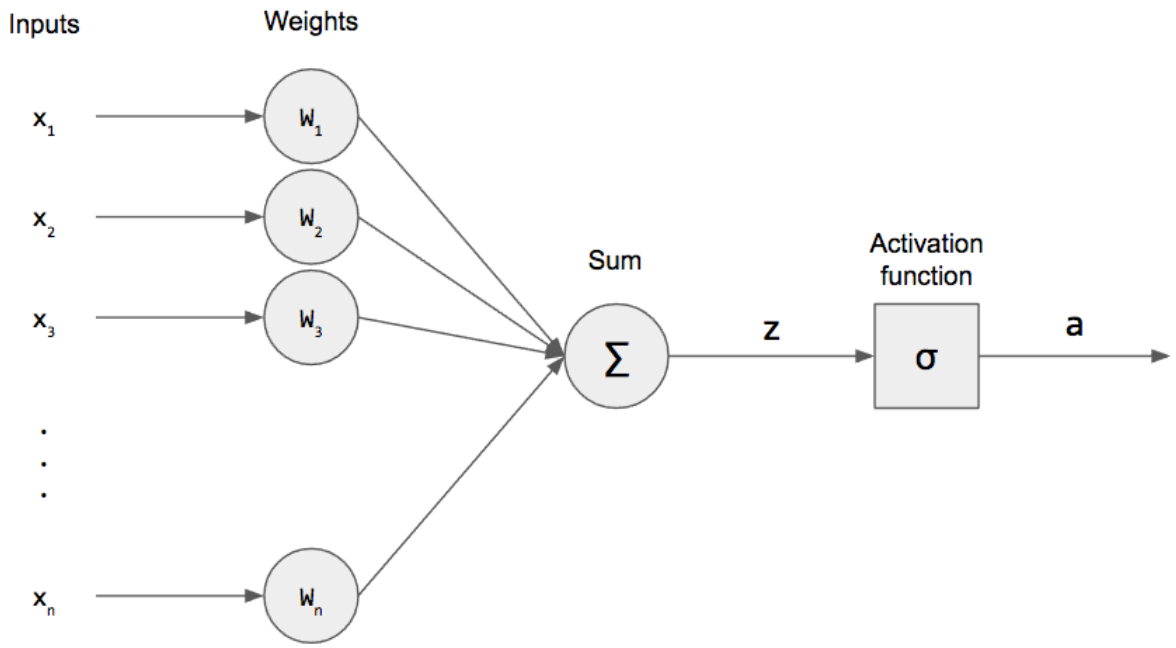


Figure 1.1: Perceptron

thus, finally we have an equation using vector notation,

$$Y = g(X \cdot W^T + b) \tag{1.3}$$

where,
X is the input vector or matrix.
W is the weight matrix.

1.3 Activation functions

According to Pedro Domingos and Michael Nielsen (Domingos, 2015; Nielsen, 2015), activation function is used to introduce non-linearity to the neural network. Without non-linear activation functions, the entire network would behave like a linear model, making it limited in its capacity to learn and represent complex relationships in data. Non-linear activation functions allow neural networks to model and represent intricate patterns and features in the data. This is particularly important in tasks where the relationship between inputs and outputs is non-linear, such as image and speech recognition. Activation functions also constrain the output range of a neuron, ensuring that the output is within a certain range. This can be important for numerical stability and preventing exploding or vanishing gradients during the training process.

Various kinds of activation functions used rigorously and frequently are:

1. **Sigmoid:** Also known as the logistic function, it is a commonly used activation function in artificial neural networks. It transforms input values into a smooth S-shaped curve, mapping them to a range between 0 and 1. One of the main advantages of the sigmoid function is its differentiability, making it suitable for gradient-based optimization during the training of neural networks. The sigmoid function is described deeply in (**Rumelhart86**) in 1986.

$$\text{The function is defined as f(x)} = \frac{1}{1+e^{-x}} \tag{1.4}$$

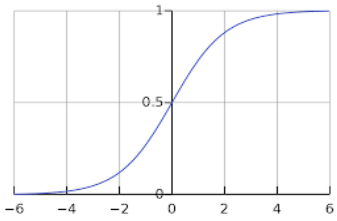


Figure 1.2: Sigmoid function

2. **ReLU function:** The Rectified Linear Unit (ReLU) is the most widely used activation function in neural networks. outputs the input directly if it is positive; otherwise, it outputs zero. ReLU introduces non-linearity to the network and is computationally efficient, contributing to faster training times. It has been particularly successful in image processing tasks and deep learning architectures due to its simplicity and ability to mitigate the vanishing gradient problem. The brilliant work of ReLU was in (Nair & Hinton, 2010).

$$\text{The function is defined as f(x)} = \max(0, x) \tag{1.5}$$

3. **Softmax Function:** The softmax function is a mathematical operation commonly used in machine learning for multiclass classification problems. It transforms a vector of real numbers into a probability distribution by exponentiating each element and normalizing the results. It is valued for its ability to convert raw scores into a probability distribution, aiding in decision-making and classification tasks. (LeCun et al., 1998) were the ones to

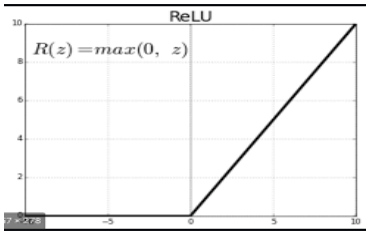


Figure 1.3: ReLU function

successfully define and use the function.

$$\text{The function is defined as } P(y = i|z) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{1.6}$$

Where,

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_n), \quad \text{where} \quad \sigma_i = \omega_i \cdot z_i \tag{1.7}$$

$$Z_i, \text{raw score or logit associated with class } i \tag{1.8}$$

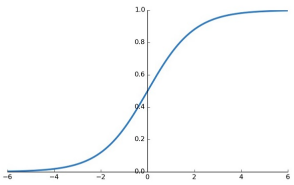


Figure 1.4: Softmax function

1.4 Learning in Neural Networks

According to (Bishop, 2006a) and (I. Goodfellow et al., 2016a), learning is a fundamental and crucial aspect of neural networks because it enables these systems to adapt and improve their performance on tasks through experience. Neural networks, inspired by the structure and function of the human brain, consist of interconnected nodes (neurons) organized in layers. Learning in neural networks involves adjusting the weights and biases of these connections to improve the network's ability to make accurate predictions or classifications.

The key reasons why learning is important in neural networks:

- **Adaptability to Data:** Neural networks are designed to learn from data. Learning allows them to capture complex patterns, relationships, and representations within the input data, making them adaptable to various tasks.
- **Generalization:** Learning enables neural networks to generalize from the training data to make accurate predictions on unseen or new data. Generalization is crucial for the model's practical utility and performance.
- **Optimization:** Through learning, neural networks optimize their parameters (weights and biases) to minimize a predefined objective function, often a loss function. This optimization process aims to improve the model's ability to make correct predictions.

- **Feature Extraction:** Neural networks learn to automatically extract relevant features from raw input data. The learning process allows the model to identify and emphasize informative patterns and characteristics in the data.
- **Non-linear Mapping:** Learning enables neural networks to perform non-linear mappings, allowing them to capture and represent complex relationships that may exist in the data.

Learning in neural networks primarily happens through the process of backpropagation, and the procedure can be explained with a simple number of steps:

1. **Forward Pass:** During the forward pass, input data is fed through the neural network, and predictions are made. Each layer's output is computed based on the weighted sum of inputs passed through an activation function.
2. **Compute Loss:** The output of the neural network is compared to the actual target values, and a loss (or error) is computed. The loss represents how far off the predictions are from the actual values or the ground truth. Few examples of famous and commonly used Loss functions are mean squared error loss, cross-entropy loss, focal loss, etc.
3. **Backward Pass (Backpropagation):** The backward pass involves computing the gradient of the loss with respect to the weights of the network. This is done using the chain rule of calculus. The gradients are propagated backward through the network, and for each weight, the algorithm computes how much it contributed to the overall error.
4. **Update Weights:** The weights of the network are then updated in the direction that reduces the error. This is typically done using an optimization algorithm such as gradient descent or one of its variants.
5. **Repeat:** Steps 1 to 4 are repeated iteratively until the model reaches a point where the loss is minimized, and the network has learned the underlying patterns in the training data.

The above process explained used the infamous Gradient descent algorithm though the algorithm was not mentioned in the paper by (Rumelhart et al., 1986) for backpropagation, the term and technique later was evolved, developed, and refined over time by multiple researchers. However, one of the early works that discussed the gradient descent method and its application to optimization problems is by (Bryson & Ho, 1969).Thus it is described as an optimization algorithm commonly used to minimize the cost or loss function in machine learning and optimization problems. The basic idea behind Gradient Descent is to iteratively move towards the minimum of a function by adjusting the parameters of the function based on the negative of the gradient.

1.4.1 Forward Pass

$\mathbf{X}$: Input Vector (features)
$\mathbf{W}^{(L)}$: Weights matrix for layer l
$\mathbf{b}^{(L)}$: Bias vector for layer l
$\mathbf{Z}^{(L)}$: Weighted Sum of inputs for layer l
$\mathbf{b}^{(L)}$: Activation for layer l
$\mathbf{Y}$: Actual output (target)
$\hat{\mathbf{Y}}$: Predicted output

1. Input Layer:

$$\mathbf{A}^{(1)} = \mathbf{X} \tag{1.9}$$

2. Hidden Layer:

$$\mathbf{Z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{A}^{(l-1)} + \mathbf{b}^{(l)} \tag{1.10}$$

$$\mathbf{A}^{(l)} = \sigma(\mathbf{Z}^{(l)}) \tag{1.11}$$

3. Output Layer:

$$\mathbf{Z}^{(L)} = \mathbf{W}^{(L)} \cdot \mathbf{A}^{(L-1)} + \mathbf{b}^{(L)} \tag{1.12}$$

$$\hat{\mathbf{Y}} = \sigma(\mathbf{Z}^{(L)}) \tag{1.13}$$

1.4.2

sectionLoss Computation where suitable loss function $\mathbf{J}(\hat{\mathbf{Y}}, \mathbf{Y})$ to compute the error.

1.4.3 Backpropagation

Backpropagation, short for "backward propagation of errors," as described in (Werbos, 1974) and (Rumelhart et al., 1986) is a supervised learning algorithm used to train neural networks. It involves the optimization of the network's weights and biases to minimize a predefined loss function, representing the difference between predicted and actual outputs. The process consists of a forward pass, computation of the loss, and a backward pass where gradients are propagated through the network for weight updates.

1. Output Layer:

$$\delta^{(L)} = \frac{\partial J}{\partial \mathbf{Z}^{(L)}} \tag{1.14}$$

$$\frac{\partial J}{\partial \mathbf{W}^{(L)}} = \delta^{(L)} \cdot (\mathbf{A}^{(L-1)})^T \tag{1.15}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(L)}} = \delta^{(L)} \tag{1.16}$$

2. Hidden Layer:

$$\delta^{(l)} = (\mathbf{W}^{(l+1)})^T \cdot \delta^{(l+1)} \cdot \sigma'(\mathbf{Z}^{(l)}) \tag{1.17}$$

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}} = \delta^{(l)} \cdot (\mathbf{A}^{(l-1)})^T \tag{1.18}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(l)}} = \delta^{(l)} \tag{1.19}$$

3. Weight Update:

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \alpha \frac{\partial J}{\partial \mathbf{W}^{(l)}} \tag{1.20}$$

$$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \alpha \frac{\partial J}{\partial \mathbf{b}^{(l)}} \tag{1.21}$$

Error ($\delta^{(l)}$): The derivative of the loss with respect to the weighted sum ($\mathbf{Z}^{(l)}$).
Gradients ($\frac{\partial J}{\partial \mathbf{W}^{(l)}}$ and $\frac{\partial J}{\partial \mathbf{b}^{(l)}}$): The derivatives of the loss with respect to weights and biases.

Learning parameter (I. Goodfellow et al., 2016a) alpha a hyper parameter which is difficult to decide on as if taken too low then too much time will be taken to reach the global minima or stuck at local minima, if too high value then we overshoot and diverge from solution thus a moderate value required.

1.5 Types of Deep Learning

The different types of Deep Learning are as follows:

- **Feedforward Neural Networks (FNN):** Basic neural networks where information moves in one direction—from input to output.
- **Convolutional Neural Networks (CNN):** Effective for image-related tasks, with convolutional layers to capture spatial hierarchies.
- **Recurrent Neural Networks (RNN):** Suited for sequential data, they maintain a memory of previous inputs.
- **Generative Adversarial Networks (GAN):** Comprising a generator and a discriminator, used for generating new data instances.
- **Long Short-Term Memory Networks (LSTM):** A type of RNN with improved memory capabilities, beneficial for sequence tasks.

1.6 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) (LeCun et al., 1989) have emerged as a revolutionary force in the field of deep learning, particularly in image processing tasks. This Neural Network explores the fundamental concepts and workings of CNNs, shedding light on how they have become the backbone of various computer vision applications. Images, with their rich and diverse content, encapsulate a wealth of information, from the subtle nuances of facial expressions to the intricate structures of celestial bodies. Harnessing the power of deep learning, researchers and practitioners have unlocked new avenues in image recognition, classification, and generation. Through convolutional neural networks (CNNs), deep learning algorithms can parse through massive image datasets with remarkable efficiency, extracting meaningful features and patterns with unparalleled accuracy.

In the 1950s and 1960s, the groundbreaking work of (Hubel & Wiesel, 1962) revealed that neurons in the cat visual cortices respond to specific regions of the visual field, forming receptive fields. These receptive fields (Hubel & Wiesel, 1977) exhibit systematic variations in size and location across the cortex, creating a comprehensive map of visual space. The researchers

identified two primary types of visual cells in a 1968 paper: simple cells, which respond maximally to specific orientations of straight edges within their receptive fields, and complex cells, with larger receptive fields and insensitivity to edge position. In 1980, inspired by Hubel and Wiesel's work, Kunihiko Fukushima introduced the "neocognitron," a neural network model that incorporated convolutional and downsampling layers (Fukushima, 1980, 1988).. Convolutional layers featured units with receptive fields covering patches of the previous layer, while downsampling layers aided in object classification by computing the average of unit activations within their patches, contributing to the correct identification of objects even when shifted in visual scenes. The papers further explained that at the core of CNNs lies as a specialized architecture designed for image-related tasks. Unlike traditional neural networks, CNNs leverage convolutional layers, pooling layers, and fully connected layers. Convolutional layers use filters to extract hierarchical features from the input image, capturing spatial hierarchies and patterns.

1.6.1 Convolution Layers

Convolutional layers form the crux of CNNs. These layers consist of filters or kernels that convolve across the input image, extracting features by performing element-wise multiplications and summations. The output, often referred to as feature maps, captures different aspects of the input, allowing the network to learn hierarchical representations. The convolution operation can be explained as a fundamental concept in signal processing and is especially crucial in the context of convolutional neural networks (CNNs) in deep learning. It involves combining two functions to produce a third, representing how one function modifies the shape of another. In image processing, convolution is used to extract features and patterns from images.

- **Local Feature Detection:** The convolution operation is used for local feature detection. Imagine sliding a small filter (also called a kernel) over an input signal or image. At each position, the filter captures the local information in the input. This process allows the network to recognize specific patterns, such as edges, corners, or textures.
- **Translation Invariance:** One key advantage of convolution is its translation-invariant property. As the filter slides across the input, it detects patterns regardless of their exact location. This makes convolutional layers effective in capturing hierarchical features in images.

1.6.2 Mathematical Explanation

Let's denote the input signal or image as I and the filter or kernel as K. The convolution operation, often denoted by *, is defined as follows for discrete signals:

$$(I * K)(x, y) = \sum_i \sum_j I(i, j) \cdot K(x - i, y - j) \tag{1.22}$$

$(I(i, j))$ represents the intensity or value at position (i, j) in the input signal.

$K(x - i, y - j)$ represents the filter's value at position $(x - i, y - j)$.

The summation is over all possible positions (i, j) where the filter overlaps with the input.

This operation involves element-wise multiplication of the filter values with the corresponding input values at each position, followed by summation. The result is a new signal that represents the presence of specific features in the input.

1.7 Pooling

Pooling is a downsampling operation commonly used in convolutional neural networks (CNNs) to reduce the spatial dimensions of the feature maps and, consequently, the computational complexity of the network. The pooling operation involves selecting a representative value from a group of neighbouring pixels, typically through operations like max pooling or average pooling. This downsampling helps to retain essential information while discarding less relevant details, making the network more computationally efficient.

1.7.1 Max Pooling:

The primary idea behind max pooling is to select the maximum value from a group of neighbouring pixels, known as the pooling window. By retaining only the maximum value, max pooling helps capture the most prominent features within each region of the input, making the network more robust to variations in spatial positions.

1.7.1.1 Mathematical Explanation

Let's denote the input feature map as X, the size of the pooling window as $k \times k$, and the output feature map as Y. The max pooling operation can be mathematically expressed as:

$$Y_{i,j} = \max_{m,n} X_{i \cdot k+m, j \cdot k+n} \tag{1.23}$$

Here:

$Y_{i,j}$ is the output of the max pooling operation at position (i, j) in the downsampled feature map.

$X_{i \cdot k+m, j \cdot k+n}$ represents the values in the pooling window at position $(i \cdot k + m, j \cdot k + n)$ in the input feature map.

The max pooling operation $\max_{m,n}$ selects the maximum value within the pooling window.

Max pooling is applied independently to different non-overlapping regions of the input feature map, resulting in a downsampled representation. The value of k determines the size of the pooling window, and common choices include 2×2 or 3×3.

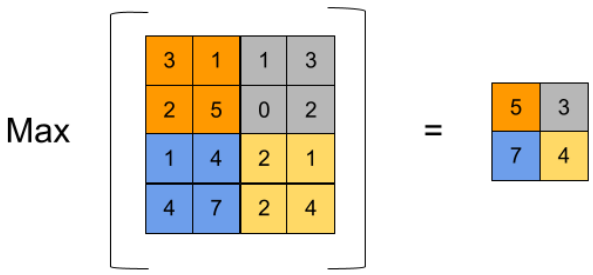


Figure 1.5: a Convolutional Neural Network

1.8 Tasks that can be performed with help of CNN

- **Regression:** Regression (Galton, 1886) models the relationship between dependent and independent variables, predicting continuous outcomes. In machine learning, it uses labeled data to train algorithms, facilitating predictions based on learned patterns. Helps to decide continuous line fitting the pattern.

- **Classification:** Classification (Bishop, 2006a) is a machine learning method that categorizes input data into predefined classes or labels. Trained on labeled datasets, classification algorithms learn patterns to assign new instances to specific categories, making it valuable for tasks such as image recognition, spam filtering, and disease diagnosis.

- **Segmentation:** In computer vision, it (Szeliski, 2010) involves partitioning an image into distinct segments or regions based on specific criteria. It is crucial for tasks like object detection and scene understanding. Segmentation helps identify and delineate individual objects or structures within an image, contributing to various applications in image processing.

1.9 Segmentation

Segmentation in computer vision involves partitioning an image into distinct regions based on specific criteria. This process aims to identify and delineate individual objects or structures within an image, facilitating tasks like object recognition, medical imaging analysis, and scene understanding. In medical images, segmentation is a critical process in extracting meaningful information from medical images. This technique involves dividing an image into distinct, semantically meaningful regions, aiding in the identification and delineation of structures such as organs, tumours, or abnormalities. Accurate segmentation is paramount for diagnosis, treatment planning, and monitoring disease progression. It plays a pivotal role in medical fields like radiology, oncology, and neurology, enabling precise analysis of anatomical structures and pathological conditions. Advances in deep learning, particularly convolutional neural networks, have significantly enhanced the accuracy and efficiency of medical image segmentation, fostering breakthroughs in diagnostic imaging and personalized medicine. Some of the early works can be found in (Boykov & Jolly, 2001; Haralick & Shapiro, 1985; Kass et al., 1988; Ronneberger et al., 2015a).

1.9.1 Type of Segmentation

There are several types of image segmentation techniques, each catering to specific requirements in computer vision and image analysis but they can be broadly classified to following categories (Zaitoun & Aqel, 2015) :

1. **Semantic Segmentation:** Semantic segmentation is a computer vision technique that involves categorizing each pixel in an image into distinct classes or semantic categories, providing a detailed understanding of the visual scene. Unlike other segmentation methods, semantic segmentation goes beyond simple object detection and assigns labels to each pixel, enabling a pixel-level analysis of the image.

   The paper introduced Fully Convolutional Networks (FCNs) for pixel-wise semantic segmentation (Long et al., 2015). By replacing fully connected layers with convolutional layers, the model generated dense predictions for arbitrary-sized input images. Skip architectures were employed to preserve fine details, demonstrating efficacy on the PASCAL VOC dataset.

2. **Instance Segmentation:** Instance segmentation is a sophisticated computer vision technique designed to identify and differentiate individual instances of objects within an image. Unlike semantic segmentation, which classifies each pixel into predefined categories, instance segmentation takes it a step further by assigning a unique label to each

distinct object instance. This method is particularly valuable in scenarios where multiple objects of the same class coexist, allowing for precise delineation and recognition of each separate entity.

A 2017 ICCV paper (He et al., 2017) , revolutionized instance segmentation. Extending Faster R-CNN, it introduced ROI Align to enhance feature alignment, enabling parallel prediction of class labels, bounding boxes, and segmentation masks. With a multi-task loss function, it achieved state-of-the-art results on the COCO dataset, advancing instance segmentation accuracy.

In this work deep learning techniques has been used for segmenting over 3 different medical image datasets, viz. CT Heart dataset ("CT Heart Segmentation dataset", 2021), Breast Ultrasound Images Dataset ("Breast Ultrasound Images dataset", 2020), LiTS ("LiTS – Liver Tumor Segmentation Challenge (LiTS17) organized in conjunction with ISBI 2017 and MICCAI 2017", 2017).

The application of segmentation techniques in medical imaging holds immense promise, revolutionizing diagnostics, treatment planning, and disease monitoring. From identifying tumours in MRI (Magnetic Resonance Imaging) scans to delineating organs in CT (Computed Tomography) images, from Ultrasounds to X-Rays, deep learning-based segmentation empowers healthcare professionals with invaluable insights, facilitating timely interventions and personalized treatments. By automating tedious tasks and augmenting human expertise, medical image segmentation not only enhances diagnostic accuracy but also expedites workflows, ultimately improving patient outcomes. Furthermore, the benefits of leveraging deep learning for medical image segmentation extend beyond mere efficiency gains. It enables the development of predictive models, paving the way for early disease detection and prognostic assessments. The U-Net and it variations are effective for small-scale medical image data but struggles with 2D segmentation's limited inter-slice information.

1.10 Problems for using deep learning in medical field

1.10.1 Clinical Acceptance and Trust

Clinical acceptance and trust in machine learning models are crucial for their integration into healthcare settings. Trust develops when clinicians feel confident that the model provides accurate, reliable, and understandable information. However, building this trust is challenging due to the complexity and unpredictability of AI systems. Effective communication about how models work, validation against clinical outcomes, and transparent reporting of model limitations are necessary. Training programs that educate healthcare providers on the strengths and limitations of AI can help in bridging the knowledge gap, thereby enhancing trust and facilitating smoother adoption into clinical practice.

1.10.2 Regulatory Compliance

Regulatory compliance for AI-driven tools in healthcare involves meeting strict guidelines set by bodies such as the FDA in the United States or the EMA in Europe. These regulations ensure that medical devices are safe and effective for patient care. AI models, due to their inherent complexity, must be rigorously validated for accuracy, reliability, and the ability to generalize across different populations and conditions. Compliance becomes particularly challenging with models that continuously learn and adapt, as these changes can affect the initial approval parameters. Developers must, therefore, implement robust monitoring and updating mechanisms that adhere to regulatory standards throughout the lifecycle of the AI product.

1.10.3 Error Analysis and Model Improvement

Error analysis in AI models helps identify the reasons behind incorrect predictions and is essential for model improvement. In the medical field, where errors can have significant consequences, understanding these missteps is crucial for patient safety. This process involves dissecting false positives and false negatives to adjust model parameters or training data accordingly. Improvements might also include enhancing data quality, increasing dataset diversity, or modifying model architectures. Regular audits and updates based on error analysis not only improve model accuracy but also build user trust by demonstrating commitment to continual improvement.

1.10.4 Data Quality and Availability

One of the most significant challenges is the availability and quality of medical data. Deep learning models require large datasets to perform effectively. However, medical data is often scarce, fragmented, and not standardized. Privacy laws and regulations, further complicate data sharing and access. Moreover, medical records often contain inconsistencies and errors, which can negatively impact the training and performance of deep learning models.

1.10.5 Data Labeling and Annotation

For supervised deep learning models, labeled data is essential. However, labeling medical data is a time-consuming and costly process, requiring expert knowledge from healthcare professionals. Annotating medical images, for instance, demands meticulous work by radiologists or other specialists. This labor-intensive process can limit the volume of labeled data available for training models, affecting their accuracy and generalizability.

1.11 Explainable AI

Explainable AI (XAI) refers to methods and techniques in the field of artificial intelligence (AI) that make the outputs and operations of machine learning algorithms understandable to humans (Doshi-Velez & Kim, 2017; Gilpin et al., 2018). As AI technologies increasingly permeate various aspects of daily life, especially in critical sectors like healthcare, finance, and legal systems, the need for transparency and understandability in AI decisions has become paramount (Ribeiro et al., 2016). The primary goal of XAI is to create a suite of machine learning techniques that produce more explainable models while maintaining a high level of accuracy, and to develop methods that enable human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners (Samek et al., 2017).

This transparency is crucial for several reasons:

- **Trust and Adoption:** For AI systems to be integrated successfully into high-stakes domains, users and stakeholders must trust that the AI systems are making decisions based on sound reasoning. Explainability fosters this trust by making the decision-making process accessible and understandable.
- **Regulatory and Ethical Compliance:** In many industries, regulations require that decisions made by automated systems be explainable to ensure fairness and accountability. XAI helps meet these regulatory requirements by elucidating how decisions are derived.

- **Debugging and Improvement:** Explainable models allow developers and users to understand the model's behavior deeply. This understanding is crucial for identifying and correcting errors, refining model parameters, and ultimately improving the performance of AI systems.

- **Mitigating Bias:** AI systems can inadvertently become biased, reflecting or amplifying existing biases in training data. Explainable AI helps identify and mitigate these biases by making it easier to see when and how biased decisions are being made.

Explainable AI is not just about making algorithms transparent but also about making them more aligned with human values and ethical principles. As AI becomes more prevalent, ensuring that these systems can be scrutinized, questioned, and adjusted is vital for their sustainable and beneficial integration into society.

1.12 Grad-CAM in Medical Imaging

Grad-CAM, or Gradient-weighted Class Activation Mapping, is a visualization technique designed to improve the interpretability of convolutional neural networks (CNNs) used in medical imaging (Selvaraju et al., 2017). This method is particularly beneficial in the medical field as it provides visual explanations of the decisions made by deep learning models when analyzing medical images, such as X-rays, CT scans, Ultrasound or MRI images. Understanding where the model is focusing its attention can help clinicians and researchers validate the AI-driven analysis and ensure that the diagnoses are based on relevant features in the images (Litjens et al., 2017). Grad-CAM works by generating heatmaps that highlight the regions of an image most influential to the model's predictions. This is achieved by capturing the gradients flowing into the last convolutional layers of the network concerning a particular class of interest—such as a type of tumor or anomaly—and using these gradients to produce a coarse localization map. This map effectively shows which areas of the image were pivotal in leading the model to its conclusion, providing a visual guide that aligns the model's 'thinking' with human interpretation (Gilpin et al., 2018).

In the context of medical imaging, the applications of Grad-CAM are profound:

- **Diagnostic Accuracy:** By revealing which features in an image are considered significant by the model, Grad-CAM helps clinicians ensure that the AI's decision-making process aligns with clinical knowledge and standards. This reassurance is critical in sensitive medical scenarios where the stakes are high.

- **Training and Validation:** Grad-CAM assists in the training phase of model development by allowing engineers and medical professionals to verify that the model focuses on appropriate anatomical markers or pathological features. This verification helps in tuning the models more effectively, ensuring they are learning from clinically relevant features rather than irrelevant artifacts.

- **Enhanced Communication:** Grad-CAM visualizations can serve as a communication tool between AI developers and medical practitioners. They provide a straightforward method for discussing how algorithms arrive at their conclusions, which is invaluable for multidisciplinary teams working on medical imaging projects.

- **Trust and Ethical Use:** In medical practice, where patient outcomes directly depend on the accuracy and reliability of diagnostic tools, Grad-CAM enhances trust in AI technologies. By making AI decisions transparent and understandable, it supports the ethical use of AI in healthcare, ensuring that technologies are used responsibly and effectively.

Overall, Grad-CAM represents a bridge between the complex computations of deep learning models and the practical needs of medical diagnostics, making it an indispensable tool in the development and deployment of AI applications in healthcare.

CHAPTER 2: LITERATURE REVIEW

Computer vision is a multidisciplinary field aimed at enabling computers to interpret visual data, akin to human visual perception. It harnesses artificial intelligence and machine learning to analyze images or videos, performing tasks such as object identification, face recognition, and anomaly detection. Medical Image Segmentation, a subset of computer vision, involves partitioning medical images into distinct segments for detailed analysis. Recent advances in deep learning, particularly convolutional neural networks, have revolutionized computer vision, simplifying tasks like medical image classification and segmentation, thus advancing medical applications. Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012) have revolutionized deep learning and computer vision, particularly in image processing tasks. They leverage convolutional layers, pooling layers, and fully connected layers to extract hierarchical features from input images, forming the backbone of various computer vision applications. Convolutional layers use filters to capture spatial hierarchies and patterns, performing element-wise multiplications and summations across the input image to generate feature maps. This convolution operation is fundamental in CNNs, extracting features and patterns crucial for image processing.

Medical image segmentation is vital for diagnostic and treatment planning, where accurate delineation of anatomical structures is essential. Convolutional Neural Networks (CNNs) have revolutionized this field by autonomously learning from raw pixel data, enhancing segmentation accuracy. They excel in discerning intricate patterns in medical images, enabling both semantic and instance-level segmentation tasks. CNN-based segmentation models have demonstrated exceptional performance in tumor detection (Ronneberger et al., 2015a), organ segmentation (Çiçek et al., 2016), and lesion localization (Dou et al., 2017), improving diagnostic accuracy and treatment planning (Litjens et al., 2017). Their adaptability to diverse imaging modalities and segmentation challenges has propelled innovations in personalized medicine (Esteva et al., 2019), ultimately benefiting patient care and clinical outcomes (Topol, 2019). A more detailed study on deep learning in medical images may be found in the survey (Litjens et al., 2017).

In our research, we selected the U-Net segmentation model (Ronneberger et al., 2015a) for its renowned accuracy and adaptability across diverse imaging datasets. Its distinctive encoder-decoder architecture effectively preserves spatial details, driving advancements in medical imaging and environmental analysis. Notably, U-Net's efficacy extends to brain tumor segmentation and cardiovascular image analysis (Havaei et al., 2017; Milletari et al., 2016a; Mortazi et al., 2017), enhancing disease detection and patient care. While U-Net excels with small-scale medical image datasets, challenges arise in 2D segmentation tasks due to limited inter-slice information. Recent work (Çiçek et al., 2016) introduces a novel strategy for 3D image segmentation, addressing some limitations, yet scalability remains an issue. To address these limitations, we explored enhancements to the U-Net model by implementing the MultiResU-Net and DCU-Net architectures. MultiResU-Net, introduced by (Ibtehaz & Rahman, 2020a), incorporates multi-resolution blocks within the U-Net framework to capture features at different scales more effectively. This enhancement improves the model's ability to handle varying

object sizes and complex structures in medical images. MultiResU-Net has shown significant improvements in segmentation accuracy for tasks such as retinal vessel segmentation and breast ultrasound image segmentation, where capturing fine details at multiple scales is crucial (Alom et al., 2018; Yap et al., 2010). DCU-Net, on the other hand, employs dense convolutional units and attention mechanisms to further refine segmentation outputs. The dense connections within DCU-Net promote feature reuse, reducing the number of parameters and enhancing the network's efficiency (Jégou et al., 2017). This model also integrates an attention mechanism that focuses on relevant features, improving the accuracy of segmentation tasks. Previous studies have demonstrated the effectiveness of DCU-Net in applications like liver tumor segmentation and lung nodule detection, highlighting its ability to handle complex medical imaging challenges (Jin et al., 2018; X. Li et al., 2018).

Our proposed model (VU-Net) is based on VNet (Milletari et al., 2016a) which comprises an encoder with multiple stages maintaining the same resolution and a decoder for gradual decompression, producing an output image of the original size. Inheriting U-Net's jump connection, VNet mitigates information loss during feature extraction. Additionally, it adopts the short-circuit connection mechanism from ResNet (He et al., 2016), adding input and output at each stage to learn the residual function. The authors in (Çiçek et al., 2016) demonstrate an application of VNet in medical imaging, specifically in 3D image segmentation utilizing a volumetric, fully convolutional neural network. The model is trained end-to-end on MRI volumes of the prostate, enabling it to predict segmentation for the entire volume simultaneously.

From our limited survey, we can understand that the integration of Convolutional Neural Networks (CNNs) into segmentation models represents a paradigm shift in image analysis, offering unprecedented capabilities for precise and efficient segmentation tasks. The versatility and adaptability of CNN architectures, coupled with the sophistication of segmentation models like U-Net, MultiResU-Net, DCU-Net and VNet have propelled advancements in medical imaging. By harnessing the power of CNNs, researchers and practitioners can achieve superior segmentation accuracy, enabling deeper insights and more informed decision-making. As we continue to explore and innovate in this field, the synergy between segmentation models and CNNs promises to unlock new possibilities and drive transformative changes in image analysis and beyond. While various methods exist, trained models may still not consistently produce desired outputs for all input data.

2.1 Explainable AI revealing BlackBoxes

Most deep learning-based segmentation methods operate as black boxes, lacking explanations for their predictions. Explainable deep learning techniques aim to clarify model decision-making through visualization and interpretation. These methodologies are often denoted as interpretable deep learning or explainable artificial intelligence (XAI) (Samek et al., 2019). In medical imaging, researchers are progressively adopting XAI techniques to elucidate algorithmic outcomes. An explanation is deemed effective if it provides understanding of the neural network's decision-making process or renders the decision comprehensible. A survey on XAI used in deep learning-based medical image analysis can be found in (Litjens et al., 2017). In the survey, XAI techniques used in medical image analysis have been categorized into three types: visual, textual, and example-based, out of which visual XAI are most popular. Gradient-weighted class activation mapping (Grad-CAM), a visual XAI, introduced by Selvaraju et al. (Selvaraju et al., 2017), extends the concept of CAM. Unlike CAM, Grad-CAM can be applied to any CNN architecture to generate post hoc local explanations without the need for global average pooling.

2.1.1 Class Activation Mapping

Introduced by (Zhou et al., 2016) in their seminal paper, CAM was developed to identify the regions of the input image that are important for predictions in CNNs equipped with global average pooling. This technique generates a heatmap for each class label considered by the final classification layer, highlighting important regions in the image for predicting these labels. CAM's simplicity and effectiveness have made it a foundational tool in visual explanations for image-based models CAM uses the global average pooling layers in CNNs to produce spatial heatmaps that highlight the discriminative regions used by the network to identify specific classes. This method, however, is often limited to specific network architectures that include global average pooling.

2.1.2 Work on Grad-CAM

In the study, (Gunashekar & Ramesh, 2019) employed an interpretable deep learning model to analyze a convolutional neural network's (CNN) predictions for segmenting prostate tumours. The CNN, based on a U-Net architecture trained on multi-parametric MRI data, automatically segments the prostate gland and tumours. To interpret the CNN's segmentation, the authors generated heatmaps using Grad-CAM (Selvaraju et al., 2017). The work by (Wang et al., 2019) utilized Grad-CAM to highlight the areas on brain MRI images that influenced the classifier's decision regarding tumour presence. In medical image segmentation, Grad-CAM serves to elucidate which regions of an input medical image contribute significantly to the segmentation outcome. By analyzing gradients flowing into the final convolutional layers of the segmentation network, Grad-CAM generates heatmaps pinpointing areas the network focuses on for segmentation decisions. In tumour segmentation from MRI or CT scans, Grad-CAM highlights regions indicative of tumour presence, aiding clinicians in understanding the model's segmentation decisions. The integration of Grad-CAM into medical imaging AI has shown promising results in enhancing model transparency. For instance, studies have applied Grad-CAM to visualize influential regions in brain tumor segmentation from MRI scans, liver and lung tumor segmentation from CT images, and breast cancer segmentation from mammography. These visualizations help radiologists and oncologists to validate the AI-assisted diagnostic decisions, providing a safety check for AI-driven recommendations.

Despite these advancements, the literature indicates ongoing challenges in the field. The variability in imaging techniques, differences in tumor morphology across patients, and inherent data imbalances present substantial hurdles in achieving generalizable and robust segmentation models. Further research is suggested to focus on incorporating domain-specific knowledge and clinician feedback directly into the training process, enhancing both the performance and the utility of visual explanation techniques (S. Li et al., 2020).

CHAPTER 3: METHODOLOGY

In this chapter, we delineate the methodological framework employed to conduct our empirical study on employing Grad-CAM in DL models for tumour segmentation and visual explanation in medical imaging. Our methodology encompasses the implementation of segmentation models, and the integration of Grad-CAM for visual interpretation of model predictions.

3.1 Motivavtion

In the realm of medical imaging, deep learning models, particularly convolutional neural networks (CNNs), have demonstrated remarkable efficacy in performing complex diagnostic tasks such as disease detection, localization, and segmentation. However, despite their high accuracy and efficiency, these models inherently lack transparency in their decision-making processes. This opacity poses a significant challenge, as it is crucial for medical practitioners to understand the reasoning behind a model's predictions to ensure accurate diagnoses and appropriate patient care.

The primary problem with deep learning models in medical applications is their "black-box" nature. These models process input data through multiple layers and complex computations, making it difficult to trace how decisions are formulated. This lack of interpretability can hinder trust and reliability, as healthcare providers may be reluctant to base clinical decisions on outputs whose rationale is unclear or unverifiable.

To address this critical issue, we propose the integration of Gradient-weighted Class Activation Mapping (Grad-CAM) into our deep learning models. Grad-CAM is an advanced visualization technique that generates heatmaps to highlight the areas in medical images that are most influential to the model's predictions. These heatmaps serve as a bridge between the high-level computations of the model and the practical, intuitive understanding required by clinicians. By implementing Grad-CAM, our goal is to:

- **Enhance Transparency:** Provide clear visual explanations of what features within the image are most important for the model's predictions, thereby demystifying the model's decision-making process.
- **Build Trust:** Improve the confidence of healthcare professionals in utilizing deep learning-based diagnostic tools by offering insights into the model's internal workings.
- **Facilitate Clinical Decision Making:** Assist medical practitioners in making more informed decisions by validating the AI-generated outputs against their clinical knowledge and experience.
- **Improve Model Reliability:** Enable the identification and correction of potential biases or errors in the model by analyzing the regions of interest highlighted by Grad-CAM, ensuring that the model attends to clinically relevant features rather than anomalies or noise.

In summary, by incorporating Grad-CAM into our deep learning models for medical imaging, we aim to not only maintain the high efficiency and accuracy of these models but also significantly enhance their interpretability and trustworthiness. This approach promises to advance the field of medical AI by merging cutting-edge machine learning technologies with the essential values of clinical transparency and patient-centered care.

3.2 Deep Learning (DL) Models for Medical image segmentation

We have used the following DL architectures: -

1. **U-Net**
2. **MultiResU-Net**
3. **DCU-Net for Image Segmentation**
4. **VU-Net (for Multiclass Segmentation)**

3.2.1 U-Net

The U-Net architecture was explained in 2015 paper (Ronneberger et al., 2015b). It is a convolutional neural network (CNN) designed for semantic segmentation tasks, particularly in the biomedical image analysis domain. Till date it is one of the best-known architectures for image segmentation

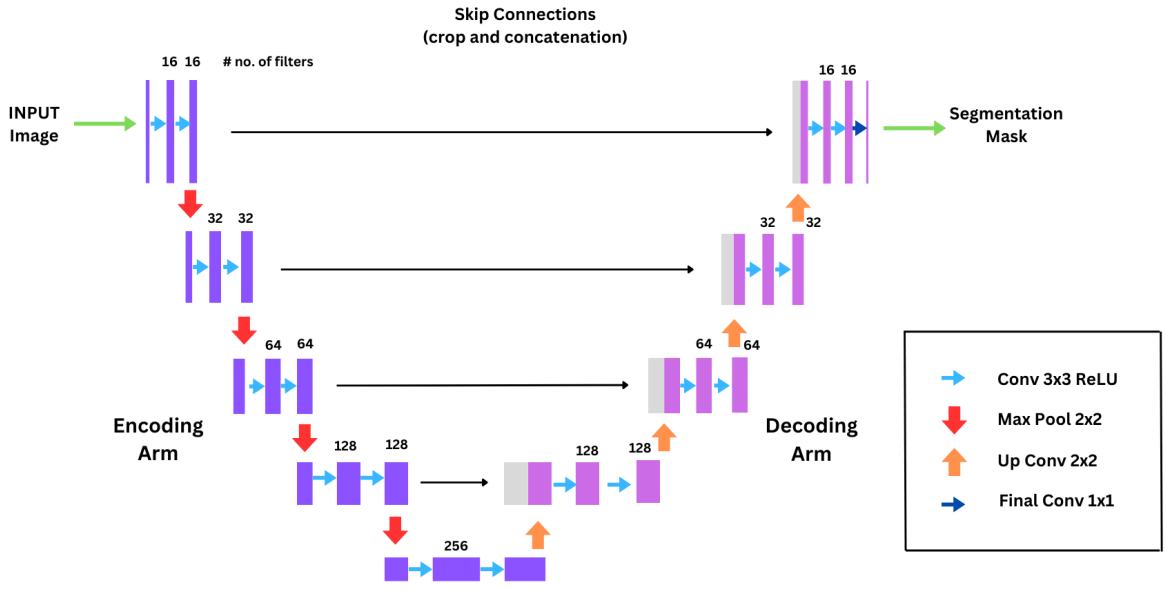


Figure 3.1: The UNet Architecture

- **Architecture Overview:**

1. **Encoder Path:** The architecture has a contracting or encoder path, which consists of convolutional blocks followed by max-pooling layers. Each block typically includes two 3x3 convolutional layers followed by batch normalization and rectified linear unit (ReLU) activation. Max-pooling layers reduce spatial dimensions.
2. **Bottleneck:** The contracting path leads to a bottleneck layer, which captures the most compressed and abstract representation of the input. It contains two 3x3 convolutions followed by batch normalization and ReLU.
3. **Decoder Path:** The decoder path is an expansive path that involves upsampling the feature maps. Each block in this path consists of two 3x3 convolutions, batch normalization, and ReLU activation. Upsampling is done through transposed convolutions.
4. **Skip Connections:** Skip connections are a key feature. Feature maps from the encoder path are concatenated with corresponding feature maps in the decoder path. These connections help preserve spatial information lost during downsampling.
5. **Final Layer:** The final layer typically involves a 1x1 convolutional layer with a sigmoid activation function for pixel-wise classification in binary semantic segmentation tasks or a softmax function for the multiclass semantic segmentation. The number of channels in this layer corresponds to the number of classes in the segmentation problem.

- **Input and Output:**
  - **Input:** U-Net is designed to handle images of arbitrary size. We have used the image size of 128x128.
  - **Output:** The output is a segmentation map where each pixel is assigned a class label. For binary segmentation, a single-channel output is used, while multi-class segmentation involves multiple channels, each corresponding to a different class.

3.2.2 MultiResU-Net

MultiResUnet (Ibtehaz & Rahman, 2020b) is a powerful deep learning architecture specifically designed for medical image segmentation. It tackles the challenges of complex anatomical structures and variations in size by incorporating multi-resolution pathways, making it a versatile tool for medical image analysis.

- **Architecture Overview:**
  1. **Encoding Pathway:**
     - **Structure:** Like U-Net, MultiResUNet has an encoding (or contracting) pathway that captures the context of the image. This pathway typically consists of several layers.
     - **Layers:** Each layer usually includes convolutional layers, which may be followed by batch normalization and a ReLU activation function. After each convolutional block, there is often a pooling layer (like max pooling) to reduce the spatial dimensions and increase the depth.
     - **MultiRes Blocks:** MultiRes blocks in MultiResUNet architecture are advanced components designed to capture multi-scale features within medical images. Each MultiRes block consists of a series of convolutional layers with varying

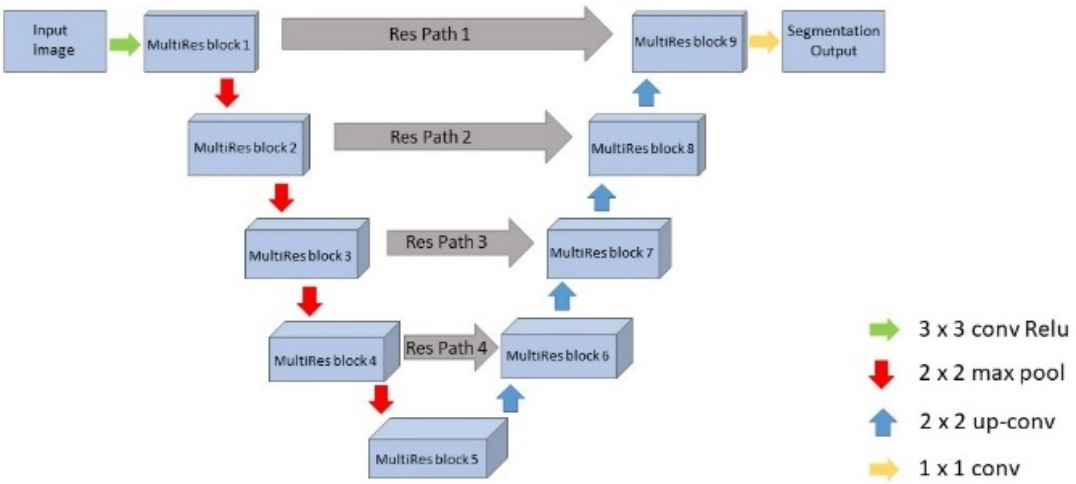


Figure 3.2: The MultiResUNet Architecture

filter sizes, enabling the extraction of features at different resolutions simultaneously. This design contrasts with traditional convolutional blocks that use a single filter size, thus limiting their resolution scope.

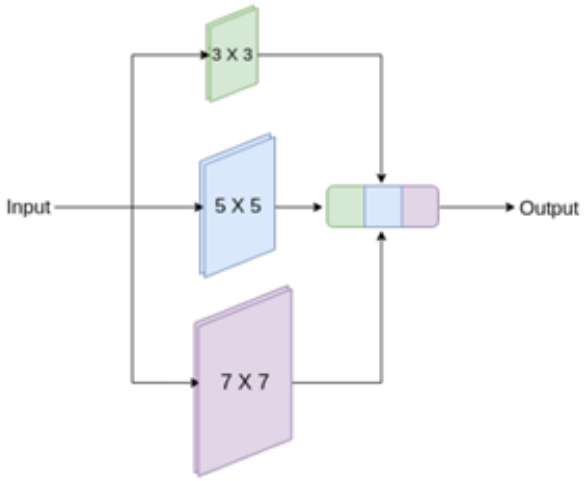


Figure 3.3: The MultiRes Block

2. **Key elements of a MultiRes block include:**
   - **Parallel Convolutional Layers:** Multiple convolutional layers with different kernel sizes (e.g., 3x3, 5x5, 7x7) operate in parallel. This setup allows the network to recognize patterns and structures of varying sizes within the same layer.
   - **Feature Fusion:** The outputs of these parallel layers are then combined, often through concatenation, to merge the multi-scale feature information.
   - **Batch Normalization and Activation:** Following the fusion of features, batch normalization is applied to standardize the activations, and a non-linear activation function, typically ReLU, is used for introducing non-linearity.
3. **Bottleneck:**
   - **Purpose:** The bottleneck is the transition zone between the encoding and decoding pathways. It's where the highest-level features are processed.

- **Composition:** It usually consists of convolutional layers (sometimes with dilation to increase the receptive field), and like in the encoding pathway, may include batch normalization and ReLU activation.

4. **Decoding Pathway:**
   - **Structure:** The decoding (or expansive) pathway is symmetrical to the encoding pathway. It progressively reconstructs the image details from the high-level features obtained in the bottleneck.
   - **Up-sampling:** In each step of the decoding pathway, the feature maps are up-sampled (using techniques like transposed convolutions or up-sampling layers).
   - **Convolutional Layers:** After up-sampling, convolutional layers are used to refine the features.
5. **Skip Connections:**
   - **Function:** Skip connections are a critical aspect of the U-Net architecture, retained in MultiResUNet. They connect the encoding pathway to the decoding pathway, bypassing the bottleneck. The Skip connections used are ResPath.
   - **Benefit:** These connections help in recovering the spatial information lost during down-sampling by concatenating or adding feature maps from the encoding layers to the corresponding decoding layers.
6. **Output Layer/Final Layer:**
   - **Final Convolution:** The last layer of the decoding pathway is followed by a final convolutional layer, which reduces the number of output channels to the desired number (usually the number of classes in segmentation tasks).
   - **Activation Function:** A softmax or sigmoid activation function is typically used in the final layer to generate the segmentation map.

- **Input and Output:**
  - **Input:** The input to the network is usually a medical image, such as Ultrasound or CT scan. The size and number of channels depend on the specific application. We have used the image size of 128x128 with only one channel.
  - **Output:** The output is a segmented image where each pixel is classified into one of the classes. Thus, the final output can be seen as a predicted mask for a given image.

3.2.3 DCU-Net

DCU-Net (Dual Channel U-Net) (Oktay et al., 2018) is an advanced architecture for medical image segmentation, building upon the traditional U-Net model. It incorporates dense connectivity, a concept borrowed from DenseNet, to enhance feature propagation and reuse.

- **Architecture Overview:**

1. **Encoding Pathway:**
   - **Function:** The encoding pathway captures the context and features of the image, progressively reducing its spatial dimensions while increasing the depth (number of channels).

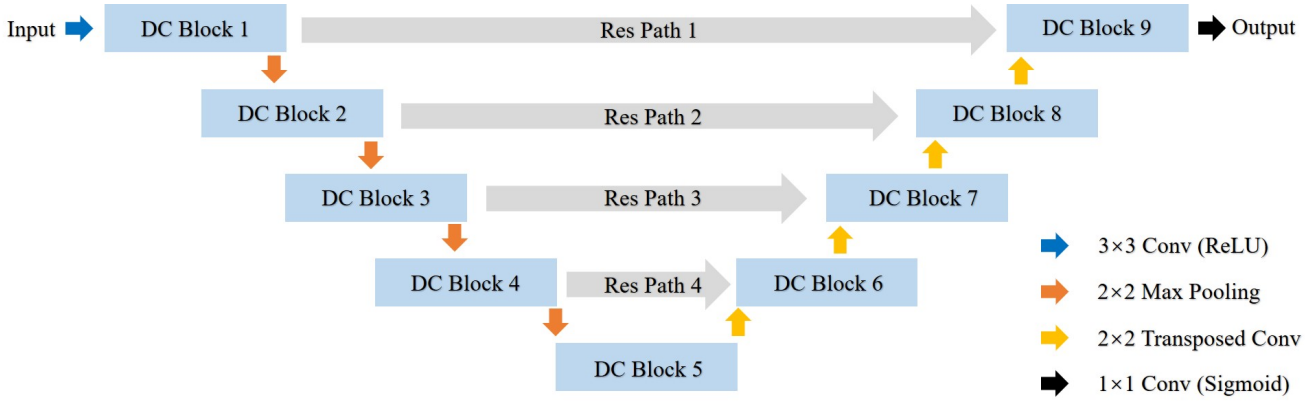


Figure 3.4: The DCUnet Architecture

- **Layers:** It typically consists of multiple DCU blocks, each containing convolutional layers (often followed by batch normalization and ReLU activation), and a pooling layer (like max pooling) to down-sample the image.
- **Dense Connectivity:** Unlike U-Net, DCU-Net introduces dense connections within each block, meaning each layer receives feature maps from all preceding layers as input. This design enhances feature propagation and reduces the vanishing gradient problem.
- **DC-Block:** The DC block introduces a critical parameter known as alpha, which plays a pivotal role in determining the quantity of filters within each convolutional layer of the DCU block. This parameter, alpha, can be adjusted based on the specific requirements of the given problem. Within the DC block, three convolutional layers, each featuring a distinct number of filters, are sequentially applied. These convolutional operations are executed linearly, and the outputs from these three operations are concatenated. Following this, batch normalization is applied, resulting in the creation of the first output. Similarly, another output is generated through a parallel linear processing of the input to the DC block. Ultimately, the two outputs are combined, and after the application of batch normalization and the rectified linear unit (ReLU) activation function, the final output of the DC block is produced.

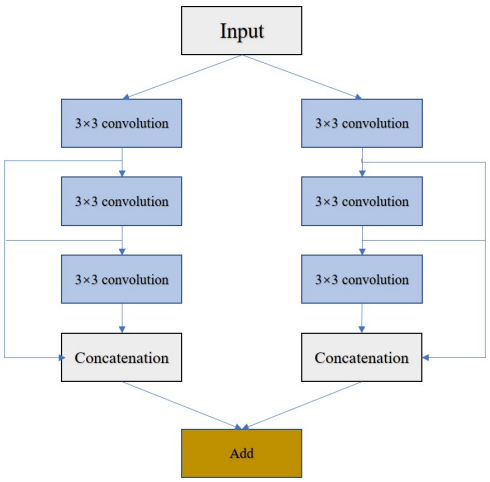


Figure 3.5: The DC-Block

2. **Bottleneck:**
   - **Role:** This is the transition zone between the encoding and decoding pathways, where the highest-level features are extracted.
   - **Composition:** The bottleneck usually comprises densely connected convolutional layers, potentially with dilation to increase the receptive field, and may include batch normalization and ReLU activation.
3. **Decoding Pathway:**
   - **Purpose:** The decoding pathway progressively reconstructs the image details from the high-level features obtained in the bottleneck.
   - **Up-sampling:** Feature maps are up-sampled (using transposed convolutions or up-sampling layers) in each step of the decoding pathway.
   - **Dense Blocks:** Similar to the encoding path, the decoding pathway also features densely connected convolutional layers to refine the up-sampled features.
4. **Skip Connections:**
   - **Function:** DCU-Net retains the concept of skip connections from U-Net, connecting the encoding and decoding pathways.
   - **Enhancement:** These connections help recover spatial information lost during down-sampling. In DCU-Net, they might also carry dense connectivity, further enriching the feature maps transferred to the decoding pathway.
5. **Output Layer/Final Layer:**
   - **Final Convolution:** The last step in the decoding pathway includes a convolutional layer to map the deep features to the desired number of classes (as in segmentation masks).
   - **Activation Function:** A softmax or sigmoid activation function is typically used to generate the final segmentation output.

- **Input and Output:**
  - **Input:** DCU-Net accepts medical images as input, whose size and channel count depend on the application. In our model, we have used images of size 128x128.
  - **Output:** The output is a segmented image, with each pixel classified into different categories of the classes present.

3.2.4 Proposed Architechture (VU-Net)

The UNet is effective for small-scale medical image data but struggles with 2D segmentation's limited inter-slice information. To address this, Wang and Wang utilized 3D images in the VNet network, employing 3D convolution operations for spatial information. While these models yield accurate results, challenges in data acquisition hinder dataset scalability. The goal is to estimate data sample distribution and generate new samples, necessitating further exploration to enhance UNet's segmentation accuracy.

- **WGAN**: The training of Generative Adversarial Networks (GANs) (I. J. Goodfellow et al., 2014) poses challenges due to the difficulty in balancing the generator and discriminator, along with a lack of suitable indicators for training assessment. Various GAN architectures, including CGAN, DCGAN, and InfoGAN, address some issues but fail to

solve training stability problems. Wasserstein GAN (WGAN) emerges as a solution by addressing training instability and introducing the Wasserstein distance, which naturally measures discrete and continuous distribution distances, mitigating common problems like stable training and gradient disappearance in GANs.

- **VNET**: The VNet architecture (Milletari et al., 2016b) comprises an encoder with multiple stages maintaining the same resolution and a decoder for gradual decompression, producing an output image of the original size. Inheriting UNet's jump connection, VNet mitigates information loss during feature extraction. Additionally, it adopts the short-circuit connection mechanism from ResNet, adding input and output at each stage to learn the residual function.

- **The architecture**: It (Ma et al., 2021) consists of 2 parts the generator and the discriminator. Generator:- The task of generating the segmentation mask lies with the generator. the generator used in our method adopts VNet as the backbone network and modifies the original network in a targeted manner. During the training process, convolution kernels with a size of 1×3×3 and 3×1×1 were used to extract the intra layer and interlayer information.

  The network comprises five convolution blocks, four deconvolution blocks, and a final convolution output layer. Its encoder path has five stages, each featuring two convolution blocks with 32 channels, employing 64, 128, 256, and 256 channels sequentially. To reduce memory usage, 1×3×3 convolution replaces pooling in downsampling. Short-range residual learning is achieved by adding input and output at each stage. Upsampling with 512 1×3×3 convolution kernels precedes decoding. The decoder, with four stages, employs 1×3×3 convolution for feature map restoration. A chain residual pooling module performs efficient pooling, retaining input information, and a 1×1×1 convolution yields an output image.

- **Discriminator:**: the discriminator network, consists of conditional pooling blocks, convolutional blocks, fully connected layers, and classification layers. It takes the generator's output and the annotated original image as inputs, fused and processed through convolutional layers, normalisation, and activation functions. The convolutional blocks utilise 1×3×3 kernels, and pooling employs 1×2×2 kernels. With 64 convolution operations in the 1st and 2nd blocks and 32 in the 3rd and 4th blocks, the output feeds into fully connected layers for classification.

- **Our Architecture**: The VU-Net framework builds upon the robust foundation of the V-Net, albeit with tailored adjustments to seamlessly accommodate 2D input images, a departure from the native 3D input characteristic of V-Net. This modification ensures compatibility with the specific requirements of our segmentation task while retaining the inherent strengths of the V-Net architecture.In particular, the final layer of our architecture, serving as the pivotal output layer, is meticulously crafted to wield three filters. This strategic choice aligns with the exigencies of our multi-class segmentation objective. To facilitate the intricate mapping of input features to distinct classes, the softmax activation function is judiciously employed in this concluding layer. The architectural prowess is further enriched through an intelligently designed connecting pathway, drawing inspiration from an influential and meticulously referenced research paper. This pathway plays a

pivotal role in capturing intricate contextual information across the input data, thereby enhancing the model's ability to discern subtle patterns and features. Moreover, the decoder segment of our architecture is seamlessly integrated from the well-established UNet architecture. This deliberate adoption contributes significantly to the holistic structure of our model, leveraging the UNet's proven efficacy in image segmentation tasks. The decoder's role in progressively reconstructing intricate details from high-level features obtained in earlier stages complements the overall segmentation process.In essence, the amalgamation of V-Net's foundational strength, a tailored adjustment for 2D inputs, a sophisticated connecting pathway, and the incorporation of UNet's decoder segment results in a comprehensive and technically sound architecture. The careful selection of design elements ensures that our model is well-equipped to handle the intricacies of multi-class segmentation tasks, making it a robust and versatile tool for image analysis.

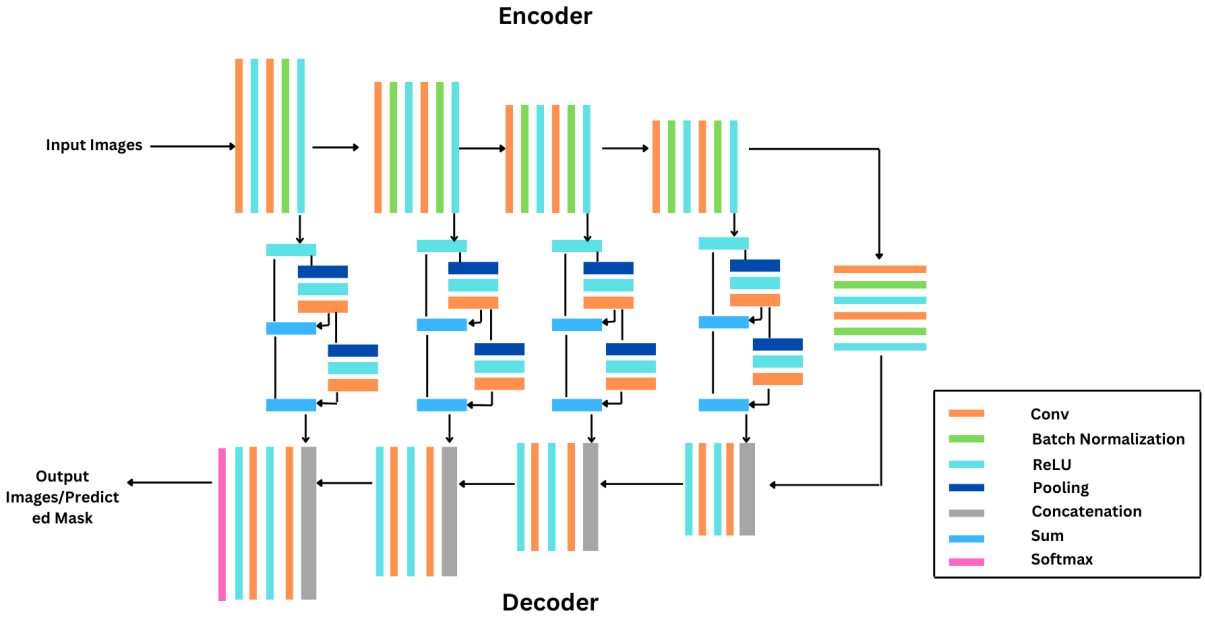


Figure 3.6: The VU-Net (Vnet + Unet)

- **Architecture Overview:**

  1. **Encoding Pathway:** The network comprises five convolution blocks, each structured with 2 convolution kernels, normalization, and a ReLU activation function, all using 32 in sequence. Convolution kernels have 16, 32, 64, 128, 256 filters with a stride of 1 to extract distinctive features.
  2. **Bottleneck:** The V-Net's bottleneck distills essential features from the encoding pathway for precise segmentation. Comprising densely connected convolutional layers with optional dilation, it captures intricate patterns and mitigates the vanishing gradient issue. Positioned between encoding and decoding, it refines high-level representations.The architecture incorporates a dual-layer convolutional block, complemented by Batch Normalization and Rectified Linear Unit (ReLU) activation functions, enhancing feature extraction and promoting non-linearity within the model.

3. **Decoding Pathway:** The decoder path involves upsampling feature maps. Each block consists of two 3x3 convolutions, batch normalization which is done during concatenation, and ReLU activation. Upsampling is done through transposed convolutions.
4. **Skip Connections:** Connecting pathways include ReLU activation, pooling module with a 5x5 filter (16 to 256 filters), ReLU activation, and another convolution layer. Outputs are combined for the first and second modules, producing the final output.
5. **Output Layer/Final Layer:** The decoding pathway ends with a convolutional layer for mapping deep features to the desired classes (as in segmentation masks) and a softmax function for multiclass semantic segmentation.

- **Input and Output:**
  - **Input:** The architecture accepts medical images of different size (we have used images of size 128x128), with channel count depending on the application.
  - **Output:**The output is a segmented image, with each pixel classified into different categories of the classes present.

3.3 Implementation of GradCAM for Segmentation

Grad-CAM is utilized to visualize influential regions in segmentation tasks, both binary and multiclass. By modifying the model to output segmentation maps and activations of a specific convolutional layer, Grad-CAM computes gradients to generate heatmaps. These heatmaps, overlaid on the original image, highlight the model's focus areas, enhancing interpretability and understanding of the model's decision-making process in medical imaging. The following sections detail the step-by-step process of implementing Grad-CAM for both binary and multiclass segmentation tasks, ensuring that the approach can be effectively applied to a wide range of medical imaging applications.

- **Model Modification for Grad-CAM:** The function `make_gradcam_heatmap` starts by modifying the existing model. It constructs a new model (`grad_model`) that outputs not only the predictions of the original model but also the outputs of a specified last convolutional layer. This layer, identified by `last_conv_layer_name`, is crucial as it captures high-level features that are directly related to the model's decision-making process.
- **Recording Operations for Differentiation:** The `with tf.GradientTape() as tape` block enables automatic differentiation, a key component for computing gradients in deep learning. This feature records all operations performed on the model's inputs, which is essential for later calculating the gradients of the output with respect to the activations of the last convolutional layer.
- **Model Forward Pass:** Inside the gradient tape block, the function performs a forward pass of the input image (`img_array`) through the `grad_model`. It captures both the last convolutional layer's outputs (`last_conv_layer_output`) and the final prediction (`preds`). In this context, the prediction is assumed to be a single-channel segmentation map typical in binary segmentation tasks.

- **Focus on Overall Influence:** The mean of the predictions (`preds`) across the image is computed using `tf.reduce_mean`, which represents the aggregate influence of all pixels on the model's prediction. This step is crucial in understanding the general areas influencing the output in binary segmentation tasks. If no specific class index (`pred_index`) is provided, the code automatically selects the class with the highest prediction from the model's output using `tf.argmax(preds[0])`. This allows focusing on the class deemed most likely by the model. The important change here is in the handling of predictions. Instead of using the mean across the entire prediction output, the activation map for the specific class index (`pred_index`) is used. This provides a focus on how the image areas specifically influence the prediction for that class.

- **Gradient Calculation:** The gradient of this mean prediction with respect to the outputs of the last convolutional layer is computed, revealing which parts of the last convolutional layer activations are most influential.

- **Pooling and Weighting:** The gradients are pooled using a global average to summarize their values across the spatial dimensions, and these pooled gradients are used to weight the last convolutional layer's activations. The sum across the channels after this weighting produces the heatmap.

- **Normalization:** Finally, the heatmap is normalized to a range between 0 and 1 to facilitate visualization, ensuring that the most influential areas are highlighted clearly.

The multiclass approach is crucial for tasks like tumor segmentation in medical imaging, where differentiating between multiple types of tissue or conditions is necessary. It allows clinicians to understand which areas of the image are decisive for each class, enhancing trust and aiding in diagnostic decisions.

CHAPTER 4: IMPLEMENTATION

This chapter provides an overview of the implementation details of the project, including the development environment, and how we implemented the Segmentation algorithm and the algorithm for Grad-CAM. During this project, we focused on the application and evaluation of three distinct deep learning models: U-Net, MultiResU-Net, and DCU-Net across three different datasets—Heart, Breast, and LiTS, each presenting unique challenges and learning opportunities.

4.1 Development Requirements

The project was developed using the following environment:

Table 4.1: Development Requirements Overview

Component	**Description**
Programming Language	Python 3.7
Development Enivronment	Google Colab (Standard Version)
Development Framework	Tensorflow
Libraries and Tools	Numpy, Keras, Matplotlib, Scikit

4.2 Implementation Process Description

4.2.1 Heart Dataset

Our initial experiments were conducted on the Heart dataset, which is primarily used for binary segmentation tasks. We began with the U-Net model, adjusting various hyper-parameters to find an optimal configuration. Through iterative training sessions, we established a learning rate of 0.0001 and a validation split of 10% as the most effective settings. The training involved monitoring key metrics such as accuracy, Intersection over Union (IoU), precision, recall, and the Dice coefficient. Graphical analyses of training versus validation performance helped us identify and mitigate overfitting; for instance, extending training beyond 20 epochs to 50, and eventually to 100 epochs, indicated a trend towards overfitting, which we addressed by introducing an Early Stopping mechanism. For loss functions, we explored Binary Cross Entropy, Binary Focal Loss, and Categorical Cross Entropy. Our findings revealed that while Binary Cross Entropy and Binary Focal Loss were more apt for binary tasks, Categorical Cross Entropy was best suited for multi-class segmentation, thereby guiding our choice of loss functions for subsequent datasets. Following U-Net, we applied MultiResU-Net and DCU-Net to the Heart dataset. Each model's performance was documented, analyzed, and compared through saved results and graphical plots.

4.2.2 Breast Dataset

Moving to the Breast dataset, we noted that the initial validation split of 10% was inadequate, necessitating an increase to 20% to enhance model validation during training. Other hyperparameters remained consistent with those set during the Heart dataset experimentation. Similar to our previous approach, we trained the models, saved the training data, and plotted the results for detailed evaluation on the testing data.

4.2.3 LiTS Dataset

The LiTS dataset, being a multi-class segmentation challenge, required a different approach, specifically the use of Categorical Cross Entropy as the loss function. Training sessions for U-Net, MultiResU-Net, and DCU-Net were conducted with an emphasis on multiclass segmentation. We meticulously saved and plotted training data for a comprehensive evaluation based on the established metrics.

4.2.4 Development of VU-Net

Informed by our extensive experimentation and the distinct characteristics of the V-Net and U-Net architectures, we developed a new model, V-UNet. This model was specifically applied to the LiTS dataset. Training data and evaluations were methodically recorded and analyzed, providing insights into the model's efficacy and areas for improvement. The result of this model was then compared to the result of U-Net, MultiResU-Net and DCU-Net.

4.2.5 Integration of GradCAM

To deepen our understanding of model behaviors and enhance interpretability, we implemented GradCAM across all models, including the newly developed V-UNet. The resulting heatmaps were instrumental in visualizing how effectively each model focused on pertinent features within the datasets, particularly in identifying and delineating tumor regions. This visualization not only confirmed the models' accuracy in targeting relevant features but also provided crucial insights that guided further refinements in our modeling approach.

This comprehensive application of multiple models across varied datasets, coupled with advanced visualization tools like Grad-CAM, has significantly enriched our understanding of deep learning's potential in medical image segmentation. It has also laid a solid foundation for future explorations and innovations in medical imaging technology.

CHAPTER 5: EXPERIMENTS AND RESULTS

5.1 Dataset

We have studied and experimented on three different dataset using three DL architectures on each and have proposed a mulitcalsss segmentation model (VU-Net) for medical image segmentation. The Datasets used are mentioned below.

5.1.1 The CT heart Disease Dataset

Globally, a substantial portion of the population, spanning various age groups, faces a considerable risk of cardiac events. Addressing the imperative need for early detection of heart attacks, numerous researchers have delved into clinical datasets sourced from reputable repositories such as PubMed and the UCI repository. However, prevalent datasets often encompass an extensive range of raw attributes, ranging from 13 to 147 in textual format, necessitating the application of traditional data mining methodologies("CT Heart Segmentation dataset", 2021).
Source: A clinical dataset comprising 25 individuals of diverse age groups has been meticulously curated from various geographical locations.
Type: CT scan
Image Format: PNG (train), DICOM (test)
Number of Images: 780 training images, 253 test images
Quality of Images: 512 by 512 pixels
Size of Dataset: 567 MB

5.1.2 Breast Ultrasound Images Dataset

Breast cancer stands as a prominent cause of mortality among women globally, underscoring the critical importance of early detection in mitigating premature fatalities. The dataset ("Breast Ultrasound Images dataset", 2020) under scrutiny pertains to medical images of breast cancer acquired through ultrasound scans. Categorized into three classes—normal, benign, and malignant—the Breast Ultrasound Dataset presents a valuable resource for leveraging machine and deep learning methodologies.
Source: Women between the ages of 25 and 75 had their breast ultrasound pictures taken as part of the baseline data collection. 2018 saw the collection of this data. There are 600 female patients in all [50].
Type: Ultrasound
Image Format: PNG
Number of Images: 780 images
Quality of Images: 512 by 512 pixels
Size of Dataset: 204 MB

5.1.3 LiTS – Liver Tumour Segmentation Challenge Dataset (LiTS17)

Liver cancer ranks as the fifth most prevalent cancer in men and the ninth in women, with over 840,000 new cases reported in 2018. The liver serves as a frequent site for the development of both primary and secondary tumours. The inherent heterogeneity and diffuse nature of these tumours pose substantial challenges in achieving automatic segmentation. Addressing the complexities associated with the delineation of tumour lesions becomes paramount, emphasizing the need for advanced methodologies in the field of medical imaging and cancer diagnosis("LiTS – Liver Tumor Segmentation Challenge (LiTS17) organized in conjunction with ISBI 2017 and MICCAI 2017", 2017).
Source:The data and mask segmentations are made available from different clinical sites across the world. The Data is of about 130 patients [52].
Type: CT scan
Image Format: NII (Neuroimaging Informatics Technology Initiative)
Number of Images: 130 NII images
Quality of Images: 512 x 512 pixels with varying depth
Size of Dataset: 49.9 GB

5.2 Performance Metrices

Performance metrics in deep learning and machine learning quantify the effectiveness of models. These metrics guide model optimization, ensuring the development of robust algorithms, they also provide valuable insights into the model's behaviour, allowing practitioners to assess its strengths, weaknesses for a given task. For Image segmentation the performance metrices that we will be using are IoU(Intersection over Union), Accuracy, Precision, Recall, Loss, Dice Coefficient. Thus, for all these metrices to be defined we first need to know about confusion matrix.

5.2.1 Confusion Matrix

A confusion matrix (Powers, 2011) is a table used in machine learning and statistical classification to assess the performance of a classification model. It is particularly useful for binary and multiclass classification problems. The matrix presents a clear summary of the model's predictions versus the actual outcomes, providing insights into the model's accuracy and error types.

The confusion matrix typically consists of four entries and with respect to our use we can define them as:

- **True Positives (TP):** Instances that were correctly predicted as positive by the model. In image segmentation, a true positive refers to a pixel or region that is correctly identified as belonging to the target class. It represents the correctly segmented or highlighted area in the image.
- **False Positives (FP):** Instances that were incorrectly predicted as positive. In our case, they occur when the model incorrectly identifies a pixel or region as belonging to the target class when it does not. In image segmentation, this corresponds to pixels or regions that are erroneously highlighted as part of the target class.
- **True Negatives (TN):** Instances that were correctly predicted as negative by the model. In image segmentation, they refer to pixels or regions that are correctly identified as not

belonging to the target class. It represents correctly identified background or non-target regions.

- **False Negatives (FN):** Instances that were incorrectly predicted as negative. They occur when the model fails to identify a pixel or region that belongs to the target class. In image segmentation, this represents areas that are part of the target class but are not highlighted by the model.

For Binary Semantic Segmentation, the Confusion Matrix appears to be a 2x2 matrix with say x – axis as predicted result and y – axis as ground truth, thus the confusion matrix will appear as –

Table 5.1: 3x3 Confusion Matrix

	Positive	**Negative**
True	True Positive (TP)	True Negative (TN)
Flase	False Positive (FP)	Flase Negative (FN)

In case for multiclass Semantic segmentation, we calculate confusion matrix of each class individually.

The Performance metrices thus used are defined further –

- **Accuracy**: Accuracy (Sokolova & Lapalme, 2009) is a common performance metric in machine learning and segmentation tasks that measures the overall correctness of predictions made by a model. It is defined as the ratio of correctly predicted instances to the total number of instances. The formulae:

$$\text{Total Instances} = \text{TP} + \text{TN} + \text{FP} + \text{FN} \tag{5.1}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} \tag{5.2}$$

  Importance: Provides an overall measure of segmentation quality. In the context of image segmentation, accuracy evaluates the model's ability to correctly classify pixels into the desired classes.

- **Precision**: Precision (Sokolova & Lapalme, 2009) is a metric that measures the accuracy of positive predictions made by a classification or segmentation model. It assesses the proportion of predicted positive instances that are actually positive. Precision is calculated as the ratio of true positives to the sum of true positives and false positives.

  The formulae:

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{5.3}$$

  Importance: Measures the accuracy of positive predictions, focusing on minimizing false positives. A high precision ensures that the predicted positive instances are accurate, reducing the likelihood of false alarms.

- **Recall**: Recall (Sokolova & Lapalme, 2009), also known as sensitivity or true positive rate, measures the ability of a classification or segmentation model to correctly identify all relevant instances belonging to a certain class. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

The formulae:

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{5.4}$$

Importance: Measures the ability to capture all positive instances, focusing on minimizing false negatives. Crucial in applications where missing positive instances is costly.

- **IoU(Intersection over Union)**: IoU (Gonzalez & Woods, 2008), also known as the Jaccard Index, is a metric commonly used in segmentation tasks to evaluate the overlap between predicted and ground truth regions. It quantifies the similarity between the two sets by measuring the ratio of their intersection to their union where intersection is The number of true positives (instances correctly predicted as positive) and union is The total number of instances that are either correctly predicted as positive or missed by the model

  The formulae:

$$\text{Intersection} = \text{TP} \tag{5.5}$$

$$\text{Union} = \text{TP} + \text{FP} + \text{FN} \tag{5.6}$$

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} \tag{5.7}$$

  Importance: IoU provides a nuanced measure of segmentation accuracy, particularly in tasks where the spatial alignment of predicted and ground truth regions is crucial. Its importance lies in capturing both the model's ability to correctly identify positive instances (precision) and its capacity to avoid missing positive instances (recall). Achieving a high IoU ensures accurate delineation of regions of interest. It is a key metric for assessing the spatial agreement between predicted and ground truth segments, offering insights into the model's localization accuracy.

- **Dice Coefficient**: The Dice coefficient (Zijdenbos et al., 2002), is a metric commonly used in segmentation tasks to quantify the similarity between predicted and ground truth regions. It balances precision and recall, providing a comprehensive evaluation of segmentation performance.

  The formulae:

$$\text{Ground Truth} = \text{TP} + \text{FN} \tag{5.8}$$

$$\text{Predicted} = \text{TPs} + \text{FP} \tag{5.9}$$

$$\text{Dice Coeff.} = \frac{2 \times \text{Intersection}}{\text{Predicted + Ground Truth}} \tag{5.10}$$

  Importance: The Dice coefficient is important in segmentation tasks as it considers both false positives and false negatives, providing a balanced measure of precision and recall. It is particularly relevant in scenarios where achieving a balance between minimizing false positives and false negatives is crucial. A high Dice coefficient indicates accurate segmentation, ensuring that the model captures the relevant structures while minimizing both over-segmentation and under-segmentation.

- **Loss**: In the context of deep learning "loss" refers to the measure of how well a model's predictions align with the actual (ground truth) values. It quantifies the difference between predicted outcomes and the true outcomes, indicating the model's performance. The loss function is an essential component of the model's objective function, which the model aims to minimize during training. Different tasks and models may require specific loss functions.

  - **Focal Loss:** Focal Loss (I. Goodfellow et al., 2016b) is a specialized loss function designed to address the class imbalance problem in binary classification tasks, particularly when one class is significantly more prevalent than the other. It was introduced by Lin et al. in the paper titled "Focal Loss for Dense Object Detection" in 2017. It is important to note that when the number of classes is two, then the focal loss applied is called binary focal loss.

    Formulae:

    $$\text{Focal Loss} = -(1 - p_t)^{\gamma} \cdot \log(p_t) \tag{5.11}$$

    where:

    p_t : predicted probability of the true class, γ : focusing parameter controlling the rate of down-weighting.

    Importance: Focal Loss is particularly relevant in segmentation tasks where the imbalance between foreground (object) and background pixels is common. In medical imaging, for instance, the number of normal pixels may significantly outweigh the number of pixels representing anomalies. Focal Loss addresses the challenge of class imbalance by assigning higher weights to misclassified examples.

  - **Dice Loss:** Dice Loss (Maier et al., 2019) is a loss function used in segmentation tasks, particularly when dealing with imbalanced datasets. It is derived from the Dice coefficient (or F1 score), which measures the similarity between predicted and ground truth regions. The Dice Loss is designed to be minimized, aligning with the goal of maximizing the Dice coefficient.

    Formulae:

    $$\text{Dice Loss} = 1 - \frac{2 \times \text{Intersection}}{\text{Predicted + Ground Truth}} \tag{5.12}$$

    Importance: Dice Loss is important in segmentation tasks, especially when there is a significant class imbalance. It is sensitive to the degree of overlap between predicted and ground truth regions. This sensitivity makes it particularly relevant in scenarios where spatial alignment is critical.

  - **Cross Entropy:** Cross entropy (Bishop, 2006b) is a widely used loss function in machine learning, particularly in classification tasks. It measures the difference between the predicted probability distributions and the true probability distributions. The goal during training is to minimize this difference, which effectively means making the predicted distribution match the true distribution as closely as possible.

    * **Binary Cross-Entropy:** Binary cross-entropy (Chollet, 2017) is used for binary classification problems where there are only two classes (0 and 1). It typically uses a sigmoid activation function in the output layer.
      Formulae:

      $$H(y, p) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \tag{5.13}$$

where:
N is the number of examples, y_i is the true label, and p_i is the predicted probability for class 1.

* **Categorical Cross-Entropy:** Categorical cross-entropy (I. Goodfellow et al., 2016b) is used for multi-class classification problems where there are more than two classes. It typically uses a softmax activation function in the output layer. Formulae:

$$H(y, p) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log(p_{i,j}) \tag{5.14}$$

where:
N is the number of examples, C is the number of classes, $y_{i,j}$ is an indicator (0 or 1) of whether class j is the true class for example i, and $p_{i,j}$ is the predicted probability for class j.

5.3 Data Preprocessing

5.3.1 Heart Dataset

The dataset comprises two directories: `"test"` containing DICOM format images without ground truth and `"train"` with PNG format images, each accompanied by corresponding ground truth. Since only the `train` directory had images with ground truth, the processing was done on them. The total number of images in the `train` directory is 2532.

To facilitate model evaluation, 10% of the `train` images (253) were reserved for testing, while the remaining 2279 were designated for training. Additionally, a validation set was derived from the training images, consisting of 10% of the training data.

To expedite the training process, the images were downscaled from 512x512 to 128x128 using the resize() [55] function from the scikit-image [54] library. This function employs bilinear interpolation [56] by default and incorporates Gaussian filtering [57] for anti-aliasing, enhancing the overall image quality.

To optimize storage and expedite training, the processed images were converted into NumPy arrays. These arrays were then stored in a .npy file format, effectively reducing the file size and enabling faster data loading during training.

5.3.2 Breast Dataset

The dataset consists of 780 PNG format images, each originally sized at 512x512 pixels. The data was strategically partitioned to facilitate effective model training and evaluation.

For rigorous model evaluation, 10% of the images, totaling 77, were randomly selected and set aside for testing. Subsequently, from the remaining 703 images, an additional 20% were randomly chosen to form the validation set. The preeminent portion of the dataset, encompassing 80% of the 703 images, was then allocated to the training set.

To expedite the training process, the images were downscaled from 512x512 to 128x128 using the same resize() function from the scikit-image library used in the heart dataset.

To optimize storage and expedite training, the processed images were converted into NumPy arrays. These arrays were then stored in a .npy file format, effectively reducing the file size and enabling faster data loading during training.

5.3.3 LiTS Dataset

The dataset initially comprised 130 NII format images, representing a 3D file format. Leveraging the nilabel [54] Python library [53], these NII images were converted into PNG format, resulting in an expansive dataset of 58,638 individual PNG images. Remarkably, a subset of 39,475 images lacked liver representation, while the remaining 19,163 images featured either the liver alone or the liver and liver tumour.

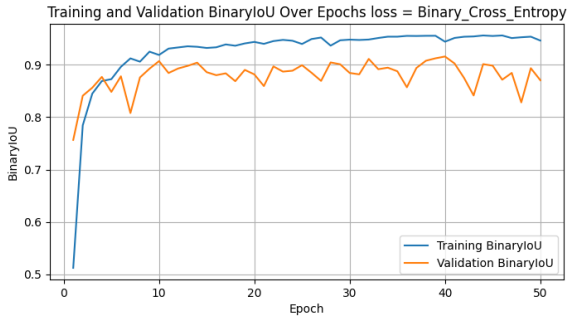
From the cohort of 19,163 images, strategic data partitioning was undertaken. A testing set was created, constituting 10% of the total images, resulting in 1,916 randomly chosen test images. Simultaneously, the training set, encompassing 17,247 images, was established. Within this training set, 10% of the images were allocated to a validation subset. Each PNG image, originally of dimensions 512x512, underwent quality-conscious downscaling to 128x128 using the resize() function from the scikit-image library. This meticulous resizing operation not only addressed computational efficiency but also maintained the essence of the images for subsequent analysis. To expedite the training pipeline, the processed images were transformed into NumPy arrays and efficiently stored in .npy file format. This reduction in file size contributes to expedited data loading during training, fostering a more streamlined and efficient deep learning workflow.

5.3.4 Heart Dataset

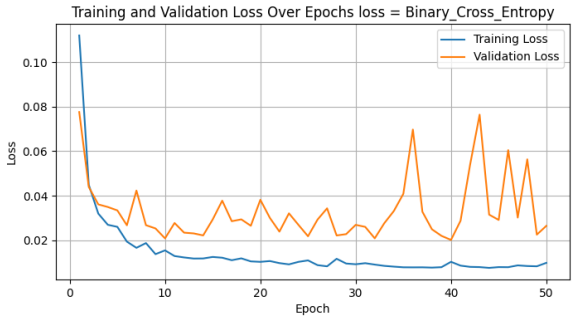
1. U-Net architecture:
   The loss functions use are Binary Cross Entropy, Binary Focl Loss and Categorical Cross Entropy.

Table 5.2: Heart Dataset - U-Net test results

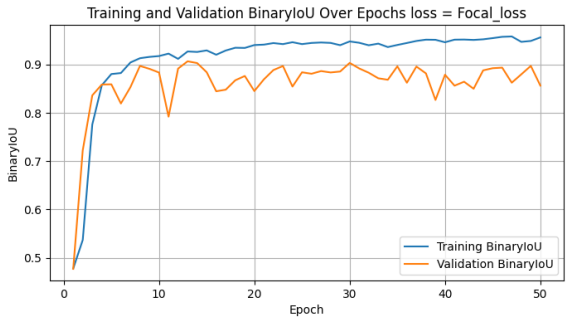
	Binary Cross Entr.	**Binary Focal Loss**	**Categorical Cross Entr.**
Accuracy	0.9953	0.9950	0.9477
Recall	0.9500	0.9183	0.9711
Precision	0.9228	0.9345	0.3989
IoU	0.9361	0.9305	0.6699
Dice coeff.	0.9360	0.9245	0.5535
Loss	0.0126	0.007	17.040



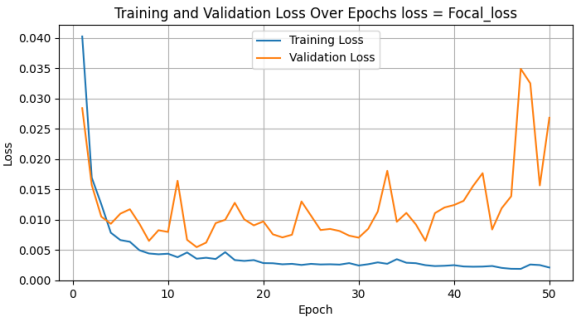
(a) IoU (Binary Cross Entropy)



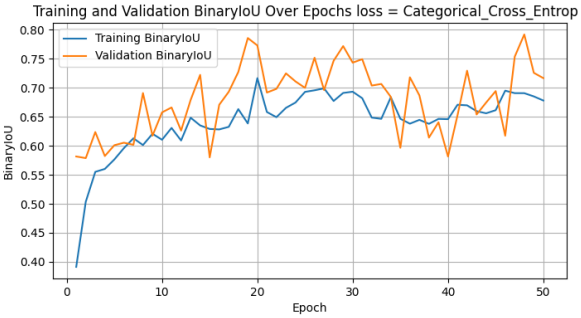
(b) Loss (Binary Cross Entropy)



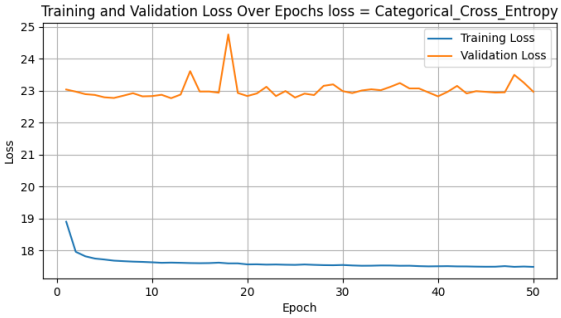
(c) IoU (Focal Loss)



(d) Loss (Focal Loss)



(e) IoU (Categorical Cross Entropy)



(f) Loss (Categorical Cross Entropy)

Figure 5.1: Heart Dataset - U-Net train graphs (Metric vs epoch)

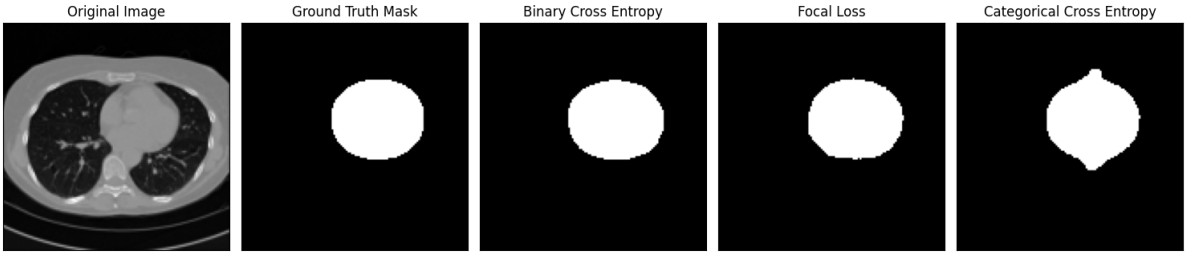
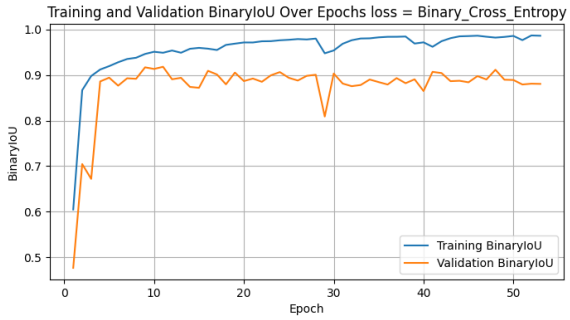


Figure 5.2: Predicted results vs Ground truth (Heart Dataset - U-Net)

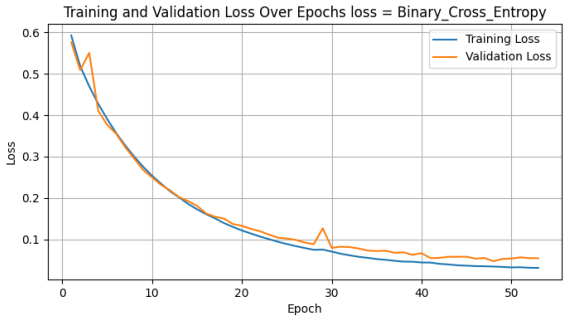
2. MultiResU-Net architecture:
   The Loss functions used are Binary Cross Entropy, Binary Focal Loss and Dice Loss. Categorical Cross Entropy proved unsuitable for binary segmentation within UNet due to suboptimal performance, emphasizing the importance of task-specific metric selection for accurate model evaluation.

Table 5.3: Heart Dataset - MultiResU-Net test results

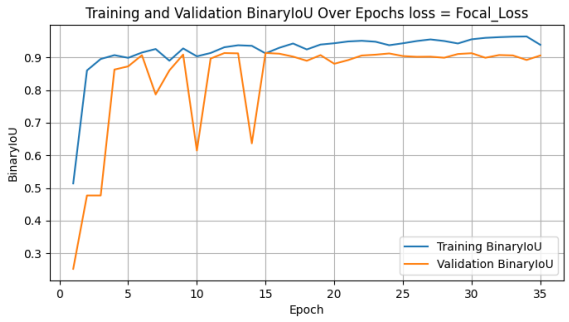
	Binary Cross Entr.	**Binary Focal Loss**	**Dice Loss**
Accuracy	0.9958	0.9939	0.9959
Recall	0.9124	0.9850	0.9382
Precision	0.9666	0.8597	0.9478
IoU	0.9405	0.9221	0.9427
Dice coeff.	0.9383	0.9177	0.9428
Loss	0.0342	0.0103	-0.666



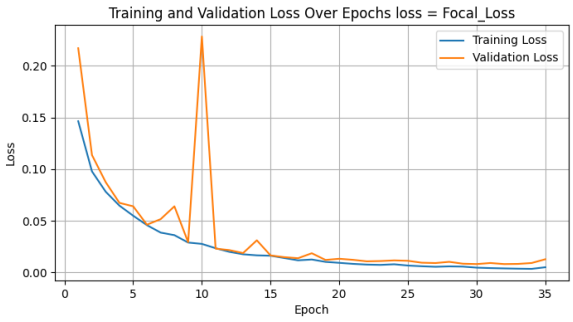
(a) IoU (Binary Cross Entropy)



(b) Loss (Binary Cross Entropy)



(c) IoU (Focal Loss)



(d) Loss (Focal Loss)

Figure 5.3: Heart Dataset - MultiResU-Net train graphs (Metric vs epoch)

(a) IoU (Dice Loss)

(b) Binary Cross Entropy (Dice Loss)

Figure 5.4: Heart Dataset - MultiResU-Net train graphs (Metric vs epoch)

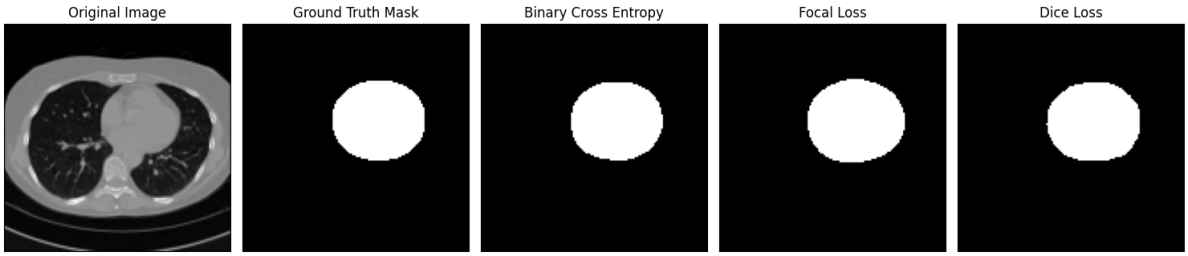
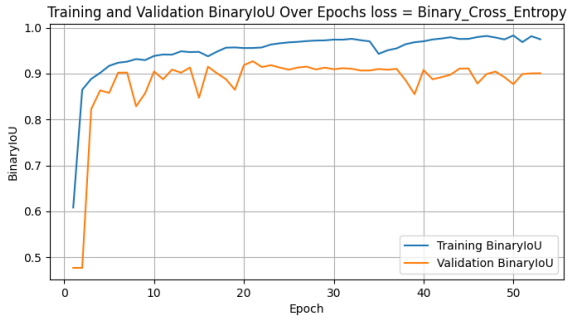


Figure 5.5: Predicted results vs Ground truth (Heart Dataset - MultiResU-Net)

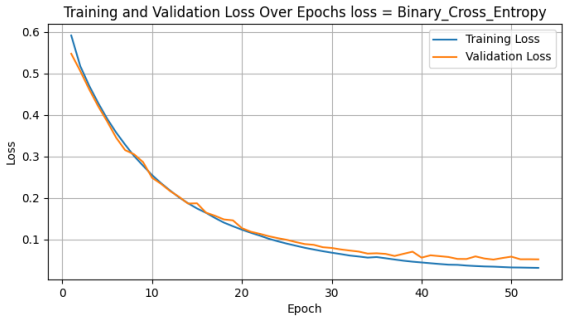
3. DCU-Net architecture:
   The loss functions used are Binary Cross Entropy, Binary Focal Loss and Dice Loss

Table 5.4: Heart Dataset - DCU-Net test results

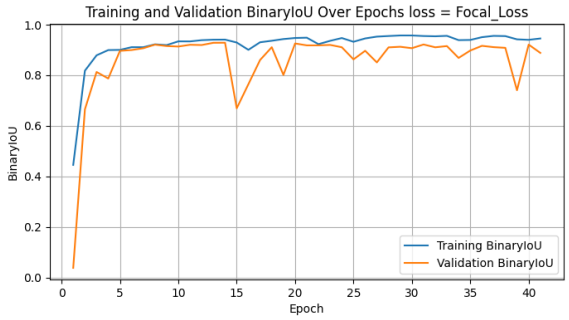
	Binary Cross Entr.	**Binary Focal Loss**	**Dice Loss**
Accuracy	0.9955	0.9953	0.9954
Recall	0.9194	0.8849	0.9082
Precision	0.9466	0.9725	0.9523
IoU	0.9365	0.9331	0.9357
Dice coeff.	0.9326	0.9261	0.9291
Loss	0.0359	0.0042	-0.645



(a) IoU (Binary Cross Entropy)

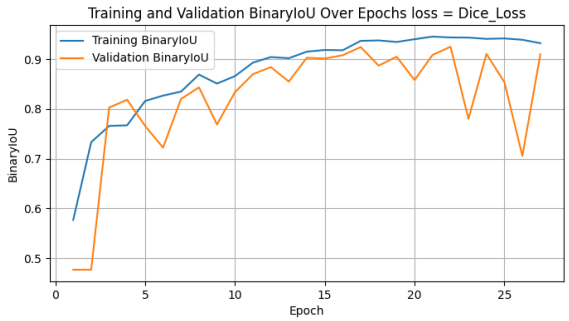


(b) Loss (Binary Cross Entropy)

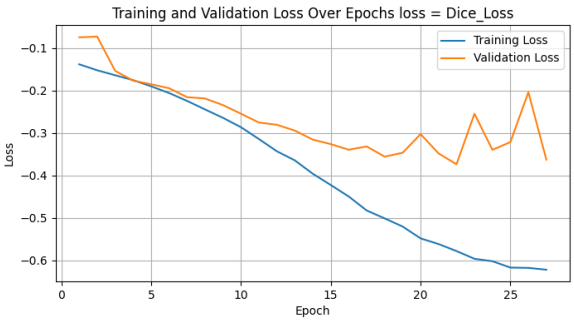


(c) IoU (Focal Loss)

(d) Loss (Focal Loss)



(e) IoU (Dice Loss)



(f) Loss (Dice Loss)

Figure 5.6: Heart Dataset - DCU-Net train graphs (Metric vs epoch)

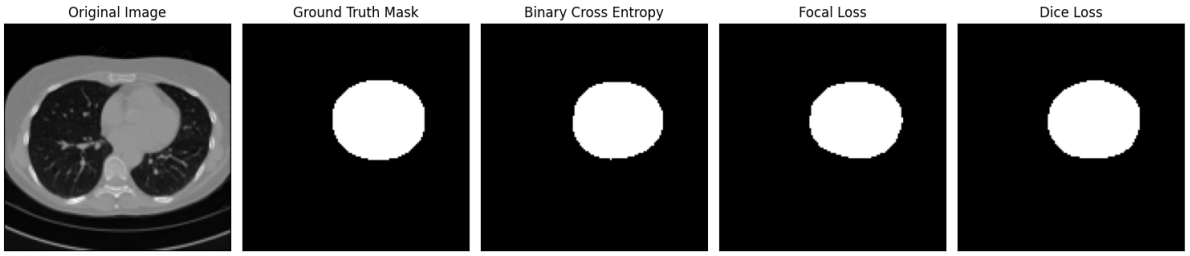


Figure 5.7: Predicted results vs Ground truth (Heart Dataset - DCU-Net)

5.3.4.1 Obervation

1. Binary Cross Entropy The analysis of the provided table reveals that the MultiResUnet architecture demonstrates superior performance compared to the other two models across four out of six metrics, notably excelling in IoU (Intersection over Union). Conversely, the Unet model managed to outperform MultiResUnet in only two metrics, with the loss metric being one of them. Consequently, based on the specific metrics considered in this evaluation, the MultiResUnet architecture emerges as the more effective choice.

2. Focal Loss In this scenario, the DCU-Net exhibits superior performance compared to the other two models across four out of six metrics, notably excelling in both IoU (Intersection over Union) and loss. Consequently, the DCU-Net emerges as the clear winner, demonstrating its effectiveness and suitability for the given task.

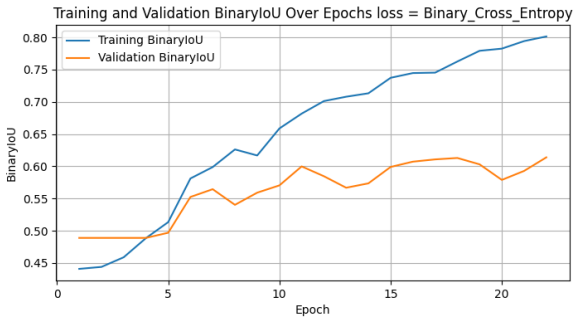
3. Dice Loss In this context, the MultiResU-Net had demonstrates superior performance across five out of six metrics, encompassing key indicators such as Intersection over Union (IoU) and loss. Consequently, MultiResU-Net is it stands as the preferred model in this comparative analysis over DCU-Net.

5.3.5 Breast Ultrasound Dataset

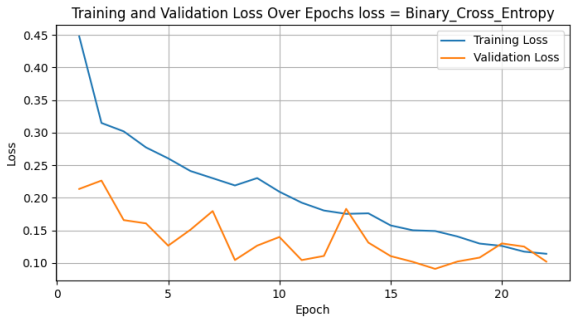
1. U-Net architecture:
   The loss functions used are Binary Cross Entropy, Binary Focal Loss and Dice Loss

Table 5.5: Breast Dataset - U-Net test results

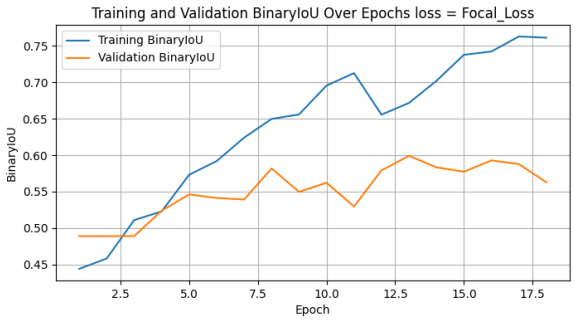
	Binary Cross Entr.	**Binary Focal Loss**	**Dice Loss**
Accuracy	0.9536	0.9437	0.9403
Recall	0.3778	0.4700	0.3659
Precision	0.6116	0.4998	0.4888
IoU	0.7292	0.7267	0.6877
Dice coeff.	0.4657	0.4809	0.4107
Loss	0.1224	0.0386	-0.5120



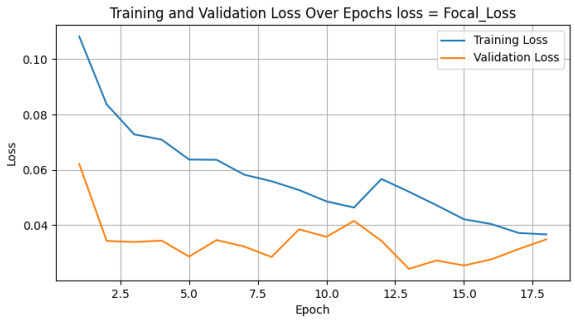
(a) IoU (Binary Cross Entropy)



(b) Loss (Binary Cross Entropy)

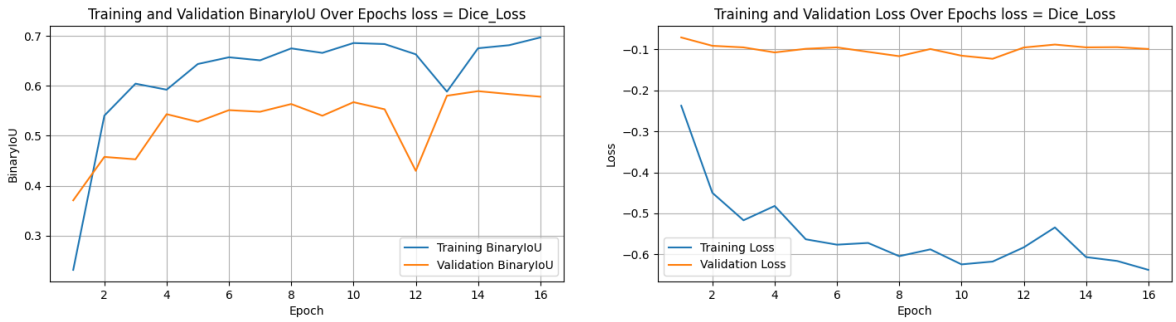


(c) IoU (Focal Loss)



(d) Loss (Focal Loss)

Figure 5.8: Breast Dataset - U-Net train graphs (Metric vs epoch)



(a) IoU (Dice Loss)

(b) Loss (Dice Loss)

Figure 5.9: Breast Dataset - U-Net train graphs (Metric vs epoch)

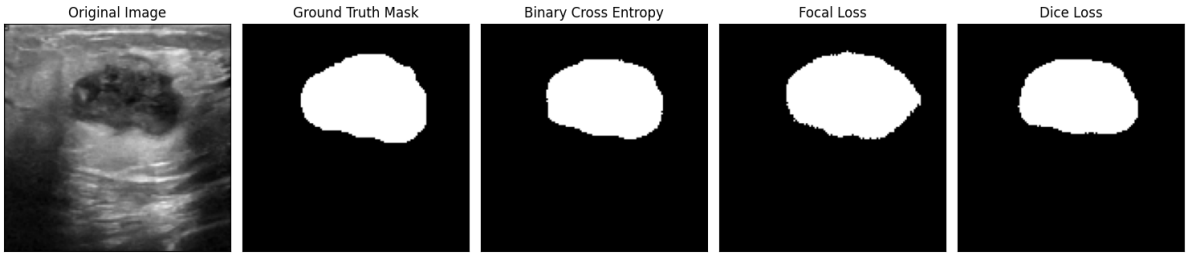
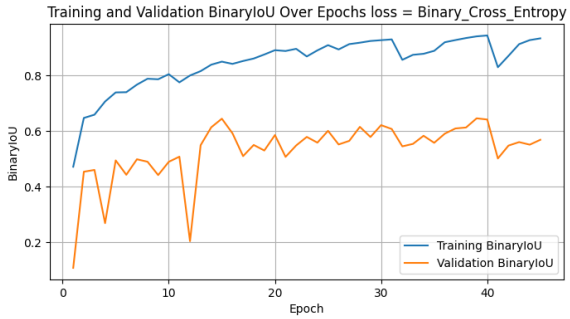


Figure 5.10: Predicted results vs Ground truth (Breast Dataset - U-Net)

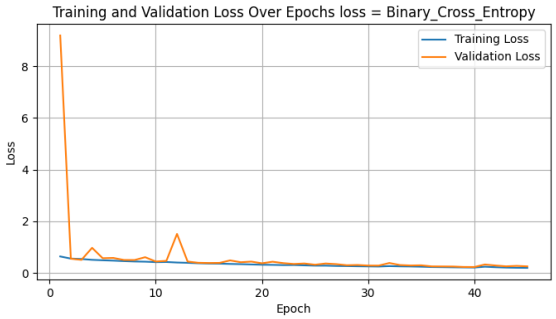
2. MultiRes-net architecture:
   The loss functions used are Binary Cross Entropy, Binary Focal Loss and Dice Loss

Table 5.6: Breast Dataset - MultiResU-Net test results

	Binary Cross Entr.	**Binary Focal Loss**	**Dice Loss**
Accuracy	0.9561	0.9399	0.9579
Recall	0.5513	0.5473	0.5407
Precision	0.5423	0.4909	0.5749
IoU	0.7836	0.7387	0.7878
Dice coeff.	0.5457	0.5151	0.5562
Loss	0.2413	0.0657	-0.3374

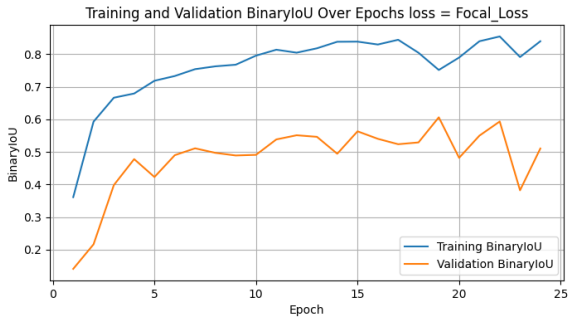


(a) IoU (Binary Cross Entropy)

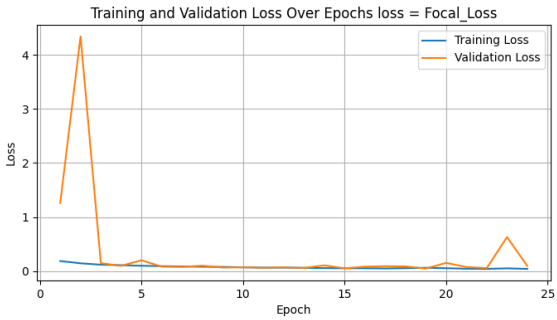


(b) Loss (Binary Cross Entropy)

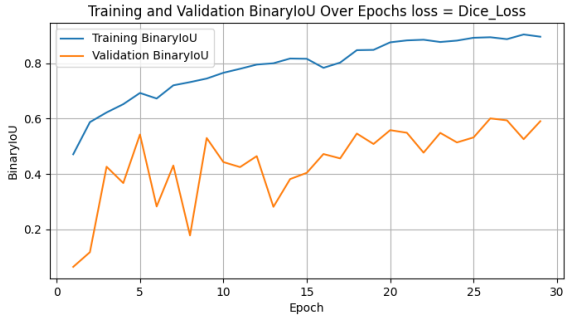
Figure 5.11: Breast Dataset - MultiResU-Net train graphs (Metric vs epoch)



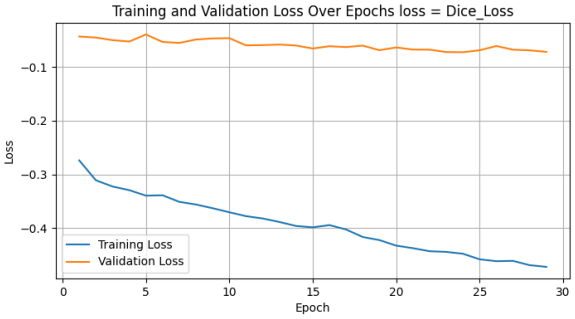
(a) IoU (Focal Loss)



(b) Loss (Focal Loss)



(c) IoU (Dice Loss)



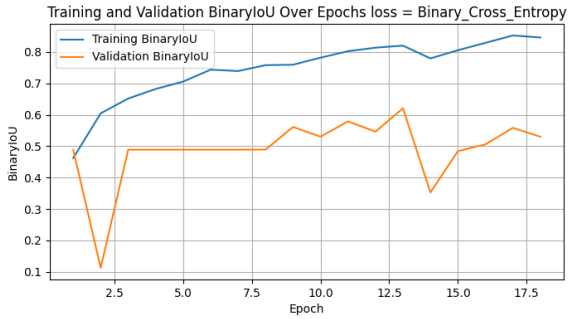
(d) Loss (Dice Loss)

Figure 5.12: Breast Dataset - MultiResU-Net train graphs (Metric vs epoch)

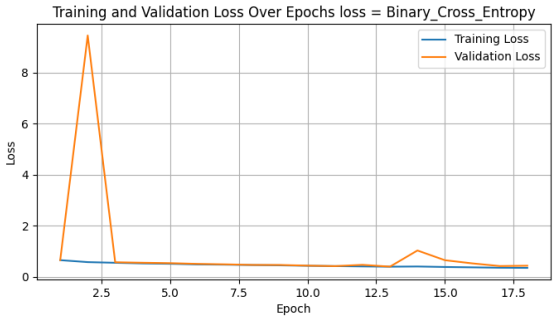
3. DCU-Net architecture:

Table 5.7: Breast Dataset - DCU-Net results

	Binary Cross Entr.	**Binary Focal Loss**	**Dice Loss**
Accuracy	0.9444	0.9083	0.9220
Recall	0.5330	0.4559	0.6082
Precision	0.5207	0.3829	0.3989
IoU	0.7462	0.6390	0.7087
Dice coeff.	0.5263	0.4044	0.4803
Loss	0.3834	0.0963	-0.3311

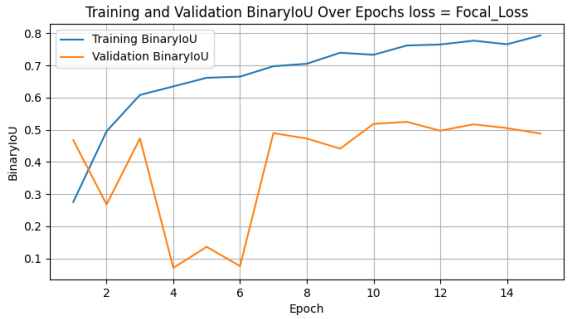


(a) IoU (Binary Cross Entropy)

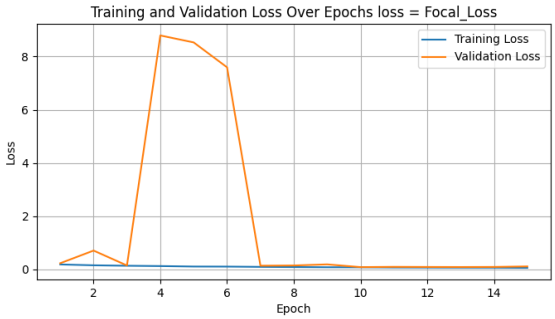


(b) Loss (Binary Cross Entropy)

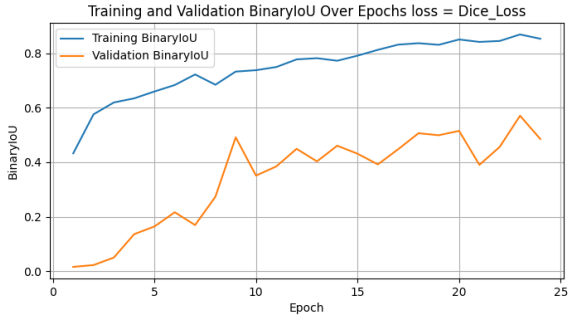
Figure 5.13: Breast Dataset - DCU-Net train graphs (Metric vs epoch)



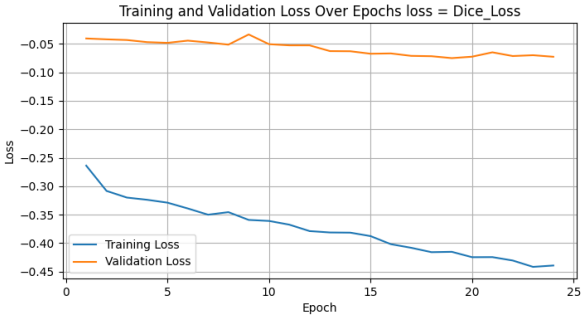
(a) IoU (Focal Loss)



(b) Loss (Focal Loss)



(c) IoU (Dice Loss)



(d) Binary Cross Entropy (Dice Loss)

Figure 5.14: Breast Dataset - DCU-Net train graphs (Metric vs epoch)

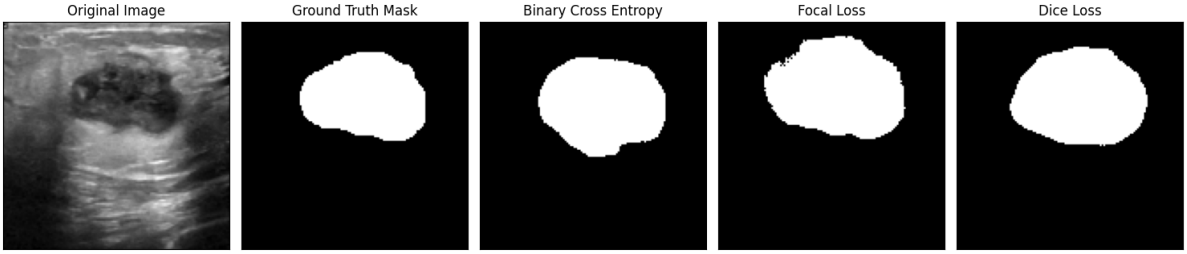


Figure 5.15: Predicted results vs Ground truth (Breast Dataset - DCU-Net)

5.3.5.1 Obervation

1. Binary Cross Entropy The MultiresUnet demonstrates superior performance compared to the other two models across four out of six evaluation metrics, notably excelling in the IoU metric, which holds particular significance in image segmentation tasks. Despite its overall success, the Unet architecture outperforms the MultiresUnet in terms of the loss metric and precision.

2. Focal Loss In this scenario, both the Unet and MultiresUnet models exhibit superior performance in three metrics each. The Unet model demonstrates better accuracy, precision, and loss, whereas the MultiresUnet model excels in recall, IoU, and Dice coefficient. To determine the most suitable model for this case, a decision must be made regarding the trade-off between emphasizing IoU or prioritizing the loss metric. This emphasizes the importance of considering specific task requirements and objectives when choosing the optimal model based on performance metrics.

3. Dice Loss The Multiresunet architecture exhibits superior performance compared to the other two architectures in the majority of the evaluated metrics, including IoU, recall, precision, accuracy, and dice coefficient. Notably, Multiresunet outperforms both architectures in these key aspects. However, it is worth noting that Unet surpasses Multiresunet in the loss metric. This suggests that while Multiresunet excels in various performance measures crucial for image segmentation tasks, Unet demonstrates a comparative advantage in minimizing the loss function. The choice between these architectures may depend on specific task requirements and the relative importance assigned to different performance metrics.

5.3.6 LiTS Dataset

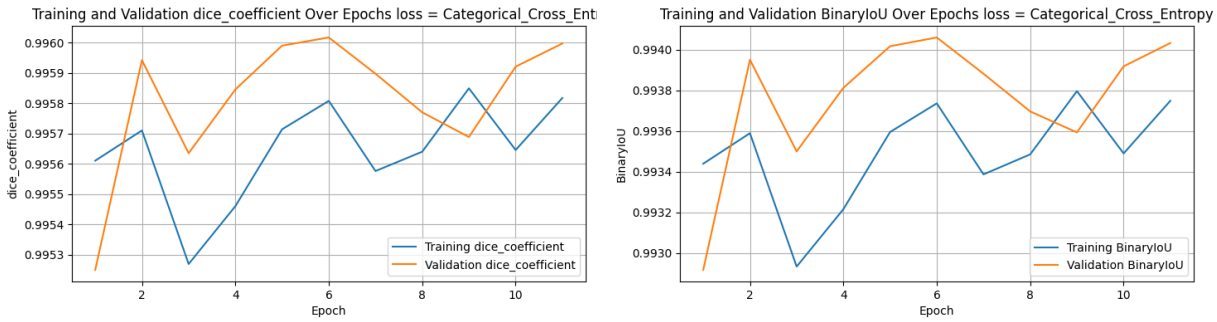
Transitioning from Binary to Multiclass Semantic Segmentation with the LiTS Dataset marked a pivotal shift. The dataset, featuring three classes—Background, Liver, and Tumor—required the adoption of Categorical Cross Entropy as the loss function. This strategic choice accommodates the nuances of multiclass segmentation, diverging from previous binary-centric approaches and underscoring the adaptability and sophistication of our segmentation strategy.

Table 5.8: Liver Dataset - Test Results

	UNet	**MultiResUNet**	**DCUNet**	**VU-Net**
Accuracy	0.9974	0.9982	0.9979	0.9934
Recall	0.9960	0.9973	0.9969	0.9895
Precision	0.9962	0.9974	0.9970	0.9908
IoU	0.9062	0.9373	0.9273	0.7964
Dice coeff.	0.9961	0.9973	0.9969	0.9901
Loss = Cat. Cross Entr.	0.0096	0.0073	0.0161	0.0428

Table 5.9: IoU for each class in Liver Dataset

	Class 1(Background)	**Class 2(Liver)**	**Class 3(Tumor)**
UNet	0.9942	0.9431	0.7813
MultiResUNet	0.9969	0.9605	0.8554
DCUNet	0.9954	0.9546	0.8318
VU-Net.	0.9847	0.8665	0.5381



(a) Dice Coeffeicient (Categorical Cross Entropy)

(b) Loss (Categorical Cross Entropy)

Figure 5.16: Liver Dataset - U-Net train graphs (Metric vs epoch)

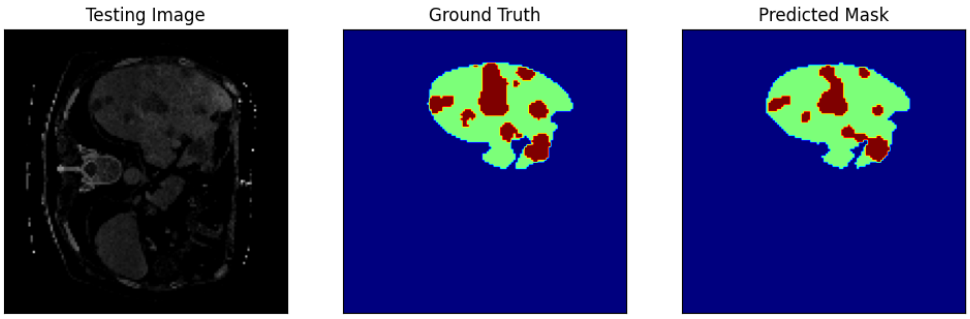
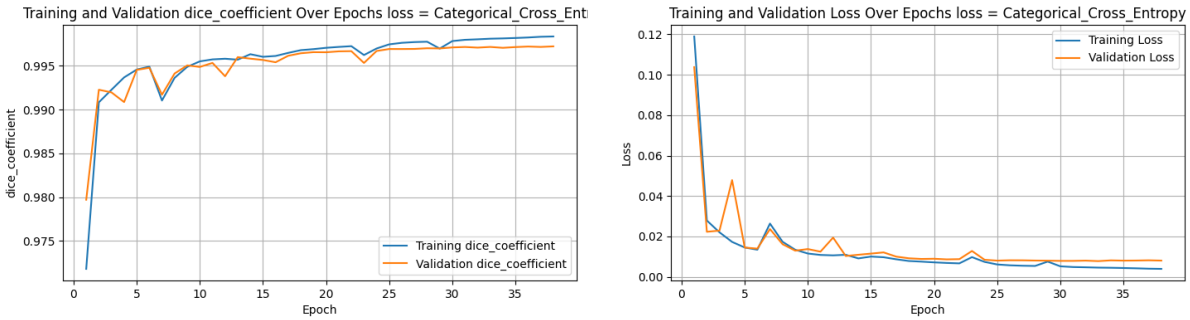


Figure 5.17: Predicted results vs Ground truth (Liver Dataset - U-Net)



(a) Dice Coefficient (Categorical Cross Entropy)

(b) Loss (Categorical Cross Entropy)

Figure 5.18: Liver Dataset - MultiResU-Net train graphs (Metric vs epoch)

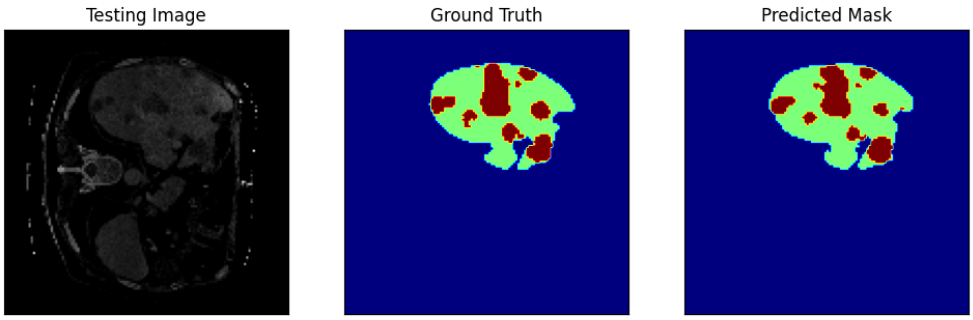
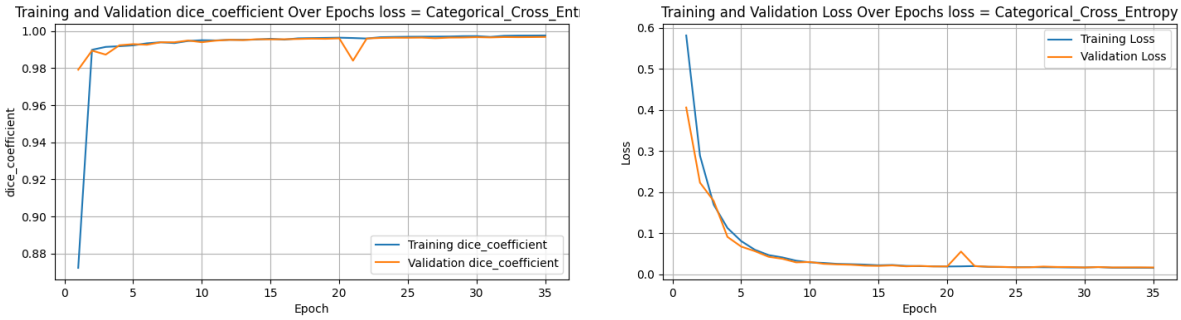


Figure 5.19: Predicted results vs Ground truth (Liver Dataset - MultiRes-Net)



(a) Dice Coefficient (Categorical Cross Entropy) (b) Loss (Categorical Cross Entropy)

Figure 5.20: Liver Dataset - DCU-Net train graphs (Metric vs epoch)

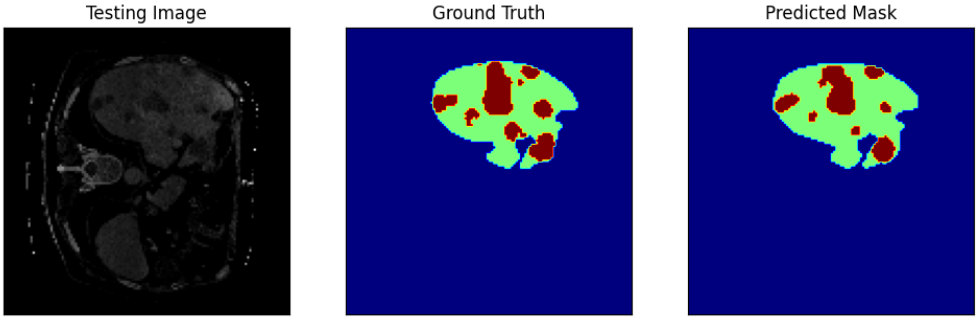
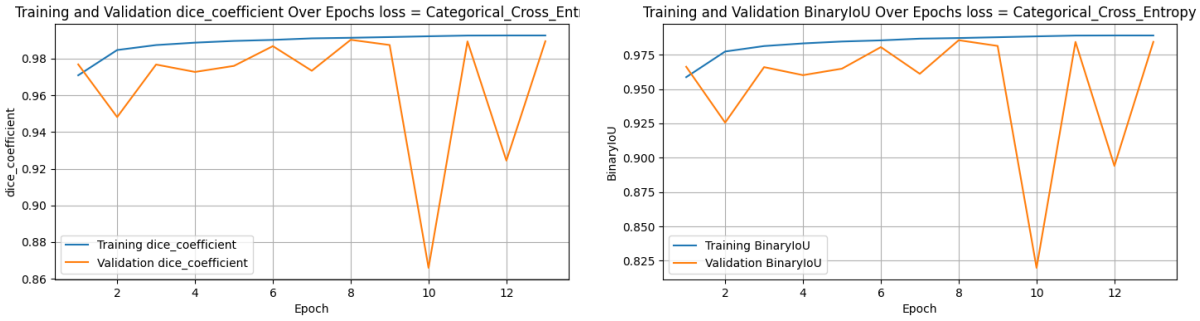
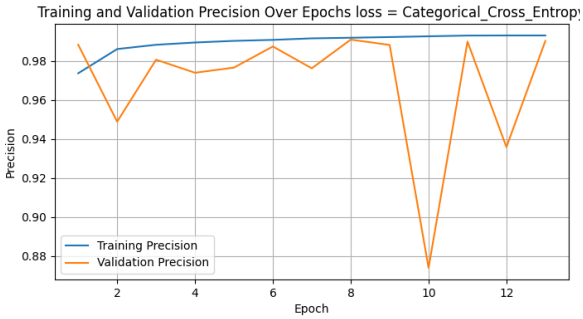


Figure 5.21: Predicted results vs Ground truth (Liver Dataset - DCU-Net)

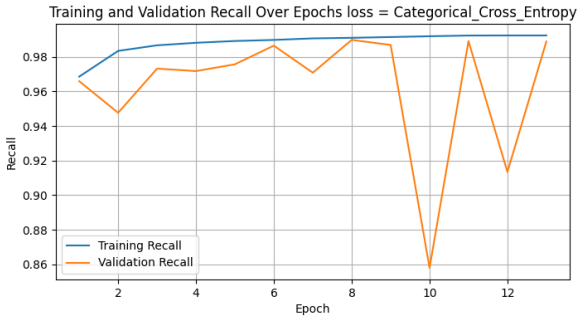


(a) Dice Coefficient (Categorical Cross Entropy) (b) Loss (Categorical Cross Entropy)

Figure 5.22: Liver Dataset - VU-Net train graphs (Metric vs epoch)



(a) Recall (Categorical Cross Entropy)



(b) Precision (Categorical Cross Entropy)

Figure 5.23: Liver Dataset - VU-Net train graphs (Metric vs epoch)

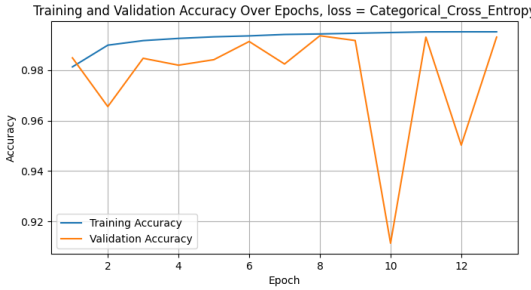


Figure 5.24: Accuracy (Categorical Cross Entropy)

Figure 5.25: Liver Dataset - VU-Net train graphs (Metric vs epoch)

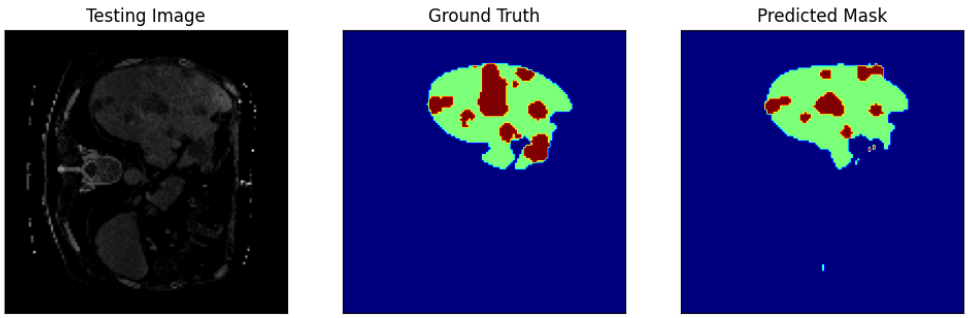


Figure 5.26: Predicted results vs Ground truth (Liver Dataset - VU-Net)

5.3.6.1 Obervation

1. In the evaluation of the Liver dataset, it is evident that the MultiResUnet architecture has demonstrated superior performance compared to all other architectures employed in the study. VU-Net, although slightly trailing behind, exhibits competitive results. To further enhance its effectiveness, minor improvements are required.

2. In the evaluation of the liver dataset, VU-Net exhibits commendable performance compared to three other models. A specific benchmark is drawn against the MultiResUnet model, which is considered the best-performing model on this dataset. The VU-Net demonstrates competitive results with only slight deviations in key performance metrics. Notably, the differences in accuracy (acc), recall, precision, and the dice coefficient are marginal, with values of 0.0048, 0.0078, 0.0066, and 0.0072, respectively. However, the intersection over union (IoU), a critical metric for segmentation tasks, exhibits a more pronounced difference, registering at 0.14 compared to the best model tested.

5.4 Results of Grad-CAM

5.4.1 Heart Dataset Grad-CAM results

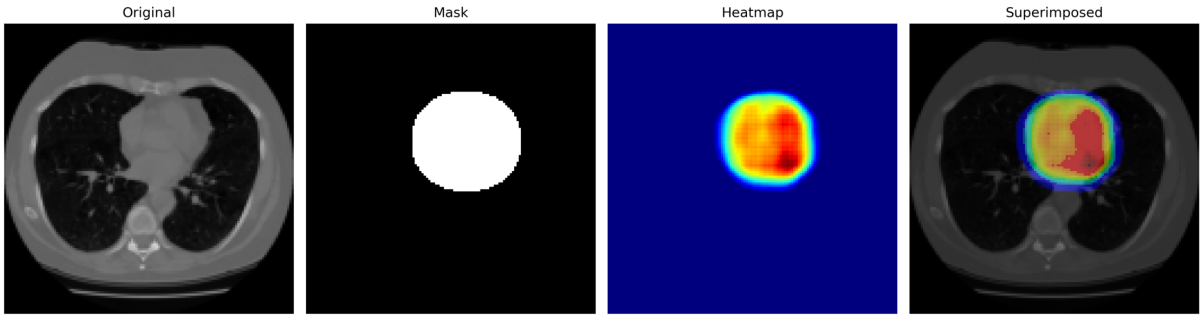


Figure 5.27: Grad-CAM on Heart - U-Net - Binary Cross Entropy

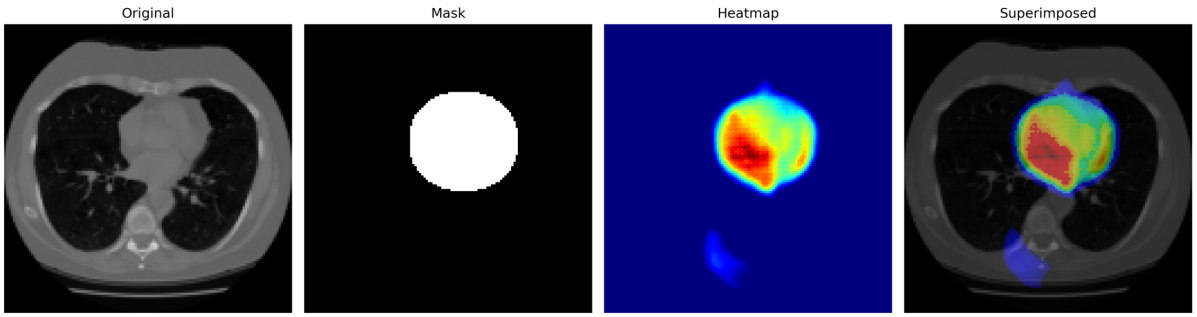


Figure 5.28: Grad-CAM on Heart - U-Net - Categorical Cross Entropy

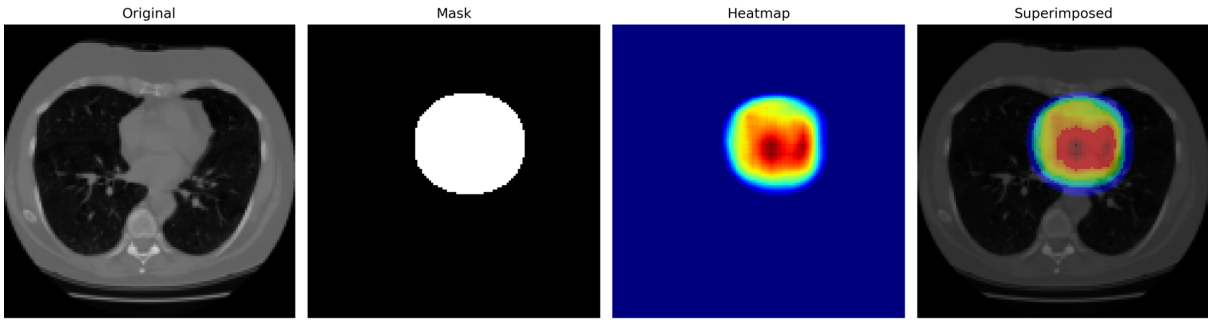


Figure 5.29: Grad-CAM on Heart - U-Net - Binary Focal Loss

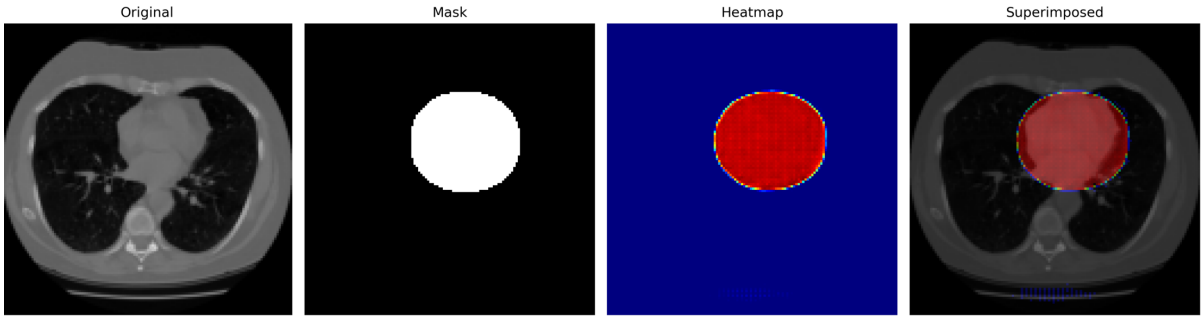


Figure 5.30: Grad-CAM on Heart - MultiResU-Net - Binary Cross Entropy

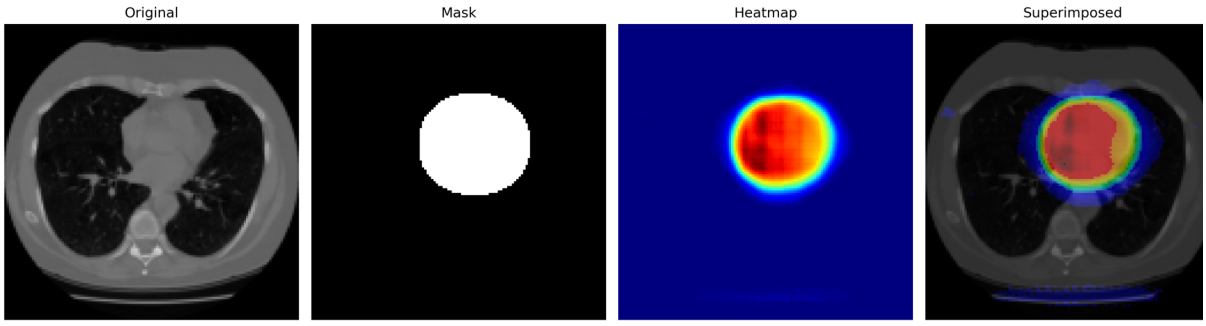


Figure 5.31: Grad-CAM on Heart - MultiResU-Net - Binary Focal Loss

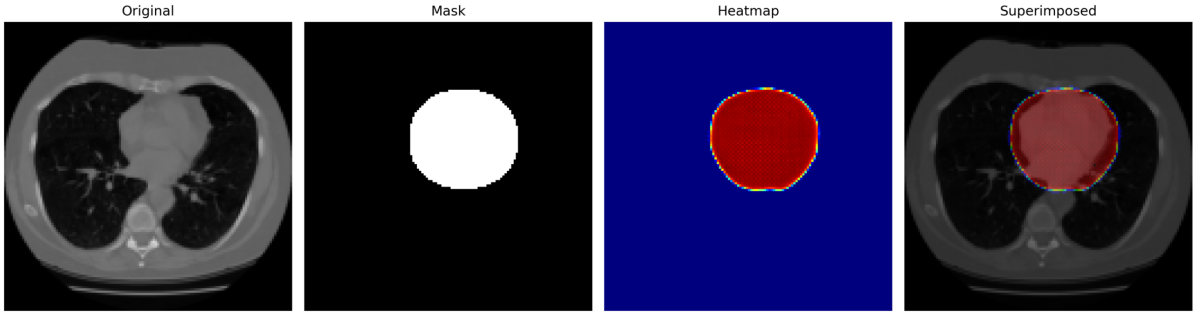


Figure 5.32: Grad-CAM on Heart - MultiResU-Net - Dice Loss

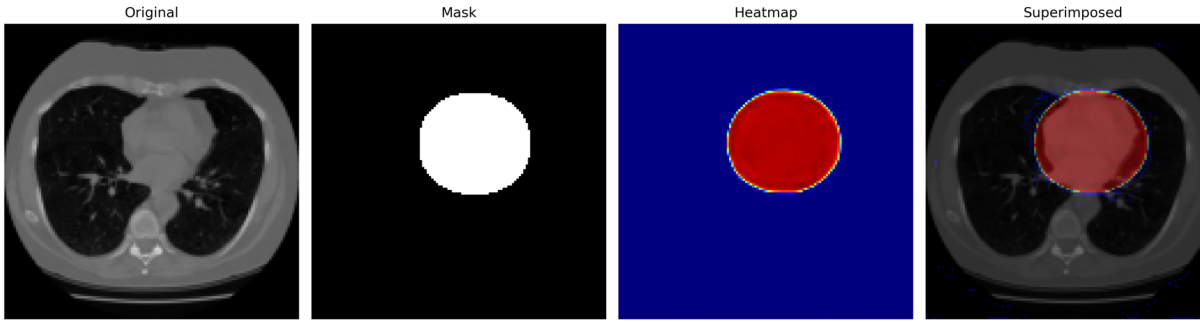


Figure 5.33: Grad-CAM on Heart - DCU-Net - Binary Cross Entropy

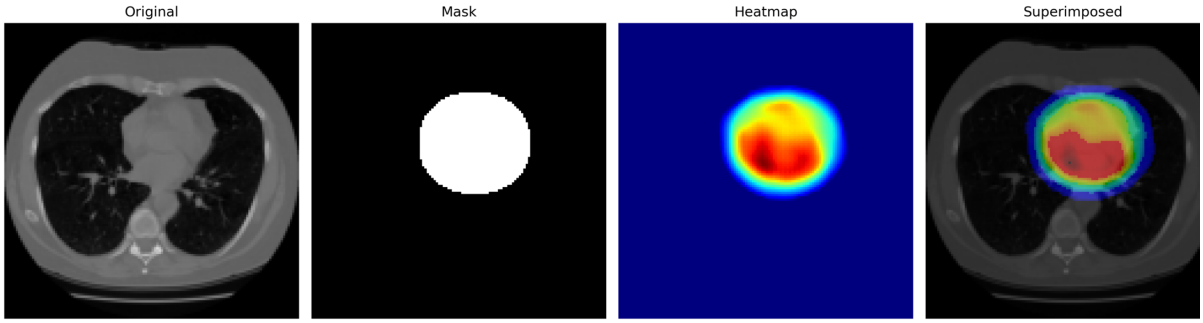


Figure 5.34: Grad-CAM on Heart - DCU-Net - Binary Focal Loss

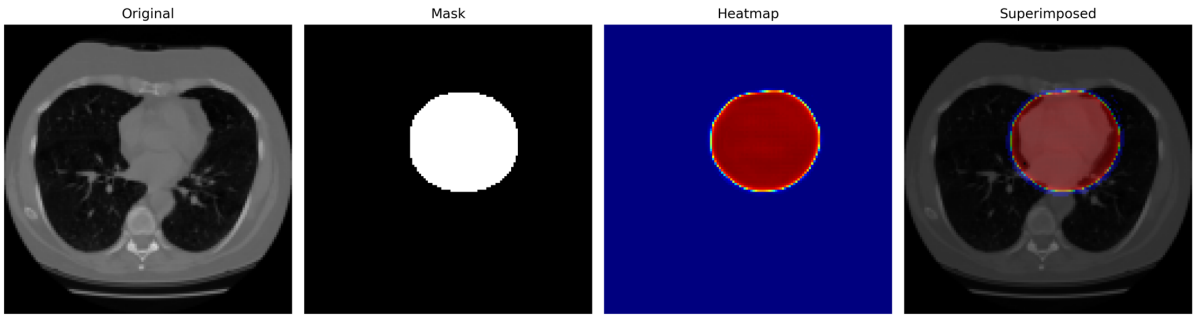


Figure 5.35: Grad-CAM on Heart - DCU-Net - Dice Loss

5.4.2 Obsevation from Grad-CAM on Heart Dataset

5.4.2.1 BCE

In our analysis of segmentation models on the Heart dataset using Binary Cross Entropy (BCE) as the loss function, GradCAM visualizations revealed distinct performance characteristics among the models, specifically U-Net, MultiResU-Net, and DCU-Net. The heatmaps indicated that MultiResU-Net and DCU-Net were particularly effective in focusing precisely on the heart. These models produced well-targeted heatmaps that closely aligned with the cardiac boundaries, suggesting superior segmentation accuracy compared to U-Net.

- **Comparison and Model Efficiency:** While U-Net adequately identified heart regions, its heatmaps were less focused, potentially including more surrounding tissue which could lead to less accurate segmentations. In contrast, MultiResU-Net and DCU-Net

demonstrated a higher level of specificity in delineating the heart, attributed to their architectural capabilities to capture multi-scale and contextual information.

- **Clinical Implications:** These observations underscore the advantages of using MultiResU-Net and DCU-Net for cardiac imaging tasks where precise segmentation is critical. Grad-CAM has proven essential in illustrating the models' effectiveness and guiding further refinements to enhance their diagnostic utility.

5.4.2.2 BFL

Our comparative analysis of DCU-Net, U-Net, and MultiResU-Net using Binary Focal Loss (BFL) as the loss function demonstrated varying degrees of efficiency in targeting the heart region. Notably, DCU-Net outperformed the other models in precisely focusing on the heart, showcasing its superior capability in segmenting cardiac structures accurately.

- **Model Ranking and Efficiency:** DCU-Net's effective targeting and segmentation precision place it at the forefront, followed by U-Net, which also demonstrated commendable performance but with slightly broader region identification. MultiResU-Net, while effective, ranked third in this specific task, suggesting that its enhancements for capturing multi-scale information might not align as effectively with the singular focus required for this dataset.

- **Clinical and Technical Implications:** These observations suggest that DCU-Net might be particularly suited for applications requiring high precision in cardiac imaging, such as detailed anatomical studies and diagnostic evaluations where exact boundary delineation is crucial. This ranking provides insights into selecting appropriate models based on specific clinical requirements and image characteristics.

5.4.2.3 Dice Loss

In our evaluation of segmentation models on the Heart dataset using Dice Loss, both MultiResU-Net and DCU-Net demonstrated high performance in accurately targeting and segmenting the heart region. This performance was validated through rigorous testing, which confirmed their effectiveness.

- **Performance and Focus:** The application of Dice Loss appears to enhance the segmentation capabilities of both MultiResU-Net and DCU-Net, with each model showing strong focus on the cardiac structures. This was evident in their ability to clearly delineate the heart from surrounding tissues, thus contributing to highly accurate segmentation results.

- **Clinical Relevance:** The success of MultiResU-Net and DCU-Net under Dice Loss conditions suggests their utility in clinical settings where precision in cardiac segmentation is vital. The testing results serve as a robust justification for their deployment in scenarios demanding high accuracy, such as in detailed diagnostic and therapeutic planning.

5.4.3 Breast Dataset Grad-CAM results

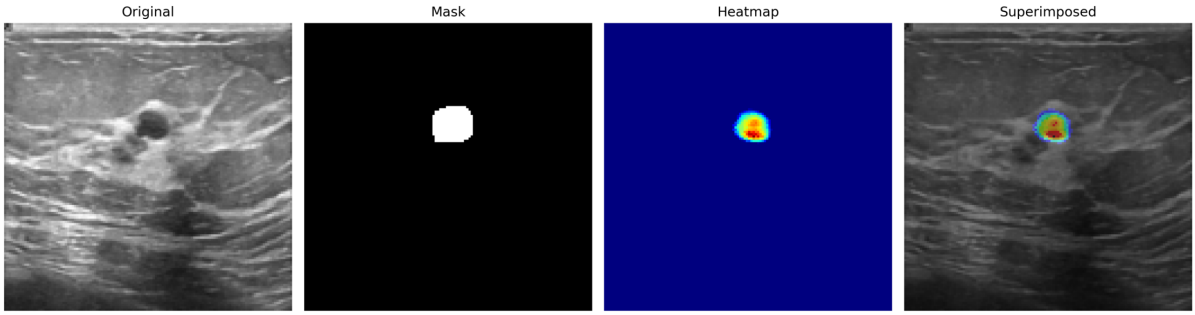


Figure 5.36: Grad-CAM on Breast - U-Net - Binary Cross Entropy

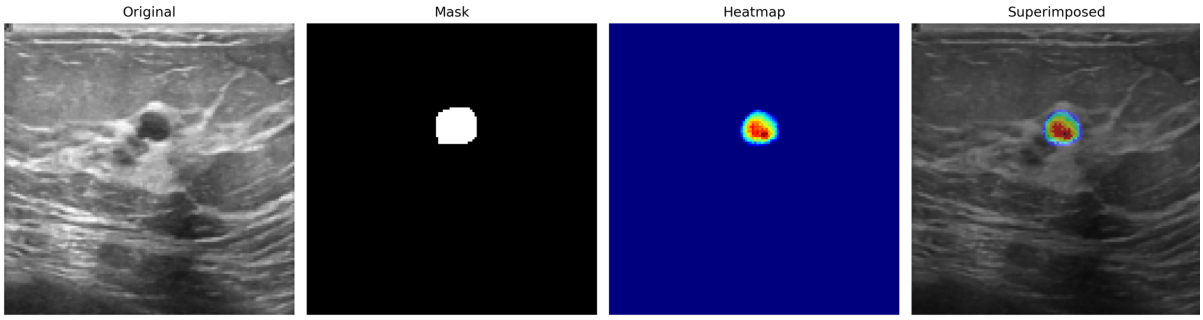


Figure 5.37: Grad-CAM on Breast - U-Net - Binary Focal Loss

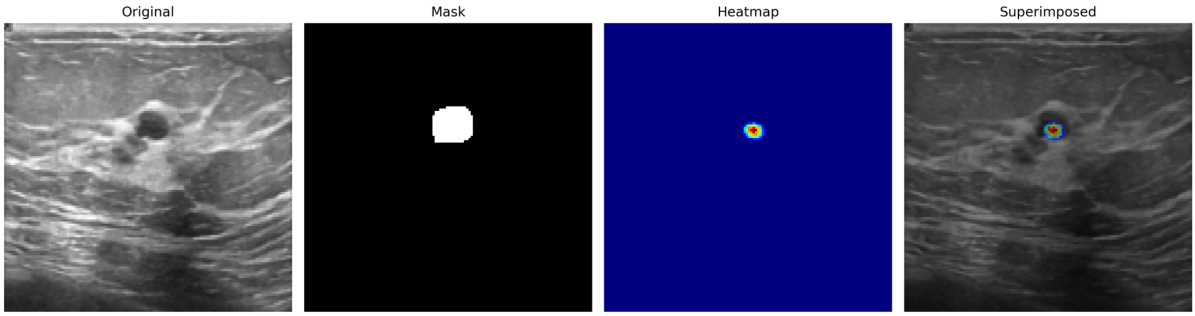


Figure 5.38: Grad-CAM on Breast - U-Net - Dice Loss

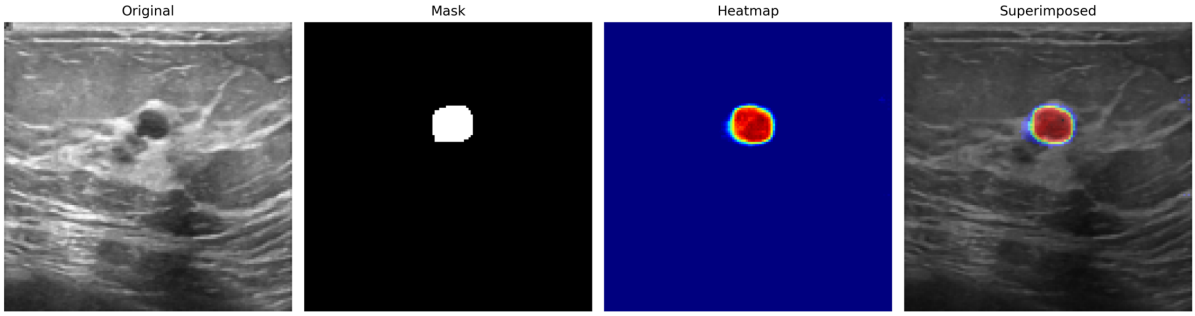


Figure 5.39: Grad-CAM on Breast - MultiResU-Net - Binary Cross Entropy

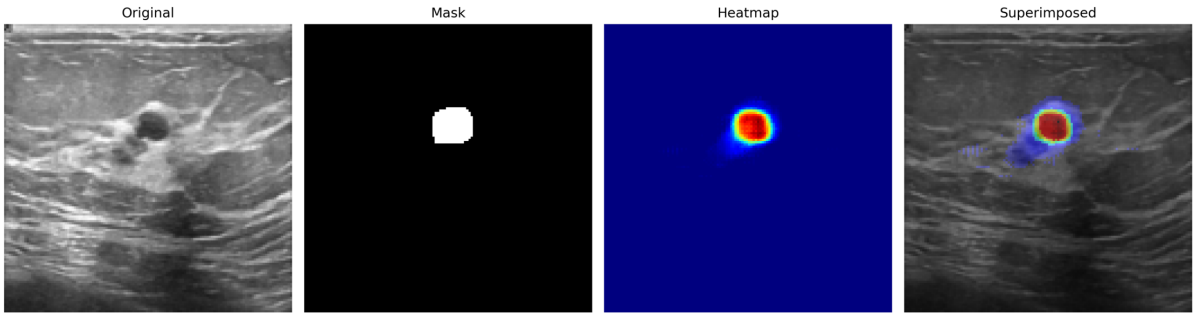


Figure 5.40: Grad-CAM on Breast - MultiResU-Net - Binary Focal Loss

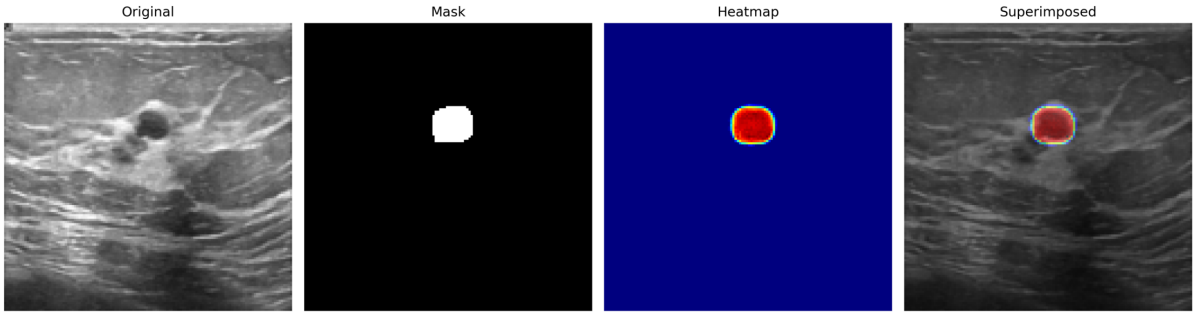


Figure 5.41: Grad-CAM on Breast - MultiResU-Net - Dice Loss

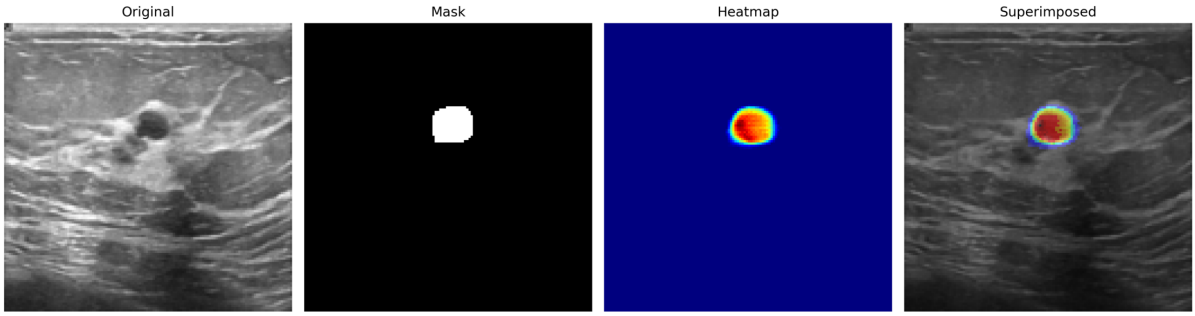


Figure 5.42: Grad-CAM on Breast - DCU-Net - Binary Cross Entropy

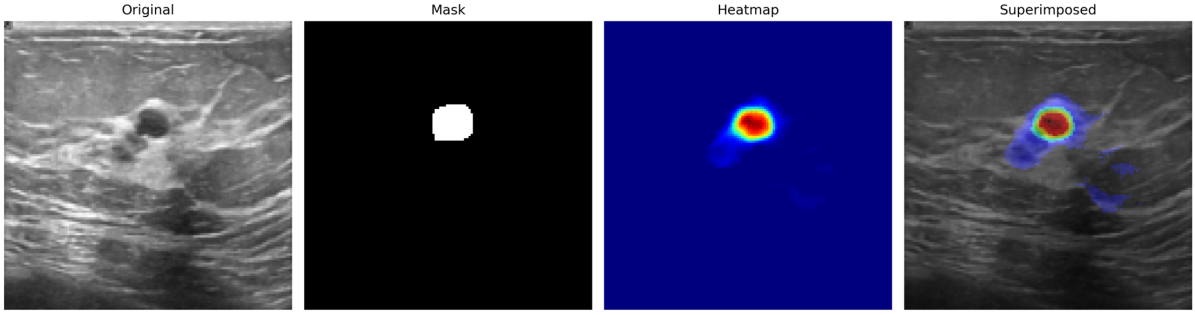


Figure 5.43: Grad-CAM on Breast - DCU-Net - Binary Focal Entropy

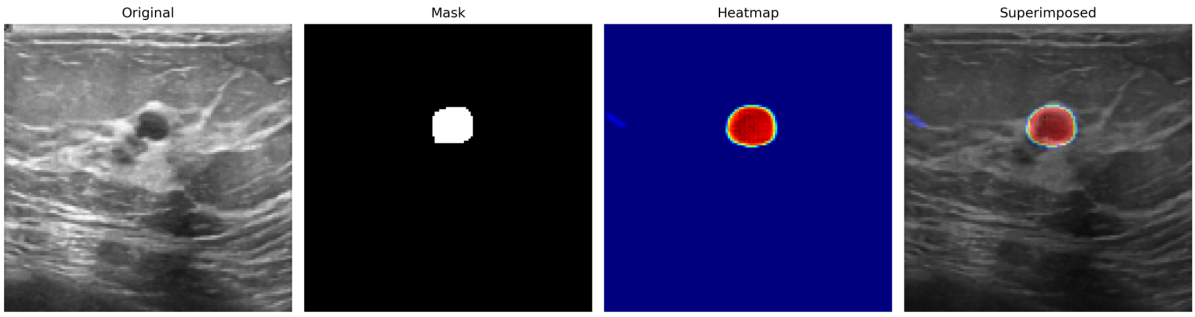


Figure 5.44: Grad-CAM on Breast - DCU-Net - Dice Loss

5.4.4 Obsevation from Grad-CAM on Breast Dataset

5.4.4.1 BCE

Our analysis of segmentation models on the Breast dataset, employing Binary Cross Entropy, highlighted notable differences in how each model targeted and segmented the tumor regions. Specifically, the heatmaps from MultiResU-Net and DCU-Net demonstrated a high degree of precision in focusing exclusively on the tumor areas, indicating effective and accurate segmentation.

- **Model Performance and Focus:** The heatmaps generated reveal that MultiResUNet has a high degree of focus on the tumour region, indicated by dark red coloration, suggesting precise targeting and effective concentration. This results in superior segmentation accuracy. DCUNet's heatmap shows an orange hue over the tumour area, reflecting a moderate level of focus. UNet's lighter colors and smaller focus area indicate less accurate and concentrated focus on the tumour. These observations explain why MultiResUNet outperforms both DCUNet and UNet, with the dark red heatmap highlighting its enhanced performance in accurately targeting and segmenting the tumour area.
- **Clinical Implications:** MultiResUNet's precise targeting and effective concentration on tumour regions, as shown by the dark red heatmap, can lead to better diagnosis and treatment planning, improving patient outcomes. DCUNet, with its moderate focus, provides reliable segmentation useful in settings where computational efficiency is crucial. UNet, while less focused, offers a baseline for improvement. Accurate and efficient tumour segmentation is vital in medical imaging, directly impacting the effectiveness of medical interventions.

5.4.4.2 BFL

Our analysis of segmentation models on the Breast dataset, employing Binary Focal Loss, highlighted notable differences in how each model targeted and segmented the tumor regions. Specifically, the heatmaps from MultiResU-Net and U-Net demonstrated a high degree of precision in focusing exclusively on the tumor areas, indicating effective and accurate segmentation.

- **Model Performance and Focus:** The heatmaps for MultiResU-Net and U-Net clearly showed that these models maintain a strict focus on the tumor regions without deviating. This targeted approach suggests that both models efficiently leverage the characteristics of Binary Focal Loss to enhance their specificity towards tumor-relevant features. In contrast, DCU-Net, while still effective, showed a tendency to include some surrounding non-tumor features in its segmentation process. This broader focus resulted in slightly lesser performance compared to MultiResU-Net and U-Net.

- **Clinical Implications:** The ability of MultiResU-Net and U-Net to precisely target tumor regions in breast imaging is crucial for clinical applications where accurate tumor delineation is necessary for effective treatment planning. The lesser performance of DCU-Net in this aspect suggests a need for further tuning of this model to improve its specificity for tumor regions. These observations guide model selection and refinement strategies for breast cancer imaging tasks, ensuring that the chosen models provide reliable and clinically useful results.

5.4.4.3 Dice Loss

When analyzing the performance of segmentation models on the Breast dataset using Dice Loss, both MultiResU-Net and DCU-Net showed exceptional ability to focus and segment the tumor regions effectively. This focus is critical for achieving high performance in segmentation tasks, as evidenced by the metrics obtained during model testing.

- **Performance and Accuracy:** The heatmaps generated by MultiResU-Net and DCU-Net clearly indicate a concentrated effort on the tumor regions within the breast images. This concentrated focus ensures that the segmentation is both precise and aligned with the actual tumor boundaries, contributing to the models' overall high performance. These models leverage Dice Loss effectively, optimizing the overlap between the predicted segmentation and the ground truth, which is crucial for medical imaging tasks where accurate tumor delineation is vital.

- **Comparison with U-Net:** In contrast, U-Net presented a notable limitation in this scenario. The heatmaps produced by U-Net were considerably smaller than the actual tumor regions, suggesting a lack of model sensitivity or inadequate learning of tumor characteristics. This misalignment implies that U-Net, under Dice Loss conditions, does not perform as effectively in capturing the full extent of tumor regions, resulting in lower performance metrics compared to MultiResU-Net and DCU-Net.

- **Clinical and Technical Implications:** The superior performance of MultiResU-Net and DCU-Net in focusing on tumor regions makes them preferable choices for clinical applications in breast cancer diagnosis and treatment planning. The underperformance of U-Net highlights the need for model adjustments or alternative strategies when using this model for tumor segmentation in breast tissue, possibly involving parameter tuning or more focused training on tumor features. These observations not only guide the selection and optimization of models for specific clinical tasks but also underscore the importance of choosing the right loss function to enhance model performance in medical image segmentation.

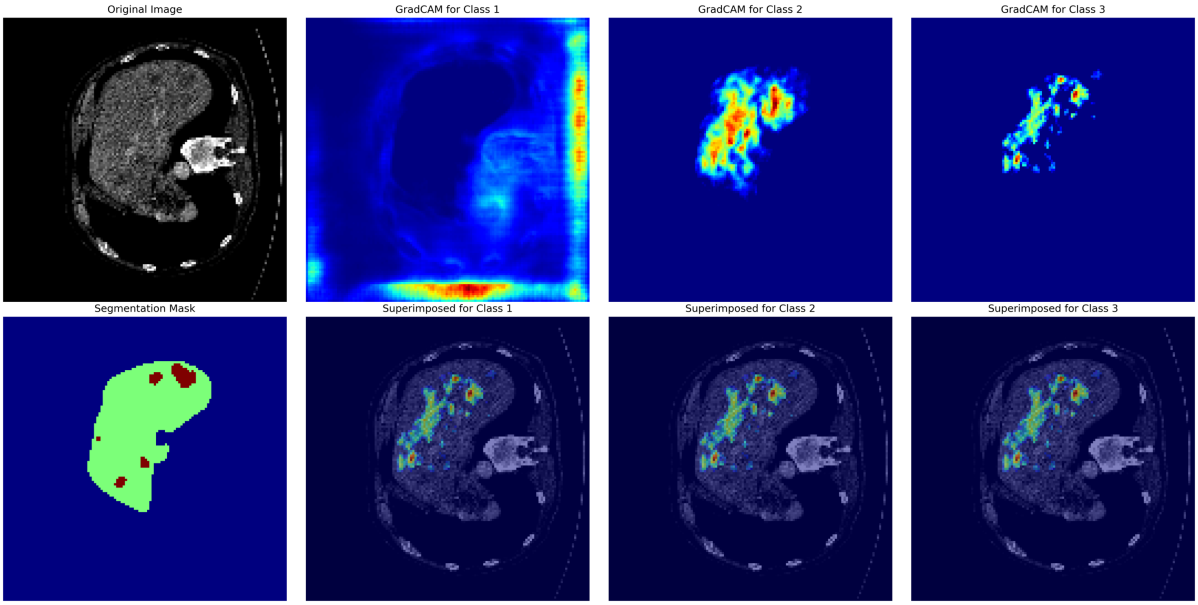


Figure 5.45: Grad-CAM on LiTS - U-Net - Categorical Cross Entropy

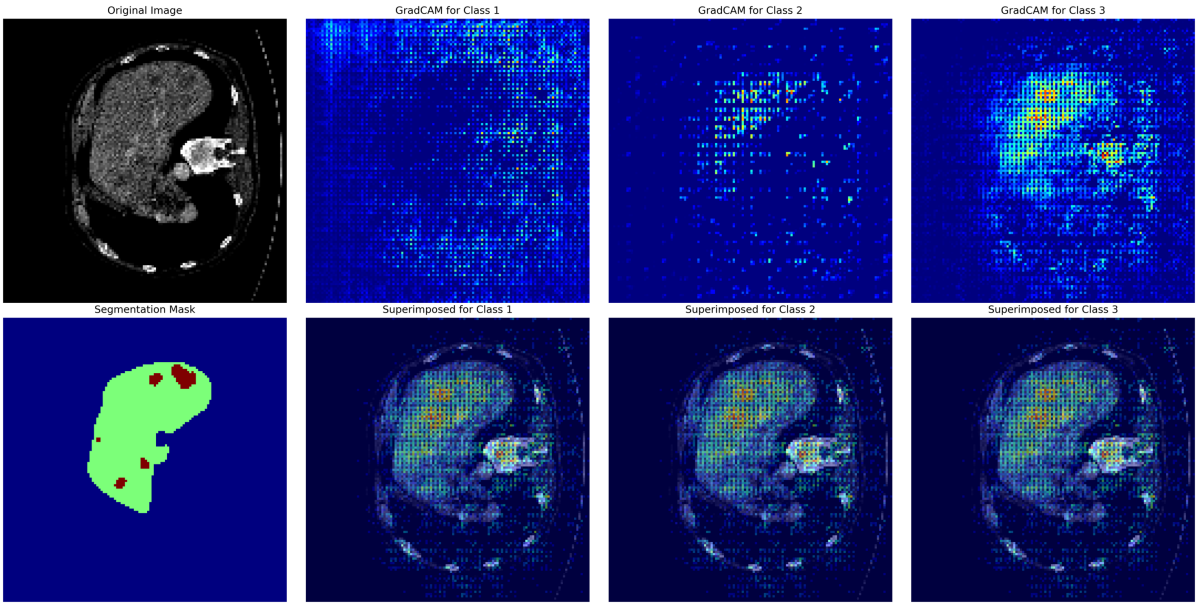


Figure 5.46: Grad-CAM on LiTS - MultiResU-Net - Categorical Cross Entropy

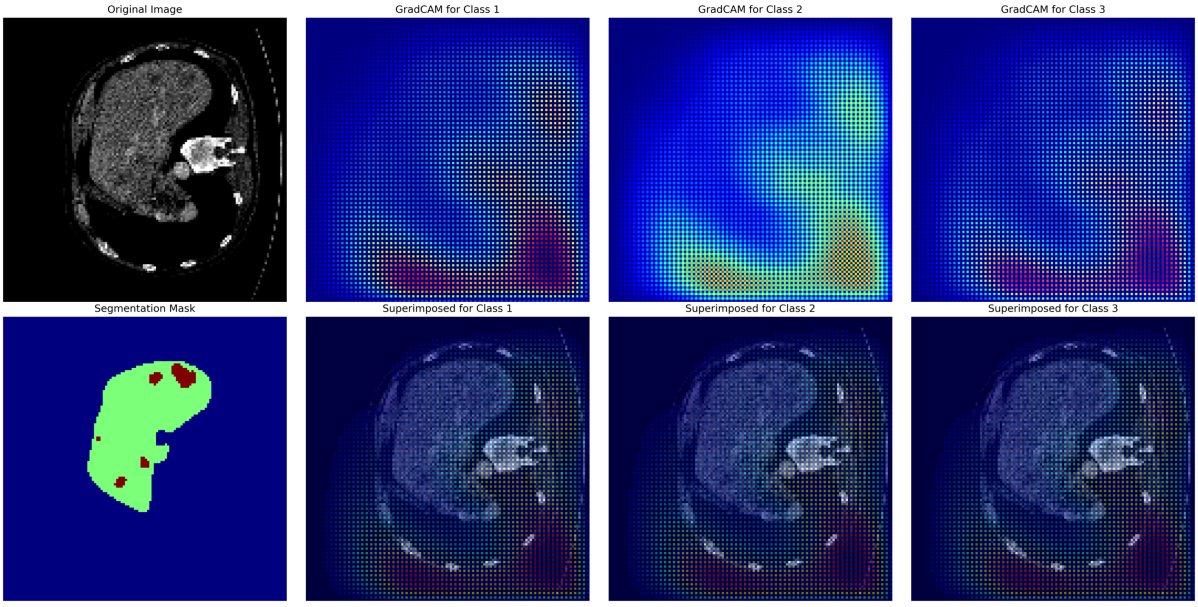


Figure 5.47: Grad-CAM on LiTS - DCU-Net - Categorical Cross Entropy

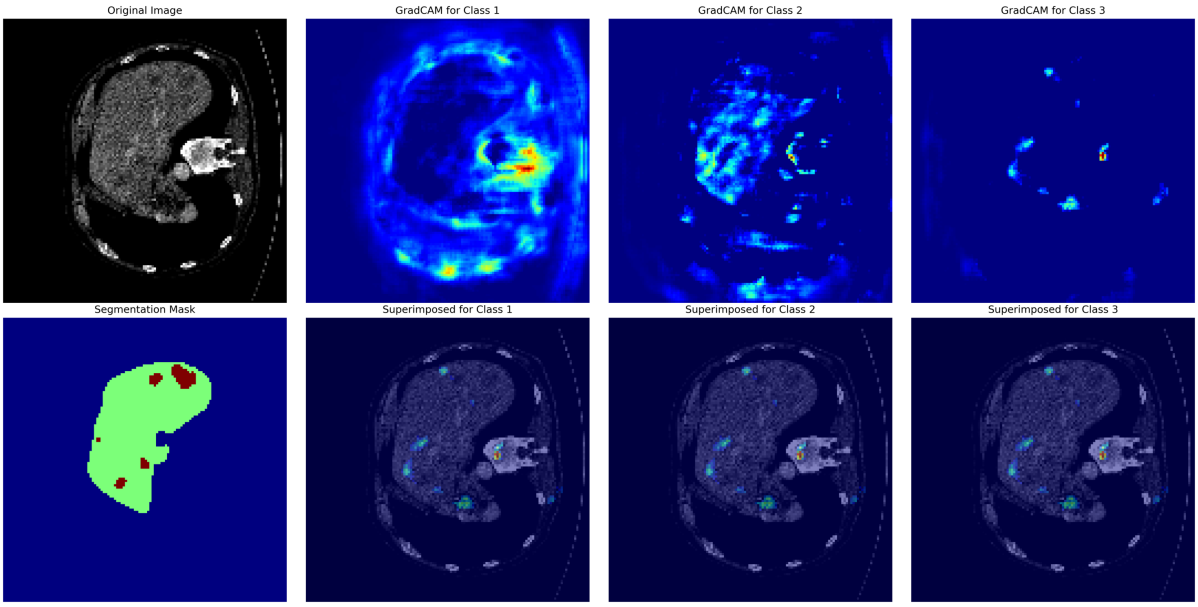


Figure 5.48: Grad-CAM on LiTS - VU-Net - Categorical Cross Entropy

5.4.5 Obsevation from Grad-CAM on LiTS Dataset

In the analysis of model performance on the LiTS dataset using Categorical Cross Entropy, significant insights were derived from the heatmaps generated by MultiResU-Net, U-Net, VUNet and DCUNet. The observations focused primarily on how effectively each model identified and segmented the liver and tumor regions.

5.4.5.1 Categorical Cross Entropy

- **Liver Segmentation Observations:**MultiResU-Net demonstrated superior performance in accurately identifying and focusing on the liver region. The heatmaps show that

MultiResU-Net effectively targets the liver without encroaching on nearby organs, maintaining a sharp focus that enhances segmentation precision. This precise targeting is evidenced by well-contained heatmap colors specifically within the liver boundaries. In comparison, U-Net and VUNet also recognized the liver region, but their focus was less concentrated than that of MultiResU-Net. U-Net, in particular, showed frequent deviations from the liver area, which could lead to less accurate segmentation results.

- **Tumor Segmentation Observations:** When it comes to tumor segmentation within the liver, MultiResU-Net again stands out by displaying highly concentrated and targeted focus, marked by vibrant red and orange patches on the heatmaps, indicative of intense model attention on the tumor areas. This targeted focus directly correlates with the best performance metrics among the models. U-Net, while capable of identifying tumor regions, exhibited less concentration in its heatmap visualization compared to MultiResU-Net, impacting its segmentation effectiveness. VUNet, although focusing decently on the tumor, did not match the intensity or preciseness of MultiResU-Net's segmentation capabilities.

- **Comparative Performance:** DCUNet, while generally effective in targeting the tumor region, showed occasional deviations that included non-tumor areas within its focus, leading to a reduced Intersection over Union (IoU) and impacting its overall segmentation performance. This contrasted sharply with MultiResU-Net's consistent accuracy and focus, underscoring MultiResU-Net's superior capability in both liver and tumor segmentation tasks.

  These observations collectively highlight MultiResU-Net's exceptional performance in segmenting complex anatomical and pathological structures within the LiTS dataset, supported by Categorical Cross Entropy loss. Its ability to maintain focus and precision in both liver and tumor regions justifies its selection for tasks requiring high accuracy and detailed anatomical delineation, making it the most effective model among those tested.

CHAPTER 6: CONCLUSION & FUTURE WORK

In this study, we explored the application of various segmentation models—U-Net, MultiResU-Net, DCU-Net, and the proposed VU-Net—across three distinct medical imaging datasets: Heart CT scans, Breast Ultrasound images, and the LiTS17 dataset. Our analysis incorporated the use of Grad-CAM to visualize the model's focus and segmentation accuracy, providing deeper insights into their performance.

6.1 Heart Dataset

- **Binary Cross Entropy:** MultiResU-Net demonstrated superior performance in four out of six metrics, particularly excelling in IoU. However, U-Net outperformed MultiResU-Net in the loss metric.
- **Focal Loss:** DCU-Net emerged as the best-performing model across four out of six metrics, highlighting its effectiveness in segmentation tasks requiring high precision.
- **Dice Loss:** MultiResU-Net again proved to be the most effective, dominating five out of six metrics.

6.2 Heart Dataset

- **Binary Cross Entropy:** MultiResU-Net excelled in four out of six metrics, particularly in IoU, while U-Net performed better in the loss metric.
- **Focal Loss:** Both U-Net and MultiResU-Net showed strengths in different metrics, necessitating a trade-off decision based on specific task requirements.
- **Dice Loss:** MultiResU-Net outperformed the other models in the majority of metrics, though U-Net showed an advantage in the loss metric.

6.3 LiTS Dataset

- **Categorical Cross Entropy:** MultiResU-Net consistently demonstrated superior performance in both liver and tumor segmentation tasks. VU-Net showed competitive results but required minor improvements to match MultiResU-Net's precision.

6.4 Grad-CAM

The Grad-CAM visualizations confirmed that MultiResU-Net and DCU-Net provide highly focused and accurate segmentations, making them suitable for clinical applications that demand precision. On the other hand the VU-Net also performed decently on the dataset. U-Net, while effective, often showed less concentration and accuracy in segmenting target regions.

6.5 Future Scope

The findings from this study highlight several areas for future research and development:

- **Model Refinement and Optimization:** The VU-Net requires further refinement. The GradCAM implementation can be used to further analyze the behaviour of the working of the model, with the help of which we can improve the model metrics. While MultiResU-Net and DCU-Net have shown superior performance, there is potential to further optimize these models, particularly in terms of computational efficiency and scalability. Fine-tuning hyperparameters and exploring more advanced architectural modifications could yield even better segmentation results.
- **Integration of Advanced Loss Functions:** Future work could explore the integration of more sophisticated loss functions that combine the strengths of Binary Cross Entropy, Focal Loss, and Dice Loss. This hybrid approach might enhance model performance across diverse medical imaging tasks.
- **Enhancements in VU-Net:** The proposed VU-Net model has shown promising results but requires further improvements. Future research could focus on refining its architecture, particularly its ability to handle multi-class segmentation tasks with greater accuracy and efficiency. The GradCAM implementation can be used to further analyze the behaviour of the working of the model, with the help of which we can improve the model metrics.
- **Real-World Clinical Trials:** While our study provides a robust evaluation of model performance, real-world clinical trials are essential to validate these findings. Collaborating with healthcare institutions to test these models in actual diagnostic and treatment planning scenarios could provide valuable insights and feedback.
- **Application to Diverse Medical Imaging Modalities:** Expanding the application of these models to other medical imaging modalities, such as MRI, PET scans, and X-rays, could demonstrate their versatility and potential for broader clinical adoption.
- **Explainability and Interpretability:** Continuing to develop and refine explainability techniques like Grad-CAM will be crucial. Enhancing the interpretability of model outputs can build greater trust among clinicians and facilitate more informed decision-making in medical practice.

In conclusion, this study underscores the importance of selecting the appropriate segmentation model based on specific clinical requirements and task characteristics. MultiResU-Net and DCU-Net have demonstrated exceptional performance and potential for significant impact in medical imaging, paving the way for future advancements in disease diagnosis and treatment planning.

CHAPTER 7: LIST OF PUBLICATIONS

- *VU-Net: An Explainable AI Approach for Liver Segmentation* 15th International IEEE Conference on Computing Communication and Networking Technologies (ICCCNT) 2024 (Communicated).
- *Employing Grad-CAM in DL models for Tumour Segmentation and Visual Explanation: An Empirical Study* - 5th International Conference on Data Science and Application (ICDSA) 2024 (Communicated)

REFERENCES

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*. https://arxiv.org/abs/1802.06955

Bishop, C. M. (2006a). *Pattern recognition and machine learning*. Springer.

Bishop, C. M. (2006b). *Pattern recognition and machine learning*. Springer.

Boykov, Y. Y., & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV)*, *1*, 105–112. https://doi.org/10.1109/ICCV.2001.937505

Breast ultrasound images dataset [Retrieved from https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset]. (2020). https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset

Bryson, A. E., & Ho, Y.-C. (1969). *Applied optimal control: Optimization, estimation, and control*. Blaisdell Publishing Company.

Chollet, F. (2017). *Deep learning with python*. Manning Publications.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. Lecture Notes in Computer Science*, *9901*, 424–432. https://doi.org/10.1007/978-3-319-46723-8_49

Ct heart segmentation dataset [Retrieved from https://www.kaggle.com/datasets/nikhilroxtomar/ct-heart-segmentation]. (2021). https://www.kaggle.com/datasets/nikhilroxtomar/ct-heart-segmentation

Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*. https://arxiv.org/abs/1702.08608

Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., & Heng, P.-A. (2017). 3d deeply supervised network for automatic liver segmentation from ct volumes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6652–6660. https://doi.org/10.1109/CVPR.2017.705

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, *25*(1), 24–29. https://doi.org/10.1038/s41591-018-0316-z

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/BF00344251

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, *1*(2), 119–130. https://doi.org/10.1016/0893-6080(88)90014-7

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246–263. https://doi.org/10.2307/2841583

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *arXiv preprint arXiv:1806.00069*. https://arxiv.org/abs/1806.00069

Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing* (3rd ed.). Prentice Hall.

Goodfellow, I., Bengio, Y., & Courville, A. (2016a). *Deep learning*. MIT Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016b). *Deep learning*. MIT Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*, 2672–2680.

Gunashekar, S., & Ramesh, D. (2019). Interpretable deep learning model for prostate tumour segmentation in mri. *Proceedings of the International Conference on Medical Imaging and Informatics (ICMII)*, 1–5. https://doi.org/10.1109/ICMII.2019.8722745

Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, *29*(1), 100–132. https://doi.org/10.1016/S0734-189X(85)90153-7

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, *35*, 18–31. https://doi.org/10.1016/j.media.2016.05.004

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2961–2969. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*(1), 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837

Hubel, D. H., & Wiesel, T. N. (1977). Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *198*(1130), 1–59. https://doi.org/10.1098/rspb.1977.0085

Ibtehaz, N., & Rahman, M. S. (2020a). Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural Networks*, *121*, 74–87. https://doi.org/10.1016/j.neunet.2019.08.025

Ibtehaz, N., & Rahman, M. S. (2020b). Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural Networks*, *121*, 74–87.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1175–1183. https://doi.org/10.1109/CVPRW.2017.156

Jin, D., Xu, Z., Tang, Y., Harrison, A. P., George, K., & Lu, L. (2018). Ct-realistic lung nodule simulation from 3d conditional generative adversarial networks for robust lung segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 732–740. https://doi.org/10.1007/978-3-030-00928-1_82

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331. https://doi.org/10.1007/BF00133570

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, *25*, 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Li, S., Dong, W., Guo, Y., & Yu, L. (2020). Visual interpretability for deep learning: A survey. *Proceedings of the International Conference on Computer Vision (ICCV)*, 690–698. https://doi.org/10.1109/ICCV47803.2021.00076

Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., & Heng, P.-A. (2018). H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE Trans-*

actions on Medical Imaging, *37*(12), 2663–2674. https://doi.org/10.1109/TMI.2018.2845918

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Lits – liver tumor segmentation challenge (lits17) organized in conjunction with isbi 2017 and miccai 2017 [Retrieved from https://www.kaggle.com/datasets/andrewmvd/lits-png]. (2017). https://www.kaggle.com/datasets/andrewmvd/lits-png

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Ma, J., Deng, Y., Ma, Z., Mao, K., & Chen, Y. (2021). A liver segmentation method based on the fusion of vnet and wgan. *Computational and Mathematical Methods in Medicine*, *2021*, Article ID 5536903. https://doi.org/10.1155/2021/5536903

Maier, A., Syben, C., Lasser, T., & Riess, C. (Eds.). (2019). *Machine learning for medical image reconstruction*. Springer.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016a). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571. https://doi.org/10.1109/3DV.2016.79

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016b). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *3D Vision (3DV), 2016 Fourth International Conference on*, 565–571.

Mortazi, A., Bagci, U., & Foruzan, A. H. (2017). Cardiac segmentation in short-axis mri using fully convolutional networks and conditional random fields. *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, 121–124. https://doi.org/10.1109/ISBI.2017.7950473

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814.

Nielsen, M. A. (2015). *Neural networks and deep learning: A visual introduction to deep learning*. Determination Press.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. (2018). Attention u-net: Learning where to look for the pancreas. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 509–517.

Powers, D. M. (2011). *Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation*. Springer.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Ronneberger, O., Fischer, P., & Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Ronneberger, O., Fischer, P., & Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K.-R. (2019). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1911.02508*.

Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*. https://arxiv.org/abs/1708.08296

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626. https://doi.org/10.1109/ICCV.2017.74

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Springer.

Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine*, *25*(1), 44–56. https://doi.org/10.1038/s41591-018-0300-7

Wang, H., Schumann, J., Wiesner, M., & Navab, N. (2019). An analytical study on deep learning for diffusion mri. *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 685–693. https://doi.org/10.1007/978-3-030-32248-9_73

Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences* [Doctoral dissertation]. Harvard University.

Yap, M. H., Edirisinghe, E. A., & Cross, S. S. (2010). Breast ultrasound image segmentation using level set method. *Proceedings of the Fourth International Conference on Functional Imaging and Modeling of the Heart (FIMH)*, 237–246. https://doi.org/10.1007/978-3-642-13824-1_29

Zaitoun, N. H., & Aqel, M. J. (2015). *Medical image segmentation: Techniques and applications*. Springer.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929. https://doi.org/10.1109/CVPR.2016.319

Zijdenbos, A. P., Forghani, R., & Evans, A. C. (2002). Automatic" pipeline" analysis of 3-d mri data for clinical trials: Application to multiple sclerosis. *IEEE transactions on medical imaging*, *21*(10), 1280–1291.