TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

Thesis no: 075/MSICE/002

Body map based wound image classification using deep learning

by

Bibek Khanal

A THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

Department of Electronics and Computer Engineering

Lalitpur, Nepal

May, 2023

Body Map based Wound Image Classification using Deep Learning

by

Bibek Khanal

075/MSICE/002

Thesis Supervisor

Prof. Dr. Shashidhar Ram Joshi

Dean, Institute of Engineering

Professor, Department of Electronics and Computer Engineering

Pulchowk Campus

A thesis report submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information and Communication Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

May, 2023

COPYRIGHT©

DECLARATION

I declare that the work hereby submitted for Master of Science in Information and Communication Engineering (MSICE) at IOE, Pulchowk Campus entitled **"Body Map based Wound Image Classification using Deep Learning"** is my own work and has not been previously submitted by me at any university for any academic award.

I authorize IOE, Pulchowk Campus to lend this thesis to other institution or individuals for the purpose of scholarly research.

Bibek Khanal
075/MSICE/002
May, 2023

RECOMMENDATION

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled **"Body Map based Wound Image Classification using Deep Learning"**, submitted by **Bibek Khanal** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Information and Communication Engineering".**

.....

Supervisor: Prof. Dr. Shashidhar Ram Joshi
Dean,
Institute of Engineering, Tribhuvan University.

.....

External Examiner: Dr. Prakash Poudyal
Asst. Professor, Department of Computer Science and Engineering,
Kathmandu University

.....

Committee Chairperson: Dr. Baburam Dawadi
Program Coordinator, M.Sc. in Information and Communication Engineering,
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University.

Date: May, 2023

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Body Map based Wound Image Classification using Deep Learning"**, submitted by Bibek Khanal in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Information and Communication Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

.....

Dr. Jyoti Tandukar

Head of the Department

Department of Electronics and Computer Engineering

Pulchowk Campus

Institute of Engineering

Tribhuvan University

Nepal

ACKNOWLEDGEMENT

It has been a real privilege for me to prepare my dissertation, and I would like to take this opportunity to thank everyone who has made it possible. I would like to express my heartfelt gratitude and heartfelt thanks to my Supervisor **Prof. Dr. Shashidhar Ram Joshi**, Dean, Institute of Engineering, Tribhuvan University, who has provided guidance and constant encouragement throughout the course of my study and in the preparation of this dissertation with his extensive knowledge and professional expertise. Working under him during my entire post graduation has been an honour and a truly inspirational experience to cherish.

I would also like to present my acknowledgement to respected program co-ordinator, **Dr. Baburam Dawadi** for his feedback and providing this precious opportunity. I would also like to present my acknowledgement to respected **Dr. Dibakar Raj Pant, Dr. Surendra Shrestha, Prof. Dr. Subarna Shakya, Dr. Nanda Bikram Adhikari, Dr. Basanta Joshi** and other faculties for their precious guidance and constant encouragement.

Sincerely,
Bibek Khanal
075/MSICE/002

ABSTRACT

Identifying different types of ulcer and surgical wounds based on their distinct features is a complex task in medical imaging. This involves the classification of ulcer and surgical wound into various labels such as diabetic ulcer, pressure ulcer, venous ulcer and surgical wounds. In order to make this process more efficient and cost-effective, there has been different study in this field. A body map based VGG 16 network is used to implement transfer learning onto two trainable dense layers for classification of wound images into five labels. The five labels include the aforementioned four types of wound and another label "Not a wound" which does not contain any wound image. The study is started with AZHMT dataset containing 4790 images. These images are classified using pre-trained inceptionV3 and VGG 16 network separately. The performance of VGG 16 was found to be better than inceptionV3 by almost 4% which was the reason for selecting VGG 16 for further study in this dataset. Also, inceptionV3 is longer and wider than VGG 16 which will learn unnecessary features from images using higher computing resources. The main aim of this thesis is to show that performance can be increased without learning unnecessary features, using fewer computing resources. and by using body map function. Therefore, VGG 16 is selected for all experiments in this thesis. The VGG 16 is used to train one dense layer using body map by varying different parameters whose results are incorporated in this thesis. Then again, VGG 16 network is used to train two dense layers as well. The optimum result was found on combination of two dense layers which were trained using Adam Optimizer using body map. Body map is manually developed with unique numbering in each image file. This resulted in average precision of 0.93, average recall of 0.95, F1 score of 0.94, average AUC of 0.97 and accuracy of 94.57%.

Keywords: *Transfer Learning, Body Map, Surgical Wounds, Ulcer Wounds, Location Labelling.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

VGG	Visual Geometry Group
VLU	Venous Leg Ulcer
DFU	Diabetic Foot Ulcer
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RoI	Region of Interest
SVM	Support Vector Machines
KSVM	Kernel Support Vector Machines
SGD	Stochastic Gradient Descent
RMSProp	Root Mean Square Propagation
TPR	True Positive Rate
FPR	False Positive Rate
MLP	Multi Layer Perceptron
LSTM	Long Short Term Memory

CHAPTER 1

INTRODUCTION

1.1 Background

Wound Image Classification is a field of study withing the medical imaging domain that involves the use of computational algorithms to classify images of wounds based on their appearance and characteristics[1]. This type of image classification has been increasingly important in recent years, as the rapid advancements in medical imaging technology have led to an increase in the number of medical images being generated.

The primary goal of wound image classification is to assist healthcare professionals in the accurate and efficient diagnosis of wounds[2]. This can be achieved by automatically categorizing wounds based on specific attributes, such as impact of wound, discoloration of wound, color tone of wound and the area of wound. This classification is vital in order to plan correct and accurate treatment course for the patient depending upon the type of wound. The process of wound image classification typically involved the digital acquisition of wounds, which are pre-processed to remove noise and improve the quality of images. The image is then augmented to create more dataset from existing dataset. Augmentation includes rotation, translation, reflection and introduction of white noise to certain extent[3].

The next step is to extract relevant features from the images which is used to train machine learning algorithms. These algorithms are then used to classify the wounds into different categories, based on their appearance and characteristics. The key challenge in wound image classification is the variability of wound appearance, as wounds can differ in size, shape and severity, and also can be affected by factors such as stage of healing and the presence of infections[4]. Despite these challenges, the use of machine learning algorithm and transfer learning has shown promising results in accurately classifying wounds, and has the potential to greatly improve patient outcomes by enabling more accurate and timely diagnosis[5]. The

wound classification task researched for this thesis will be based on classification of images into five classes of wounds which are Surgical Wound, Venous Ulcer, Diabetic Ulcer, Pressure Ulcer and Not a wound.

The researched thesis includes the body map categorical input along with the image in order to effectively classify the type of wound with utmost accuracy. Body map is created through extensive research by breaking down body parts into various labels. The labels are assigned to integer values and trained in the network along with their respective images so that the network understands and assigns respective labels to respective wound probability in order to classify the wound with higher accuracy.

1.2 Problem Statement

Wound classification process is the basis of wound treatment. The process of wound classification includes location of wound and the type of wound associated with it. Wound classification is a very important aspect in wound healing process because correct classification of wound helps in planning and developing the correct course of wound treatment and care plan. However, accurate wound classification can be challenging as it requires expertise and experience in wound assessment[6]. Another challenge in automated wound classification using deep learning is the erroneous classification of wound which might lead to wrong treatment. Since most of the wounds have similar shapes, contours and features, even deep learning algorithms might fail to correctly classify it. Another challenge is the availability of fewer dataset and scarcity of complex algorithm processing hardware which might make training and testing of wound image more sophisticated[7].

The goal of this thesis to eradicate the aforementioned problems by introduction of categorical body map in order to indicate the location of wound along with the wound image. For instance, diabetic ulcers generally occur on bottom of the foot for most of the patients. This result is not only confined to bottom of foot but may also result in other body parts as well[8]. Similarly, venous ulcers generally occur below knee upto around ankle and lower leg[9]. There are various deep learning models which have shown encouraging results for image classification task

and the same result can be applied with wound image classification task as well. The proposed system will use wound classifier block and location classifier block parallely in order to determine the type of wound for proper care and accurate treatment

1.3 Objective

Image wound analysis is a process of analysing type of wound based on wound image and body map location. Body map location corresponds to the certain part of the body where wound is located. The objectives of this thesis work are:

- To develop a transfer learning approach for classifying wound images into five classes with use of body map.
- To validate the implemented network using a reliable outcome measure that can be used to assess the type of wound.
- To compare the performance of transfer learning network with state of the art CNN architectures.

1.4 Contribution of this thesis

The contribution of this work involves the use of 256 neurons dense layers on top of VGG 16 architecture in order to implement transfer learning for wound image classification task. The main contributions of this thesis are:

- The wound images related to ulcers and surgical wounds are collected from various sources forming the wound image dataset which contains 4790 labelled image and categorized into five different classes
- The naming convention for implementation of body map is manually performed in each dataset.
- A dense layer containing 256 neurons is trained on top of VGG 16 network by freezing its top layer along with categorical body map using transfer learning.

1.5 Originality of this work

There has been several researches on medical image wound classification using different pre-trained network such as AlexNet, InceptionV3 and VGG-16/19 architectures. However, there are very few researches in medical imaging by using transfer learning using these pre-trained models. Also, there have been no effective research on how body location can be used to predict type of wound. Transfer learning uses pre-trained weights to update new weights on another network within few epochs, training time and few datasets. This thesis studies about using concept of transfer learning into another dense layer and use of body map in order to further improve performance of the network by using precise location of wound. A more detailed list of various contribution is provided below:

- Use of transfer learning through pre-trained VGG-16 network to classify wounds from images.
- Investigate different learning optimizations and vary number of dense layers for effective parameter tuning.
- Integrate body map for further improving the performance of model.

1.6 Thesis Outline

The further part of documentation is organized as below:

Chapter 2 describes the state of the art of image wound classification techniques and pre-trained models.

Chapter 3 describes the theoretical background of this thesis work.

Chapter 4 describes the methodology used in researching this thesis, performance metrics and tools used and implementation of body map and pre-trained models.

Chapter 5 discusses the results and graphs observed for different parameters of transfer learning for classifying wounds.

Chapter 6 concludes the thesis with future scopes.

CHAPTER 2

LITERATURE REVIEW

2.1 Classification of wound image using ensemble-based CNN

Acute and chronic wounds pose challenges to healthcare systems around the world and affect the lives of many people every year. Wound classification is an important step in wound diagnosis which helps medical personnel to find the best treatment procedure. Therefore, a strong classifier can help experts in the field to classify wounds at less cost and time cost. Various wound classification methods based on machine learning and deep learning have been proposed at academic conferences and literatures. The published paper by Mahbod et. al. used an ensemble deep based convolutional neural network-based classifier for classifying scars images including multiclass surgery, diabetes, and venous ulcers[10]. The classification values output from the two classifiers (patchwise and framewise) are input to a multi-layer perceptron. It provides excellent classification performance. The proposed method used a 5-fold cross-validation approach which obtained maximum and average classification accuracy scores of 84.4% and 84.28% for the binary and 81.9% and 83.21% for the three-class classification problem[10]. The result is the proposed method can be effectively used as a decision support tool wound images or other relevant clinical classification systems application.

2.2 Automatic Wound Segmentation using Deep CNN

Acute and chronic wounds have many causes and place an economic burden on healthcare systems worldwide. The advanced wound care market is expected to exceed USD 22 billion by 2024 worldwide[11]. Wound care professionals rely heavily on imaging and imaging documentation for proper diagnosis and treatment. Unfortunately, lack of expertise can lead to misdiagnosis of wound etiology and inaccurate wound management and documentation. Fully automated segmentation of wound areas in natural images is an important part of diagnostic and

care protocols, as it is important to measure wound area and provide quantitative parameters in treatment[12]. Various deep learning models have been proven for image analysis, including semantic segmentation. This manuscript proposed a new convolutional framework to segment wound regions from natural images based on MobileNetV2[13] and connected component labeling[14]. The advantage of this model is a lightweight and computationally intensive architecture which is comparable to deeper neural networks without compromising performance. This paper created an annotated wound image dataset consisting of 1109 images of leg ulcers from 889 patients to train and test a deep learning model. The deep CNN demonstrated the effectiveness and mobility by performing extensive experiments and analyzes on various neural segmentation networks.

2.3 Wound Image Classification

Chen et al. used a deep learning approach to segment and classify wounds in clinical care by proposing a U-Net architecture for wound segmentation and a CNN for wound classification. The authors evaluated their approach on a dataset of 320 wound images and achieved an accuracy of 92.5% for wound segmentation and 91.3% for wound classification[15]. Thurnhofer et. al. proposed a patch based classifier with CNN to segment diabetic foot ulcer with a dice index of 0.848. They concluded that CNN based approach is better for foot ulcer segmentation[16].

Liu et al. presented an all-encompassing approach calle WoundSeg for segmentation of wound images. The authors suggest a three-stage framework for wound segmentation using deep convolutional networks. In the first stage, image enhancement techniques are employed to enhance the contrast and sharpness of wound images. In the second stage, a pre-trained VGG-16 network is utilized to extract important features from the preprocessed wound images. Finally, in the classification stage, a fully connected layer with a sigmoid activation function is used to categorize individual pixels as either wound or non-wound.%[17]..

Huimin et al. used either automated or semi-automated techniques to select the wound area before performing tissue classification, which involved utilizing one or more image descriptors and classification. The most commonly employed features

included color histograms and texture parameters such as entropy, sum of squares variance, wavelet, and local binary patterns[18]. Venkatesan et al. developed a new deep convolutional neural network called DFU QUTNet for binary patch classification of normal skin versus abnormal skin (DFU). The authors created a new dataset consisting of 754-foot images from a diabetic hospital center in Iraq, which generated 542 normal skin patches and 1067 DFU patches[19]. They augmented their dataset by 13 times using flipping, rotating, and scaling transformations. Their proposed network had 58 layers, with 17 of them being convolutional. The performance of their proposed method was compared to other deep CNNs like GoogLeNet, VGG16, and AlexNet, with the maximum reported F1-Score being 94.5% obtained by combining the DFU QUTNet architecture with SVM[19].

Rostami et al. proposed an ensemble DCNN-based classifier to categorize entire wound images into multiple classes, including surgical, diabetic, and venous ulcers[20]. The researchers brought upon a new dataset containing 538 wound images from four different classes of wounds. The resulting classification metrics of two classifiers based on patch-wise and image-wise method were fed into a Multi-Layer Perceptron to improve the classifier's performance[20]. The maximum and average classification accuracy values were 91.4% and 90.28% for binary and 91.9% and 87.7% for 3-class classification, respectively. On the other hand, Sarp et al. used an explainable artificial intelligence approach to classify chronic wounds into four classes (diabetic, lymphovascular, pressure injury, and surgical)[21] The authors employed transfer learning on the VGG16 network as the classifier model and augmented the dataset with mirroring, rotation, and horizontal flip techniques. The dataset was composed of 8690 wound images collected from the eKare, Inc. data repository. The researchers reported an average F1-score of 0.76 as the test result.

In another research by D.N Anussuzzaman et al. the proposed method consists of two stages[22]. In the first stage, the researchers used a VGG 19 to extract features from the wound images. The researchers fine-tuned the pre-trained VGG 19 CNN architecture on their wound image dataset and achieve an accuracy of 88.35% for binary classification and 84.75% for multi-class classification[22]. In the second stage, the researchers incorporated the wound location information with the

image features obtained from the first stage. The authors use a LSTM architecture to combine the location features and image features and achieve an accuracy of 97.34% for binary classification and 93.22% for multi-class classification.

2.4 Research Gap

Researches performed in field of medical wound segmentation included use of CNN and RNN architectures for classification problem [13] [15] [19]. Recently there have been increasing researches on use of transfer learning for wound classification[23]. The motivation behind use of transfer learning in wound image classification is due to the lack of availability of large dataset and no standarization in images protocol. This can lead to variability in image quality and affect the performance of classification model. Also, there are few but limited researches in use of transfer learning in wound image classification, the results are promising since use of transfer learning has increased accuracy and avoided the need for complicated hardware for training of large dataset[24]. The concept of transfer learning is to use some pre trained model such as InceptionV3[25] or VGG network[26] which has already been trained on vast amount of image dataset and apply it some another classification task.

There has been limited research on applying transfer learning to wound image classification, which could be a promising avenue for improving the performance of these models. So this thesis is focused on use of transfer learning and apply it on top of dense layer to increase classification accuracy and avoid the need of complicated hardware. Also, there has been limited research on use of body maps in order to predict and study which part of body are vulnerable to different type of wound. This thesis also focuses on use of body map and use it to classify wounds based on location and type of wound.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 Convolutional Neural Network

CNNs are an optimized and efficient method for processing images, as they are able to learn and extract key features and abstractions. These networks were first proposed by Fukushima in 1988, who suggested a hierarchical network called neocognitron, which used multiple layers of neurons to recognize patterns by learning[27]. This network's lower layers extracted local features, while the higher layers integrated this information to identify more specific and global features. However, due to computational limitations, neocognitron did not gain much attention until the 1990s, when LeCun et al. used a gradient descent algorithm to achieve successful results for handwritten digit recognition[28].

In essence, a CNN consists of multiple convolution and pooling layers applied in sequence to extract important features from input data. Convolution involves taking the sum of element-wise products between a tensor and a kernel. Two important operations that are applied before convolution are padding and stride. Padding is used to ensure that the kernel takes into account corner pixels or boundary data, while stride refers to the step size taken over an input tensor during convolution[29]. The larger the stride, the more the input size shrinks. Let (nH, nW, nC) be the dimension of the input image and (f, f, nK) be the kernel dimension where f is generally an odd dimension. Let s and p represent the size of stride and padding then, the dimension of the output after convolution operation is

$$output = [(nH + 2pf) = s + 1; (nW + 2pf) = s + 1; nK] \tag{3.1}$$

The convolutional layer of a CNN has filters that contain weights and biases, which are learned and updated during the back-propagation phase[30]. An activation function is applied after the convolutional process. The choice of activation function depends on the problem at hand, but the ReLU activation function is

commonly used in hidden layers due to its computational efficiency compared to sigmoid or tanh functions.. A ReLU activation can be represented mathematically as

$$f(z) = max(0; z) \tag{3.2}$$

Once convolution and activation have been applied, the feature size is reduced using a pooling operation. Pooling involves moving a filter with no learnable parameters over a tensor and calculating either the maximum or average value of the window of the filter. This operation only affects nH and nW. The dimension of an image after pooling with a fxf filter is given by

$$output = [(nH + 2pf) = s + 1; (nW + 2pf) = s + 1; nK] \tag{3.3}$$

where, p and s are padding and stride values for the pooling operation. We apply convolution and pooling a number of times depending on the requirement of the feature extraction and then connect the output to a fully connected network which is essentially a feed-forward neural network with weights and biases[31]. These are updated during the back propagation phase. The architecture of CNN is shown as in fig 3.1.

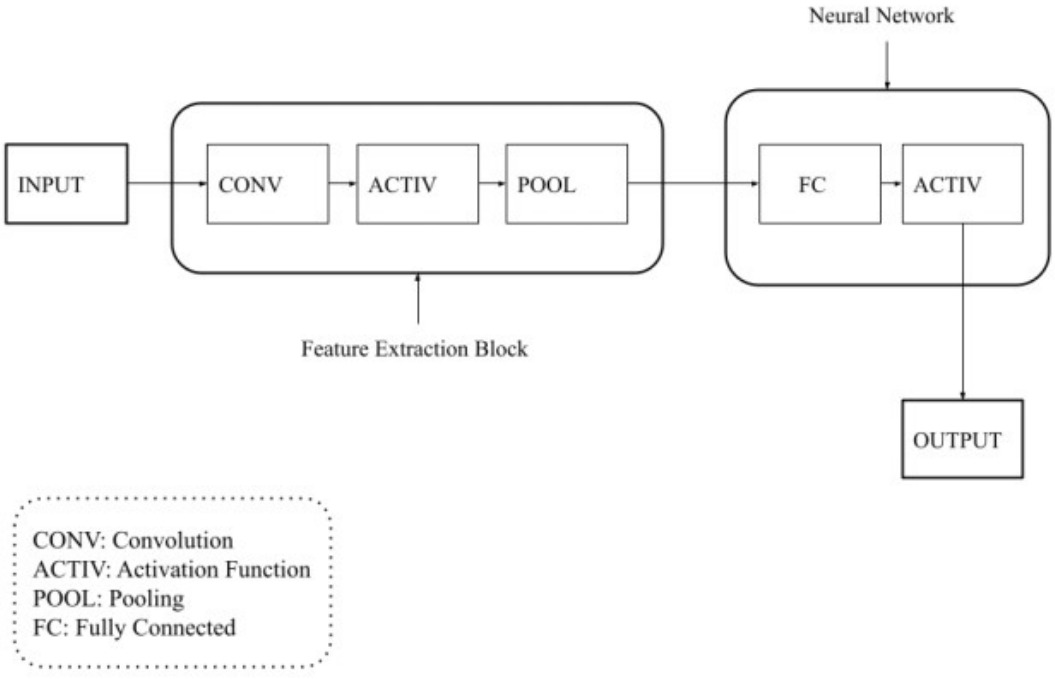


Figure 3.1: Convolutional Neural Network Architecture

3.2 Stochastic Gradient Descent Optimization

Stochastic gradient descent is a popular optimization algorithm used in machine learning to find the optimal set of parameters that minimize the cost function. The main idea behind SGD is to update the model parameters in small batches using the gradients of the cost function with respect to the parameters, rather than the entire dataset[32].

The SGD algorithm works by randomly selecting a mini-batch of training examples from the dataset and computing the gradients of the cost function with respect to the parameters using the selected mini-batch. SGD is used to minimize neural network loss function $J(\theta)$ which is parameterized by a neural model's parameter $\theta \in R^d$ with respect to the parameters. The learning rate η determines the size of the steps taken to reach the minimum value[33]. Stochastic gradient descent performs the optimization by computing the gradient of the loss function with respect to the parameters θ for each training sample $x^{(i)}$ and $y^{(i)}$ given by the equation 3.4:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{3.4}$$

However, one of the drawbacks of SGD is that the learning process can get stuck in local minima. Another drawback is that SGD can have a slow convergence rate when the cost function has a large number of flat regions, plateaus, or ridges. In such cases, the algorithm may take a long time to reach the minimum [33].

A different optimization technique, known as Mini-batch gradient descent, combines the advantages of both Batch gradient descent and Stochastic gradient descent by updating the parameters based on a batch of n training examples at a time. This allows for a more stable convergence of the objective function[34]:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \tag{3.5}$$

However, traditional mini-batch gradient descent has some limitations. Firstly, it can be difficult to select the best learning rate. A learning rate that is too small can lead to slow convergence, while a learning rate that is too large can prevent the best convergence to the minimum. This can result in the loss function either fluctuating or diverging away. Although a learning rate scheduler can help

solve this problem, it is pre-defined and does not take into account changes to the objective function that occur between epochs of training, making it an unadaptive solution[34]. Furthermore, in traditional mini-batch gradient descent, the same learning rate is applied to all parameters of the neural model. However, if there is sparse data with multiple parameters, each parameter can benefit more if an individual learning rate is applied to it. If the loss function is highly non-convex, the neural network can get trapped in suboptimal local minima with mini-batch gradient descent[35].

Momentum approach can help accelerate SGD in the relevant direction by damping the oscillations of the objective function. To achieve this, it adds a fraction γ of the update vector of the past time step to the current update vector as shown in equation 3.6 and 3.7:

$$v_t = \gamma v_{t-1} + \eta \cdot \nabla_\theta J(\theta) \tag{3.6}$$

$$\theta = \theta - v_t \tag{3.7}$$

Adagrad is a gradient-based optimization algorithm that adjusts the learning rate of parameters. It updates the parameters with larger values for those that are associated with infrequent features and smaller values for those associated with frequently occurring features. Unlike the traditional method where all parameters of the model use the same learning rate η, Adagrad uses a different learning rate for each parameter at each time step[36]. The gradient at time step t is denoted as g_t, and the partial derivative of the loss function with respect to the parameter θ_i at time step t is denoted as $g_(t, i)$.

$$g_{(t,i)} = \nabla_\theta \cdot J(\theta_t, i) \tag{3.8}$$

The parameter update for every θ_i is:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \tag{3.9}$$

The general learning rate is modified by Adagrad at each time step for every parameter based on the past gradients that have been computed for:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i} \tag{3.10}$$

In the given equation, $G_t \epsilon R^{d*d}$ is a diagonal matrix where each diagonal element i, i is the sum of squares of the squares of the gradients with respect to θ_i. RMSprop resolves the radically diminishing learning rates of Adagrad optimization.

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \tag{3.11}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{3.12}$$

Adam is a contemporary approach that can calculate adaptive learning rates for every parameter. It combines the characteristics of momentum and RMSprop by retaining both an exponentially decaying average of past squared gradients and an exponentially decaying average of past gradients[37]. To obtain the decaying averages of past and past squared gradients, we can compute m_t and v_t, respectively, using the following method.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.13}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.14}$$

where, mt and vt are the mean and variance of the gradients respectively. Then, to update the parameters.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{3.15}$$

Adam computes individual learning rates for different parameters, which means that it adapts the learning rate based on the gradient statistics of each parameter. This feature enables Adam to converge faster and more efficiently than traditional optimization algorithms. Adam optimizer uses concept of momentum which helps the optimizer to continue its previous direction of update, which enables it to move quickly through the flatter regions of the loss landscape and to avoid getting stuck in local minima. This is the reason adam optimizer is widely used[38].

3.3 Dropout

Dropout is a regularization technique used in neural networks to prevent overfitting of the model to the training data. It works by randomly dropping out some of the neurons in a layer during training[39]. This is done with a specified probability, typically around 0.5, which means that each neuron has a 50% chance of being dropped out.In process of dropout, the network is forced to learn redundant representations of the input data, which makes it more robust to noise and helps prevent overfitting. In essence, dropout can be seen as a way to ensemble different models, as it trains several different models with subsets of the available neurons, each of which learns to make predictions on its own[40]. In this thesis, to avoid overfitting, dropout of 0.7 is used which means randomly dropping out 70% of the total neurons during learning.

3.4 VGG 16 (Visual Geometry Group)

VGG-16 is a convolutional neural network (CNN) architecture that was developed by the Visual Geometry Group at the University of Oxford[26]. It is one of the most popular and widely used deep learning models for image recognition and classification tasks. The architecture of VGG-16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers use small 3x3 filters with stride 1 and padding 1, and are followed by max-pooling layers with 2x2 filters and stride 2. The fully connected layers at the end of the network are used for classification[26]. The VGG-16 model has achieved state-of-the-art performance on a number of benchmark datasets, including ImageNet, which contains millions of labeled images from thousands of categories. It has also been used as a pre-trained model for transfer learning in various computer vision applications. Convolutional networks have got a great success in large scale-image and video recognition which became possible due to the large public image repositories such as ImageNet and high-performance computing system such as GPUs or large-scale distributed clusters[41]. Depth of ConvNet architecture and used of very small (3×3) convolutional filters in all layers has been addressed. As a result, a significant ConvNet architecture (VGG16) has been achieved which got

the state-of-the-art accuracy on ILSVRC classification and other image recognition datasets. The model achieved 92.7% top-5 best accuracy in Imagenet, which is a dataset of over 14 million images belonging to 1000 classes[26]. The architecture of VGG 16 is shown in fig 3.2.

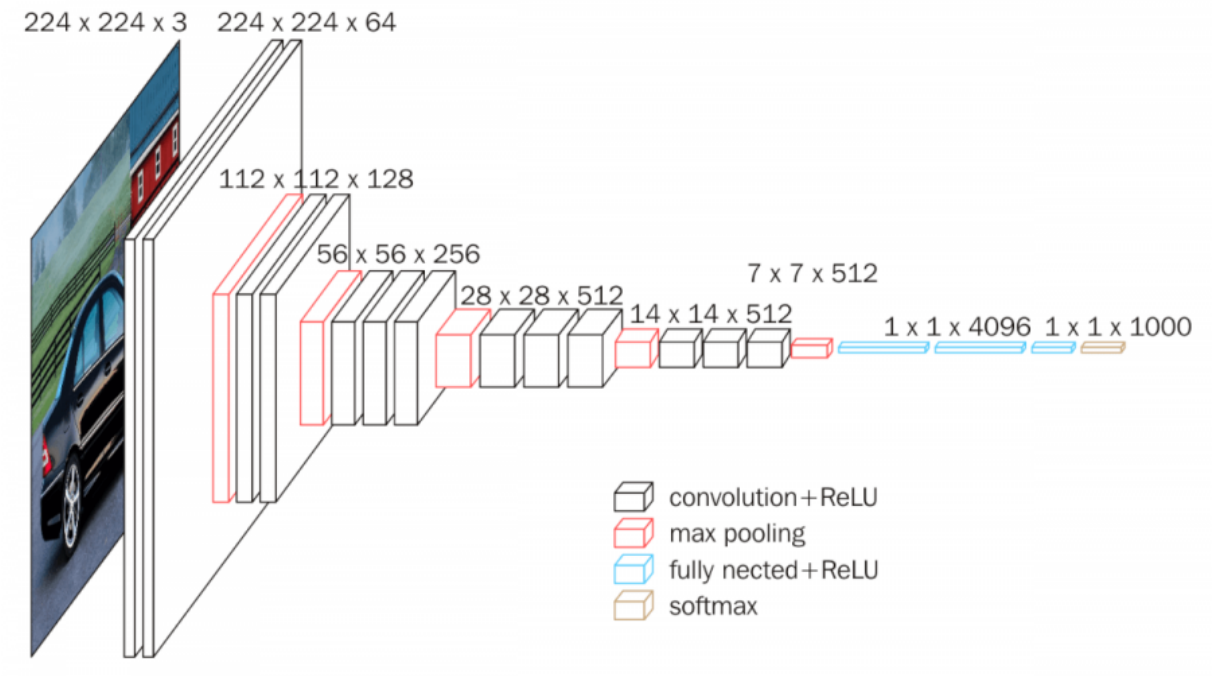


Figure 3.2: VGG 16 Architecture

The input to VGG16 224*224 RGB image is provided to VGG16. The provided image is passed to stack of convolutional layers where the image is convolved with fixed sized filter of 3*3. The convolutional stride performed on convolutional layer is 1 pixel and padding performed is such that spatial resolution is preserved after convolution. Spatial pooling is carried out by five max- pooling layers. Max-pooling is performed over a 2*2 pixel window, with stride 2 pixel[26]. Three Fully-Connected layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer where classifications of images are obtained[26].

The mathematical equation of VGG-16 can be expressed as a series of convolutional and pooling operations followed by fully connected layers for classification.

Let X be an input image of size W x H x C, where W is the width, H is the height, and C is the number of channels which is typically 3 for RGB channels.The first layer applies a 3x3 convolution with 64 filters, followed by a rectified linear activation function (ReLU):

$$Z_1 = ReLU(X_1 * W_1 + b_1) \tag{3.16}$$

where W_1 is the weight matrix for the first layer, b_1 is the bias term, and $*$ represents the convolution operation. The second layer applies another 3x3 convolution with 64 filters and a ReLU activation:

$$Z_2 = ReLU(Z_1 * W_2 + b_2) \tag{3.17}$$

The third layer applies a 2x2 max pooling operation with stride 2:

$$P_1 = maxpool(Z_2) \tag{3.18}$$

The fourth layer applies a 3x3 convolution with 128 filters and a ReLU activation:

$$Z_3 = ReLU(P_1 * W_3 + b_3) \tag{3.19}$$

And so on, for a total of 13 convolutional layers, followed by 3 fully connected layers for classification. The final output of the network is a probability distribution over the possible classes, which can be obtained using a softmax activation function on the output of the last fully connected layer. However, for the implementation of transfer learning this layer is frozen and then fed into another dense layer which contains trainable neurons.

3.5 Evaluation Metrics

In order to determine the consistency and correctness of a classification model, typical assessment metrics are determine according to:

1. True Positive (TP): A true positive is when the model predicts the positive class accurately

2. False Positive (FP): A false positive occurs when the model estimates the positive class inaccurately.

3. True Negative (TN): A true negative is when the model predicts the negative class accurately.

4. False Negative (FN): A false negative is when the model predicts the negative class inaccurately.

3.5.1 Precision

Precision is used to measure the number of right positive predictions among all positive predicted values. For example, if the model predicts 100 wound images as diabetic wounds then precision gives the number of correct predicted wounds as diabetic among predicted values. It is calculated by using formula:

$$Precision = \frac{TP}{TP + FP} \tag{3.20}$$

3.5.2 Recall

Recall is used to measure the number of right positive predictions among all the true actual class. For example, if the model predicts 100 wound images as diabetic wound then recall gives us the correct predicted diabetic wound among the true values. It is calculated by using formula:

$$Recall = \frac{TP}{TP + FN} \tag{3.21}$$

3.5.3 F1 Score

F1 score is used to measure the test accuracy. It is the weighted mean of precision and recall. This score takes both false positives and false negatives into account. It is calculated using formula:

$$F1Score = \frac{Precision * Recall}{Precision + Recall} \tag{3.22}$$

3.5.4 Accuracy

Accuracy is used to measure the performance of the model. It is the simple ratio of the correctly predicted observations to the total observations. It can be calculated using formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.23}$$

3.5.5 Confusion Matrix

A confusion matrix is a table that shows the results of the prediction model. The number of observations made by the model where it categorized the groups correctly or incorrectly is expressed by entry in a confusion matrix. The confusion matrix has peculiar table organization which helps the output to be visualized, usually supervised learning. It shows not only a predictive model's results, but also which groups are correctly predicted, which are incorrectly predicted, and what types of errors are being made as shown in fig 3.3.

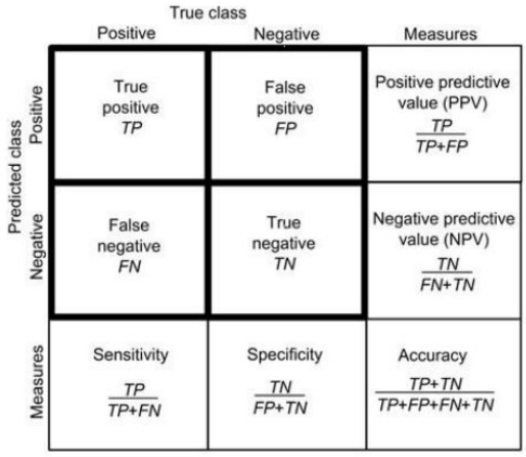


Figure 3.3: Confusion Matrix

We have five classes of wound namely "Surgical Wound", "Venous Ulcer", "Diabetic Ulcer", "Pressure Ulcer" and "Not a wound". Each cell in the confusion matrix corresponds to each respective label. An example of confusion matrix inclusive of these five labels is shown in figure 3.4. The vertical labels represent the true label of wound image and the horizontal label represents the predicted label of wound image by the learning network. The highlighted cell symbolizes for the true and predictions of each wound images. The confusion matrix is developed upon comparison with with each labels so it is helpful in calculating evaluation matrix (Accuracy, Precision, Recall and F1 Score).

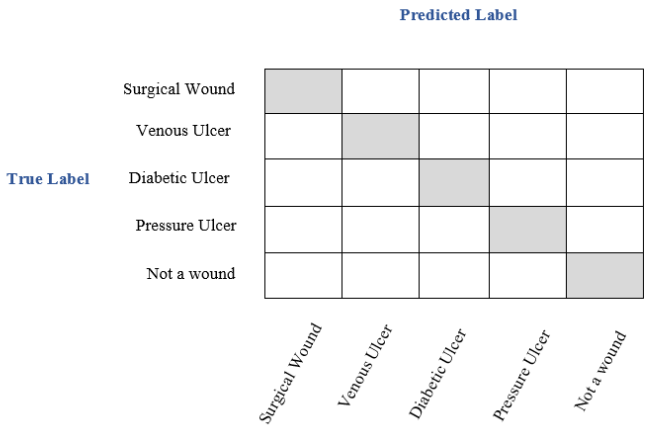


Figure 3.4: Confusion Matrix for five classes

3.5.6 ROC-AUC Curve

The ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model. It is a plot of the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings. The TPR is also known as sensitivity or recall, and it measures the proportion of true positives that are correctly classified by the model. The FPR is the proportion of false positives that are incorrectly classified by the model.ROC-AUC curve helps to see how the threshold plays out the decision of the model at different threshold settings[42]. The VGG model will predict the True Positives and Negatives and all other metrics mentioned above based on the assumption of a threshold that dictates what output label is considered positive and negative. This value of the threshold however could have been any random number hence this is an arbitrary choice and should not affect the decision provided by our model. It provides us the measure of goodness of fit, summarizes the model output across all thresholds, and provides a good sense of the discriminative power of a given model[43]. The ROC curve helps to visualize the trade-off between TPR and FPR at different classification thresholds. A perfect classifier would have a TPR of 1 and an FPR of 0, resulting in a point in the upper left corner of the plot. A random classifier would have a diagonal line, as the TPR and FPR would be equal at all threshold settings.

The AUC is a single numerical value used to evaluate the effectiveness of a classification model. It calculates the area under the ROC curve, and its value varies between 0 and 1. When the value is 0.5, it indicates a random classification, whereas a value of 1 represents a perfect classification. Essentially, the AUC measures the likelihood of a positive example being ranked higher by the model than a negative example chosen at random[43]. ROC AUC curve is depicted in fig 3.5.

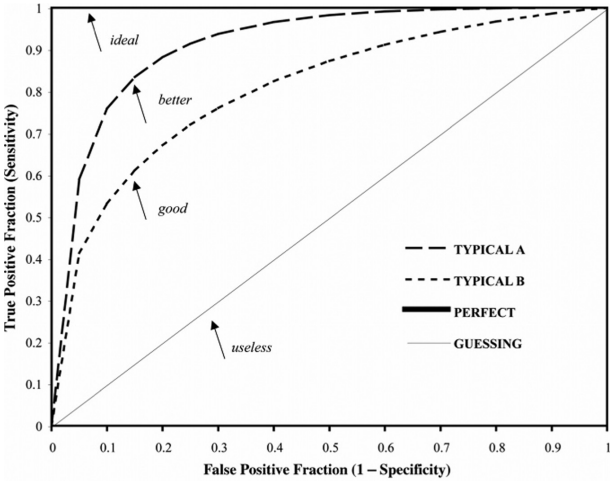


Figure 3.5: ROC-AUC Curve

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

4.1 Block Diagram

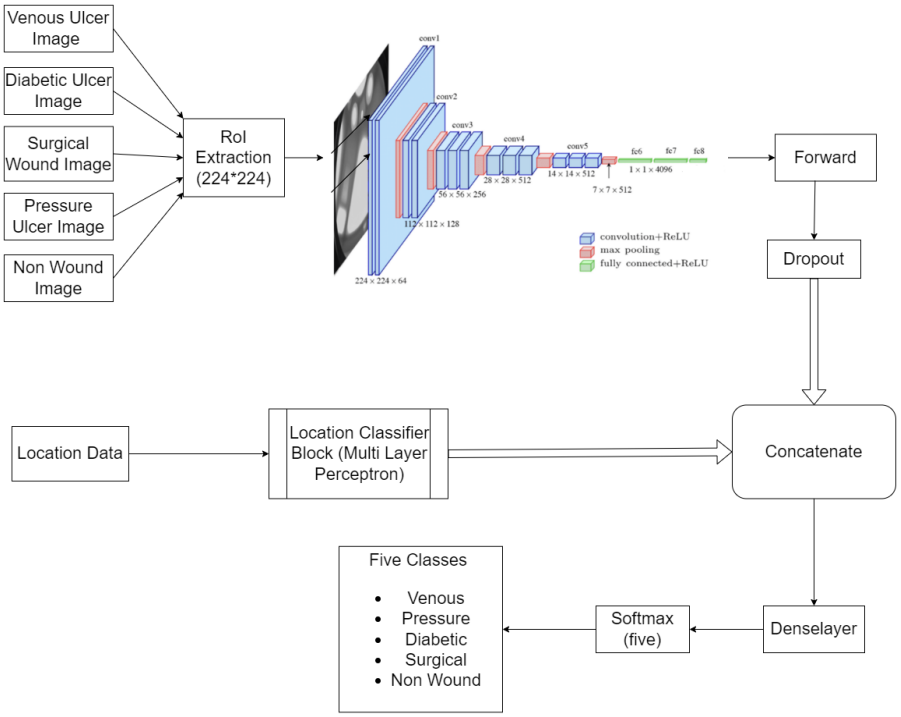


Figure 4.1: Block Diagram

The block diagram system is as shown in figure 4.1 is for the classification task which takes wound image as input. The wound image is of size 224*224 in order to make it compatible for VGG 16. The wound image is labelled as 0,1,2,3,4 for venous ulcer, surgical wound, diabetic ulcer, pressure ulcer and not a wound. The system contains VGG16 network to extract feature maps from images and VGG16 has been implemented to test its performance in this dataset and got good performance. All the layers except the output of VGG16 has been frozen in order to train Dense layer based on transfer learning. There are two layers of Dense layer trained based on input dataset which uses ReLU activation. Dense layer uses dropout after pooling in order to discard random neurons to avoid overfitting problem. At the same time, another block processes body map in order to identify

which part of the body contains wound. Body map contains different breakdown categorical breakdown of the body to build a probabilistic model of where the wound is likely to take place. For instance, Diabetic ulcer is likely to happen on the bottom of the feet. Similarly, venous ulcer are like to occur between ankles and knee cap. However, location map alone does not confirm wound type and is only used in order to increase accuracy of the system. The categorical value of location map will be processed using multi layer perceptron and will be concatenated with the final output of VGG 16 layer and fed into dense layer for training it. The final output will have softmax function divided into five classes based on their probabilities.

4.2 Dataset Collection

The datasets are carefully collected from AZHMT and Medetec dataset which contains images of different categories ulcer and surgical wound. Only relevant dataset are collected for formulating our problem. These dataset have images varying from 320 to 700 pixels in width and 240 to 560 pixels in height. The AZHMT dataset is robust and reliable for our problem. The AZH dataset which we will be using was gathered during a clinical period of two years at the AZH Wound and Vascular Center located in Milwaukee, Wisconsin. The dataset comprises images of four distinct types of wounds, namely venous, diabetic, pressure, and surgical wounds. The images are captured using an iPad Pro and a Canon SX 620 HS digital camera. The wound specialist at AZH Wound and Vascular Center is responsible for labeling the images. In the majority of cases, each image is taken from a different patient. The samples of dataset belonging to five classes are shown from fig 4.2 to fig 4.6.

Figure 4.2: Diabetic Ulcer Image Data

Figure 4.3: Venous Ulcer Image Data

Figure 4.4: Surgical Wound Image Data

Figure 4.5: Pressure Ulcer Image Data

Figure 4.6: Not a Wound Image Data

4.3 Body Map

A body map for wound image classification is a graphical representation of the human body that is used to identify the location of a wound on a patient. The body map is typically divided into different anatomical regions, such as the head

and neck, upper extremities, trunk, lower extremities, and genital area[44].Using a body map can be helpful in organizing and classifying wound images based on their location, which can be important for clinical and research purposes. For example, it can help to identify patterns in the occurrence of certain types of wounds in different anatomical regions, or to monitor the healing process of a wound over time[45].

A body map has been developed in order to increase accuracy of wound prediction since there are different cases of ulcers that develop in different part of the body. For instance, diabetic ulcer generally occurs in lower feet and venous ulcer occur in ankles area[8] [9]. Surgery generally takes place in joint area such as elbow, chest, ankles, knee etc[46]. These places are generally susceptible to surgical wound. Also, pressure ulcer occurs in muscular areas such as gluteus maximus, latissimus dorsi etc. However, this does not limit the occurrence of wound in specific areas only. Hence, in order to increase accuracy of prediction, both body map number and wound image are fed into network for effective classification. The detailed breakdown of body maps is as shown in figure 4.7, 4.8 and 4.9.

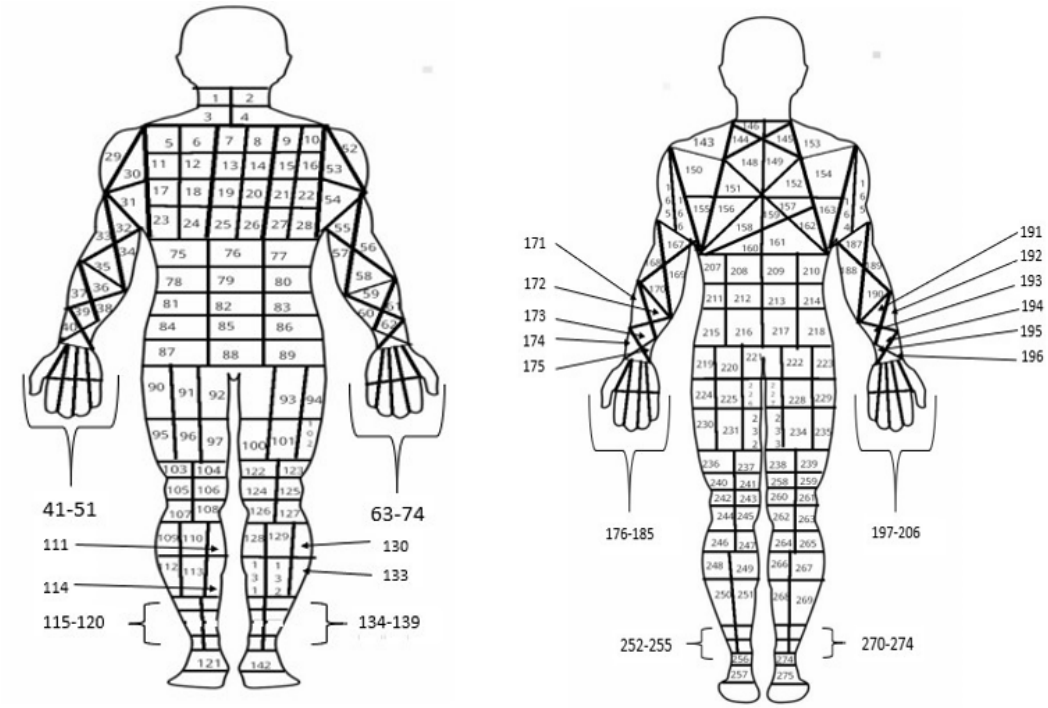


Figure 4.7: Body Map front/back

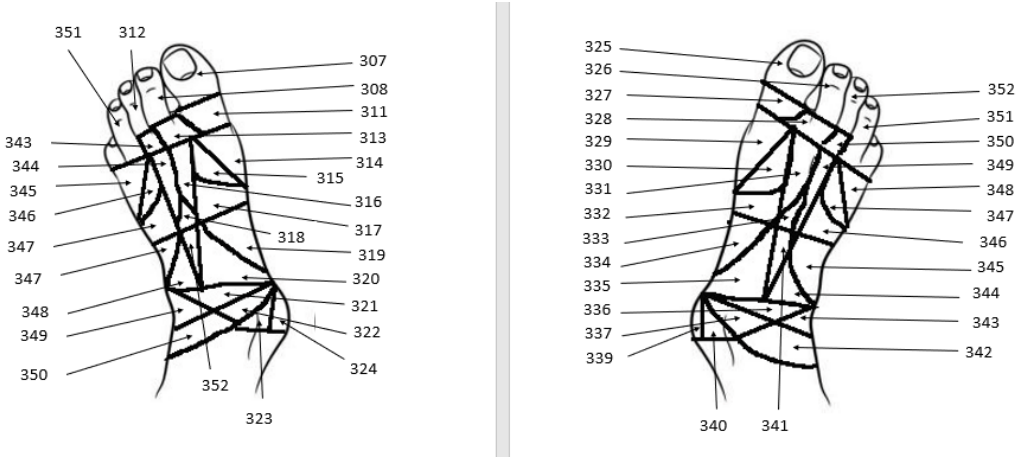


Figure 4.8: Body Map Dorsal

Table 4.1: Body map location labelling sample

Hand Location		Foot Location		Back Location	
Right Proximal Palmer	44 - 45	Right Medial Plantar First Toe	276, 281	Right Dorsum	213, 214
Right Hypothenar	43	Left Medial Plantar First Toe	301, 302	Right Lumbar	217
Left Distal Interphalangeal	71 - 73	Right Dorsal Proximal Toe	327	Right Gluteal	228
Left Wrist Crease	65	Right Dorsal Distal Toe	325	Left Branchial	165, 166

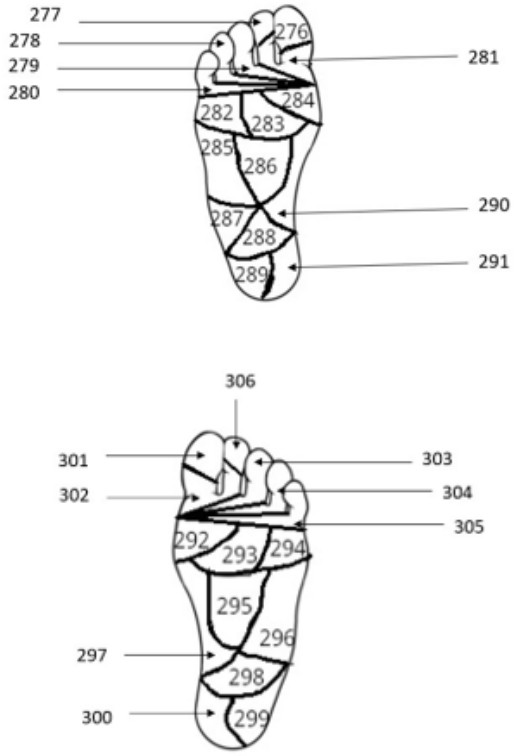


Figure 4.9: Body Map Plantar

4.4 Pre-processing

The dataset collected were originated from various sources, which is why they differed in size, resolution and color intensity. Therefore, a detailed pre processing technique was applied to dataset to maintain uniformity.

- **Adaptive Histogram Equalization (AHE)**: The wound images were subjected to the AHE transformation technique in order to enhance their contrast. AHE addresses the problem of overamplifying noise in uniform areas of an image by limiting the amplification. Initially, a neighborhood histogram was computed for each pixel in the image. The histograms were then clipped at a predefined threshold value and the clipped histogram was redistributed evenly among all the histogram bins. Next, the Cumulative Distribution Function (CDF) and transformation function were computed for each pixel based on the clipped histogram. Finally, the transformation function was applied to each pixel to obtain the equalized image.

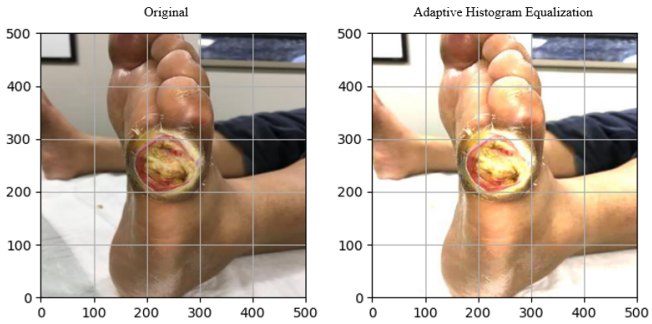


Figure 4.10: Original vs AHE

- **Scaling**: The variable wound and non wound image size were then converted to 224*224 size since the VGG 16 network only reads the respective image size. The mathematical representation of scaling operation can be defined as follows:

$$x' = S_x + x \tag{4.1}$$

$$y' = S_y + y \tag{4.2}$$

Here,

(x,y) are the two dimensional co-ordinates of image pixels.

(x',y') are two dimensional co-ordinates after scaling.

S_x and S_y are scaling factors.

$$S_x = \frac{New\ width\ of\ rescaled\ image}{Actual\ width\ of\ the\ original\ image} \tag{4.3}$$

$$S_y = \frac{New\ height\ of\ rescaled\ image}{Actual\ height\ of\ the\ original\ image} \tag{4.4}$$

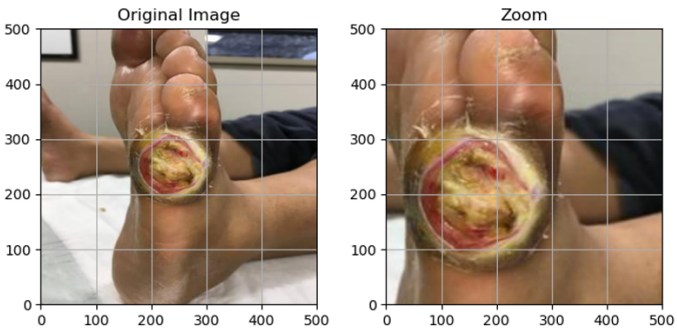


Figure 4.11: Original image vs Scale by 0.4

4.5 Data Augmentation

After the pre-processing step, in order to prevent the model from the problem of overfitting, different techniques of data augmentations were used to introduce variation in images of dataset. Some of the methods of augmentations were:

- **Lateral Inversion**: The wound images were horizontally flipped. The mathematical expression of lateral inversion is given as:

$$x' = -1 * x \tag{4.5}$$

$$y' = y \tag{4.6}$$

- **Rotation**: The wound images were randomly rotated upto ±45°. The mathematical expression of rotation is given as:

$$x' = \cos\theta * x + \sin\theta * y \tag{4.7}$$

$$y' = -\sin\theta * x + \cos\theta * y \tag{4.8}$$

  Here, the angle θ varies from -45°to +45°

- **Translation**: The wound image were shiften in lateral and vertical direction with shift range interval of [-0.4,+0.4] of the total width and height, respectively.

$$x' = x + t_x \tag{4.9}$$

$$y' = y + t_y \tag{4.10}$$

  Here,t_x and t_y are the translational factors in horizontal and vertical directions, respectively

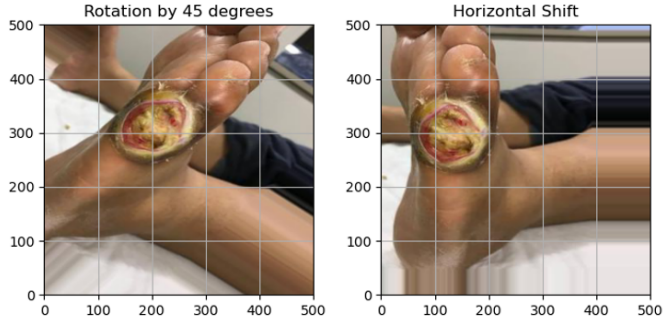


Figure 4.12: Rotation by 45°and Horizontal Shift

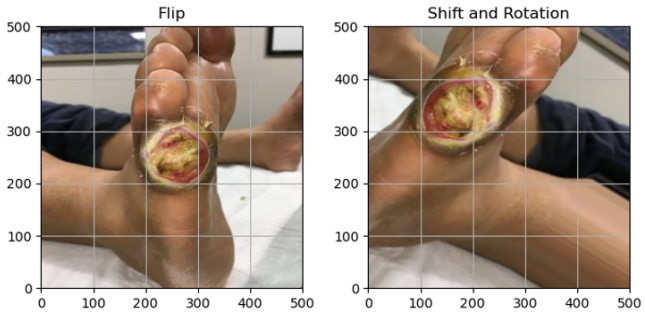


Figure 4.13: Flip and Rotation + Shift

4.6 Wound Classifier Block (VGG 16)

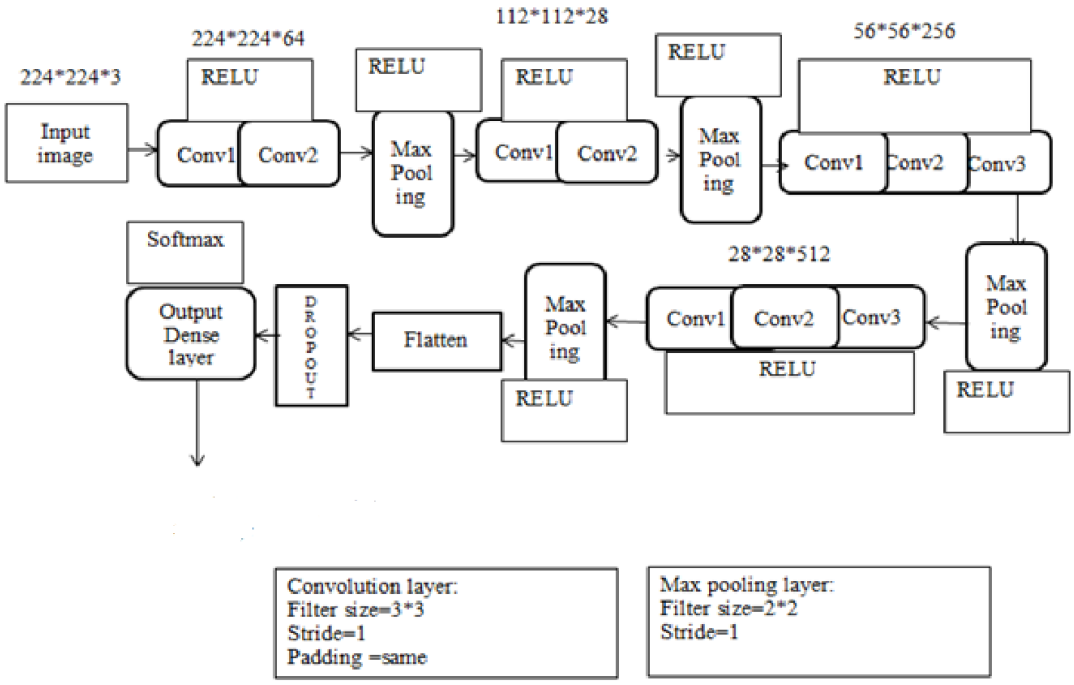


Figure 4.14: VGG Image Modality

The VGG16 network for wound image classification has been successfully completed using dense layer and transfer learning. The block for VGG16 training is named as wound classifier block since it classifies type of wound based on input as shown in fig 4.14. The output of the VGG 16 will be fed into dense layer in form of flattened and 0.5 dropout with dense layer containing 256 neurons. Paralelly, there is another simple MLP block which encodes the location maps of wound and merge with output of VGG 16 in order to improve accuracy of the system. The

performance of VGG 16 was highest as compared to other deep networks such as inceptionV3. 4165 numbers of annotated images for five different categories were used to train the VGG16 model. The evaluation of performance of the models on the image dataset concluded that transfer learning model using VGG16 returned the best accuracy. The dataset consists 4790 numbers of wound image with five different classes with training and for validation 625 number of wound images were taken.

In order to categorize the images into different classes, the pretrained deep convolutional network was first downloaded where were able to reuse the weights of VGG16 by providing the weights parameter as 'imagenet' and can be fine-tuned by removing the first and last layers of VGG16 and compute the performance of the classifier according to our task. Here, simple supervised learning method implemented to train the model with our available dataset to perform our required task. This CNN model consists of 13 convolutional layers, 5 max pooling layers with RELU activation functions and 3 fully connected layers. The last layer of VGG16 is of 1000 categories and we have the problem of classifying the images into 5 categories, so another dense layer with 256 neurons with ReLU activation will be added in the output with softmax activation function to classify the output into 5 classes. Next, another dense layer of 256 neurons will be added and dropout of 0.5 in order to train the network more accurately.

4.7 Location Classifier Block (Multi-Layer Perceptron)

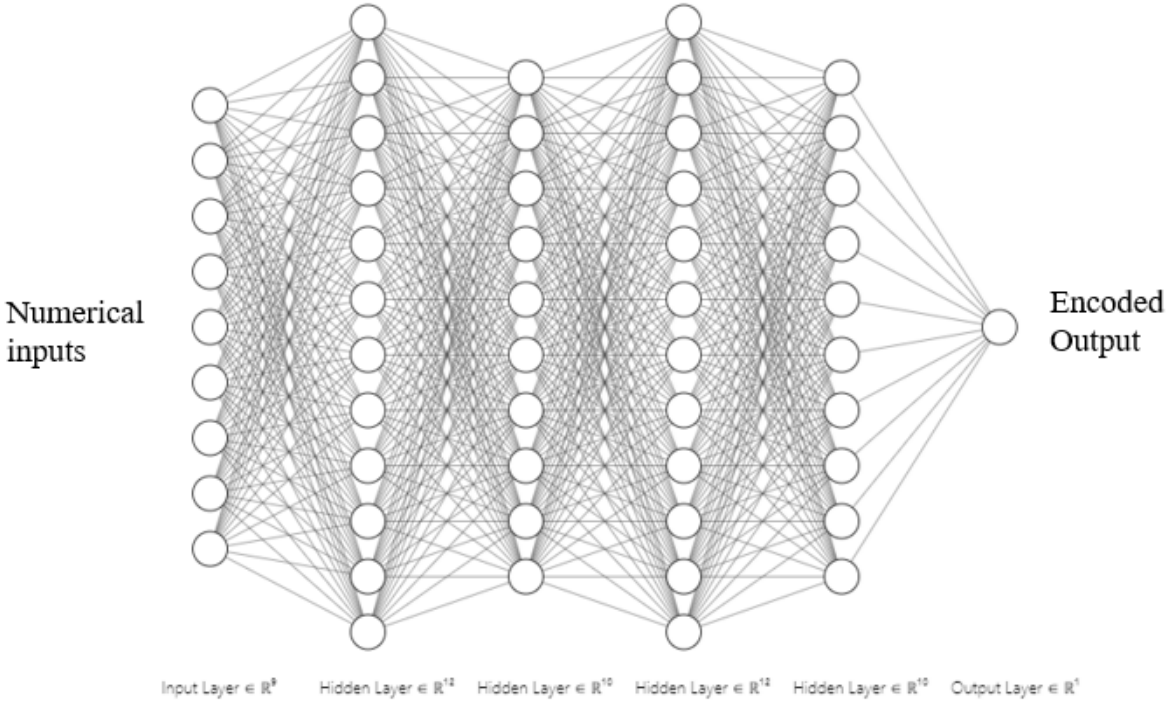


Figure 4.15: MLP Location Classifier

The figure 4.15 depicts the location classifier block used to encode the body maps location. Each of the image dataset contains its respective body map number after the underscore sign which have been embedded manually for each images. For example, a wound image of diabetic ulcer in left medial plantar first toe will have the body map number of 301. The filename for this image will have a format of <filename>_301 so that the program will automatically clip off the number after _ sign to extract the location number of that particular image. The MLP contains 9 input layers with four hidden layers in between. The four hidden layers consists of 12,10,12,10 neurons and one neuron on output layer. The network is trained at the same time as that of dense layer of wound classifier block. All the neighbouring body maps are parallelly fed into the 9 neurons at the same time for proper training so that wound at the nearest areas will have the same encoding to avoid any conflicts.

For example, the neighbour of 301 body map is 302 and 306. So, 301, 302 and

306 will be fed to three input neurons will rest of the input being zero. 9 neurons are preferred in the input since one body map can have maximum of 9 neighbors. The table of neighbors in developed and maintained as shown in table 4.2.

Table 4.2: Table depicting latent neighbors of each body map

Body Map Number	**Neighbors**
334	332, 333, 335, 341
286	283, 285, 290
213	208, 209, 210, 212, 214, 216, 217, 218
260	258, 259, 261, 262, 263

4.8 Dataset Split

A total of 4790 dataset belonging to five classes are available for training and testing the VGG 16 transfer learning block. The dataset will be split in 85% - 15% train and test split. That means out of 4790 images, 4165 images belonging to five classes will be used for training and rest 625 images of 5 classes will be used for validating the network as shown in table 4.3.

Table 4.3: Dataset Description

Wound Class	**Training Set**	**Test Set**	**Total**
Diabetic Ulcer	831	125	956
Pressure Ulcer	836	125	961
Surgical Wound	829	125	954
Venous Ulcer	833	125	958
Not a Wound	836	125	961
Total	**4165**	**625**	**4790**

4.9 Tools and resources used

The entire experiment was done on google colaboratory since it offers 12GB RAM free with NVIDIA K80 GPU processor. The main programming tool used was python with its numerical libraries. The python libraries used for the study are listed below:

- **Keras**: Keras is a high-level open-source neural network library written in Python. It is designed to enable fast experimentation with deep neural networks and to provide an easy-to-use and modular interface for building and training neural networks.

- **Tensorflow**: TensorFlow is an open-source software library for numerical computation and machine learning developed by the Google Brain team. It is designed to be a flexible and scalable platform for building and deploying machine learning models, from research to production.

- **matplotlib**: Matplotlib is a data visualization library for Python that allows users to create a wide variety of static, animated, and interactive visualizations in Python.

- **Scikit Learn**: Scikit-learn, also known as sklearn, is a Python library for machine learning. It provides a wide range of tools for machine learning tasks such as classification, regression, clustering, and dimensionality reduction, as well as tools for data preprocessing, model selection, and evaluation.

- **Open CV**: OpenCV provides a wide range of tools and algorithms for processing images and video data, including functions for image and video capture, image and video manipulation, object detection and recognition, and machine learning.

CHAPTER 5

Results and Discussion

The VGG16 network for wound image classification is successfully completed using dense layer and transfer learning. The block for VGG16 training is named as wound classifier block since it classifies type of wound based on input. The second block is called location classifier block which will encode the location data to be concatenated with output of VGG 16. The purpose of location classifier is to encode the location maps accordingly and merge into final Dense layer's output in order to improve accuracy of the system. The performance of VGG 16 was highest as compared to other deep networks such as inceptionV3. 4165 numbers of annotated images for five different categories were used to train the VGG16 model. The evaluation of performance of the models on the image dataset concluded that transfer learning model using VGG16 returned the best accuracy. The dataset consists 4165 numbers of wound image with five different classes with training and for validation 625 number of wound images and non wound images are taken. For the initial study, both networks VGG 16 and inceptionv3 were run for upto 15 epochs only. This was done to check which network would perform better for our given dataset. It was evident from study that VGG 16 outperformed inceptionV3. Even though inceptionV3 is wider and longer than VGG 16, for small dataset in order to avoid unnecessary feature learning, VGG 16 is preferred over inceptionV3. Therefor We had decided to move on with VGG 16 for task at hand since we will be using body map as well for improving performance of network and there will be no need to learn unnecessary features of wound images.

5.1 Comparison of baseline VGG 16 model and final VGG 16 model to decide optimal number of epochs

The reason of comparing accuracy and loss vs epoch depicted in figure 5.1 and 5.2 respectively is to decide the optimal number of epochs for further study due to limitations of computing resources. The figures contains accuracy and loss of

baseline and final models and it reveals that the network saturates after 15 epochs of training and does not increase or decrease significantly after that. This study was performed earlier than the ablation study between various tuned parameter to show that training the network upto 15 epochs or upto 40 epochs shows similar result. In order to avoid huge training time, the ablation study will be later performed at 15 epochs since it reveals result similar to that at 40 epochs.

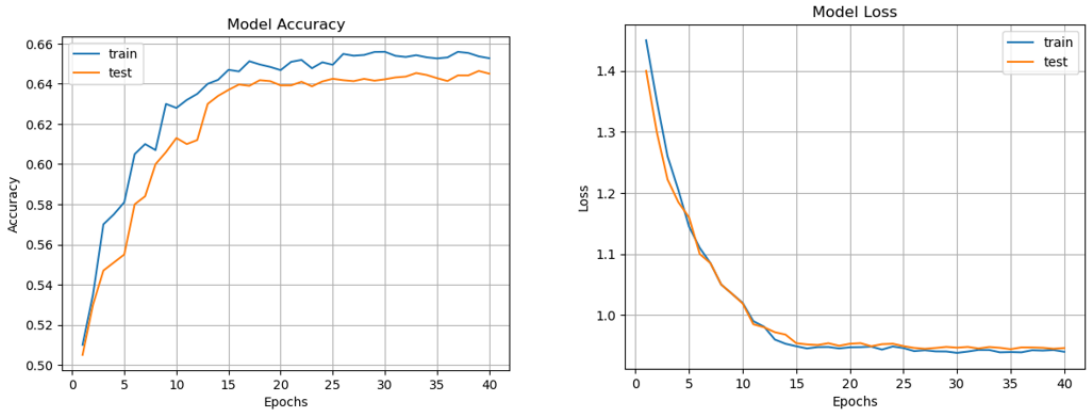


Figure 5.1: Accuracy and Loss of baseline VGG 16 model upto 40 epochs

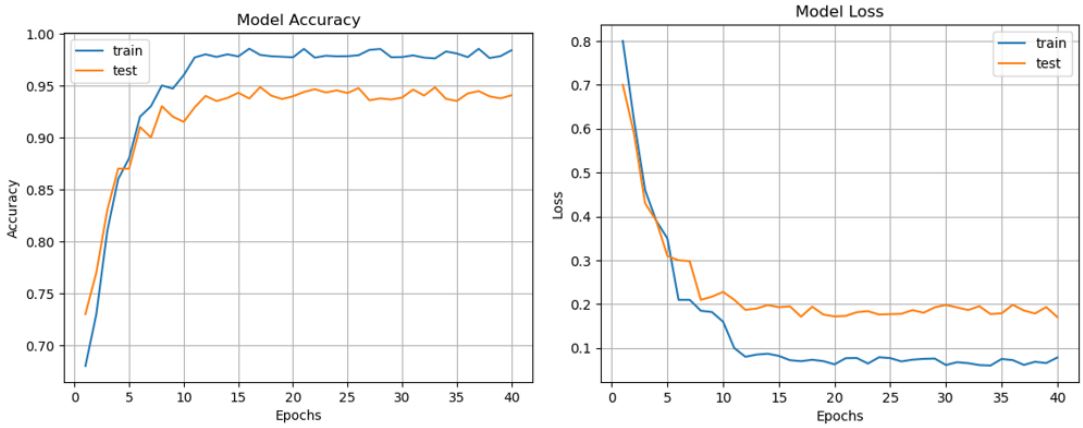


Figure 5.2: Accuracy and Loss of final VGG 16 model with dense layer upto 40 epochs

5.2 Baseline Model with VGG 16 and five softmax

The baseline model for the wound image classification task is a VGG 16 architecture containing five softmax function at the output as shown in fig 5.3. The purpose of five softmax function is to classify the wound images into five classes based on their probability. In baseline implementation, all the layers of VGG 16

are frozen and non-trainable and there is no use of dense layer or body map. The VGG 16 used is already pre trained on imagenet database which contains over 1.2 million dataset over 1000 object categories. However, it is emperical to note that since the VGG 16 is non trainable with all the layers frozen, is does not result in significant improvement of accuracy for variying dataset.

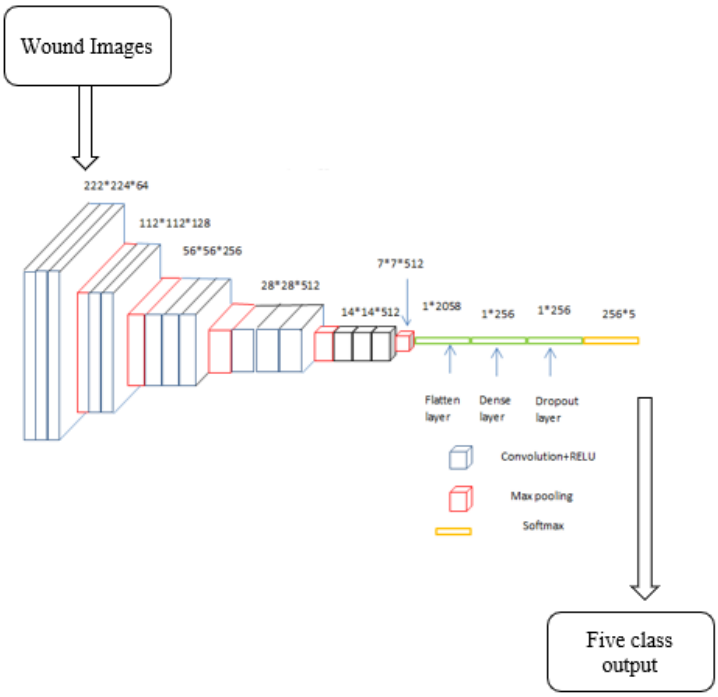
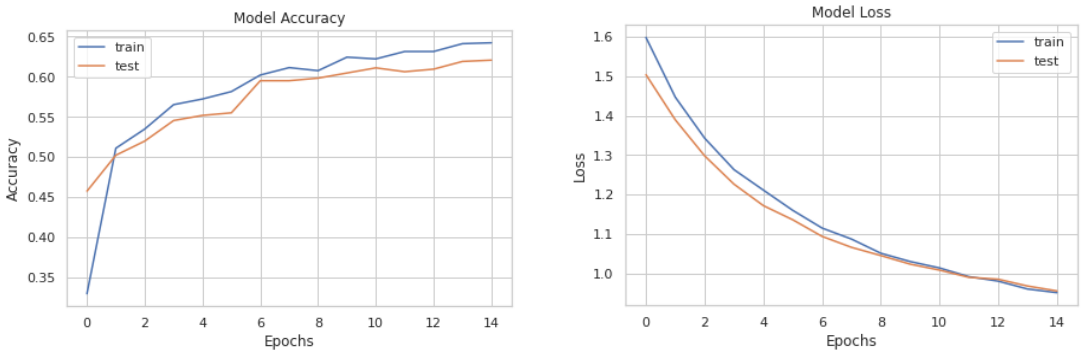


Figure 5.3: Baseline Model for wound image classification

The VGG16 network was trained and validated using the same 4165 train datasets of five classes and 625 validation dataset for validation of result. In order to maintain fair comparison, the hyperparameters for training the network will be maintained constant for next experiments as well.. I have used Adam optimizer for adaptive learning rate of 10^{-4} with batch size of 32 for 15 epochs. This resulted in VGG 16 to return the accuracy of 62.23% and loss of 0.95. The average precision in VGG 16 was found to be 0.62 and the average recall was found to be 0.62. The reason for selecting 15 epochs is because of high training time for each epoch and that most of the saturation had occurred within 15 epochs of training. The graphs related to this study is shown in figure 5.4 and 5.5 respectively.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.4: Baseline VGG 16 without Dense layer and without Body Map

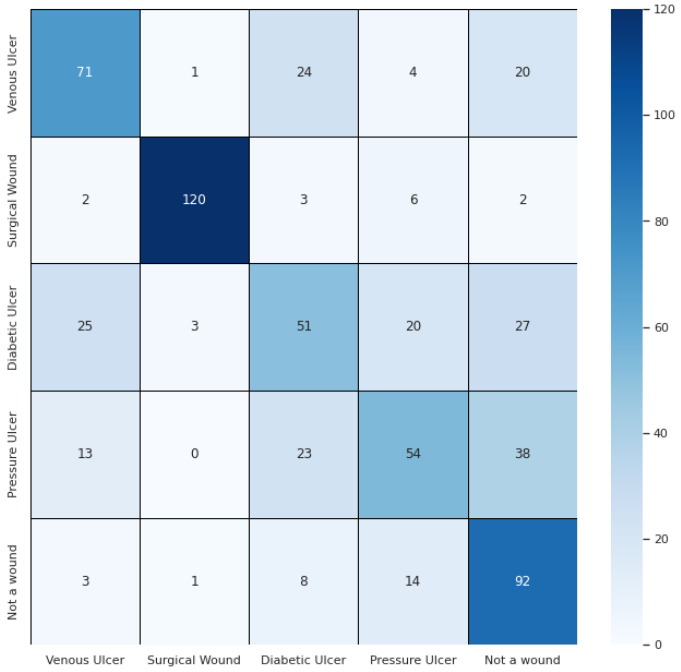


Figure 5.5: VGG 16 Confusion Matrix without Dense layer and without Body Map

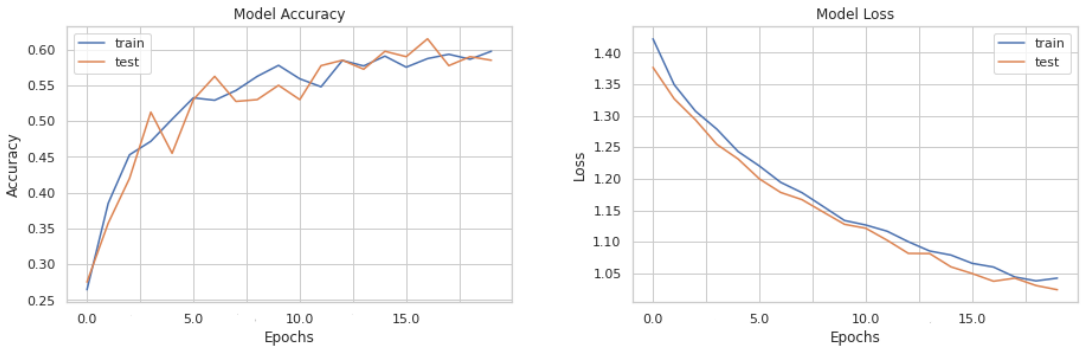
The results of the experiment can be summarized in the table 5.1.

Table 5.1: Performance Metrics for VGG 16 without Dense layer and without Body map

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.62	0.59	0.61
Surgical Wound	0.96	0.90	0.93
Diabetic Ulcer	0.47	0.4	0.43
Pressure Ulcer	0.55	0.42	0.48
Not a Wound	0.51	0.78	0.62
Accuracy			**62.23%**

5.3 Baseline Model with InceptionV3 and five softmax

In order to compare the baseline model of VGG 16, I have experimented the same dataset using inception V3 as well. This was done to ensure which of the model outperformed on the given wound dataset. From the experiment it was evident that VGG 16 outperformed inception V3 in terms of accuracy and loss as well. Implementation of inceptionv3 was done after converting the input images to size of 299 * 299 since the input of inceptionv3 only accepts the aforementioned format. One of the key differences between VGG-16 and Inception-v3 is their architecture. VGG-16 has a relatively simple architecture with many layers of 3x3 convolutional filters followed by max-pooling layers. In contrast, Inception-v3 employs a more complex architecture that uses a combination of convolutional filters of different sizes and pooling operations, and also incorporates a technique called "inception modules" to enable efficient and effective feature extraction. However, this complex architecture does not always guarantee better results. The results are entirely dependent on specific dataset and task at hand and shown as in fig 5.6.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.6: Baseline InceptionV3 without Dense layer and without Body Map

The results of the experiment with Inceptionv3 are summarized in the table 5.2.

Table 5.2: Performance Metrics for InceptionV3 without Dense layer and without Body map

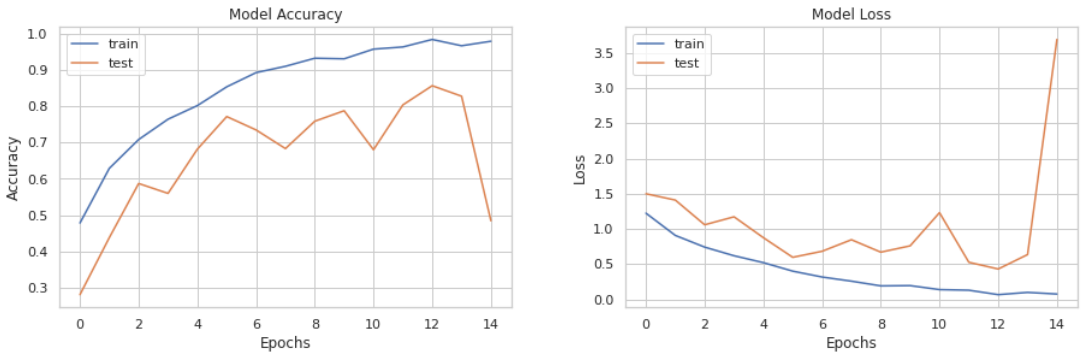
Wound type	**Precision**	**Recall**	**F1 Score**
Venous Ulcer	0.56	0.68	0.62
Surgical Wound	0.5	0.56	0.53
Diabetic Ulcer	0.68	0.36	0.47
Pressure Ulcer	0.64	0.77	0.7
Not a Wound	0.47	0.4	0.43
Accuracy			**58.3%**

The baseline comparison resulted in VGG 16 outperforming inceptionV3 in terms of accuracy and loss. The average accuracy in inceptionv3 was found to be 58.3% which was 4% lower than that of VGG 16 with accuracy of 62.23%. In case of loss as well, the inceptionv3 resulted in loss of 1.03 whereas VGG 16 had returned loss of 0.95. It was evident that the difference in loss of inceptionv3 was higher by margin of 0.08 than that of VGG 16. This was the motivation of this thesis to move ahead with VGG 16 for further experiments since baseline model of VGG 16 outperformed inceptionv3 on the same dataset and with same parameters. The training parameters used in inception v3 was Adam optimizer of learning rate 10^{-4}, Batch size of 32 and 15 epoch. Even though inceptionv3 is longer than wider than VGG 16 network, for our study, we don't need to learn unnecessary

features from image for wound classification task. Also, use of body map is done so that less computing resources will be used in order to achieve better performance. Since, VGG 16 network uses less resources and does not learn unnecessary features than inceptionV3, the further experiment will be performed using VGG 16 so that better performance will be achieved using fewer resources.

5.4 VGG 16 with one added dense layer, 0.5 dropout, five softmax and SGD optimizer (Overfitting)

The results of VGG16 with only 62.23% accuracy was still unsatisfactory. So in order to perform transfer learning to one dense layer, a dense layer with 256 neurons and dropout of 0.5 with ReLU activation was added at end of the network. The optimizer used was SGD optimizer. However, it was noticed that that there was sudden surge increase in model training accuracy but the validation accuracy did not increase accordingly rather decreased at last moment. So it was obvious that model had overfitted. The validation accuracy had dropped to 50% when training accuracy stayed at 97%. Overfitting occurs when the network tries to fit each and every training data but fails to do so for validation data. The cause of overfitting in this case was because of low dropout of neurons at 50%. So, for the next experiment this dropout was increased to 0.7, which means 70% of the neurons are randomly dropped out or set to 0. The result of overfitting in shown in figure 5.7.



(a) Accuracy vs Epoch

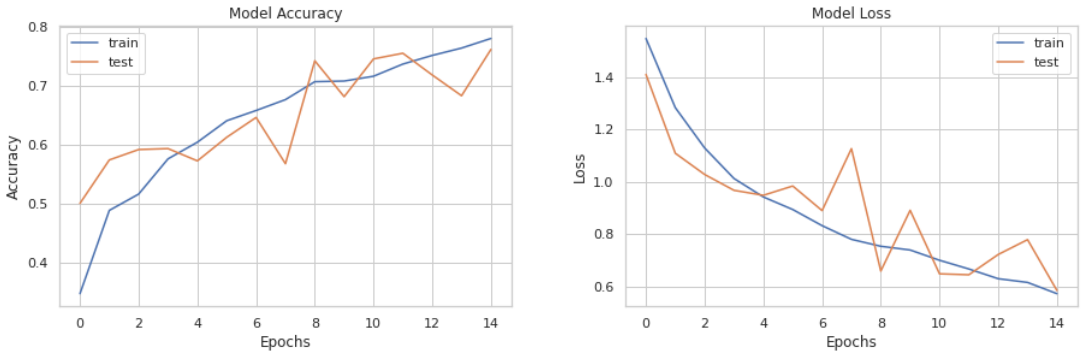
(b) Loss vs Epoch

Figure 5.7: VGG 16 accuracy with one dense layer and dropout of 0.5 and SGD (Overfitting)

The model had overfitted for dropout of 0.5 in 256 neuron dense layer. This had occured because of significantly higher value of dropout. Hence, for the next experiment, regularization value of drop out was increased to 0.7 so that the network does not learn unnecessary features and unnecessary noises.

5.5 VGG 16 with one added dense layer, 0.7 dropout, five softmax and SGD optimizer (Overfitting Removed):

After addition of one dense layer and dropout of 0.5, the graph depicted that the algorithm had overfitted. The options were either to increase training data through further augmentations or to increase the value of regularization. But since the training time for one epoch is around 300ms it was evident that increasing training data will be time consuming. So, I chose the latter and increased the dropout of fully connected 256 neurons dense layer from 0.5 to 0.7. The observed accuracy was 76.08% with precision and recall of 0.79 and 0.78 respectively. This increment in accuracy, precision and recall was due to the fact that the unnecessary features were not learnt by the network because of increase in dropout value. The dense layer are trainable and this trainable layer resulted in increased accuracy. To validate the results, the training was initially performed with SGD optimizer of learning rate 10^{-3}.The observed result was as follows in figure 5.8 and 5.9:



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.8: VGG 16 with one dense layer and dropout of 0.7 and SGD

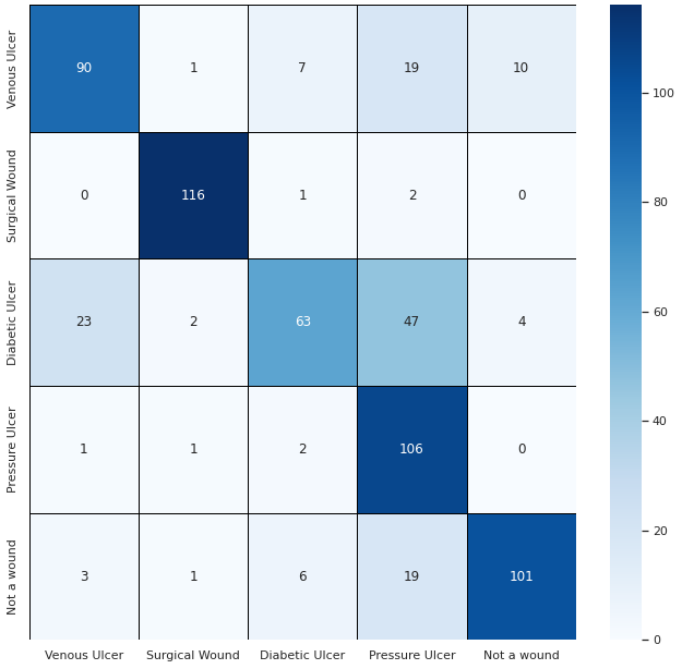


Figure 5.9: VGG 16 Confusion Matrix with one dense layer and dropout of 0.7 (SGD)

The values of performance metrics of the network are depicted in table 5.3.

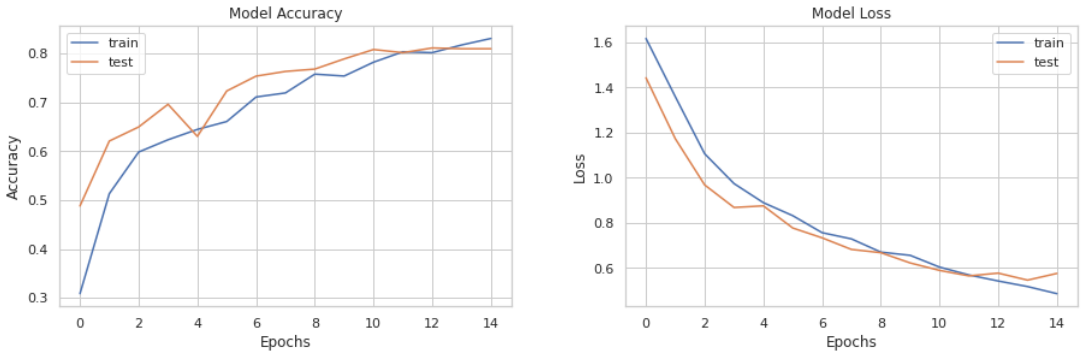
Table 5.3: VGG 16 metrics with one dense layer and dropout of 0.7 (SGD)

Wound type	**Precision**	**Recall**	**F1 Score**
Venous Ulcer	0.77	0.71	0.74
Surgical Wound	0.96	0.97	0.97
Diabetic Ulcer	0.80	0.45	0.58
Pressure Ulcer	0.55	0.96	0.7
Not a Wound	0.88	0.78	0.82
Accuracy			**76.08%**

The accuracy of the wound classification was found to be at 76.08% using SGD Optimizer. Now we will check the change in accuracy by using Adam Optimizer instead of SGD Optimizer.

5.6 VGG 16 with one added dense layer, 0.7 dropout, five softmax and Adam optimizer

The use of Adam optimizer with learning rate 10^{-4} resulted in accuracy of 81.39% which was 5% higher than that of SGD. Adam is more accurate than SGD in wound image classification is because it adapts the learning rate for each parameter based on the estimate of the first and second moments of the gradients. This allows Adam to converge faster and more accurately than SGD. In wound image classification, there are large number of parameters and features that need to be learned. In such cases, Adam's adaptive learning rate can help in fine-tuning the model's parameters, making it better suited to the specific task of wound image classification. This increment can be seen after increment of accuracy by almost 5% after using Adam Optimizer as shown in fig 5.10.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.10: VGG 16 with one dense layer and dropout of 0.7 and Adam

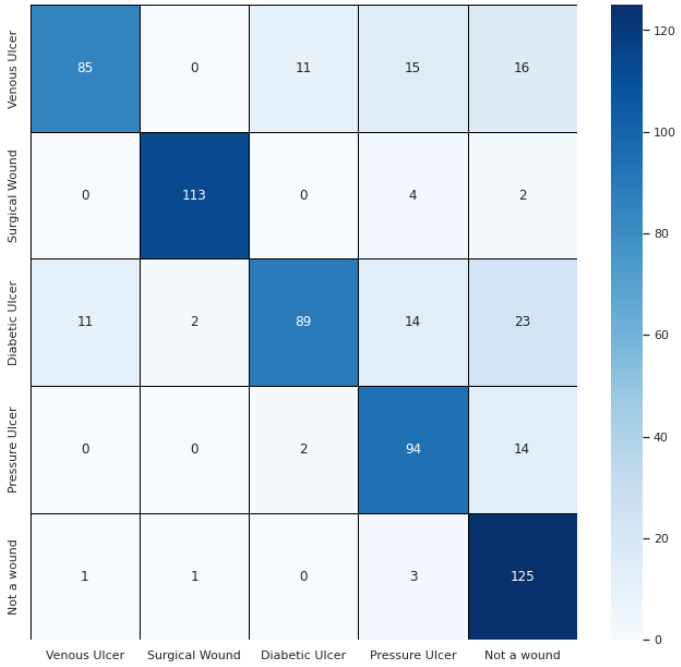


Figure 5.11: VGG 16 Confusion Matrix with one dense layer and dropout of 0.7 (Adam)

Table 5.4: VGG 16 metrics with two dense layer and dropout of 0.7 (SGD)

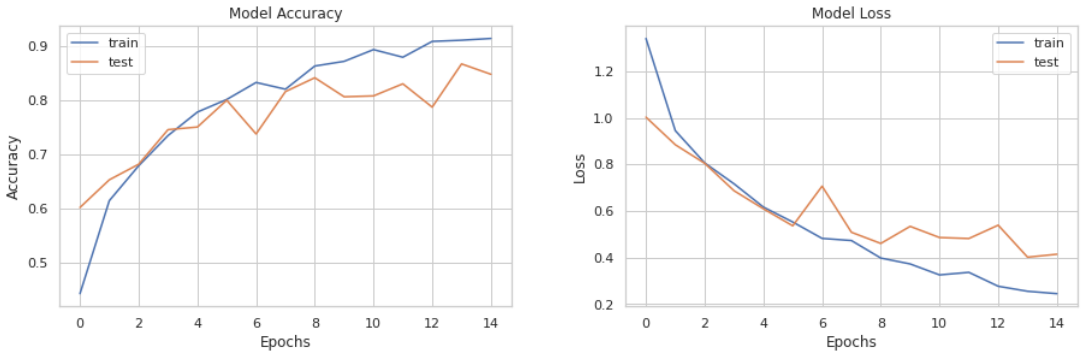
Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.88	0.67	0.76
Surgical Wound	0.97	0.95	0.96
Diabetic Ulcer	0.87	0.64	0.74
Pressure Ulcer	0.72	0.85	0.78
Not a Wound	0.69	0.96	0.81
Accuracy			**81.39%**

5.7 VGG 16 with two added dense layer, 0.7 dropout, five softmax and SGD optimizer

After addition of dense layer with 256 neurons on top of previous dense layer with Relu activation and 0.7 dropout, there was significant increase in training time for the algorithm. However, the validation accuracy had now reached upto 85.4% which is 4% higher than that of previous one dense layer layer. Hence, it is evident that adding another dense layer in another dense layer increased the

overall accuracy of the system. The overall value of precision and recall is found to be 0.85 and 0.87 respectively which is higher than that of one dense layer only.

Two dense layers in a neural network was found to be more accurate than one dense layer because they allowed for more complex representations of the input data to be learned. Each layer in a neural network learns to extract different features and representations of the data, and adding more layers allows for increasingly abstract and complex representations to be learned. Addition of another dense layer is done by cascade connection with previous dense layer. With just one dense layer, the model may not be able to capture all the relevant features and patterns in the data. However, by adding a second dense layer, the model has the ability to learn more complex representations that can better discriminate between different classes of wound.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.12: VGG 16 with two dense layer and dropout of 0.7 and SGD

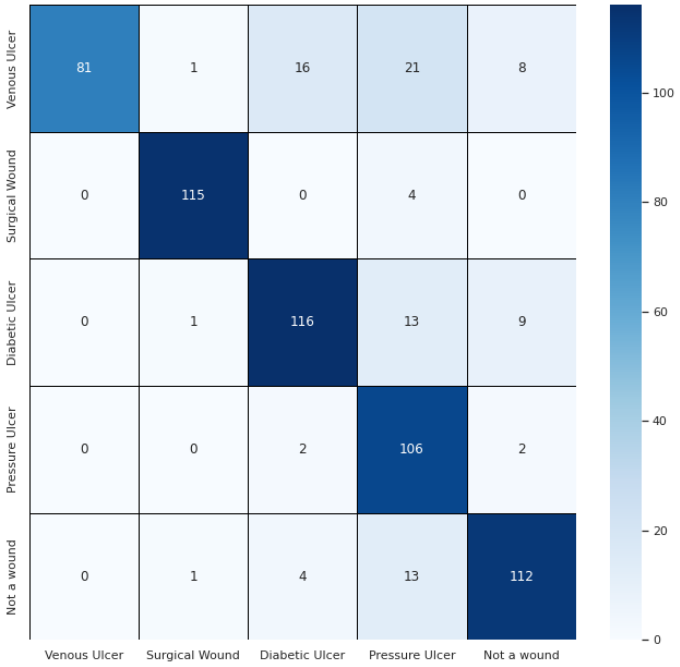


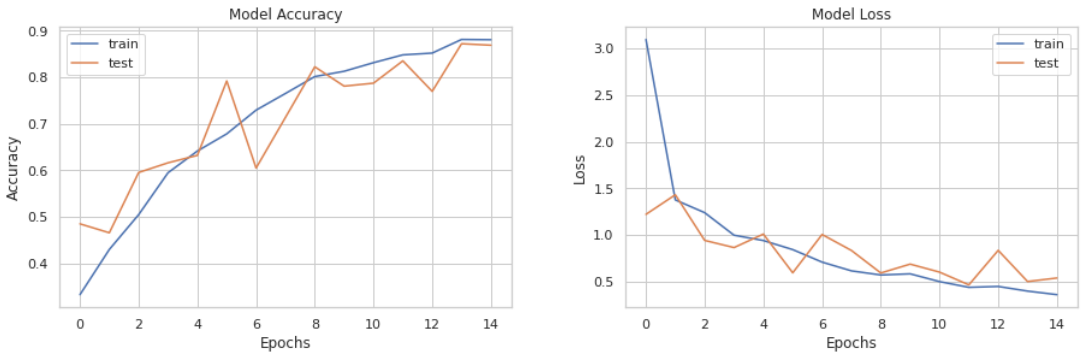
Figure 5.13: VGG 16 Confusion Matrix with two dense layer and dropout of 0.7 (SGD)

Table 5.5: VGG 16 metrics with two dense layer and dropout of 0.7 (SGD)

Wound type	**Precision**	**Recall**	**F1 Score**
Venous Ulcer	1.0	0.64	0.78
Surgical Wound	0.97	0.97	0.97
Diabetic Ulcer	0.84	0.83	0.84
Pressure Ulcer	0.68	0.96	0.79
Not a Wound	0.85	0.86	0.86
Accuracy			**85.4%**

5.8 VGG 16 with two added dense layer, 0.7 dropout, five softmax and Adam optimizer

The experiment conducted by using Adam optimizer with learning rate of 10^{-4} resulted in increased accuracy by 2% at 87.44% than when using SGD optimizer. The precision increased by 1% and recall increased by 2% and reached 0.88 and 0.87 respectively.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.14: VGG 16 with two dense layer and dropout of 0.7 and Adam

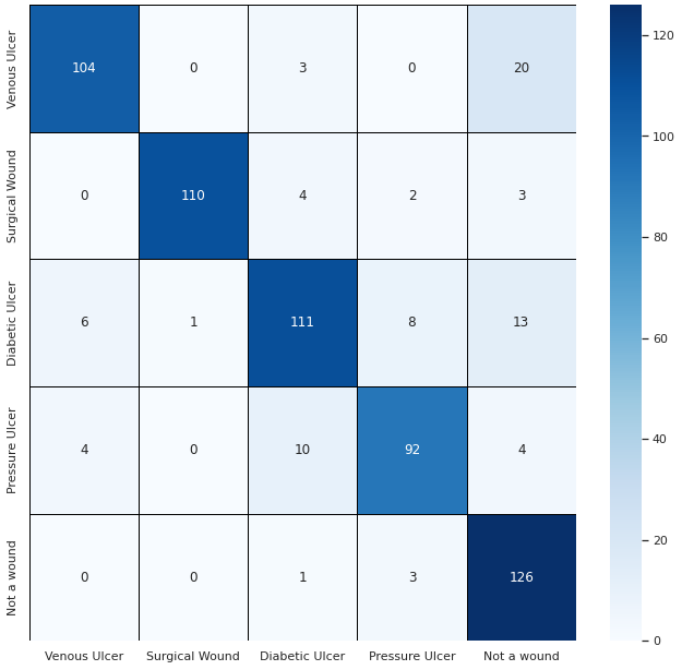


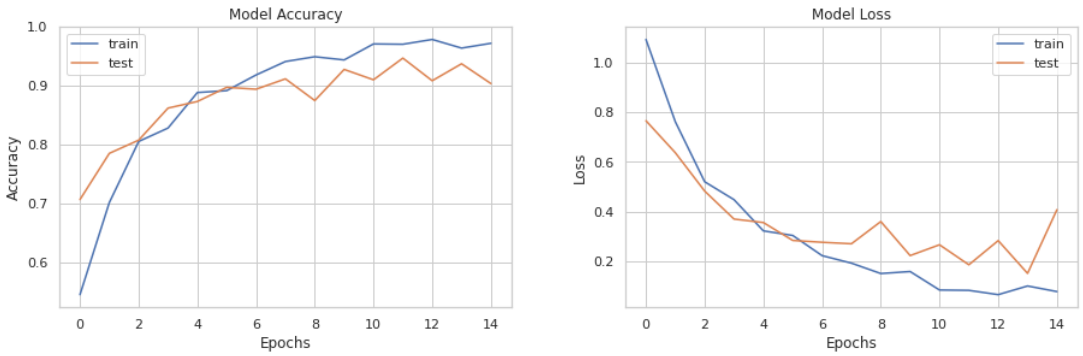
Figure 5.15: VGG 16 confusion matrix with two dense layer and dropout of 0.7 (Adam)

Table 5.6: VGG 16 metrics with two dense layer and dropout of 0.7 (Adam)

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.91	0.82	0.86
Surgical Wound	0.99	0.92	0.96
Diabetic Ulcer	0.86	0.80	0.83
Pressure Ulcer	0.88	0.84	0.86
Not a Wound	0.76	0.97	0.85
Accuracy			**87.44%**

5.9 VGG16 with one dense layer and body map, 0.7 dropout, 5 softmax and SGD Optimizer

The experiment with VGG16 was implemented now with body map using multi-layer perceptron but keeping the other parameters as constant since the comparison framework should be constant. Initially, the experiment was conducted by using SGD optimizer with learning rate of 10^{-3} and one dense layer. The body map number of wound location was initially fed into a multi-layer perceptron. The output of the perceptron was appended into that of VGG 16 to feed them both in dense layer. It was observed that the implementation of body map increased the accuracy upto 90.82% in 15 epochs which was even higher than that of two dense layer without body map. The introduction of body map improved accuracy by 3% in the single dense layer. The metrics and charts are shown in figure 5.16 and 5.17 respectively:



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.16: VGG 16 with one dense layer, bodymap and dropout of 0.7 and SGD

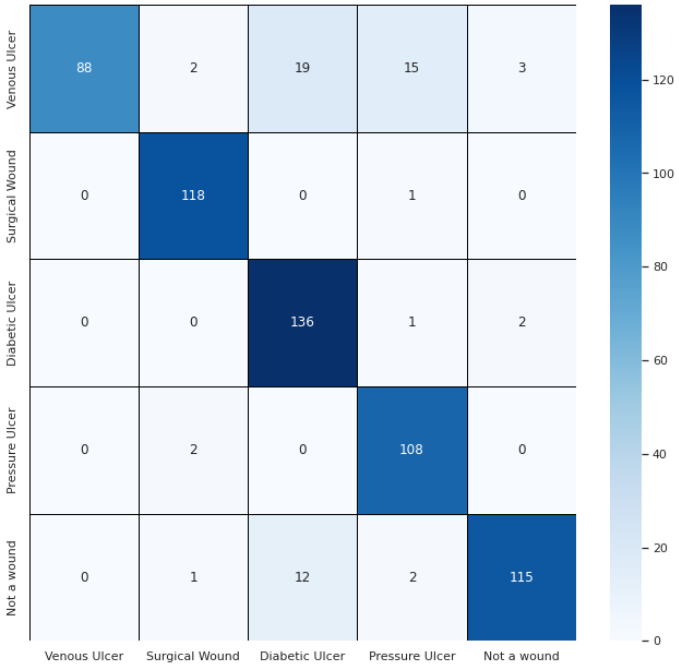


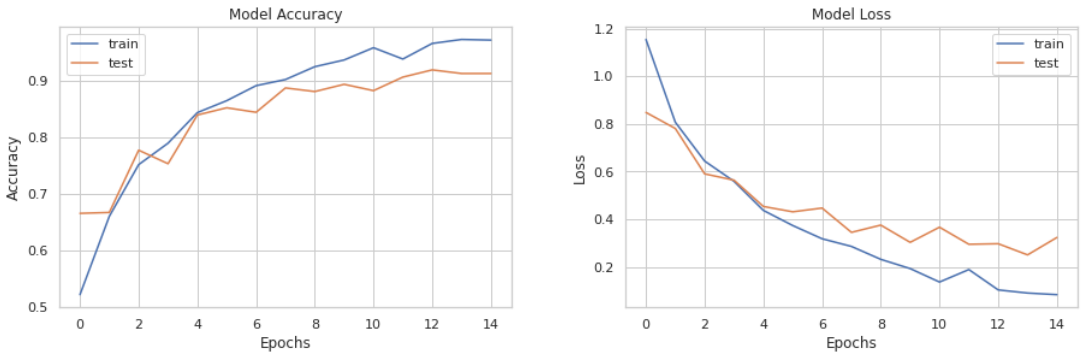
Figure 5.17: VGG 16 Confusion Matrix with one dense layer, bodymap and dropout of 0.7 (SGD)

Table 5.7: VGG 16 metrics with two dense layer and dropout of 0.7 (Adam)

Wound type	Precision	Recall	F1 Score
Venous Ulcer	1.0	0.69	0.82
Surgical Wound	0.96	0.99	0.98
Diabetic Ulcer	0.81	0.98	0.89
Pressure Ulcer	0.85	0.98	0.91
Not a Wound	0.96	0.88	0.92
Accuracy			**90.82%**

5.10 VGG16 with one dense layer and body map, 0.7 dropout, 5 softmax and Adam Optimizer

The above experiment was repeated with Adam optimizer with learning rate of 10^{-4} and all the other parameters were constant. It was then observed that the learning curve was smooth and 1% better than that of SGD optimizer. That means the total accuracy of the network was improved to 91.02%. The reason why Adam has a smoother learning curve than SGD is that Adam incorporates a momentum term and adapts the learning rate for each weight based on the history of gradients. The momentum term helps Adam to dampen oscillations and noise in the gradients, allowing the optimization process to converge more smoothly towards the optimal solution. Additionally, Adam can adjust the learning rate separately for each weight, making it more adaptive and flexible than SGD, which uses a fixed learning rate for all weights.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.18: VGG 16 with one dense layer, body map, dropout of 0.7 and Adam

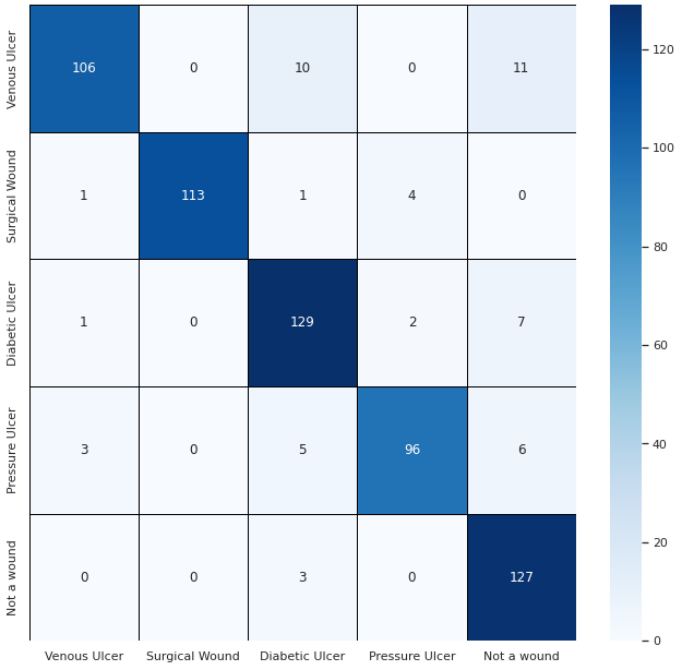


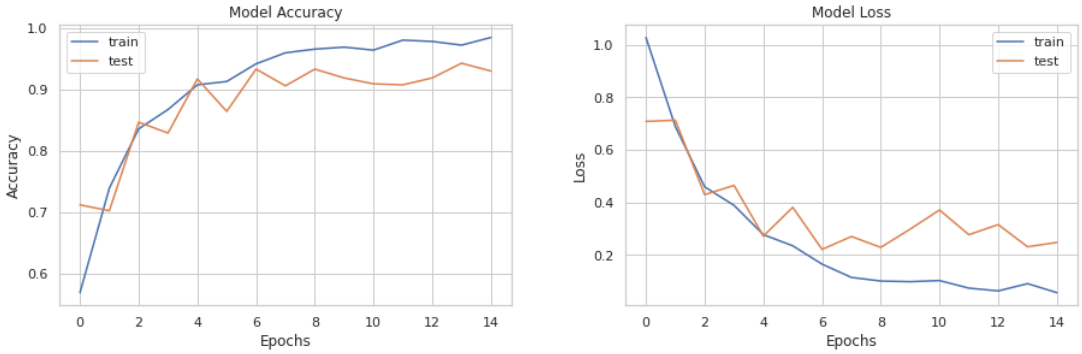
Figure 5.19: Confusion Matrix of VGG 16, one dense layer, body map, dropout of 0.7 and Adam Optimizer

Table 5.8: Performance metrics of VGG 16, one dense layer, body map and Adam Optimizer

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.95	0.83	0.89
Surgical Wound	1.0	0.95	0.97
Diabetic Ulcer	0.87	0.93	0.9
Pressure Ulcer	0.94	0.87	0.91
Not a Wound	0.84	0.98	0.9
Accuracy			**91.02%**

5.11 VGG16 with two dense layer and body map, 0.7 dropout, 5 softmax and SGD Optimizer

The experiment included two dense layers for better learning and implementation of body map. The denselayer contains 256 neurons each and the network was trained upto 15 epochs. Implementation of two dense layers improved the accuracy by 2% than previous learning with one dense layer. The dense layer are added using sequential model and the output of first dense layer is input into second layers. Both of the dense layer are trainable with dropout of 0.7 each and initial training is done using SGD optimizer of learning rate 10^{-3}. The overall accuracy of the network is 93.25%.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.20: VGG16 with two dense layer, body map, dropout of 0.7 and SGD

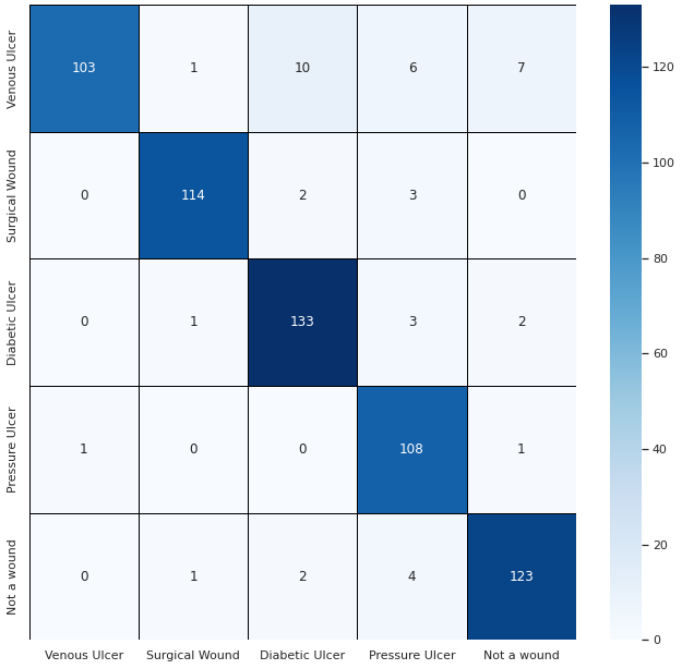


Figure 5.21: VGG16 confusion matrix with two dense layer, body map, dropout of 0.7 and SGD

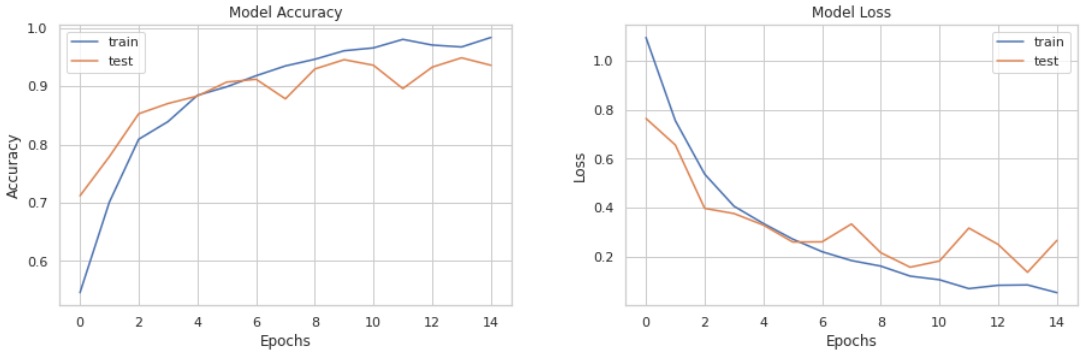
Table 5.9: Performance metrics of VGG 16, two dense layer, body map and SGD Optimizer

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.99	0.81	0.89
Surgical Wound	0.97	0.96	0.97
Diabetic Ulcer	0.90	0.96	0.93
Pressure Ulcer	0.87	0.98	0.92
Not a Wound	0.92	0.95	0.94
Accuracy			**93.25%**

5.12 VGG16 with two dense layer and body map, 0.7 dropout, 5 softmax and RMSProp Optimizer

The experiment was conducted using RMSprop in order to check whether the accuracy improves or degrades using RMSprop Optimizer. The RMSprop optimizer increased the accuracy by 1% but the learning curve was too fluctuated. The main idea behind RMSProp is to divide the learning rate by a running average of the root mean squared (RMS) of the past gradients for each weight. This allows the

algorithm to adjust the learning rate for each weight based on the history of the gradients, which helps to prevent the learning rate from being too large or too small. The learning curve is fluctuated because RMSProp is designed to adjust the learning rate for each weight based on the magnitude of the gradients.Another reason for fluctuation is because RMSProp optimization is sensitive to hyperparameters. The batch size for this experiment is 32 for 15 epochs and the resulting accuracy is 94.09%.



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.22: VGG16 with Body map, dropout of 0.7, two dense layer and RMSprop

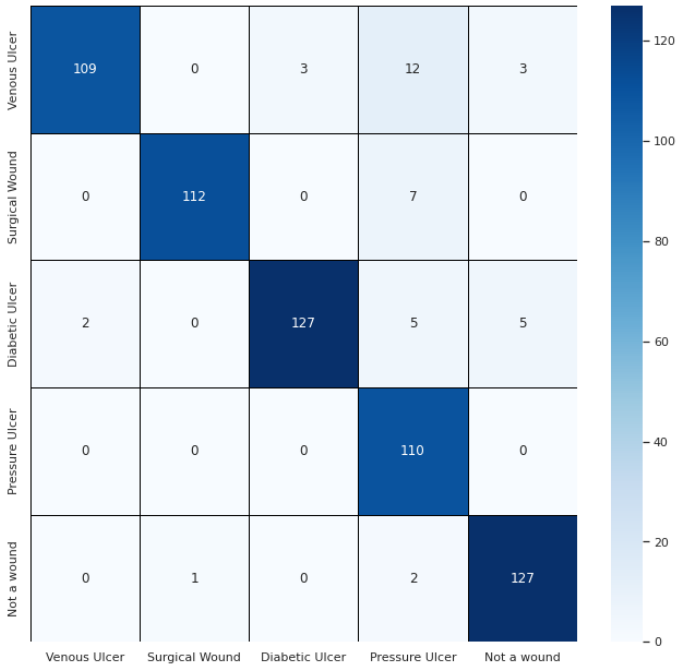


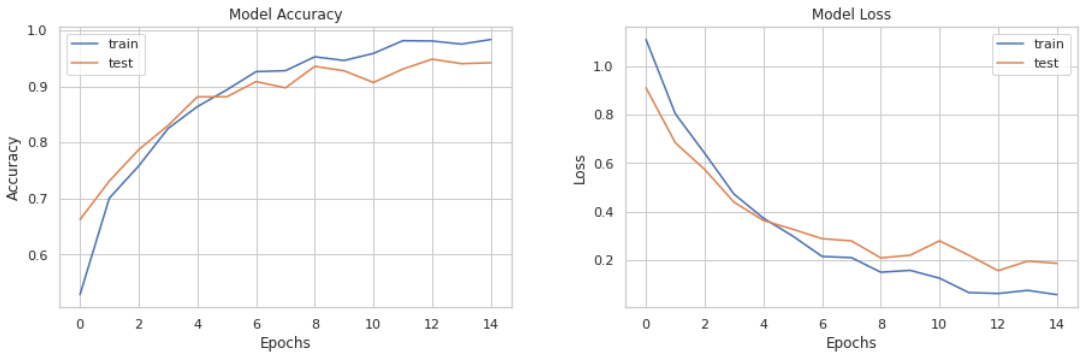
Figure 5.23: VGG16 confusion matrix, Body map, dropout of 0.7, two dense layer and RMSprop

Table 5.10: Performance metrics of VGG16 with two dense layer and body map and RMSProp Optimizer

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.98	0.86	0.92
Surgical Wound	0.99	0.94	0.97
Diabetic Ulcer	0.98	0.91	0.94
Pressure Ulcer	0.81	1.0	0.89
Not a Wound	0.94	0.98	0.96
Accuracy			**94.09%**

5.13 VGG16 with two dense layer and body map, 0.7 dropout, 5 softmax and Adam Optimizer

The final study was conducted using body map and two dense layer using Adam optimizer. The accuracy was only 0.57% more than RMSProp but the learning curve was smoothened out. The final accuracy observed was found to be 94.57% with 15 epochs. The parameters used for this study is batch size of 32, categorical cross entropy loss and dropout of 0.7. The graphs related to the study are depicted in fig 5.24 and 5.25:



(a) Accuracy vs Epoch

(b) Loss vs Epoch

Figure 5.24: VGG16 performance with Body map, dropout of 0.7, two dense layer and Adam

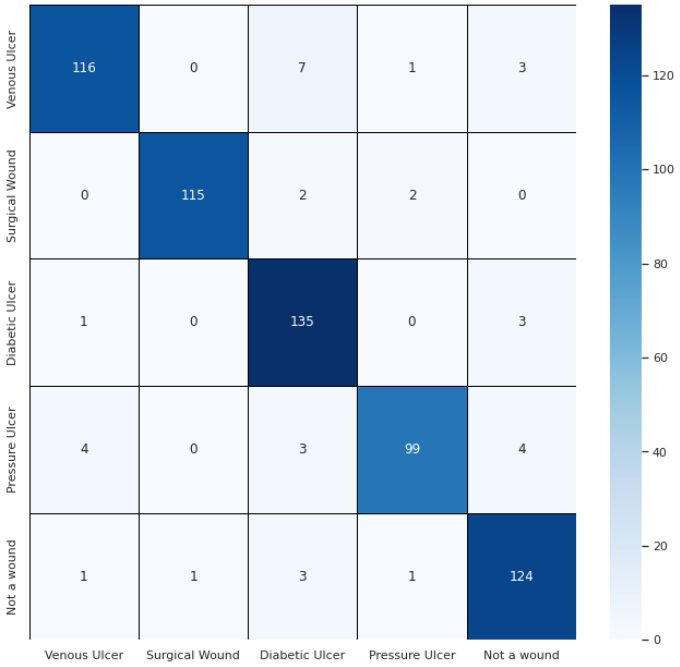


Figure 5.25: VGG16 confusion matrix, Body map, dropout of 0.7, two dense layer and Adam

Table 5.11: Performance metrics of VGG16 with two dense layer and body map and RMSProp Optimizer

Wound type	Precision	Recall	F1 Score
Venous Ulcer	0.95	0.91	0.93
Surgical Wound	0.99	0.97	0.98
Diabetic Ulcer	0.90	0.97	0.93
Pressure Ulcer	0.9	0.9	0.93
Not a Wound	0.93	0.95	0.94
Accuracy			**94.57%**

5.14 Quantitative Analysis

The performance of transfer learning using two dense layer of 256 neurons and body map was evaluated by varying the optimization hyperparameter by using Adam optimizer, SGD optimizer and RMSprop optimizer. Table 5.12 depicts the macro average precision and recall and weighted average precision and recall along with accuracy for these three optimization parameters. The test dataset is shuffled to calculate weighted precision and recall for imbalanced class whereas

same number of test dataset are used in macro average precision and recall.

Table 5.12: Performance result of VGG 16 + Dense Layers + Body Map for varying optimization techniques

	Macro Average		**Weighted Average**		
Optimizer	**Precision**	**Recall**	**Precision**	**Recall**	**Accuracy**
SGD	0.934	0.931	0.934	0.932	93.25%
RMSProp	0.947	0.944	0.948	0.946	94.09%
Adam	**0.951**	**0.947**	**0.945**	**0.944**	**94.57%**

The quantitative analysis of the three optimization techniques used in VGG 16 with two dense layer and body map showed that the best result was returned by use of Adam optimizer. Even thought there was increment of only 0.57% which is not visible in two decimal places, overall macro and weighted precision and recall have higher values for Adam optimizer than the rest of two optimizer.

The ROC curve obtained for VGG 16 network classification with two dense layer and use of body map is depicted in figure 5.26. The experimental settings done for the final study is as follows:

- Image Size: 244 * 244
- Batch Size = 32
- Loss = Categorical Cross entropy loss
- Epoch = 15
- Optimization= Adam optimizer 10^{-4}
- Regularisation = Dropout of 0.7

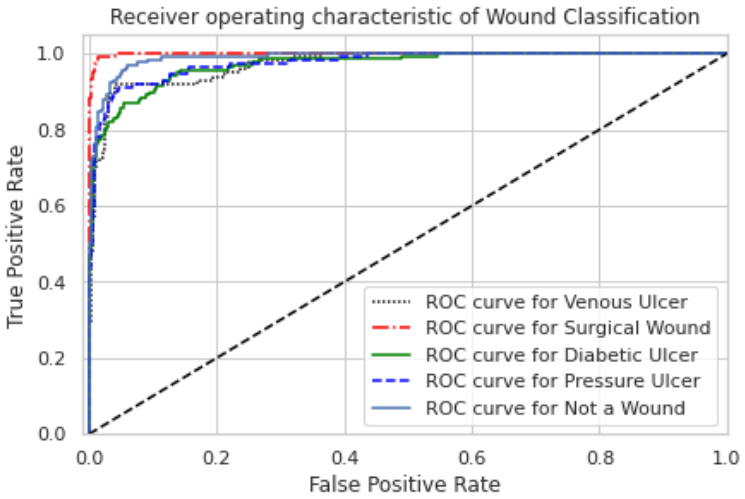


Figure 5.26: ROC Curve for five class classification

The value of Area under the curve (AUC) for class surgical wound was found to be highest at 0.99. The next greater value of AUC was found to be of class "Not a wound" 0.98. The next corresponding values of AUC for class "Pressure Ulcer" and "Venous Ulcer" were found to be at 0.97 whereas the AUC for the class "Diabetic Ulcer" was found to be at 0.96. The reason behind surgical wound having the highest AUC was because the surgical wound has different image texture when compared to other wounds. For example, A surgical wound has an elongated texture with no significant blood or red colors which makes it distinguishable from other wound types. Also, since surgical wound are generally located in joints of body, the use of body map has increased its true positive rate when compared to others because pressure, venous and diabetic ulcer only occur in rare cases in locations where surgical wound are most likely to happen.

Similarly, The class "Not a wound" has AUC of 0.97 because there was no body map used for training the dataset of corresponding label. Even though, "Not a Wound" label is highly distinguishable in terms of features and texture from other wound images, its AUC is comparatively low than that of surgical wound because during training there was no use of body map in the Multi layer perceptron and all the values were initialized to zero. This concludes that use of body map has significant positive impact on network in context of wound classification. The value of FPR in "not a wound" indicates that there are some wound images which

as identified as non- wound by the network. This happens when wound images are very small or the texture is similar to healed wound.

5.15 Ablation Study

The study with various changes and introduction of dense layer involves ablation study of the thesis. The ablation study of VGG 16 model for wound classification was performed using different values of dropouts and change in network parameters. The evaluation metrics used for ablation study are precision, recall, F1 score and accuracy for different cases. Table 5.13 shows the performance results of the network with integration of trainable dense layer and body map into the pre-trained VGG 16 network and inceptionv3 network.

Table 5.13: Ablation Study

Network Type	**Precision**	**Recall**	**Accuracy**	**F1-Score**
InceptionV3	0.57	0.56	58.3%	0.56
VGG 16	0.62	0.62	62.23%	0.61
VGG 16 + one dense layer(SGD)	0.79	0.78	76.08%	0.76
VGG 16 + one dense layer(Adam)	0.83	0.82	81.39%	0.81
VGG 16 + two dense layer(SGD)	0.87	0.85	85.4%	0.85
VGG 16 + two dense layer(Adam)	0.88	0.87	87.44%	0.87
VGG 16 + one dense layer + body map (SGD)	0.92	0.91	90.82%	0.90
VGG 16 + one dense layer + body map (Adam)	0.92	0.91	91.02%	0.91
VGG 16 + two dense layer + body map (SGD)	0.93	0.93	93.25%	0.93
VGG 16 + two dense layer + body map (RMSProp)	0.94	0.94	94.09%	0.94
VGG 16 + two dense layer + body map (Adam)	**0.95**	**0.94**	**94.57%**	**0.94**

The analytical study of VGG 16 with different integration showed that the network

improved its performance whenever the trainable dense layer was added. This is because although VGG 16 is a pre-trained model on 1.2 million images and it avoids the need to build a training model from scratch, it does not perform well for different datasets as per our needs. Initial study of VGG 16 showed accuracy of only 62.23% which was very low. So the need to transfer learning was then realized by adding some extra dense layer on top of VGG 16 to improve its performance. Initial study with inceptionV3 and VGG 16 showed that VGG 16 was more effective for our study since it returned accuracy of 4% higher than inceptionV3. So, rest of the study was conducted with help of VGG 16. In the next study, after body map was introduced in VGG 16, the performance improved significantly. Even one dense layer with body map outperformed the VGG 16 with two dense layer and no body. The best performance was returned by VGG 16 with use of two dense layer and a body map using Adam optimizer at 94.57%. In order to maintain fairness in ablation study, all the networks have are trained with same parameters. The parameters include 4165 train dataset, 625 test dataset, 15 epoch, 32 batch size and categorical cross entropy loss.

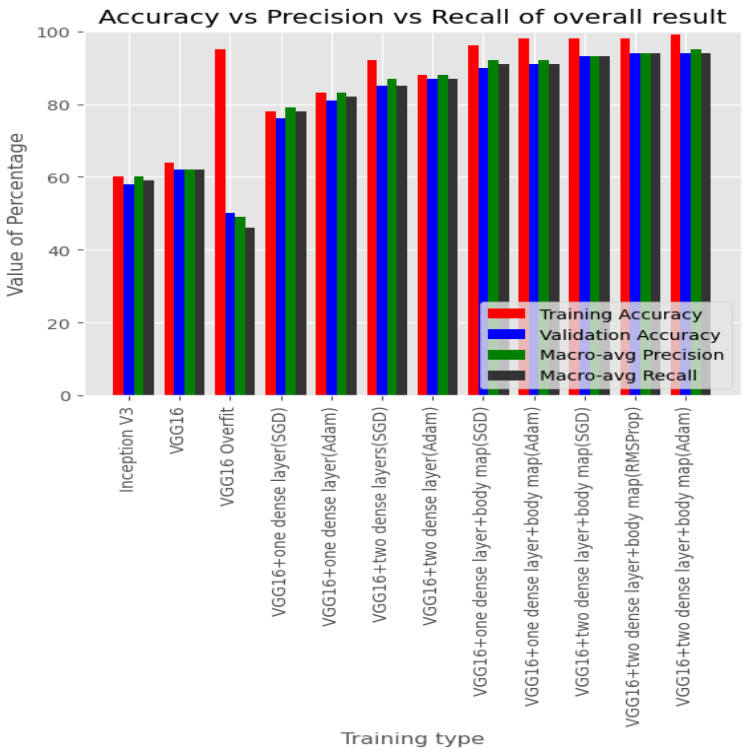


Figure 5.27: Bar diagram of overall result

The figure 5.38 is the overall depiction of each type of networks used for studying the behaviour of transfer learning. It is evident from diagram that the overall performance of the system increases as we introduce different relevant parameters such as body map, optimization and dense layer in the network. The study is also performed for each wound type which is shown in figure 5.28, 5.29 and 5.30.

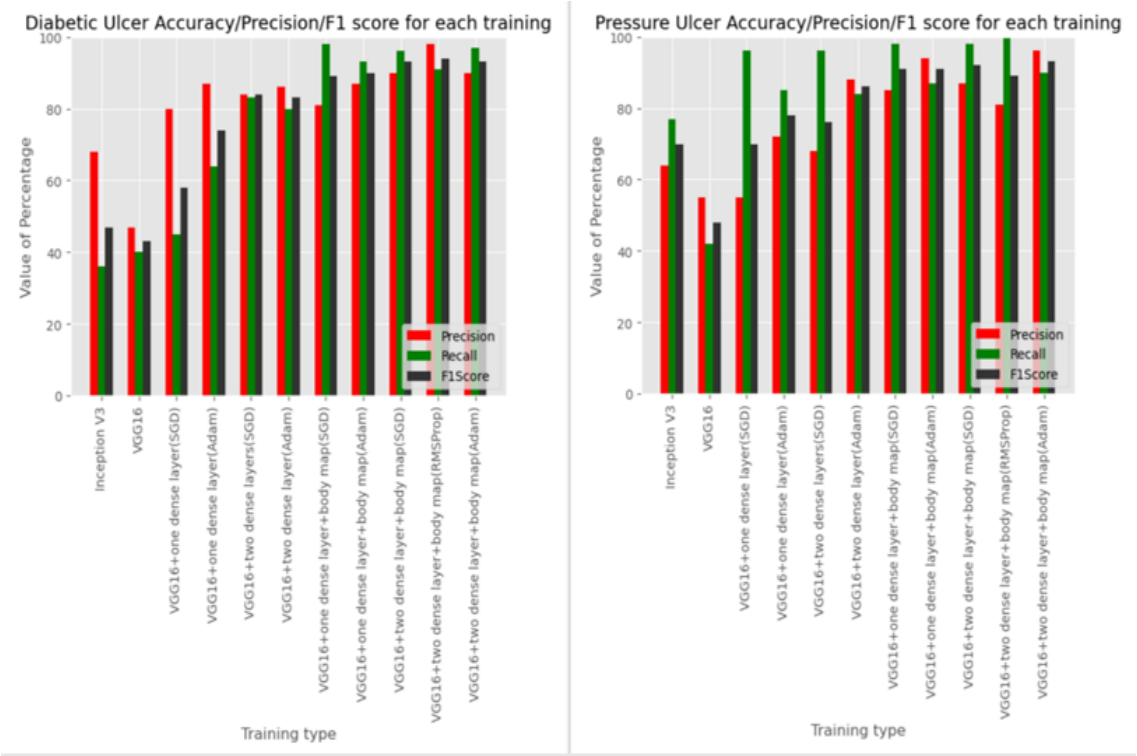


Figure 5.28: Diabetic Vs Pressure metrics for all networks

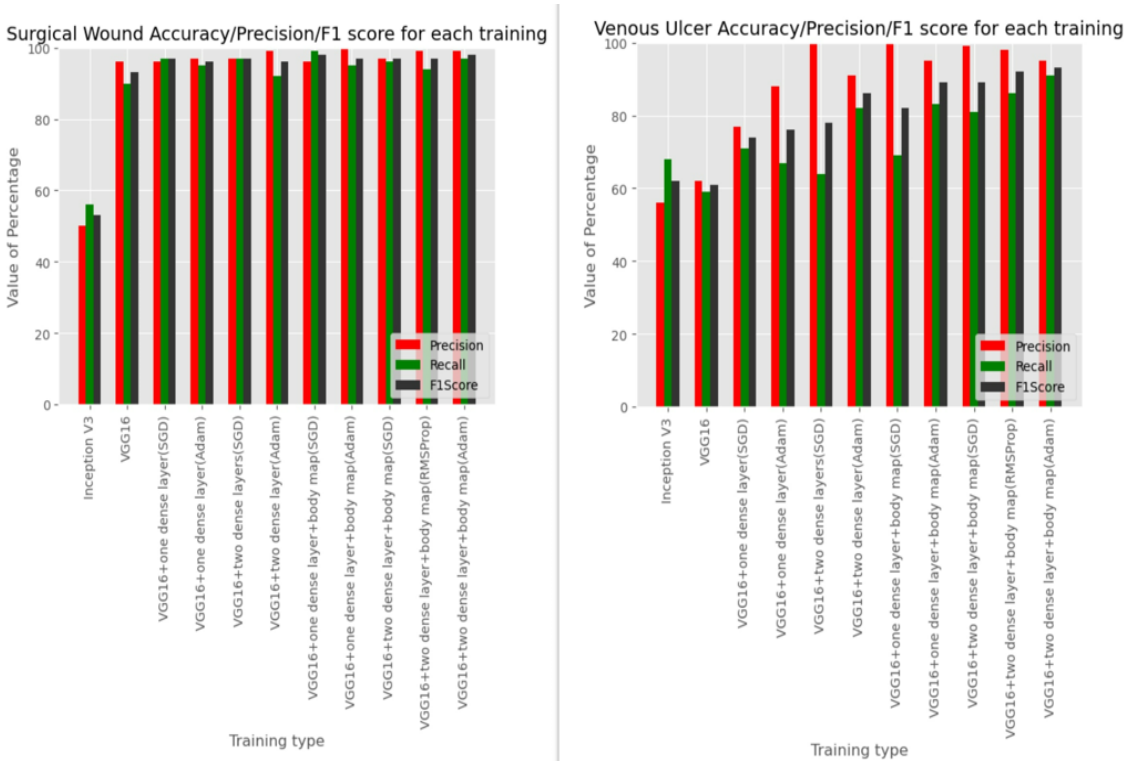


Figure 5.29: Surgical vs Venous metrics for all networks

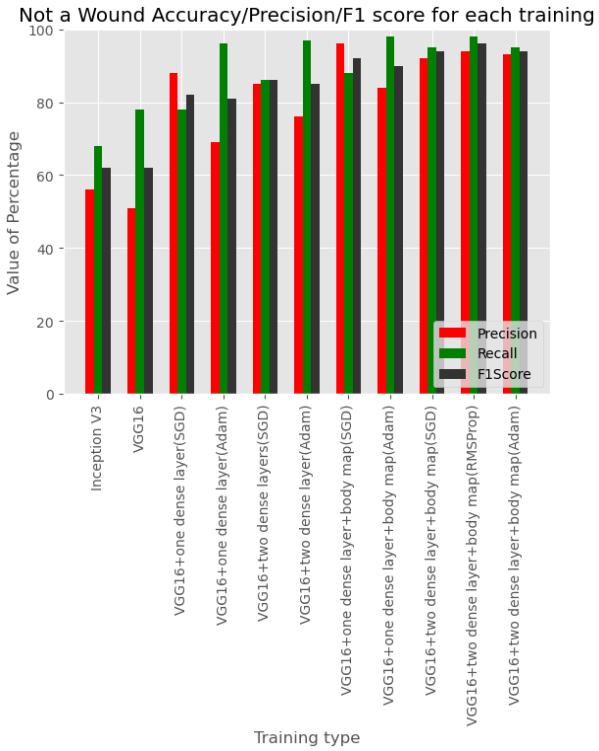


Figure 5.30: Not a wound label metrics of all metrics

5.16 Comparative Analysis

The comparative analysis study on AZHMT dataset is done in order to relate this study with some past works. The study is performed on three other state of the art CNN models with same number of epochs. Table 5.14 shows performance results of different network architectures that were considered in the study

Table 5.14: Comparison study of wound image classifier with SOTA CNN architectures

Network Type	Precision	Recall	Accuracy	F1-Score
Ensembled DCNN[20]	0.92	0.91	91.9%	0.91
VGG19 + LSTM[22]	0.88	0.91	93.22%	0.89
VGG 16 + Dense Layers	**0.95**	**0.94**	**94.57%**	**0.94**

From the comparative study, it can be inferred that the pre-trained model performs better when the output of these models are used for transferring knowledge to another network. Even through the pre-trained models are trained for millions of images, the network might not perform well for specific dataset if the trainable layers are frozen. Hence, it is evident that the use of transfer learning to train any other trainable network outperforms any other network built from scratch with few epochs and low dataset as well. In table 5.14, VGG 16 is the base of transfer learning to dense layer and has outperformed VGG 19 even though VGG 19 is longer network than VGG 16.

Second result to note is that the use of body map along with VGG 16 has outperformed ensembled based DCNN within lesser number of epochs. The ensembled based DCNN[20] used 20 epochs whereas our network uses only 15 epochs. It concludes that use of body map plays significant role in improving performance of the network.

5.17 Observed output classifications

Figure 5.31 shows the VGG 16 with dense layers and body map classifying ten types of images correctly according to their labels. Similarly, some samples of

incorrectly classified examples are shown as well in figure 5.32. Their predicted label are shown on top of images whereas actual labels are shown on side of images.

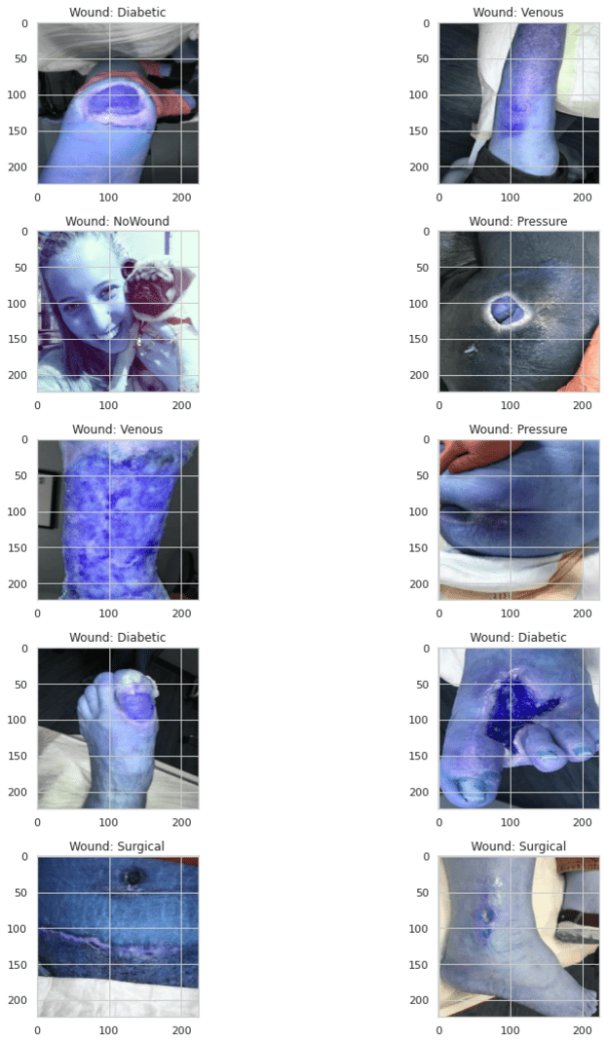


Figure 5.31: Correct Classification of wounds

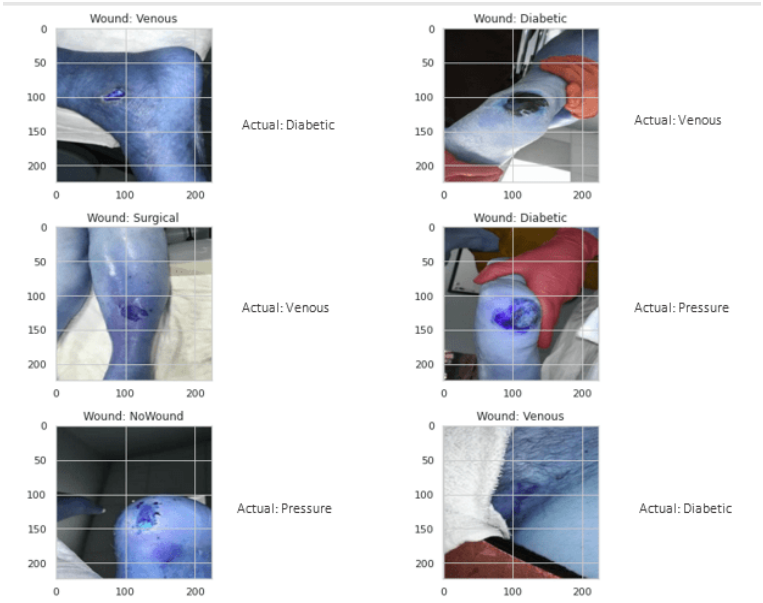


Figure 5.32: Incorrect Classification of wounds

CHAPTER 6

CONCLUSION

6.1 Conclusion

A VGG 16 pre-trained architecture is effectively implemented for the multi-class classification of wound images which are labelled as "Diabetic Ulcer", "Venous Ulcer", "Pressure Ulcer", "Surgical Wound" and "Not a wound". The study consists of use of frozen paramters for VGG 16 initially which returned low accuracy of only 62.23% only. So it was realized that transfer learning must be implemented on this network in order to improve performance. Transfer learning was implemented in VGG 16 by adding two trainable dense layers with ReLU activation on top of VGG 16 and using body map to further enhance performance. Through multiple study with different optimization techniques and body maps, the performance gradually increased in each case and finally the best accuracy was achieved at 94.57%. This result was compared to other implementations of VGG 19 and Ensembled DCNN. These implementations did not add different trainable layers but instead modified the internal layers to be trainable. It was evident from the results that adding extra trainable layers on top will be more effective than making internal layers trainable. This was concluded because even though VGG 16 is only 16 layers deep, adding only two trainable layer on top outperforms VGG 19 which is 19 layers deep for the same AZHMT dataset. Automated wound classification of wound images is potentially applicable for use in real medical environment. The study suggests that this network can be used for practical application as well in order to correctly classify the type of wound. This helps medical personnel to identify type of wound correctly and plan the course of treatment accordingly.

6.2 Challenges

The most challenging part in the study was to manually label the body maps in each image dataset. It took approximately 2 months to collect dataset from various

sources and label the body map accordingly. The collected dataset was validated with help of medical officer and the body map value was assigned to each images. Another challenging part to this study was implementation of transfer learning to train a dense layer from pre-trained model.

6.3 Future Works

The study is limited to classification of wound images through transfer learning and body map only. For the future scope, with advanced hardware, the study can be continued in order to predict the healing rate of ulcers as well. Similarly for surgical wound, the study can be extended to check whether any infections has occured or not through image. These extensions can be done by collecting large amount of dataset and through advanced hardware for future purpose.

References

[1] S. Sarp, M. Kuzlu, E. Wilson, U. Cali, and O. Guler, "The enlightening role of explainable artificial intelligence in chronic wound classification," *Electronics*, vol. 10, no. 12, p. 1406, 2021.

[2] J. Hurlow and P. G. Bowler, "Acute and chronic wound infections: Microbiological, immunological, clinical and therapeutic distinctions," *Journal of wound care*, vol. 31, no. 5, pp. 436–445, 2022.

[3] N. PHOLBERDEE *et al.*, "Wound-region segmentation from image by using deep learning and various data augmentation methods," Ph.D. dissertation, Silpakorn University, 2019.

[4] N. Ohura, R. Mitsuno, M. Sakisaka, *et al.*, "Convolutional neural networks for wound detection: The role of artificial intelligence in wound care," *Journal of Wound Care*, vol. 28, no. Sup10, S13–S24, 2019.

[5] S. Sarp, M. Kuzlu, E. Wilson, U. Cali, and O. Guler, "A highly transparent and explainable artificial intelligence tool for chronic wound classification: Xai-cwc," 2021.

[6] W. J. Jeffcoate, L. Vileikyte, E. J. Boyko, D. G. Armstrong, and A. J. Boulton, "Current challenges and opportunities in the prevention and management of diabetic foot ulcers," *Diabetes care*, vol. 41, no. 4, pp. 645–652, 2018.

[7] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," *Classification in BioApps: Automation of Decision Making*, pp. 323–350, 2018.

[8] D. G. Armstrong, A. J. Boulton, and S. A. Bus, "Diabetic foot ulcers and their recurrence," *New England Journal of Medicine*, vol. 376, no. 24, pp. 2367–2375, 2017.

[9] S. B. Millan, R. Gan, and P. E. Townsend, "Venous ulcers: Diagnosis and treatment," *American family physician*, vol. 100, no. 5, pp. 298–305, 2019.

[10] A. Mahbod, G. Schaefer, R. Ecker, and I. Ellinger, "Automatic foot ulcer segmentation using an ensemble of convolutional neural networks," in *2022*

26th International Conference on Pattern Recognition (ICPR), IEEE, 2022, pp. 4358–4364.

[11] C. K. Sen, "Human wound and its burden: Updated 2020 compendium of estimates," *advances in wound care*, vol. 10, no. 5, pp. 281–292, 2021.

[12] A. C. Johnson, E. P. Buchanan, and D. Y. Khechoyan, "Wound infection: A review of qualitative and quantitative assessment modalities," *Journal of Plastic, Reconstructive & Aesthetic Surgery*, vol. 75, no. 4, pp. 1287–1296, 2022.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[14] C. Wang, D. Anisuzzaman, V. Williamson, *et al.*, "Fully automatic wound segmentation with deep convolutional neural networks," *Scientific reports*, vol. 10, no. 1, p. 21 897, 2020.

[15] J.-T. Hsu, Y.-W. Chen, T.-W. Ho, *et al.*, "Chronic wound assessment and infection detection method," *BMC medical informatics and decision making*, vol. 19, no. 1, pp. 1–20, 2019.

[16] C. Cui, K. Thurnhofer-Hemsi, R. Soroushmehr, *et al.*, "Diabetic wound segmentation using convolutional neural networks," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2019, pp. 1002–1005.

[17] X. Liu, C. Wang, F. Li, X. Zhao, E. Zhu, and Y. Peng, "A framework of wound segmentation based on deep convolutional networks," in *2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, IEEE, 2017, pp. 1–7.

[18] H. Lu, B. Li, J. Zhu, *et al.*, "Wound intensity correction and segmentation with convolutional neural networks," *Concurrency and computation: practice and experience*, vol. 29, no. 6, e3927, 2017.

[19] C. Venkatesan, M. Sumithra, and M Murugappan, "Nfu-net: An automated framework for the detection of neurotrophic foot ulcer using deep convolu-

tional neural network," *Neural Processing Letters*, vol. 54, no. 5, pp. 3705–3726, 2022.

[20] B. Rostami, D. Anisuzzaman, C. Wang, S. Gopalakrishnan, J. Niezgoda, and Z. Yu, "Multiclass wound image classification using an ensemble deep cnn-based classifier," *Computers in Biology and Medicine*, vol. 134, p. 104 536, 2021.

[21] S. Sarp, Y. Zhao, and M. Kuzlu, "Artificial intelligence-powered chronic wound management system: Towards human digital twins," 2022.

[22] D. M. Anisuzzaman, Y. Patel, B. Rostami, J. Niezgoda, S. Gopalakrishnan, and Z. Yu, "Multi-modal wound classification using wound image and location by deep neural network," 2021. DOI: 10.1038/s41598-022-21813-0. eprint: arXiv:2109.06969.

[23] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, *et al.*, "Towards a better understanding of transfer learning for medical imaging: A case study," *Applied Sciences*, vol. 10, no. 13, p. 4523, 2020.

[24] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, 2015. eprint: arXiv:1512.00567.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[28] S. Haykin and B. Kosko, "Gradientbased learning applied to document recognition," in *Intelligent Signal Processing*. 2001, pp. 306–351. DOI: 10.1109/9780470544976.ch9.

[29] J. Gu, Z. Wang, J. Kuen, *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[30] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[31] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks," *arXiv preprint arXiv:1503.02351*, 2015.

[32] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[33] A. Jentzen, B. Kuckuck, A. Neufeld, and P. von Wurstemberger, "Strong error analysis for stochastic gradient descent optimization algorithms," *IMA Journal of Numerical Analysis*, vol. 41, no. 1, pp. 455–492, 2021.

[34] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 2880–2887.

[35] X. Qian and D. Klabjan, "The impact of the mini-batch size on the variance of gradients in stochastic gradient descent," *arXiv preprint arXiv:2004.13146*, 2020.

[36] A. Lydia and S. Francis, "Adagrad—an optimizer for stochastic gradient descent," *Int. J. Inf. Comput. Sci*, vol. 6, no. 5, pp. 566–568, 2019.

[37] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, Ieee, 2018, pp. 1–2.

[38] A. Tato and R. Nkambou, "Improving adam optimizer," 2018.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] L. Wu, J. Li, Y. Wang, *et al.*, "R-drop: Regularized dropout for neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 890–10 905, 2021.

[41] L. Tuggener, J. Schmidhuber, and T. Stadelmann, "Is it enough to optimize cnn architectures on imagenet?," 2022.

[42] Z. H. Hoo, J. Candlish, and D. Teare, *What is an roc curve?* 2017.

[43] S. Narkhede, "Understanding auc-roc curve," *Towards Data Science*, vol. 26, no. 1, pp. 220–227, 2018.

[44] J. Elfehri, F. Boussu, V. Koncar, and C. Vasseur, "Novel approach of ulcer prevention based on pressure distribution control algorithm," in *2011*

IEEE International Conference on Mechatronics and Automation, IEEE, 2011, pp. 265–270.

[45] E. M. Tottoli, R. Dorati, I. Genta, E. Chiesa, S. Pisani, and B. Conti, "Skin wound healing process and new emerging technologies for skin wound care and regeneration," *Pharmaceutics*, vol. 12, no. 8, p. 735, 2020.

[46] S. Ding, F. Lin, and B. M. Gillespie, "Surgical wound assessment and documentation of nurses: An integrative review," *Journal of Wound Care*, vol. 25, no. 5, pp. 232–240, 2016.